Fibonacci Numbers, Vector Programs and a new kind of science

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

Example: $f_5 = 5$

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

Example: $f_5 = 5$

4 = 2 + 2

2 + 1 + 1

1 + 2 + 1

1 + 1 + 2

1 + 1 + 1 + 1

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

f_1

f_2

f_3

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

$f_1 = 1$

0 = the empty sum

$f_2 = 1$

1 = 1

$f_3 = 2$

2 = 1 + 1

2

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

$$f_{n+1} = f_n + f_{n-1}$$

Sequences That Sum To n

Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

$$f_{n+1} = f_n + f_{n-1}$$

# of sequences beginning with a 1

# of sequences beginning with a 2

Leonardo Fibonacci

In 1202, Fibonacci proposed a problem about the growth of rabbit populations.

Rules

1. in the first month there is just one pair
2. new-born pairs become fertile after their second month
3. each month every fertile pair begets a new pair, and
4. the rabbits never die

Inductive Definition or Recurrence Relation for the Fibonacci Numbers

Stage 0, Initial Condition, or Base Case:

Fib(0) = 0; Fib (1) = 1

Inductive Rule

For n>1, Fib(n) = Fib(n-1) + Fib(n-2)

n	0	1	2	3	4	5	6	7
Fib(n)	0	1	1	2	3	5	8	13

Fibonacci Numbers Again

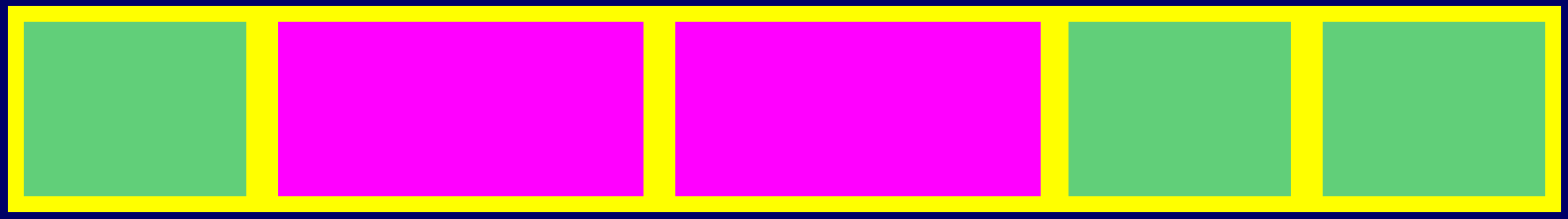
Let f_{n+1} be the number of different sequences of 1's and 2's that sum to n.

$$f_{n+1} = f_n + f_{n-1}$$

$$f_1 = 1 \qquad f_2 = 1$$

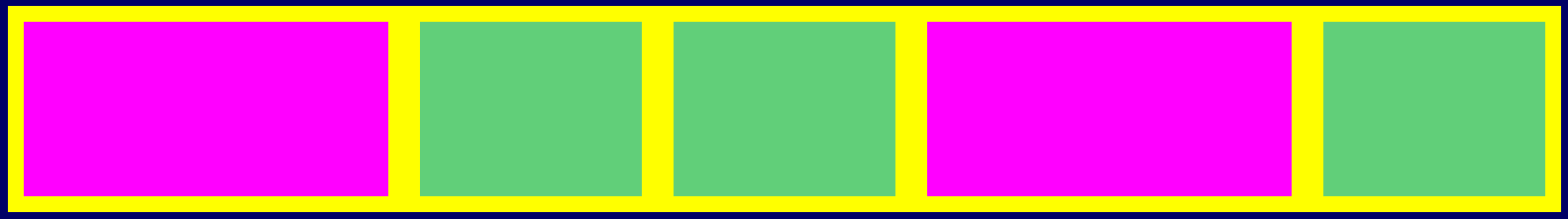
Visual Representation: Tiling

Let f_{n+1} be the number of different ways to tile a $1 \times n$ strip with squares and dominoes.



Visual Representation: Tiling

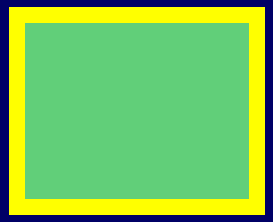
Let f_{n+1} be the number of different ways to tile a $1 \times n$ strip with squares and dominoes.



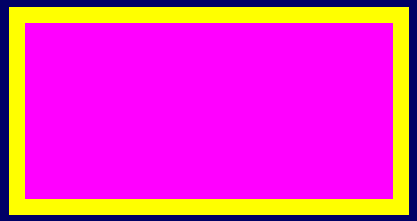
Visual Representation: Tiling

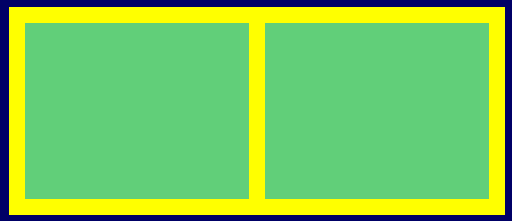
1 way to tile a strip of length 0

1 way to tile a strip of length 1:



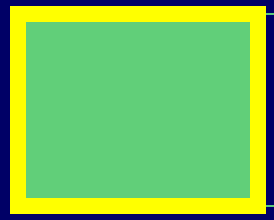
2 ways to tile a strip of length 2:



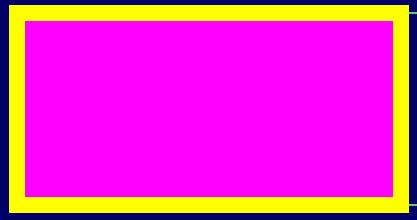


$$f_{n+1} = f_n + f_{n-1}$$

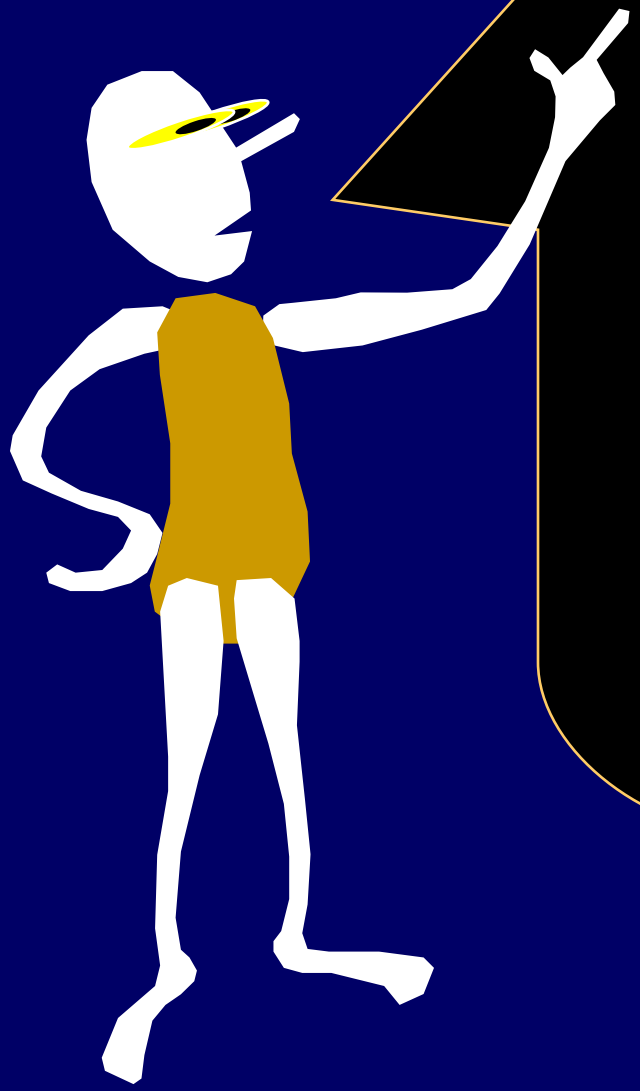
f_{n+1} is number of ways to tile length n.



f_n tilings that start with a square.



f_{n-1} tilings that start with a domino.



Let’s use this visual representation to prove a couple of Fibonacci identities.

Fibonacci Identities

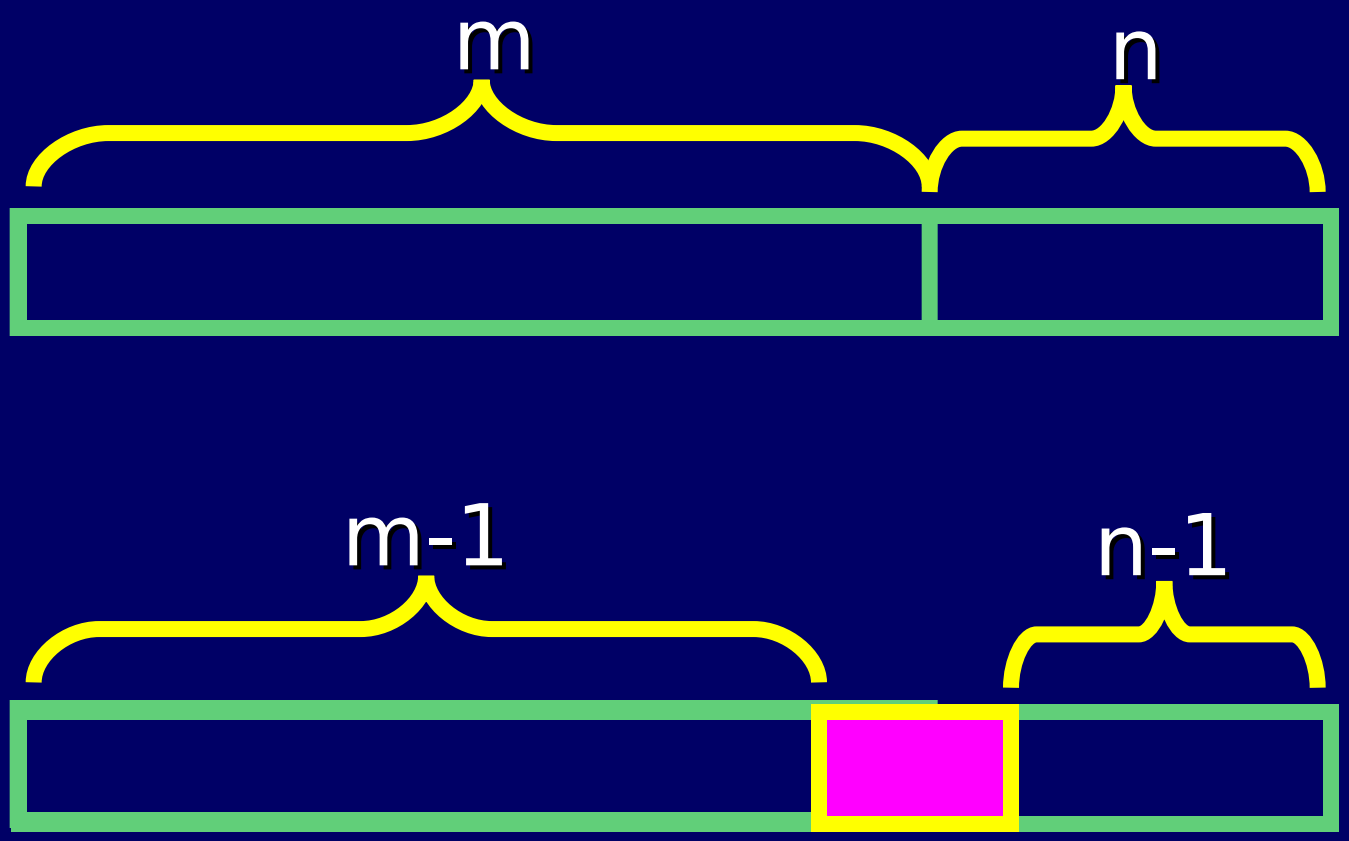
Some examples:

$$F_{2n} = F_1 + F_3 + F_5 + \ldots + F_{2n-1}$$

$$F_{m+n+1} = F_{m+1} F_{n+1} + F_m F_n$$

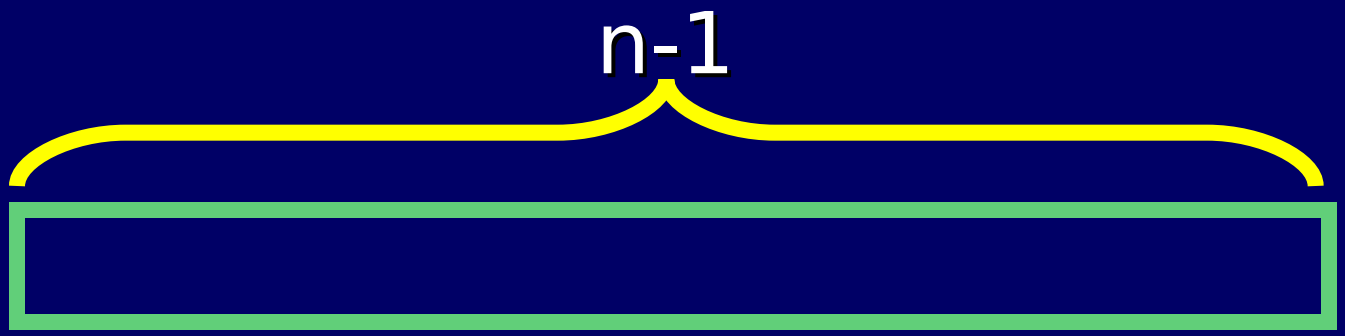
$$(F_n)^2 = F_{n-1} F_{n+1} + (-1)^n$$

$$F_{m+n+1} = F_{m+1}\,F_{n+1} + F_m\,F_n$$



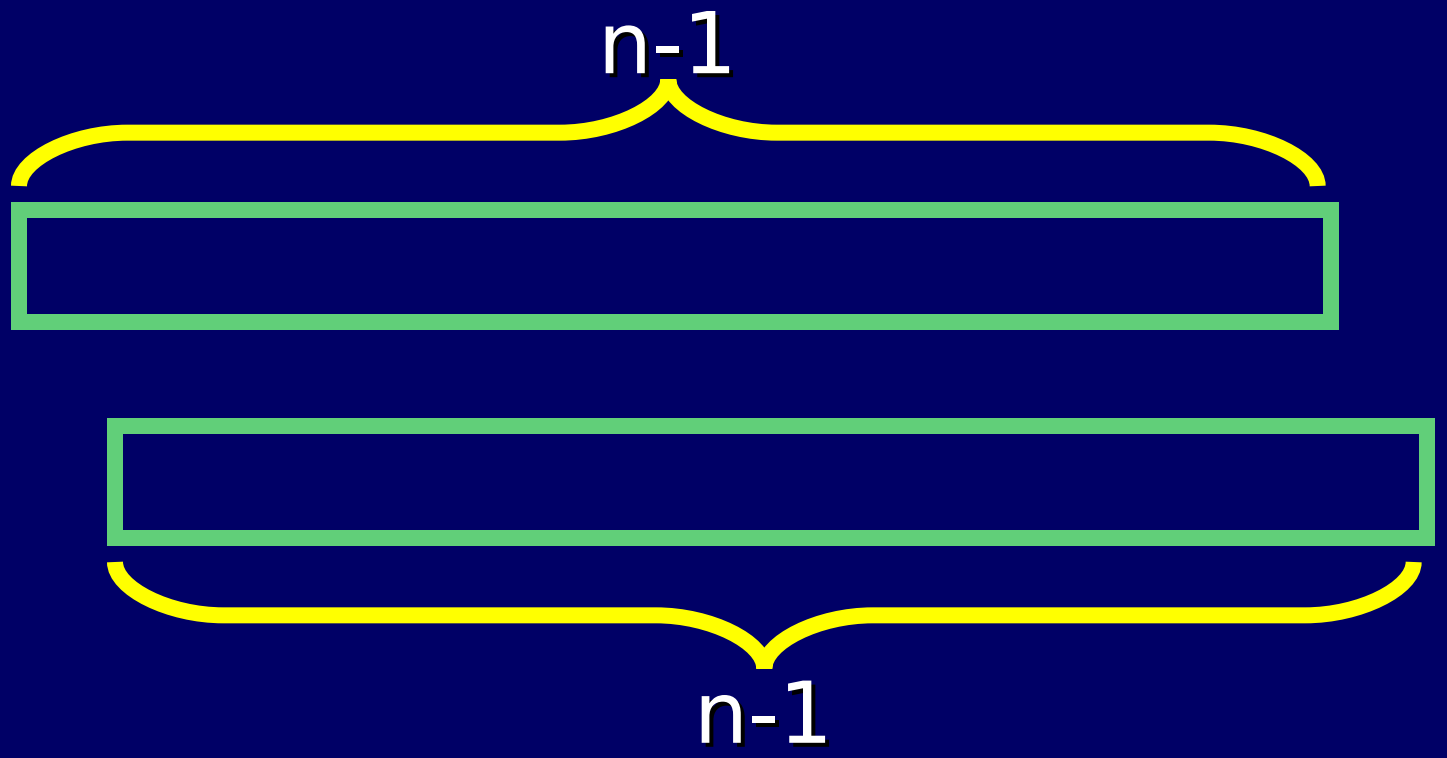
$$(F_n)^2 = F_{n-1} F_{n+1} + (-1)^n$$

$$(F_n)^2 = F_{n-1}\,F_{n+1} + (-1)^n$$

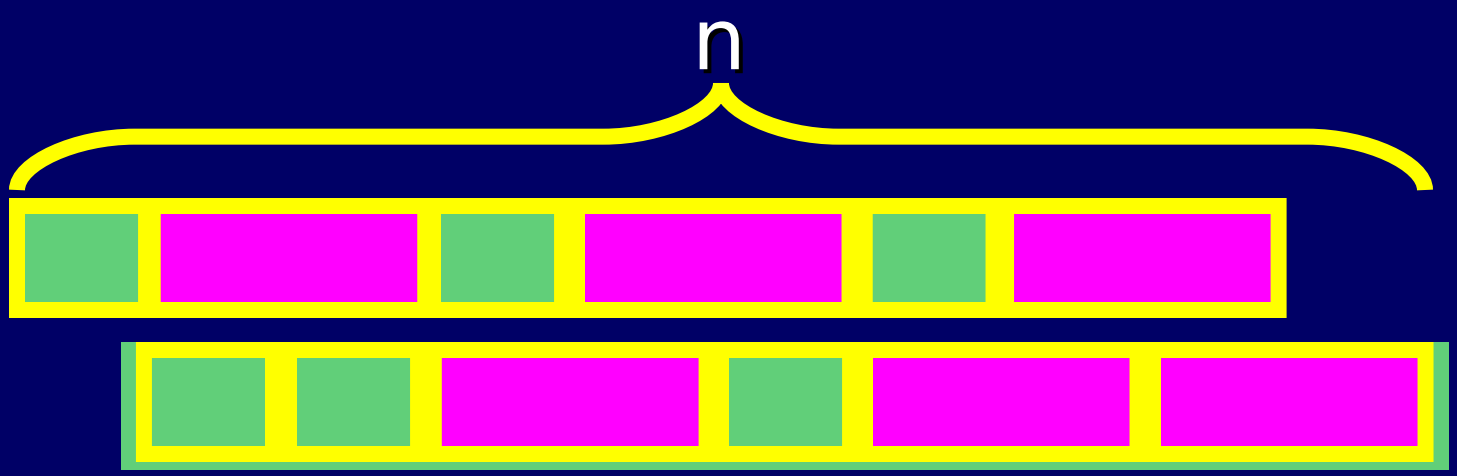


F_n tilings of a strip of length n-1

$$(F_n)^2 = F_{n-1}\,F_{n+1} + (-1)^n$$

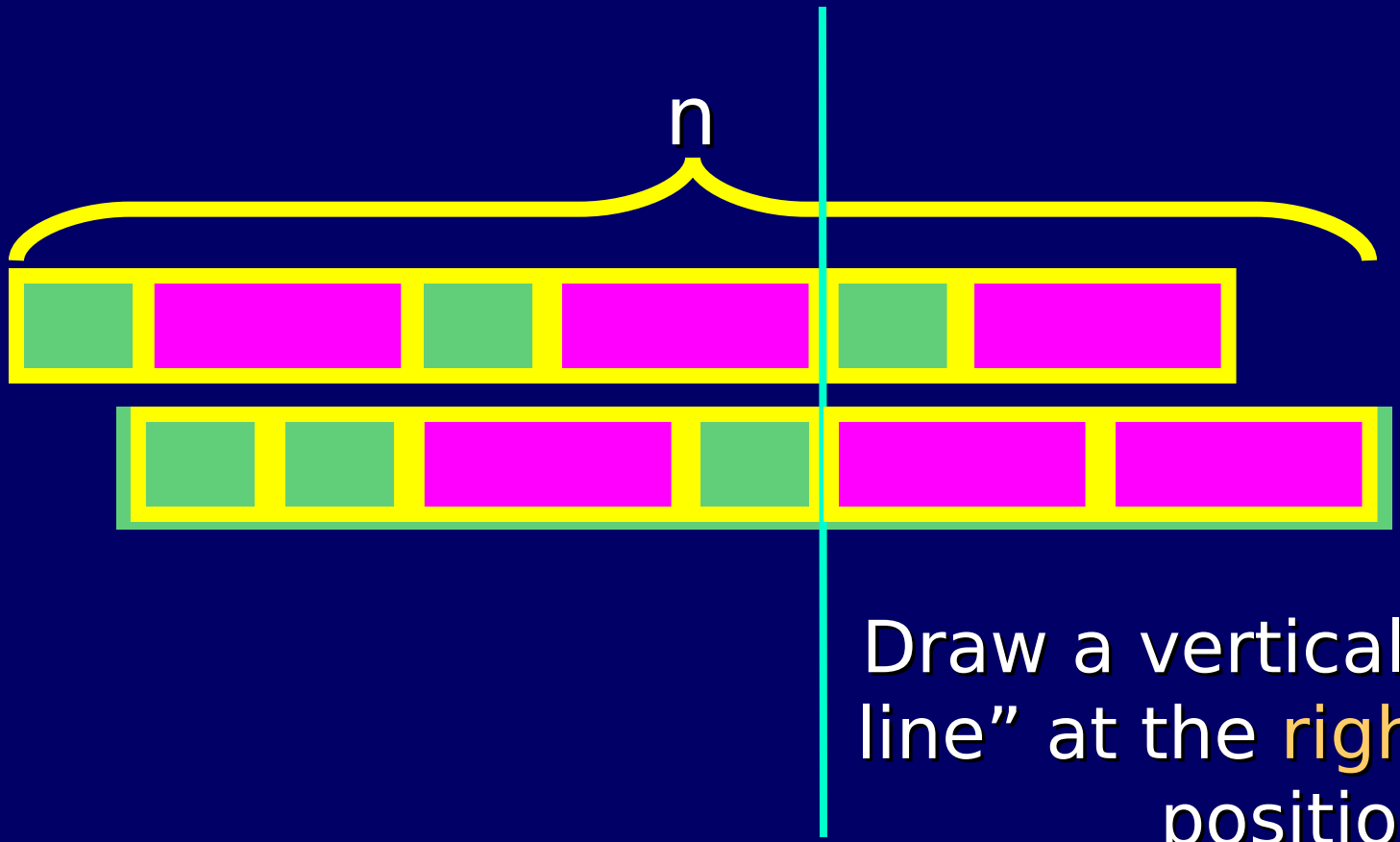


$$(F_n)^2 = F_{n-1}\,F_{n+1} + (-1)^n$$



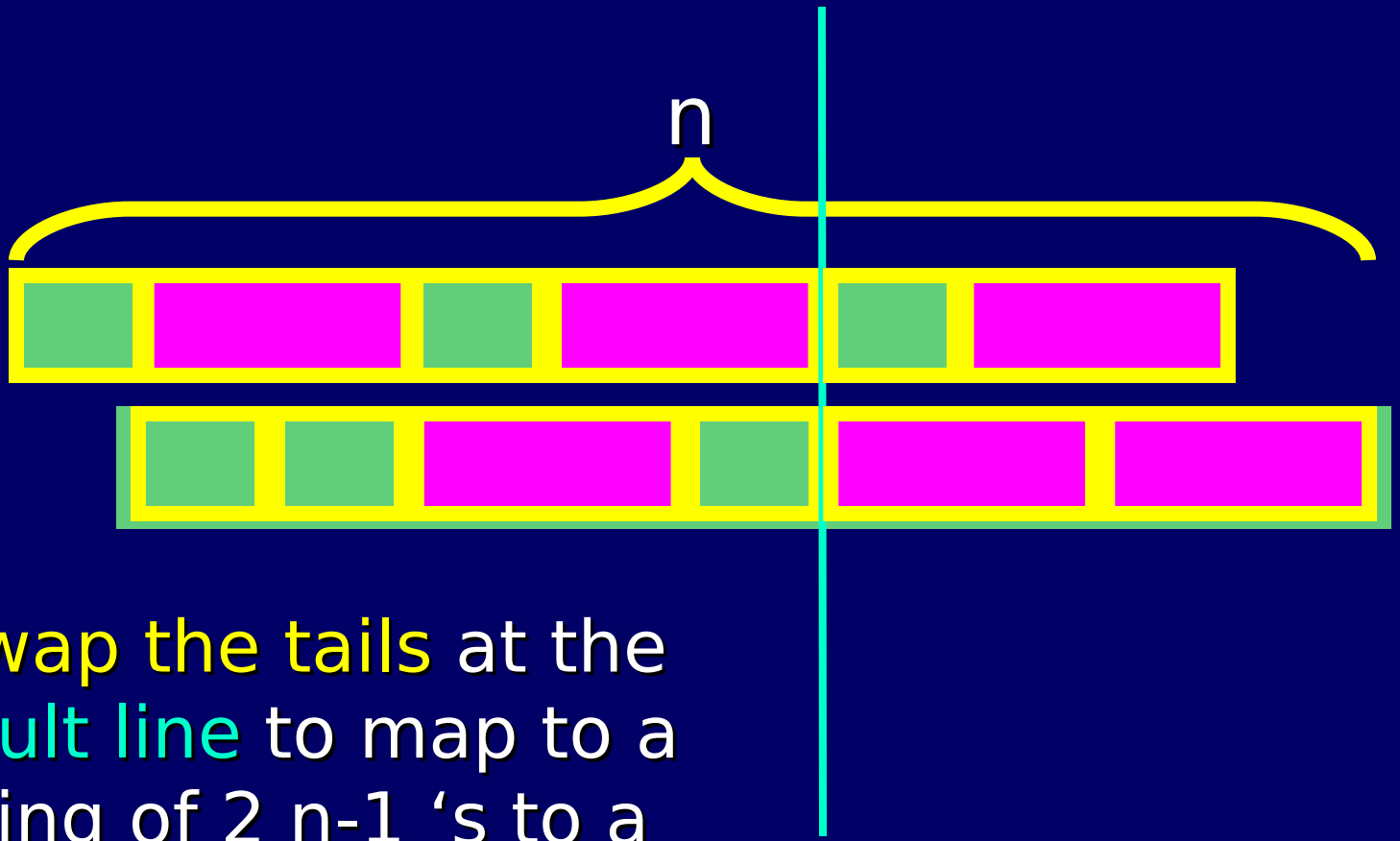
$(F_n)^2$ tilings of two strips of size n-1

$$(F_n)^2 = F_{n-1}\,F_{n+1} + (-1)^n$$



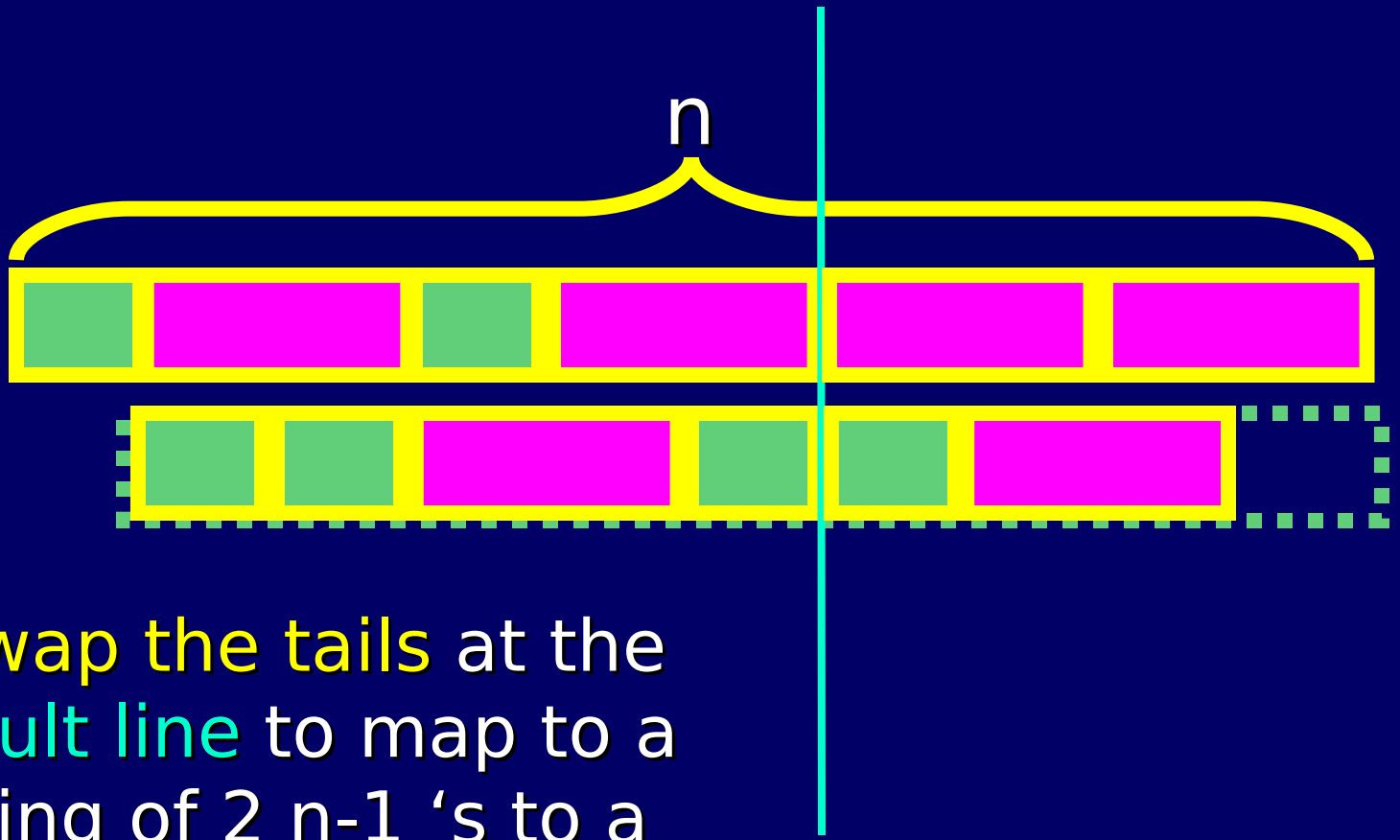
Draw a vertical “fault line” at the rightmost position (<n) possible without cutting any

$$(F_n)^2 = F_{n-1} F_{n+1} + (-1)^n$$



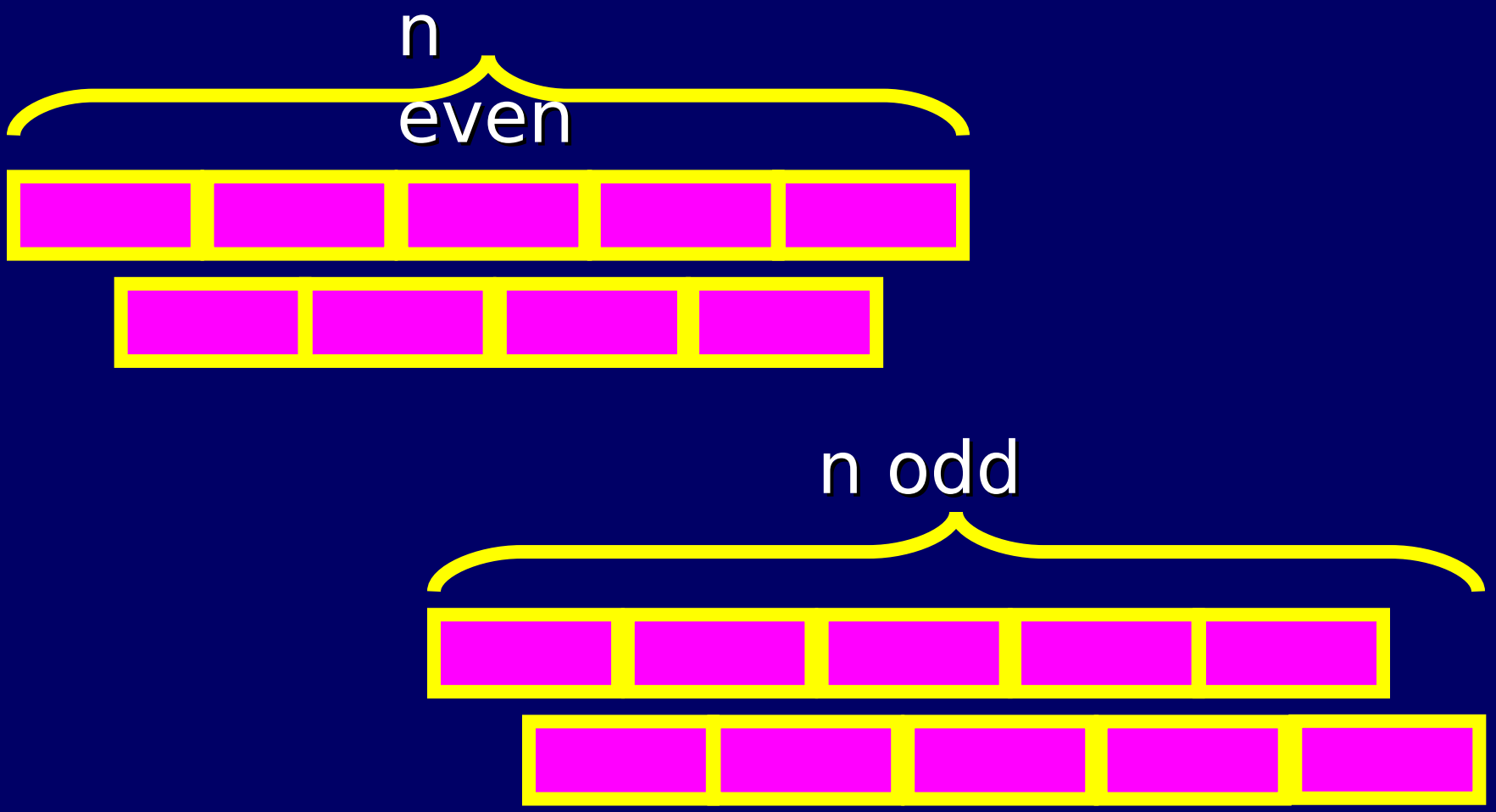
Swap the tails at the fault line to map to a tiling of 2 n-1 ‘s to a tiling of an n-2 and an n.

$$(F_n)^2 = F_{n-1} F_{n+1} + (-1)^n$$



Swap the tails at the fault line to map to a tiling of 2 n-1 ‘s to a tiling of an n-2 and an n.

$$(F_n)^2 = F_{n-1}\,F_{n+1} + (-1)^{n-1}$$



The Fibonacci Quarterly

Vector Programs

Let’s define a (parallel) programming language called VECTOR that operates on possibly infinite vectors of numbers. Each variable V' can be thought of as:

< *, *, *, *, *, *, ... >

0 1 2 3 4 5 . . . . . .

Vector Programs

Let k stand for a scalar constant

<k> will stand for the vector <k,0,0,0,...>

<0> = <0,0,0,0,....>

<1> = <1,0,0,0,...>

$V^{!} + T^{!}$ means to add the vectors position-wise.

<4,2,3,...> + <5,1,1,....> = <9,3,4,...>

Vector Programs

RIGHT($V^{!}$) means to shift every number in $V^{!}$ one position to the right and to place a 0 in position 0.

RIGHT(<1,2,3, ...>) = <0,1,2,3,. ...>

Vector Programs

Example:

$V^{!} := \langle 6 \rangle;$

$V^{!} := RIGHT(V^{!}) + \langle 42 \rangle;$

$V^{!} := RIGHT(V^{!}) + \langle 2 \rangle;$

$V^{!} := RIGHT(V^{!}) + \langle 13 \rangle;$

Store

$V^{!} = \langle 6,0,0,0,.. \rangle$

$V^{!} = \langle 42,6,0,0,.. \rangle$

$V^{!} = \langle 2,42,6,0,.. \rangle$

$V^{!} = \langle 13,2,42,6,.. \rangle$

$V^{!} = < 13, 2, 42, 6, 0, 0, ... >$

Vector Programs

Example:

$V^{!} := \langle 1 \rangle;$

Loop n times:

$V^{!} := V^{!} + RIGHT(V^{!});$

Store

$V^{!} = \langle 1,0,0,0,.. \rangle$

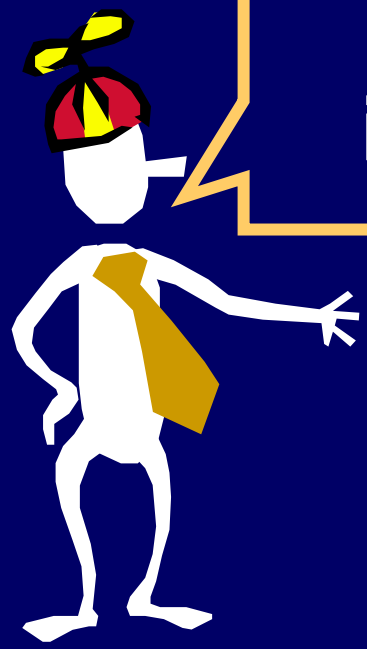
$V^{!} = \langle 1,1,0,0,.. \rangle$

$V^{!} = \langle 1,2,1,0,.. \rangle$

$V^{!} = \langle 1,3,3,1,.. \rangle$

$V^{!} = n^{th}$ row of Pascal's triangle.

Vector programs can be implemented by polynomials!



Programs -----> Polynomials

The vector V' = < a_0, a_1, a_2, . . . > will be represented by the polynomial:

$$P_V = \sum_{i=0}^{i=\infty} a_i X^i$$

Formal Power Series

The vector V' = < a_0, a_1, a_2, . . . > will be represented by the formal power series:

$$P_V = \sum_{i=0}^{i=\infty} a_i X^i$$

$V^! = < a_0, a_1, a_2, \ldots >$

$$P_V = \sum_{i=0}^{i=\infty} a_i X^i$$

<0> is represented by 0

<k> is represented by k

$V^! + T^!$ is represented by $(P_V + P_T)$

RIGHT($V^!$) is represented by $(P_V\ X)$

Vector Programs

Example:

$V^{!} := \langle 1 \rangle;$ $\qquad P_V := 1;$

Loop n times:

$V^{!} := V^{!} + RIGHT(V^{!});$ $\qquad \mathbf{P_V := P_V + P_V X;}$

$V^{!}$ = n^{th} row of Pascal's triangle.

Vector Programs

Example:

$V^{!} := \langle 1 \rangle;$ $\quad P_V := 1;$

Loop n times:

$V^{!} := V^{!} + RIGHT(V^{!});$ $\quad P_V := P_V (1+ X);$

$V^{!}$ = n^{th} row of Pascal's triangle.

Vector Programs

Example:

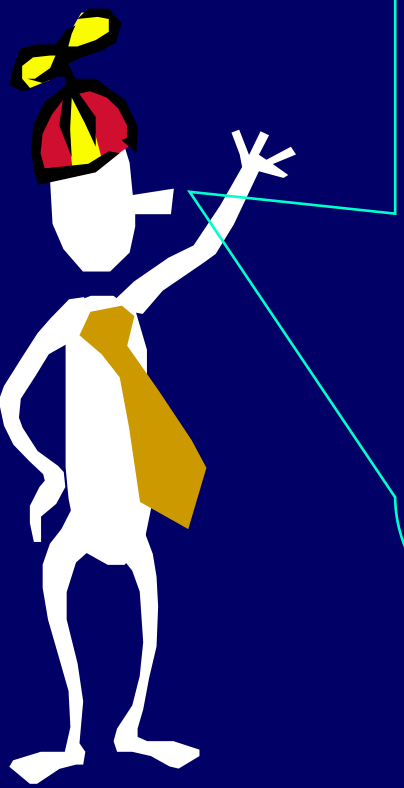
$V^! := \langle 1 \rangle;$

Loop n times:

$V^! := V^! + \text{RIGHT}(V^!);$

$P_V = (1+ X)^n$

$V^!$ = n^{th} row of Pascal's triangle.



Let’s add an instruction called PREFIXSUM to our VECTOR language.

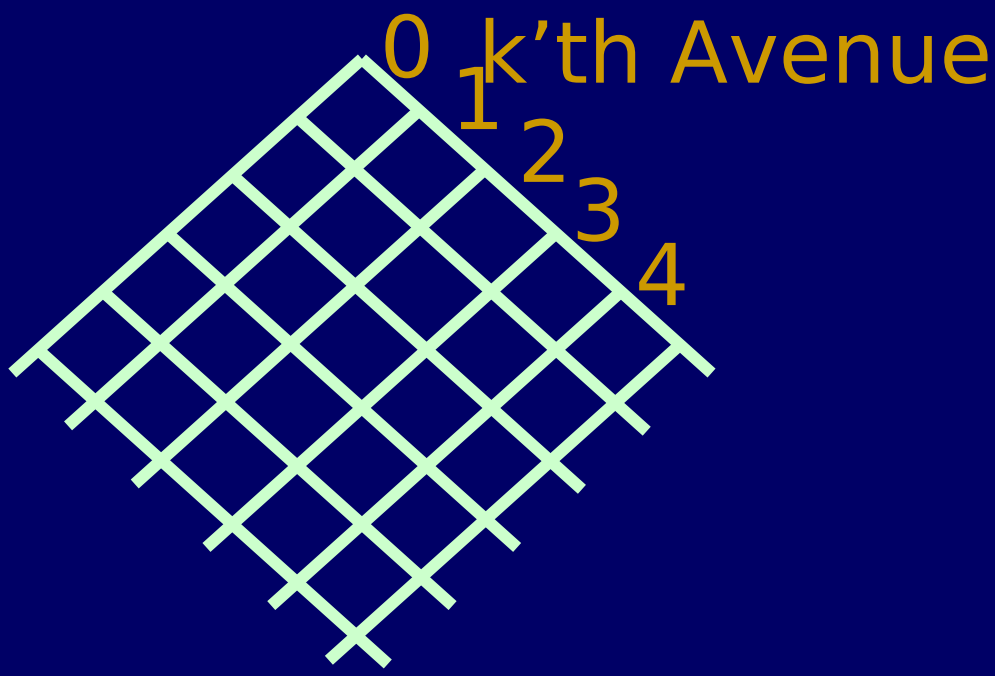
$W^{!} := \text{PREFIXSUM}(V^{!})$

means that the i^{th} position of W contains the sum of all the numbers in V from positions 0 to i.

What does this program output?

$V^! := 1^!$;

Loop k times: $V^! :=$ PREFIXSUM($V^!$) ;



Al Karaji Perfect Squares

```
Zero_Ave    := PREFIXSUM(<1>);
First_Ave   := PREFIXSUM(Zero_Ave);
Second_Ave :=PREFIXSUM(First_Ave);

Output:=
      RIGHT(Second_Ave) + Second_Ave
```

Second_Ave = <1, 3, 6, 10, 15

RIGHT(Second_Ave) = <0, 1, 3, 6, 10,..

Output = <1, 4, 9, 16, 25

Can you see how PREFIXSUM can be represented by a familiar polynomial expression?

How to divide polynomials?

$$\frac{1}{1 - X} \; ?$$

$$
\begin{array}{r|l}
 & 1 + X + X^2 \\
\hline
1 - X & 1 \\
 & -(1 - X) \\
\hline
 & X \\
 & -(X - X^2) \\
\hline
 & X^2 \\
 & -(X^2 - X^3) \\
\hline
 & X^3 \quad \ldots
\end{array}
$$

$$= 1 + X + X^2 + X^3 + X^4 + X^5 + X^6 + X^7 + \ldots$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^n + \ldots = \frac{1}{1 - X}$$

The Infinite Geometric Series

$W^! := \text{PREFIXSUM}(V^!)$

is represented by

$$P_W = P_V / (1-X)$$
$$= P_V (1+X+X^2+X^3+ \ldots)$$

Al-Karaji Program

$Zero\_Ave = 1/(1-X);$

$First\_Ave = 1/(1-X)^2;$

$Second\_Ave = 1/(1-X)^3;$

Output =

$$1/(1-X)^3 + X/(1-X)^3$$

$$= (1+X)/(1-X)^3$$

$(1+X)/(1-X)^3$

```
Zero_Ave    := PREFIXSUM(<1>);
First_Ave   := PREFIXSUM(Zero_Ave);
Second_Ave :=PREFIXSUM(First_Ave);

Output:=
      RIGHT(Second_Ave) + Second_Ave
```

Second_Ave = <1, 3, 6, 10, 15

RIGHT(Second_Ave) = <0, 1, 3, 6, 10,..

Output = <1, 4, 9, 16, 25

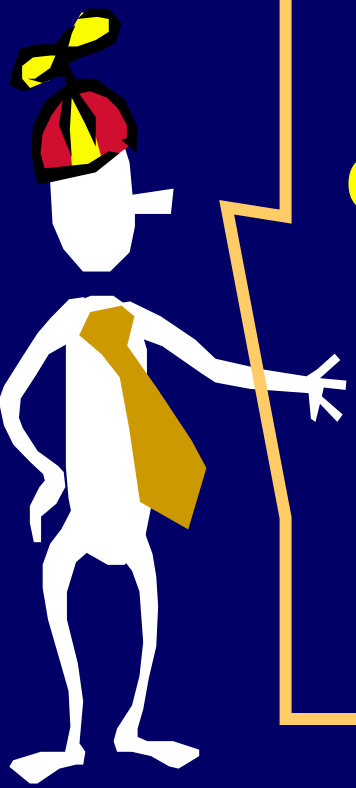
$(1+X)/(1-X)^3$

outputs <1, 4, 9, ..>

$X(1+X)/(1-X)^3$

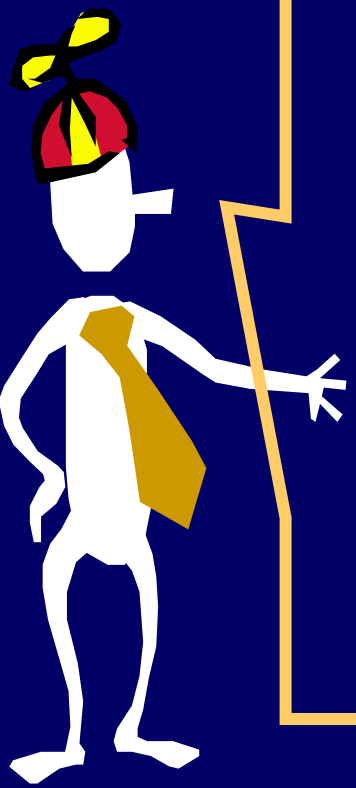
outputs <0, 1, 4, 9, ..>

The k^{th} entry is k^2



$$X(1+X)/(1-X)^3 = \sum k^2 X^k$$

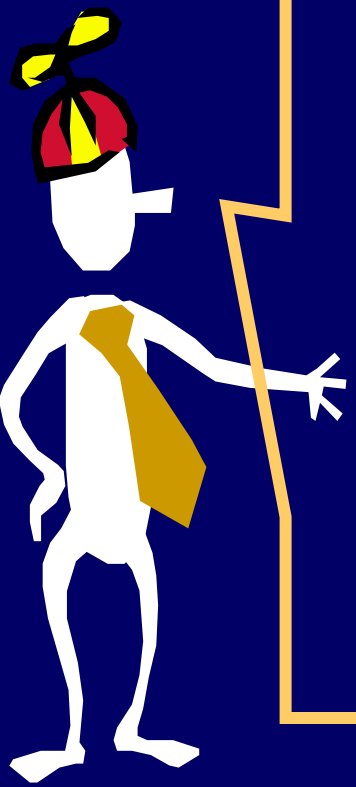
What does $X(1+X)/(1-X)^4$ do?

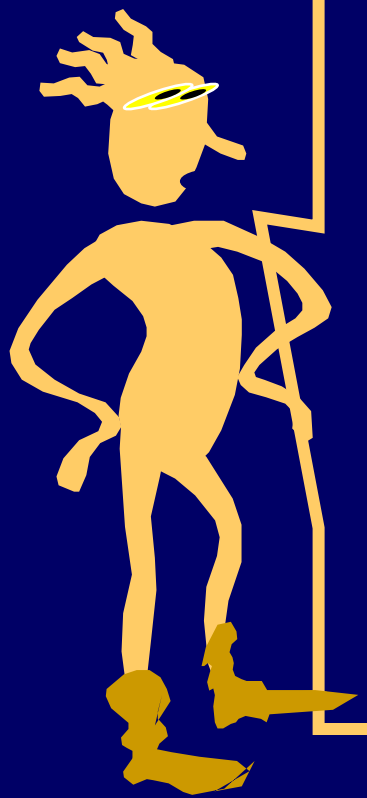


$X(1+X)/(1-X)^4$ expands to :

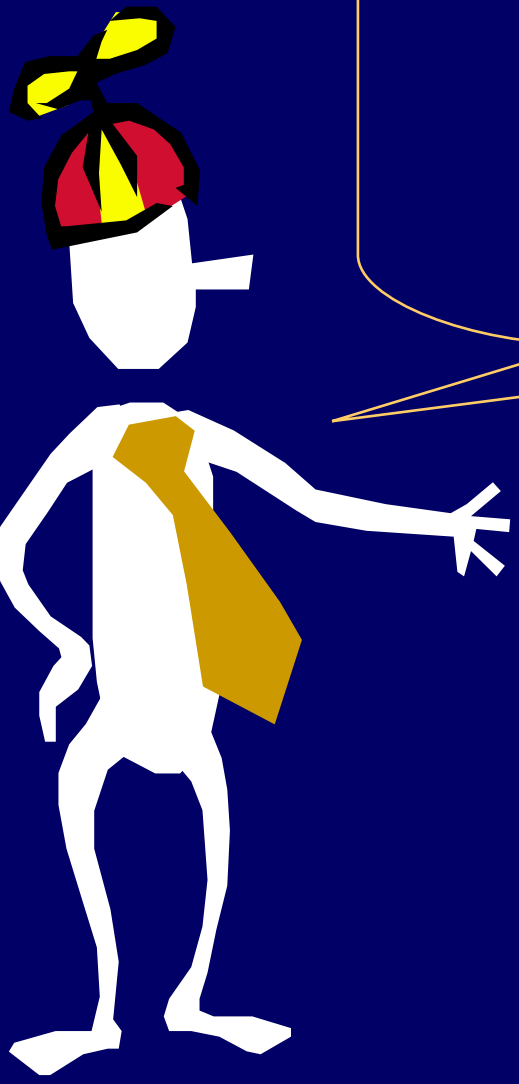
$$\sum S_k X^k$$

where S_k is the sum of the first k squares





Aha! Thus, if there is an alternative interpretation of the k^{th} coefficient of $X(1+X)/(1-X)^4$ we would have a new way to get a formula for the sum of the first k squares.



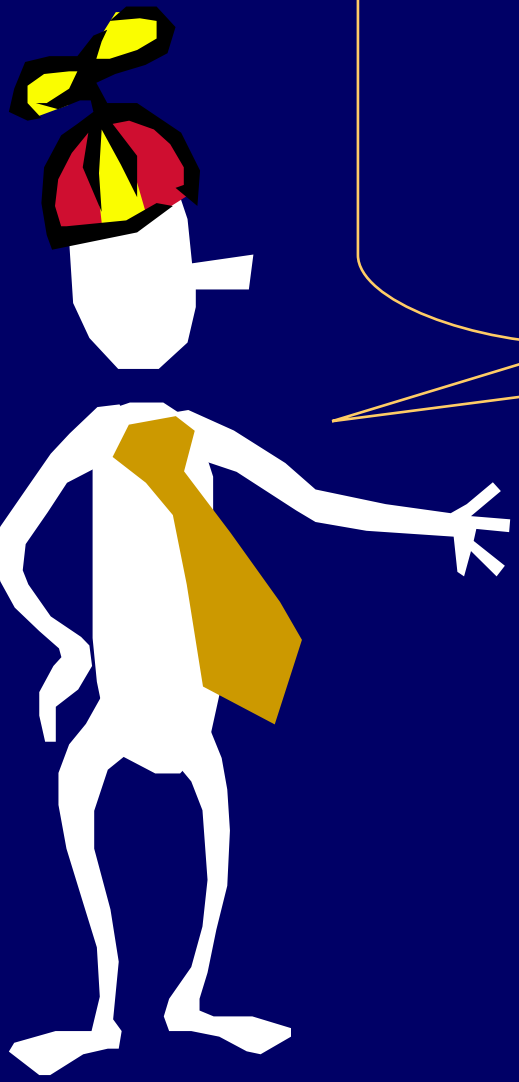
What is the coefficient of X^k in the expansion of:

$$(1 + X + X^2 + X^3 + X^4 + \ldots)^n \; ?$$

Each path in the choice tree for the cross terms has n choices of exponent $e_1, e_2, \ldots, e_n$ ¸ 0. Each exponent can be any natural number.

Coefficient of X^k is the number of non-negative solutions to:

$$e_1 + e_2 + \ldots + e_n = k$$

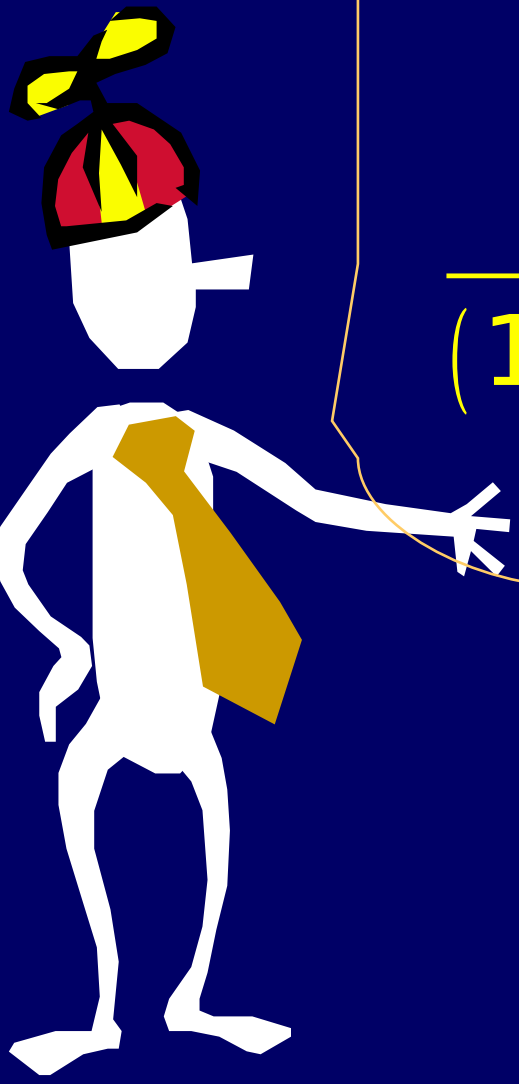


What is the coefficient of X^k in the expansion of:

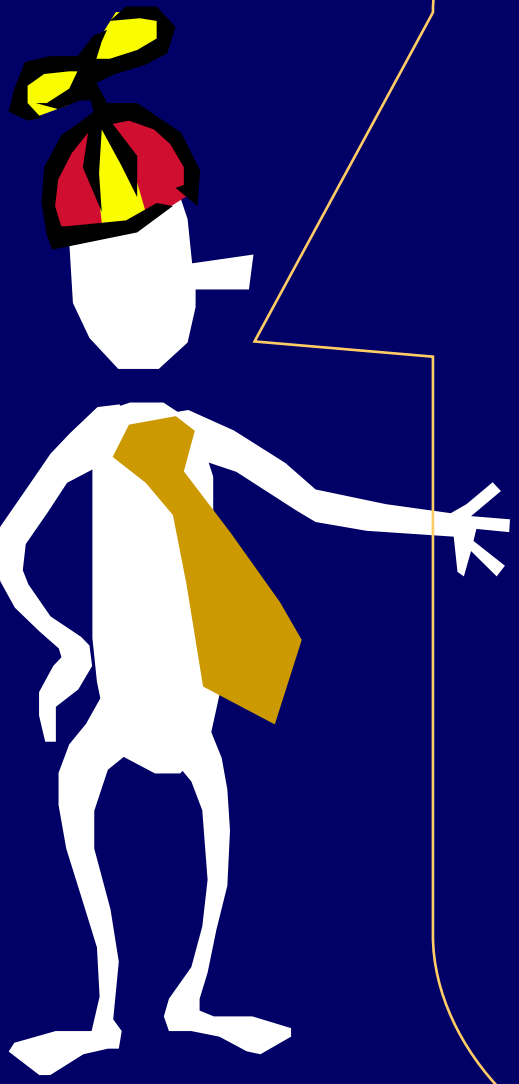
$$(1 + X + X^2 + X^3 + X^4 + \ldots .)^n \; ?$$

$$n + k - 1$$

$$n - 1$$



$$(1 + X + X^2 + X^3 + X^4 + \ldots)^n = \frac{1}{(1-X)^n} = \sum_{k=0} \binom{n+k-1}{n-1} X^k$$



Using pirates and gold we found that:

$$\frac{1}{(1-X)^n} = \sum_{k=0} \binom{n+k-1}{n-1} X^k$$

THUS:

$$\frac{1}{(1-X)^4} = \sum_{k=0} \binom{k+3}{3} X^k$$

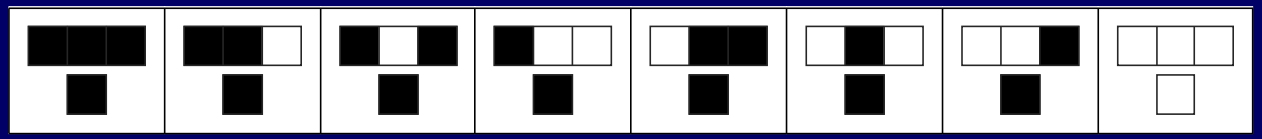
Vector programs -> Polynomials -> Closed form expression

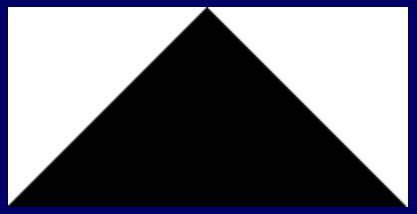
$$\frac{X^2 + X}{(1 - X)^4} = \sum_{k=0}^{\infty} (\binom{k+2}{3} + \binom{k+1}{3}) X^k$$

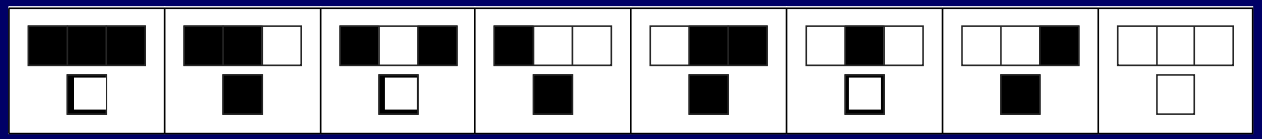
A big jump

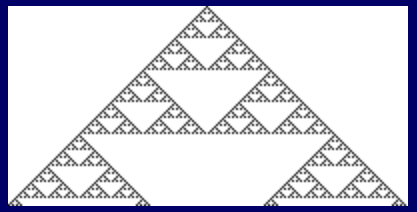
Let’s jump into the world of simple programs

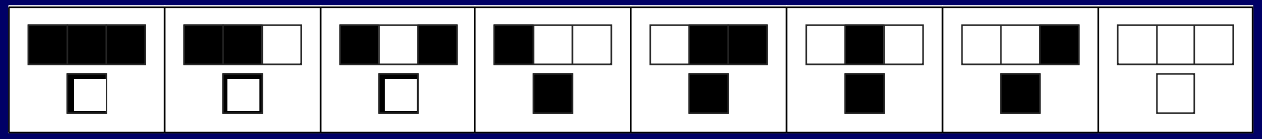
Cellular automata

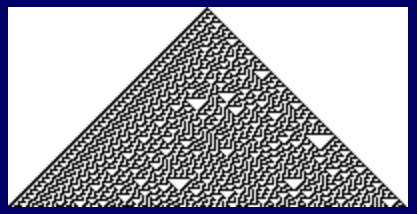


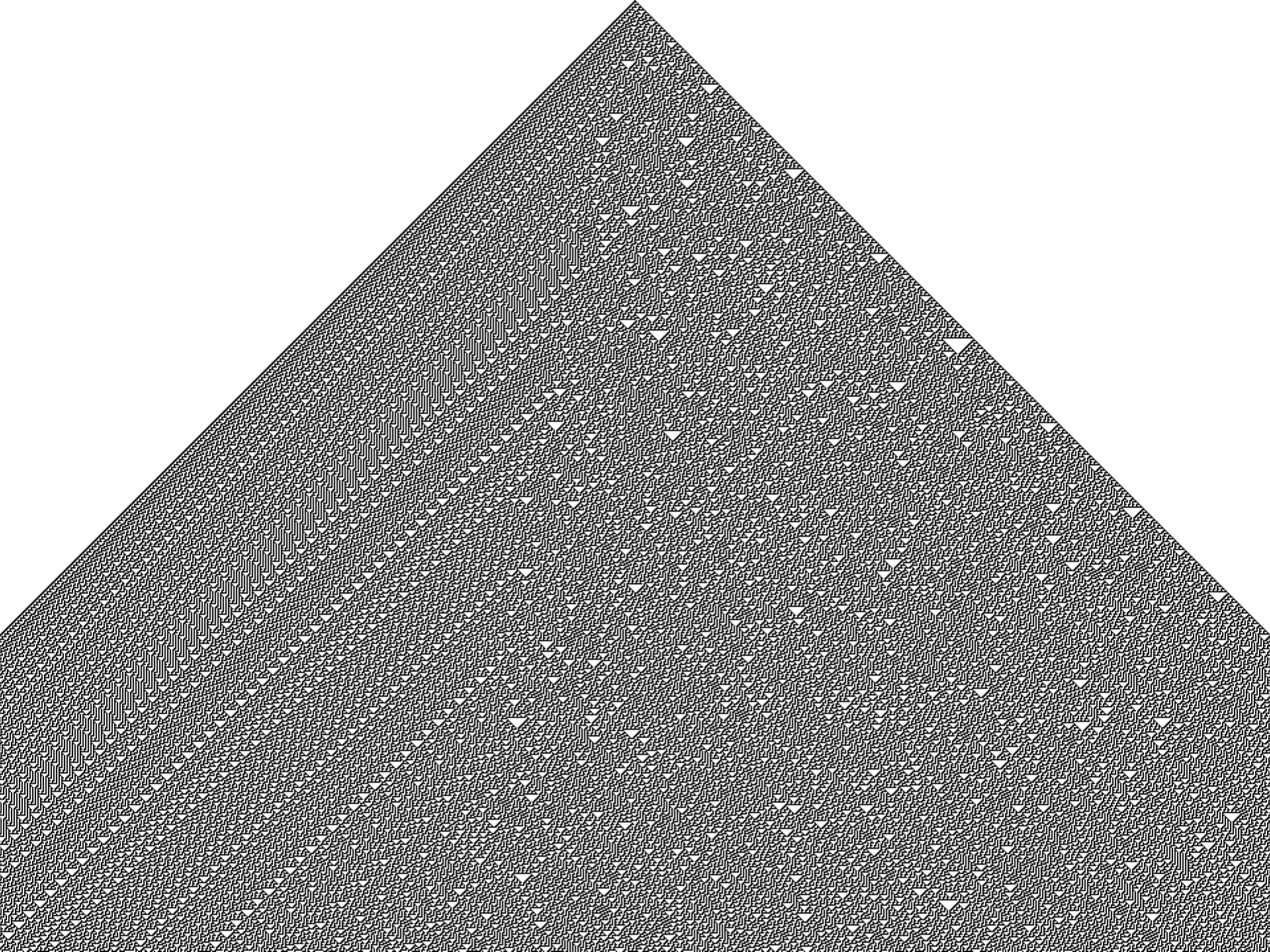








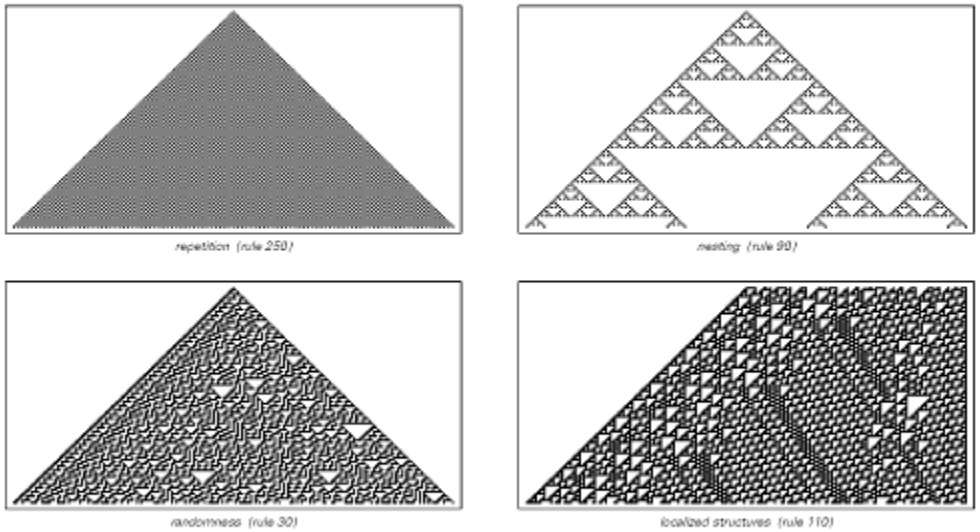




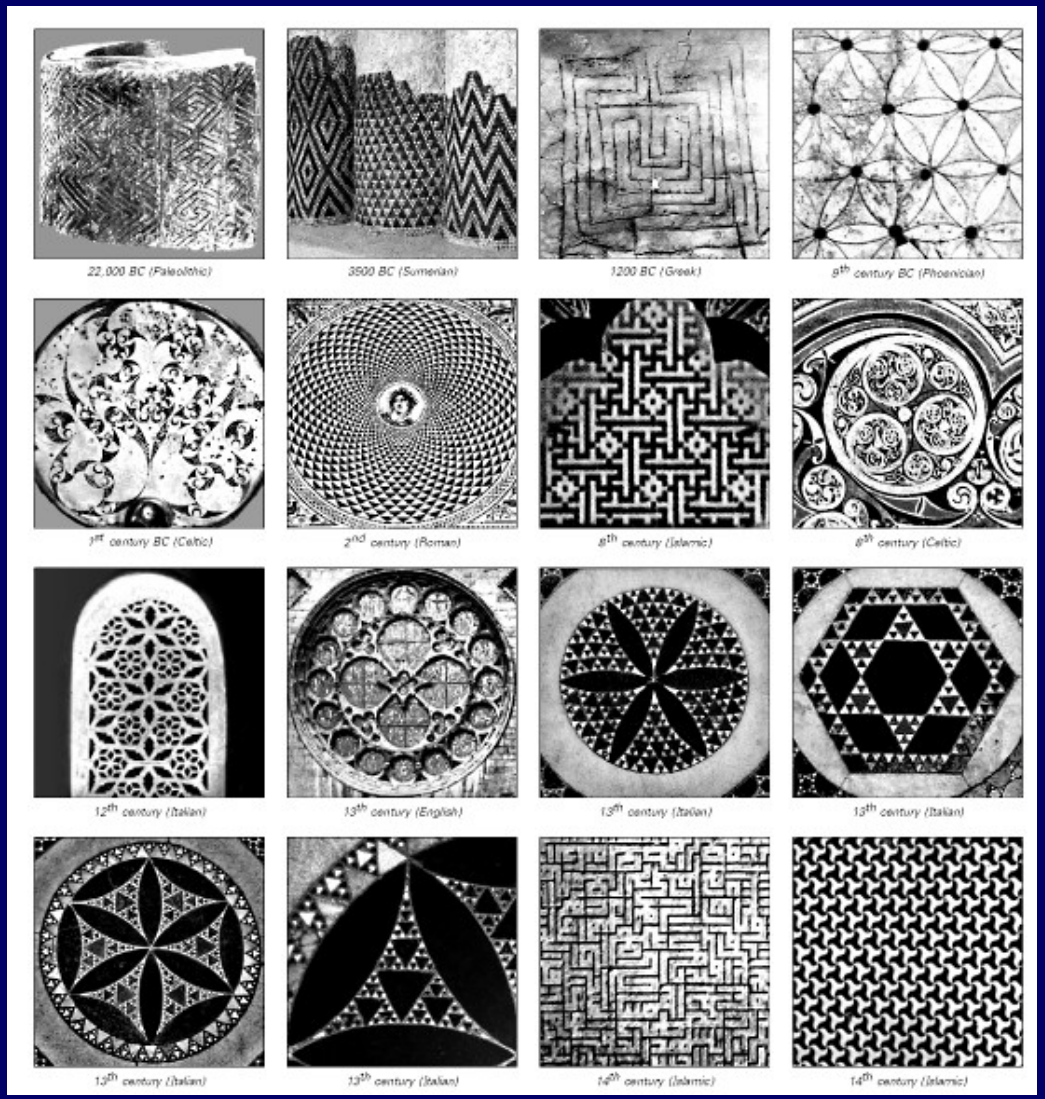
The main discovery

A simple program can create complex output

4 kinds of behavior



Why these discoveries were not made before?



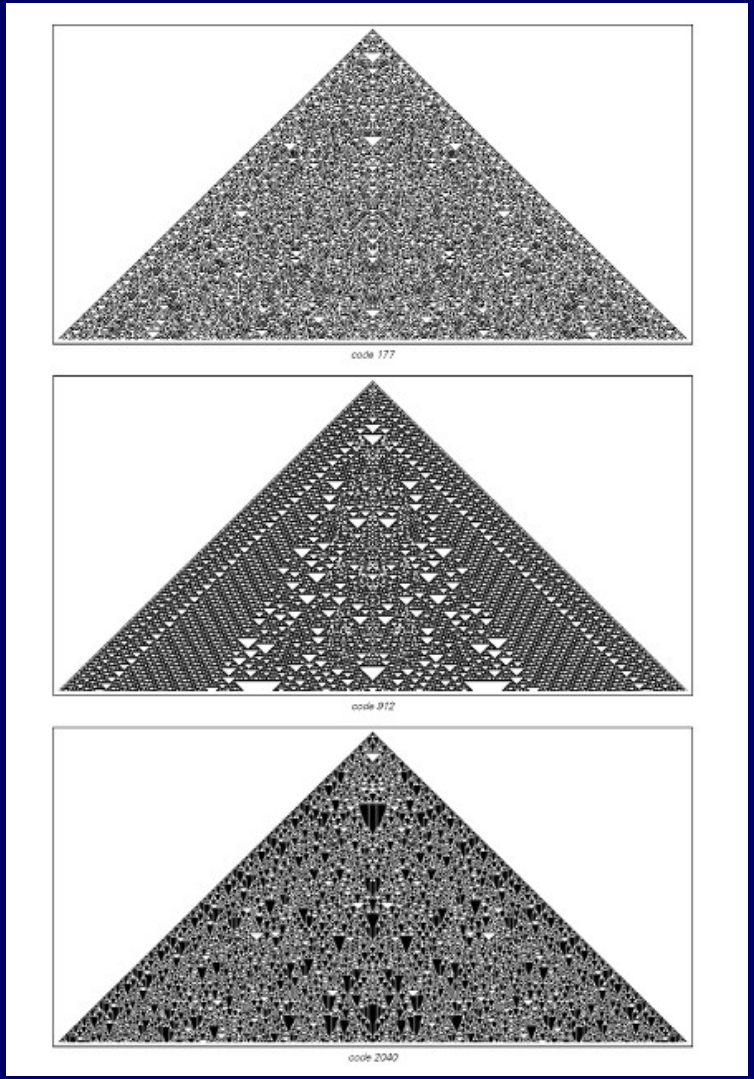
Ne
echno

A hypothesis

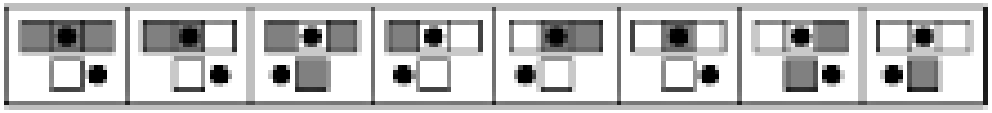
Cellular automata are an exception!

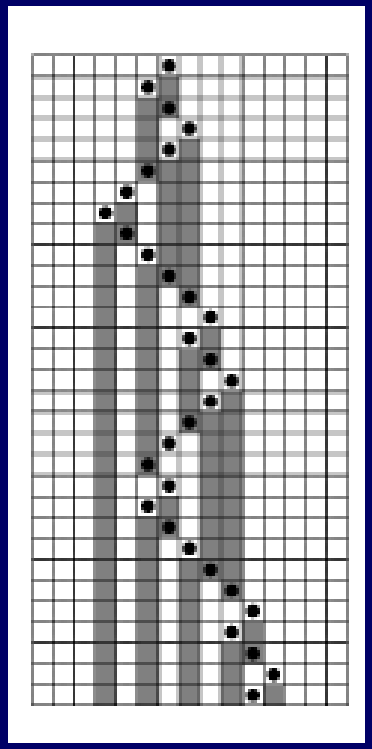
Other simple programs

3 colors



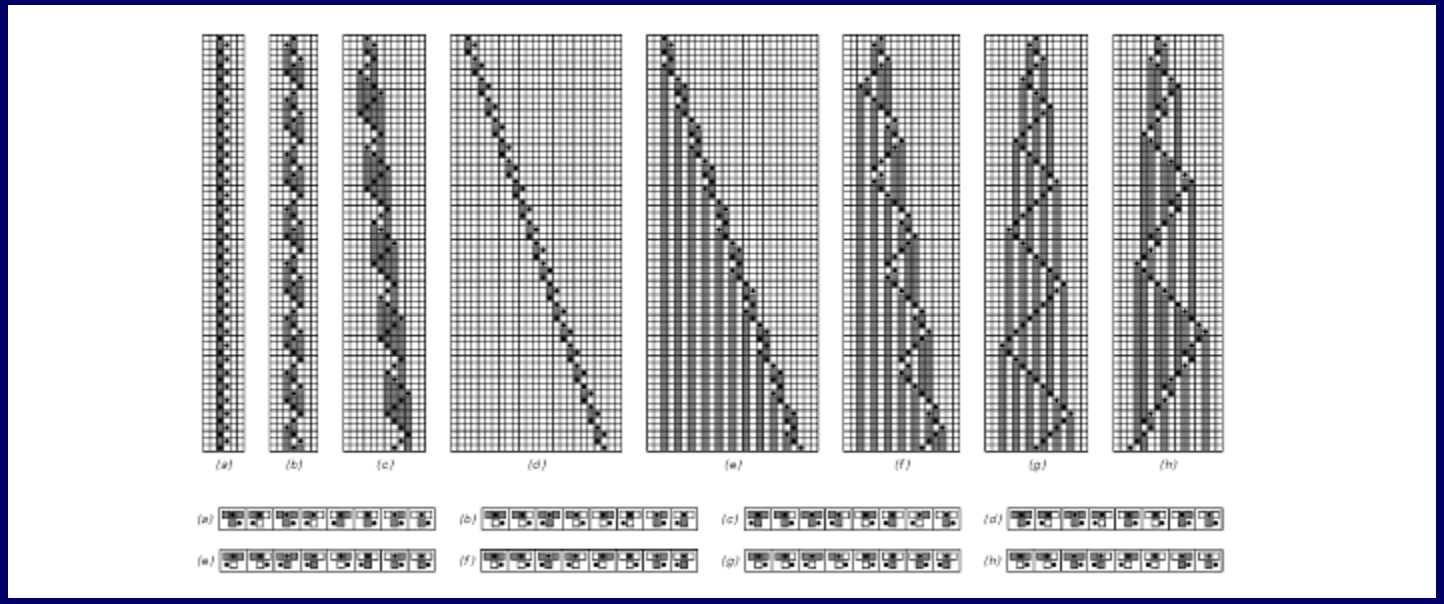
Being mobile

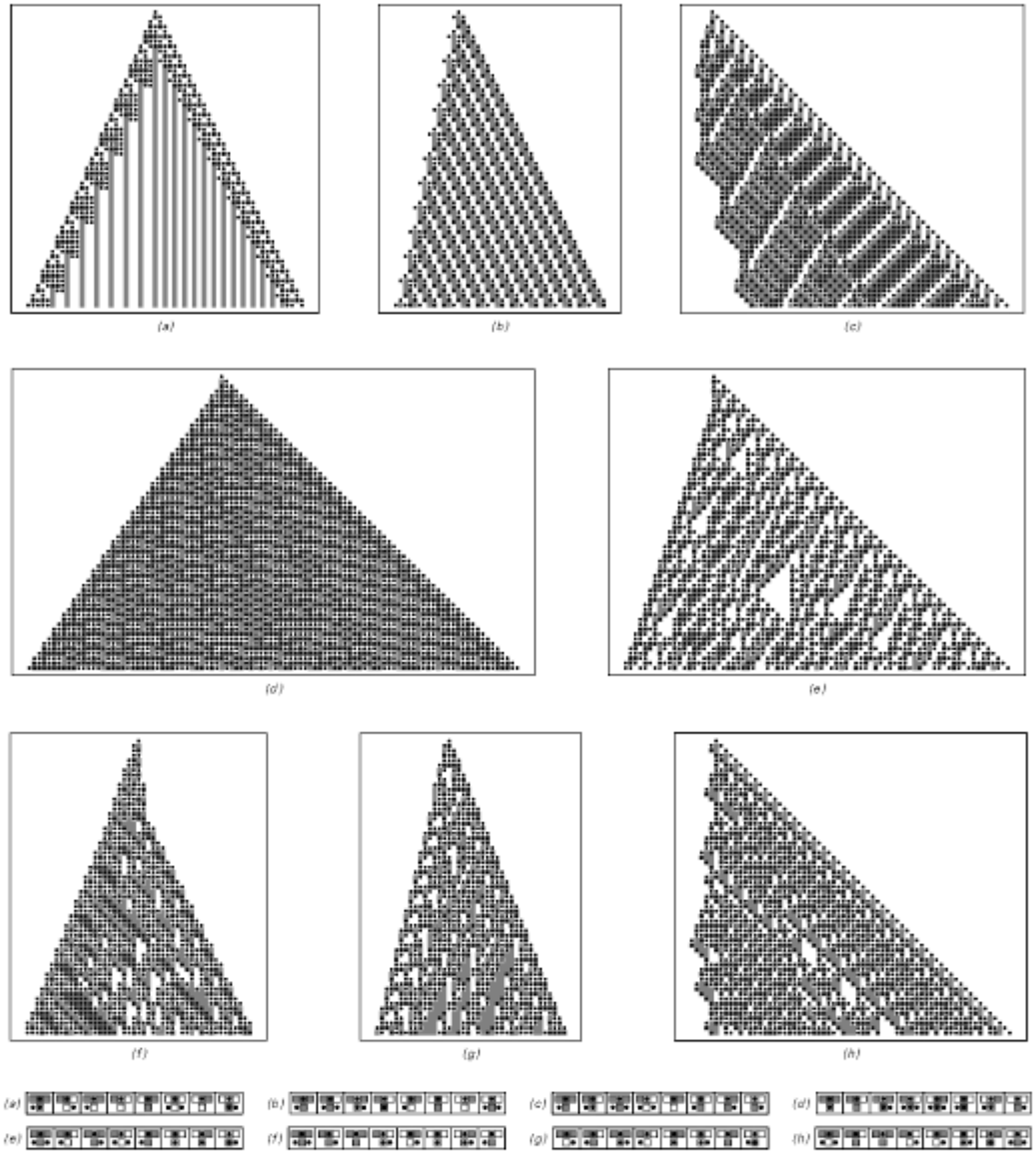




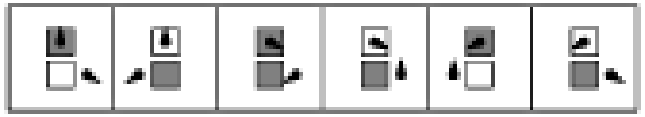
Not parallel!

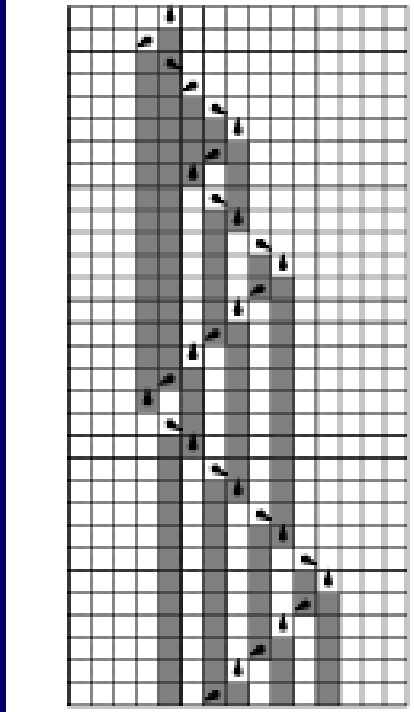
Mobile Automata

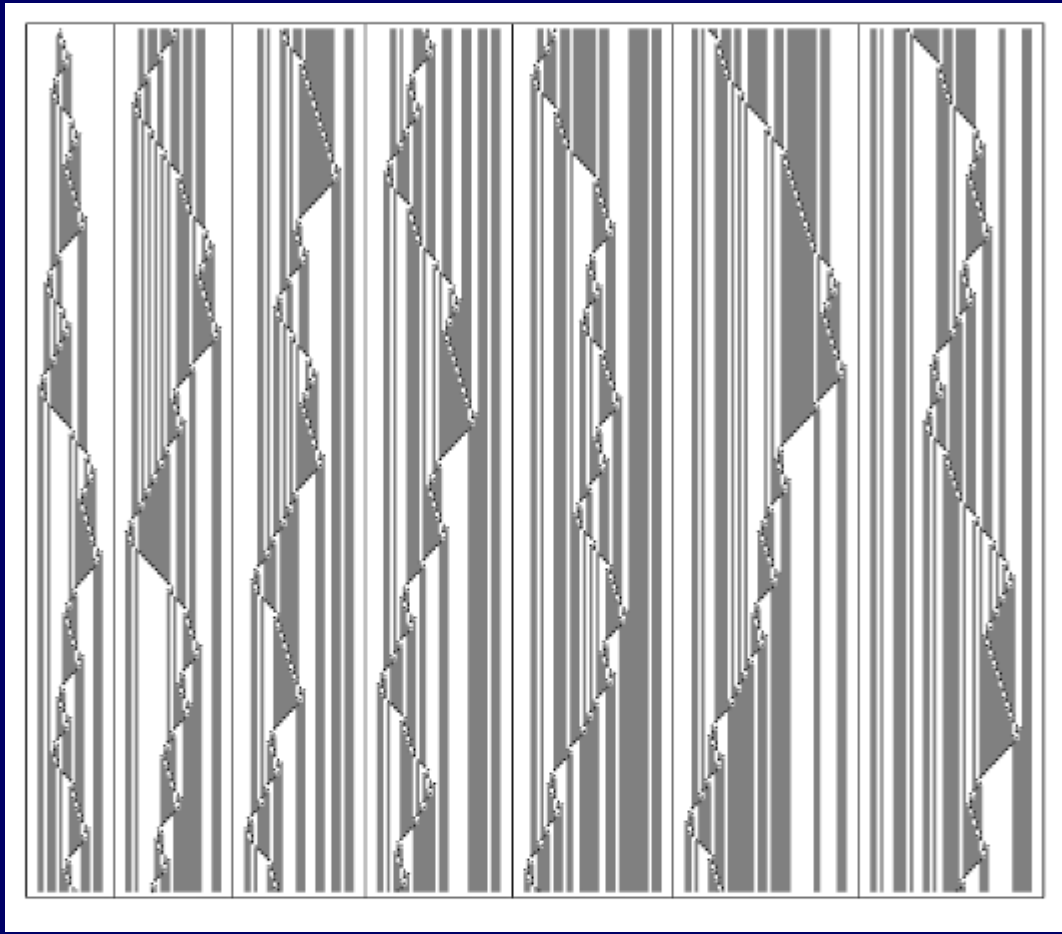



(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)

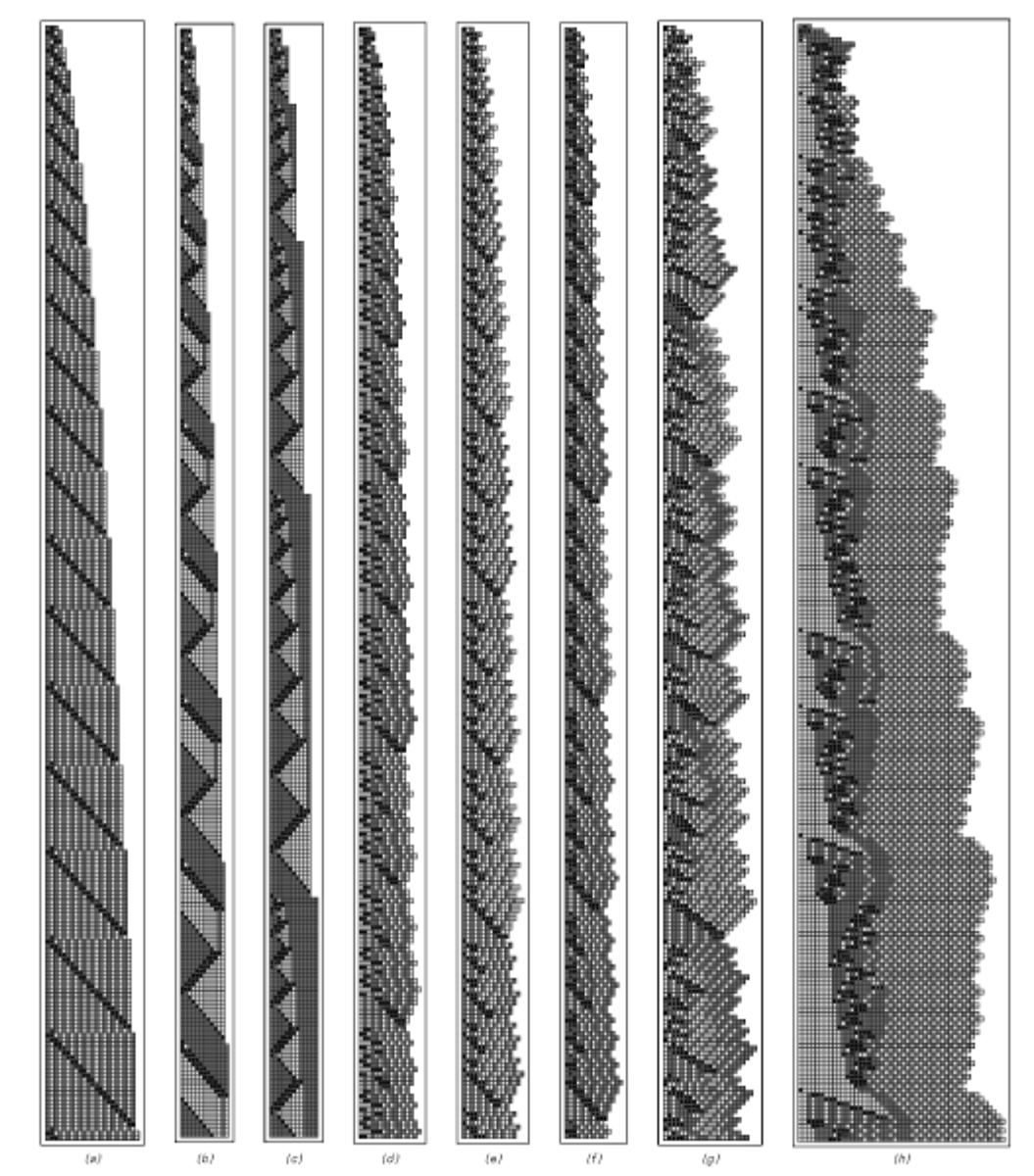
Turing machines







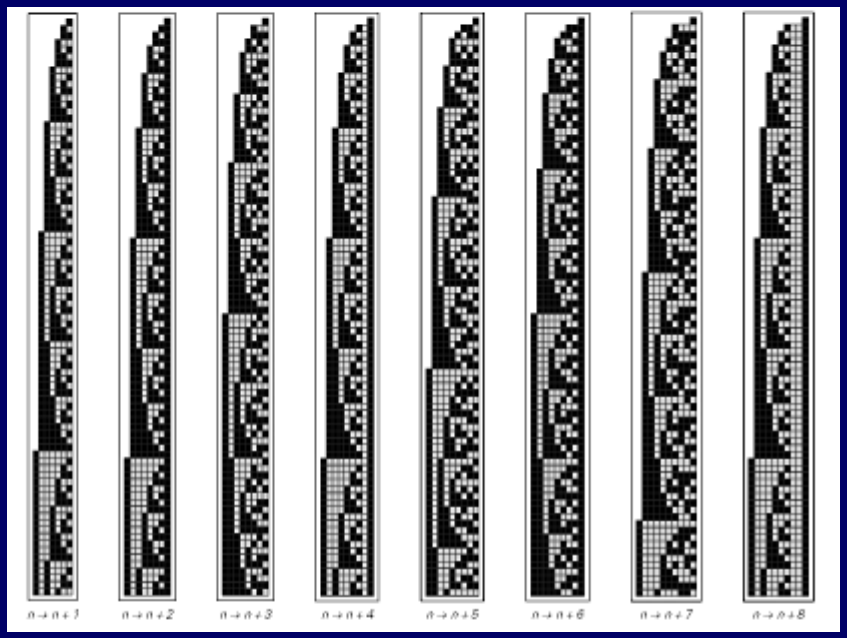
Other



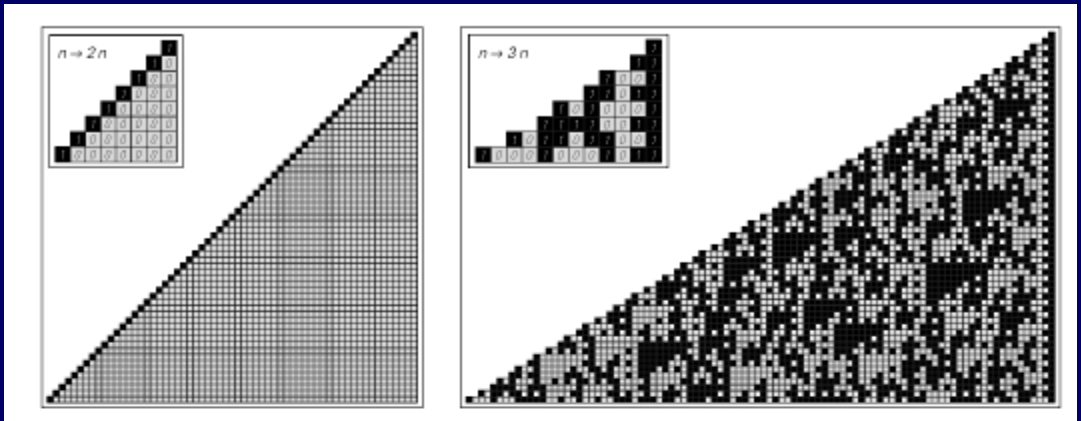
First conclusions

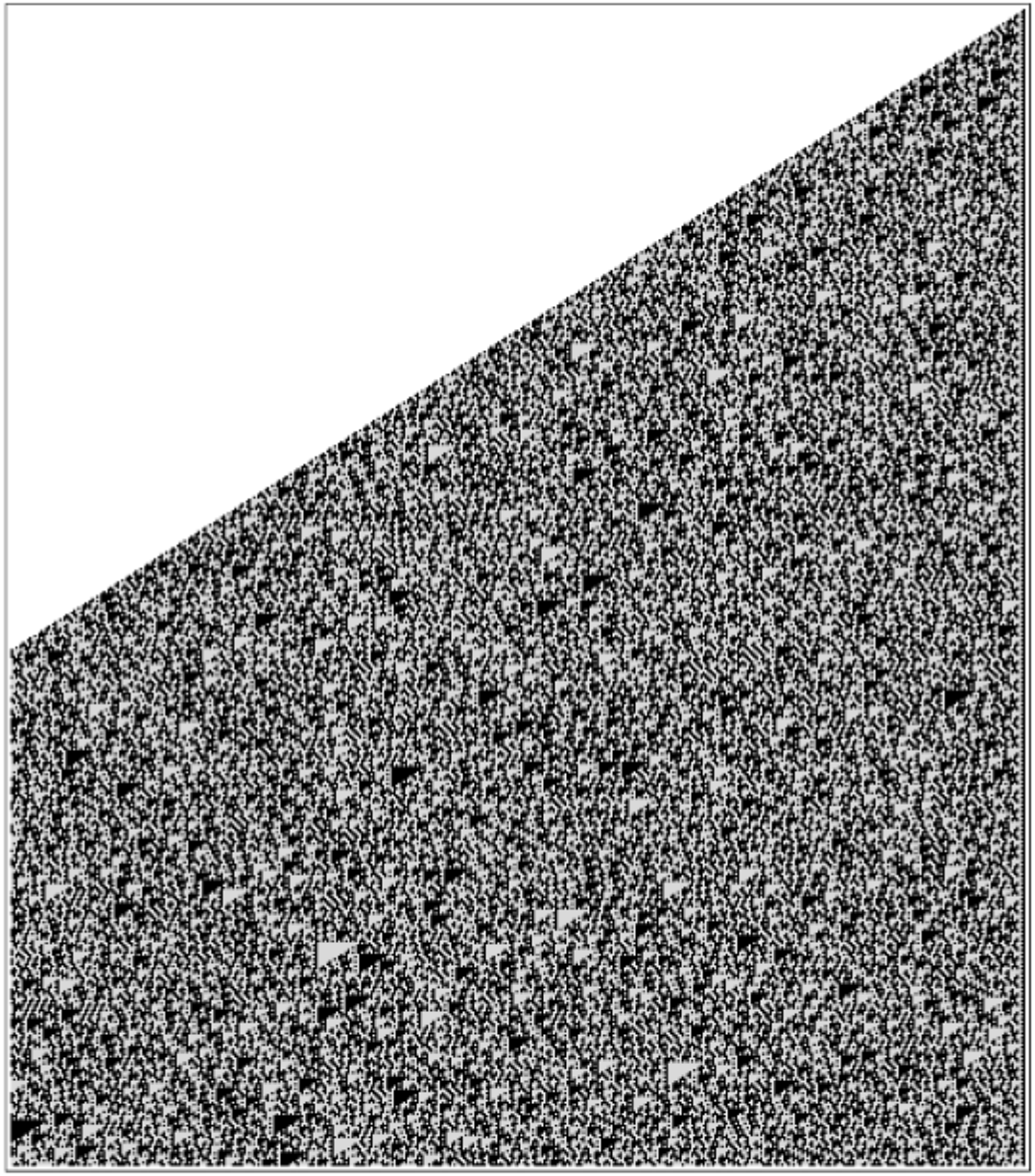
Phenomena of Complexity can be found in a variety of simple programs!

Systems based on Numbers



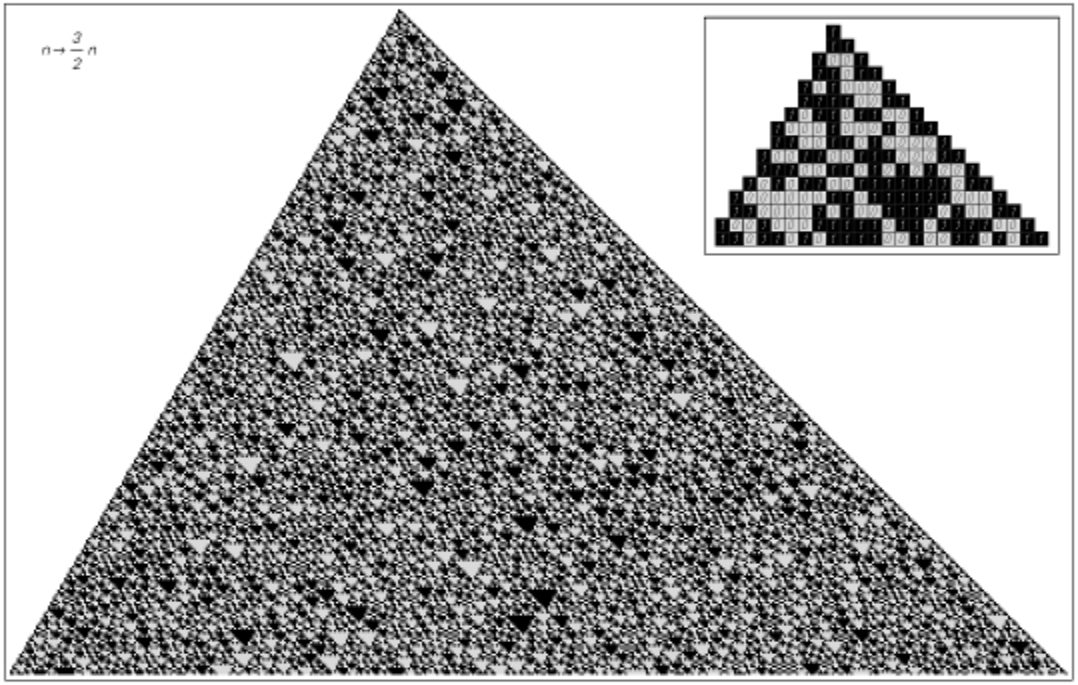
But...

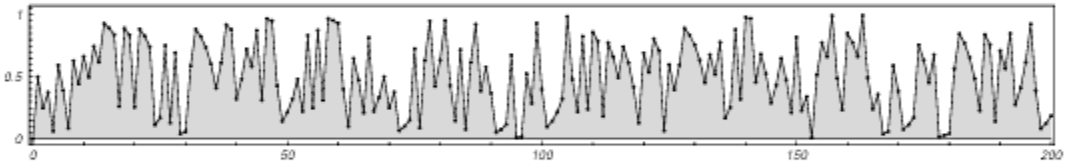




A hypothesis

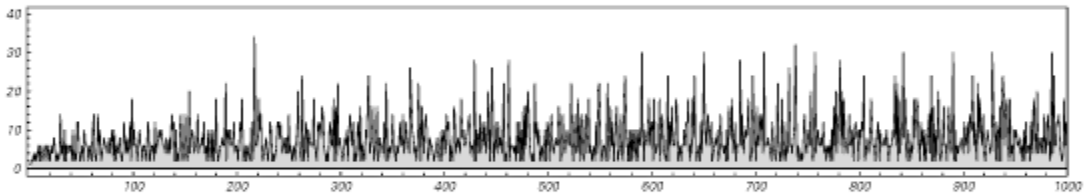
This is all because of the representation in base 2!





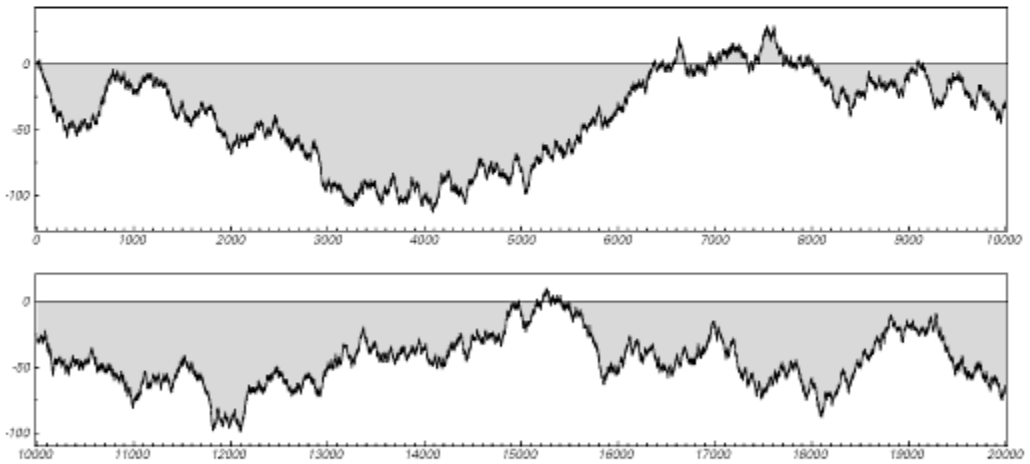
Sizes of the fractional parts of successive powers of 3/2. These sizes are completely independent of what base is used to represent the numbers. Only the dots are significant; the shading and lines between them are just included to make the plot easier to read.

Primes

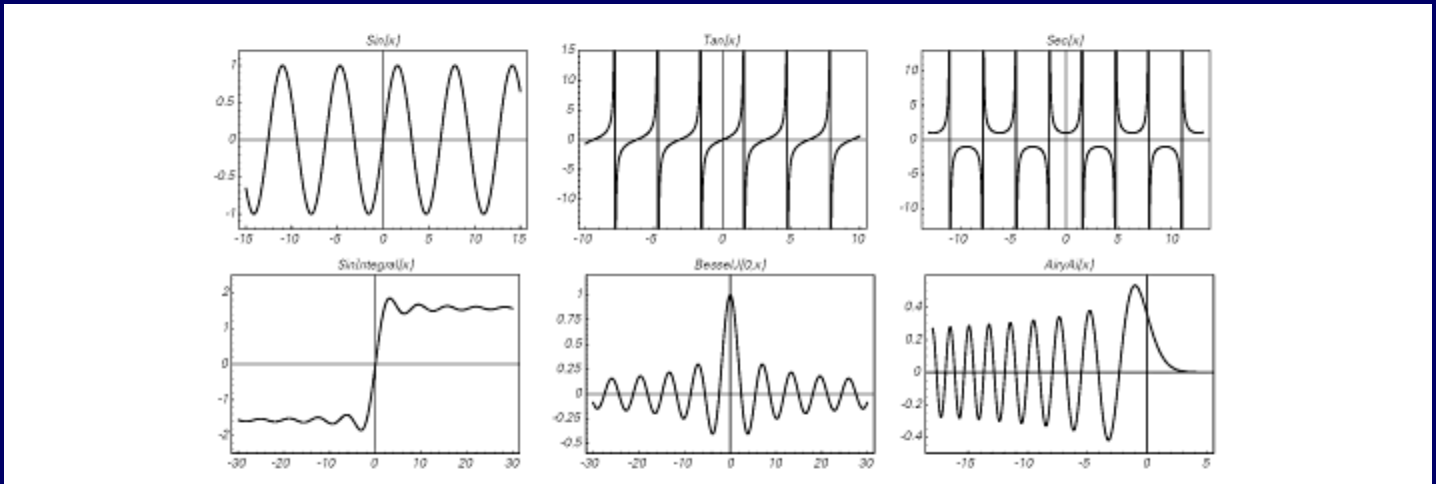


(f) Gaps between successive primes

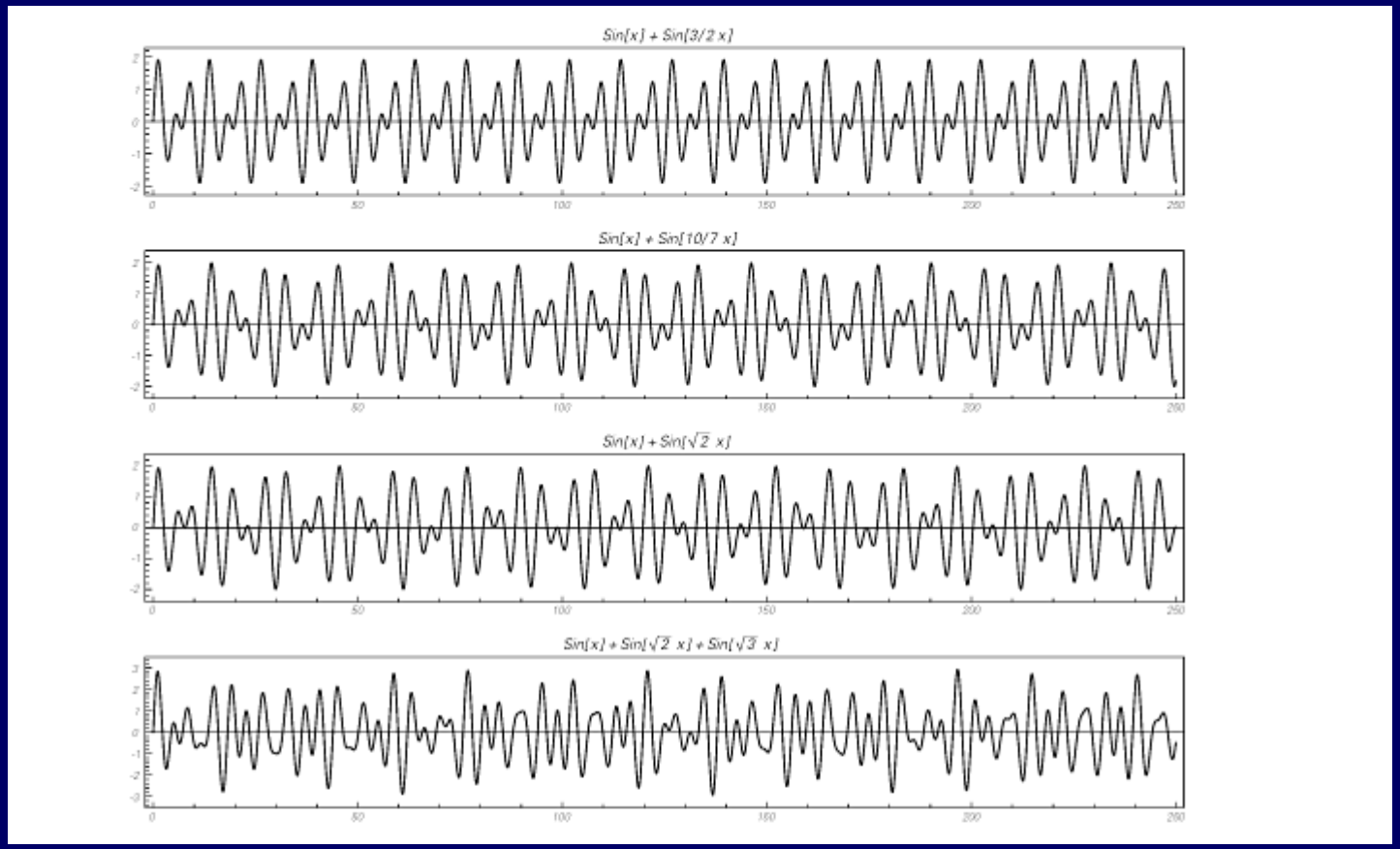
Pi


0
-50
-100
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
-50
-100
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000

Functions



Functions

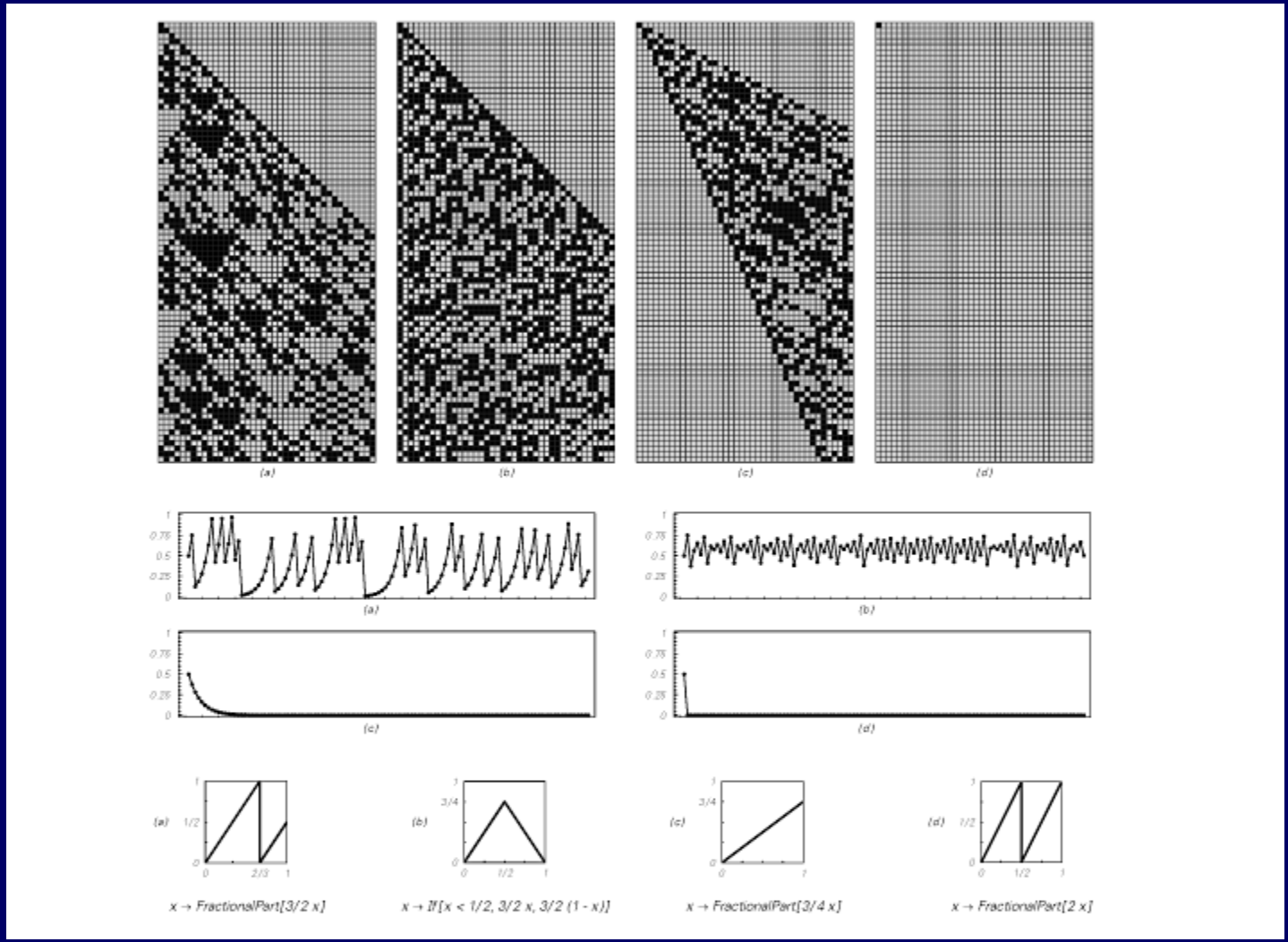


Conclusion

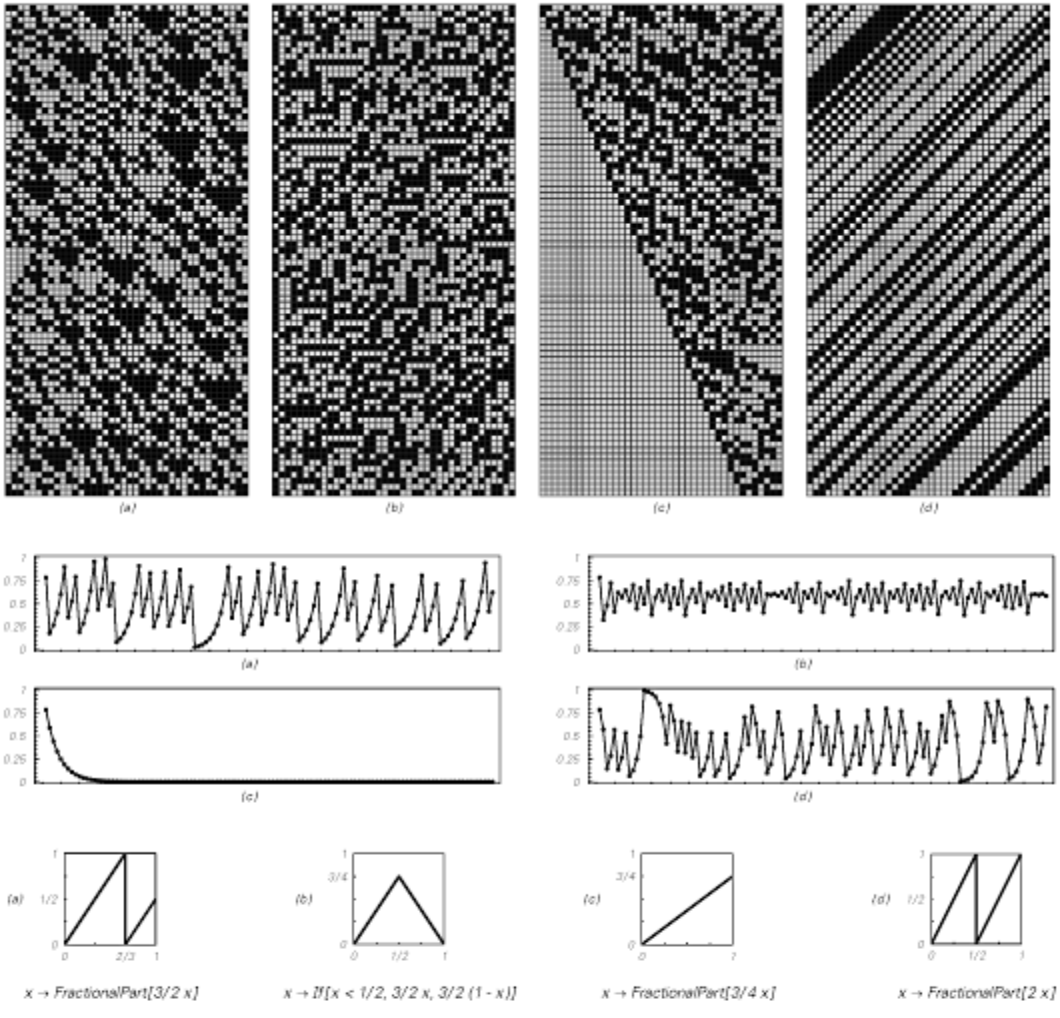
Other systems can exhibit the same behavior as cellular automatas

Chaos phenomena

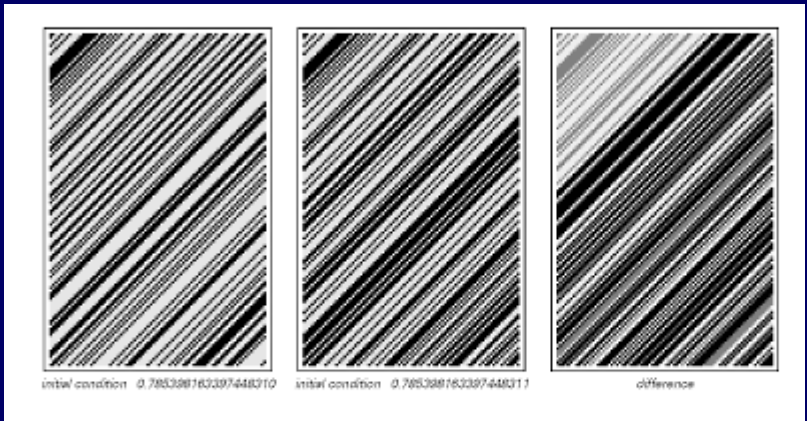
Start with 1/2

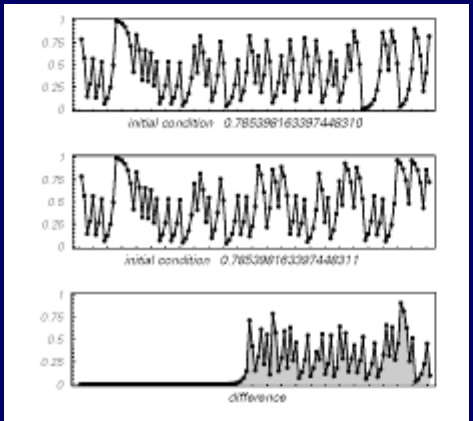


Start with random value

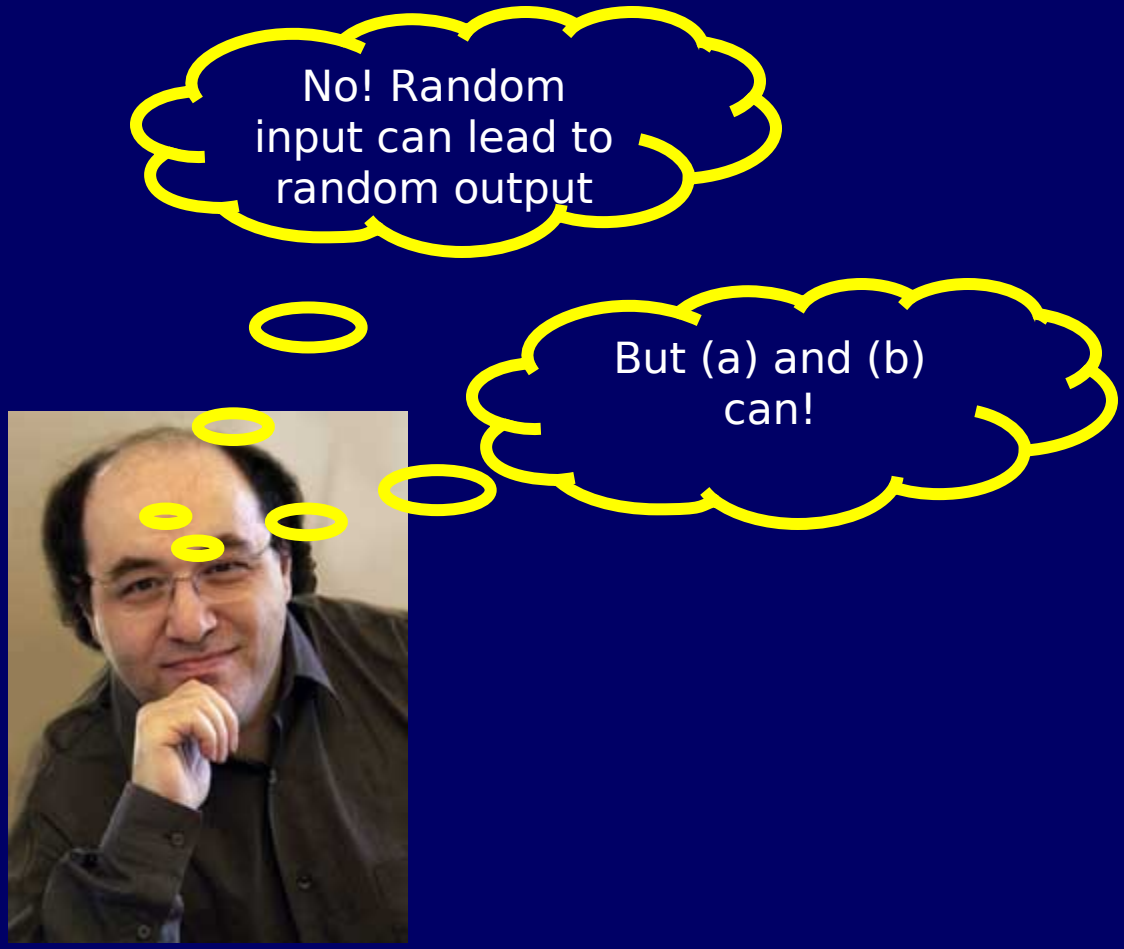


Tiny perturbations of the input

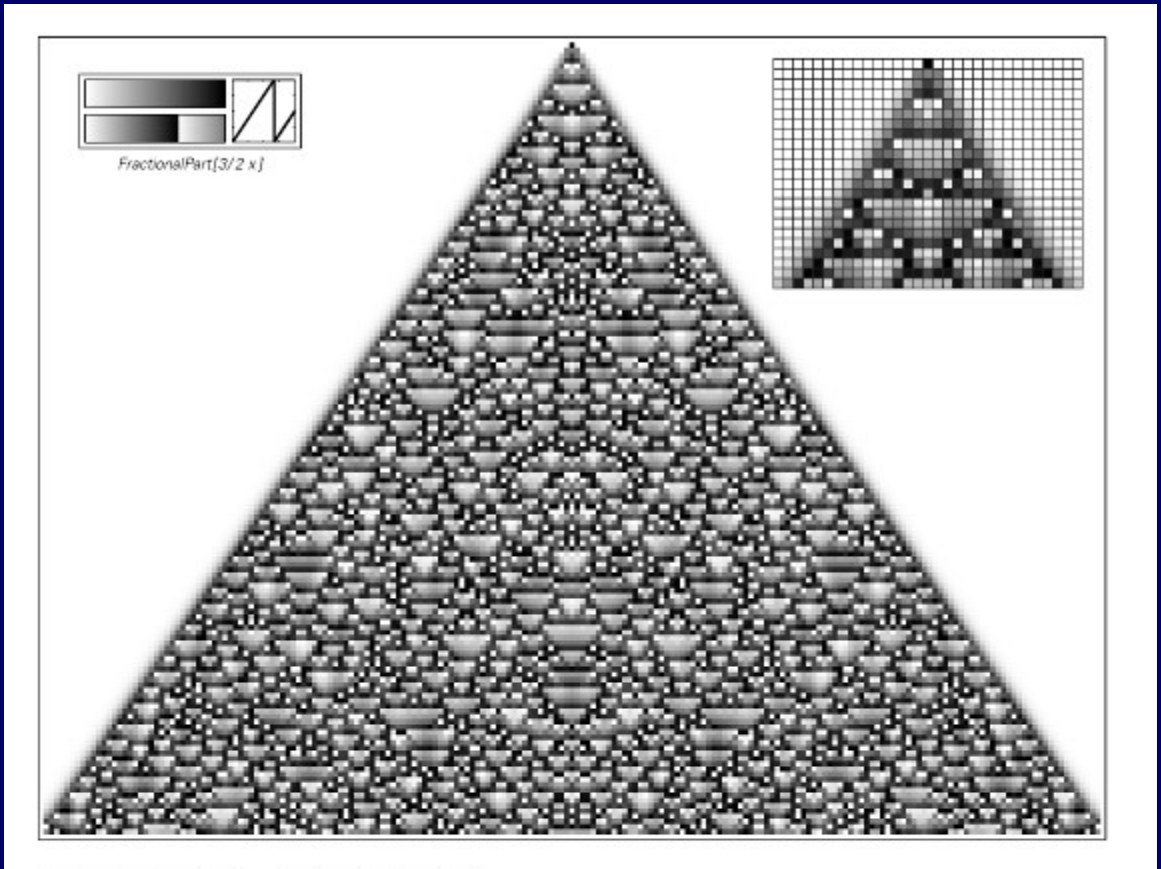




Can (d) create randomness?



Continuous



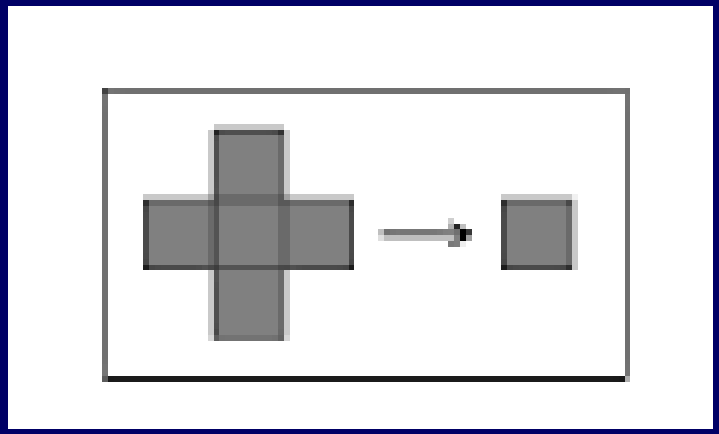
Conclusion

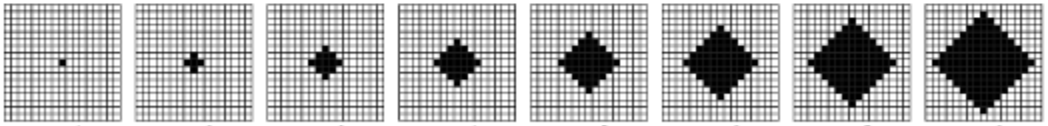
Same results in continuous and discrete

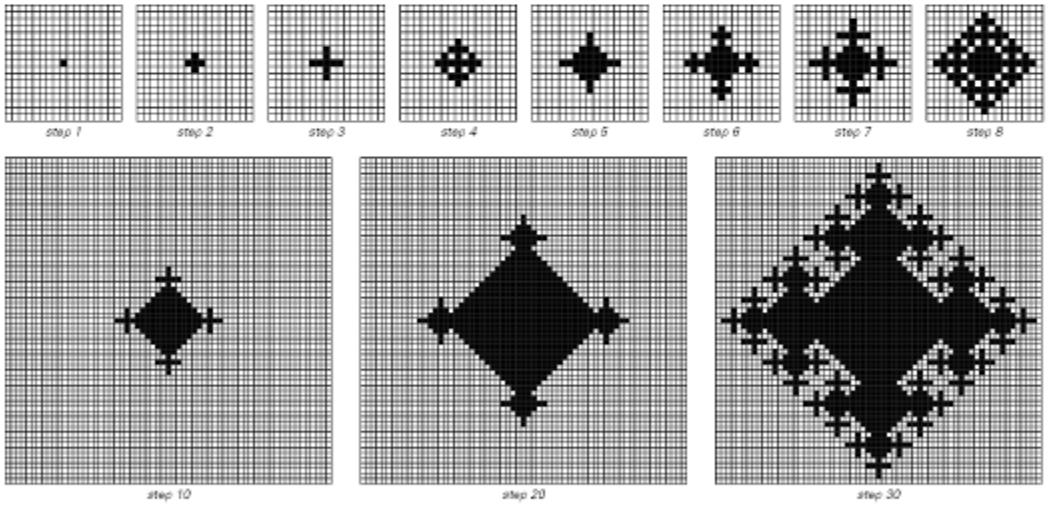
First discovered in discrete because easier to investigate

Continuous is like average of discrete

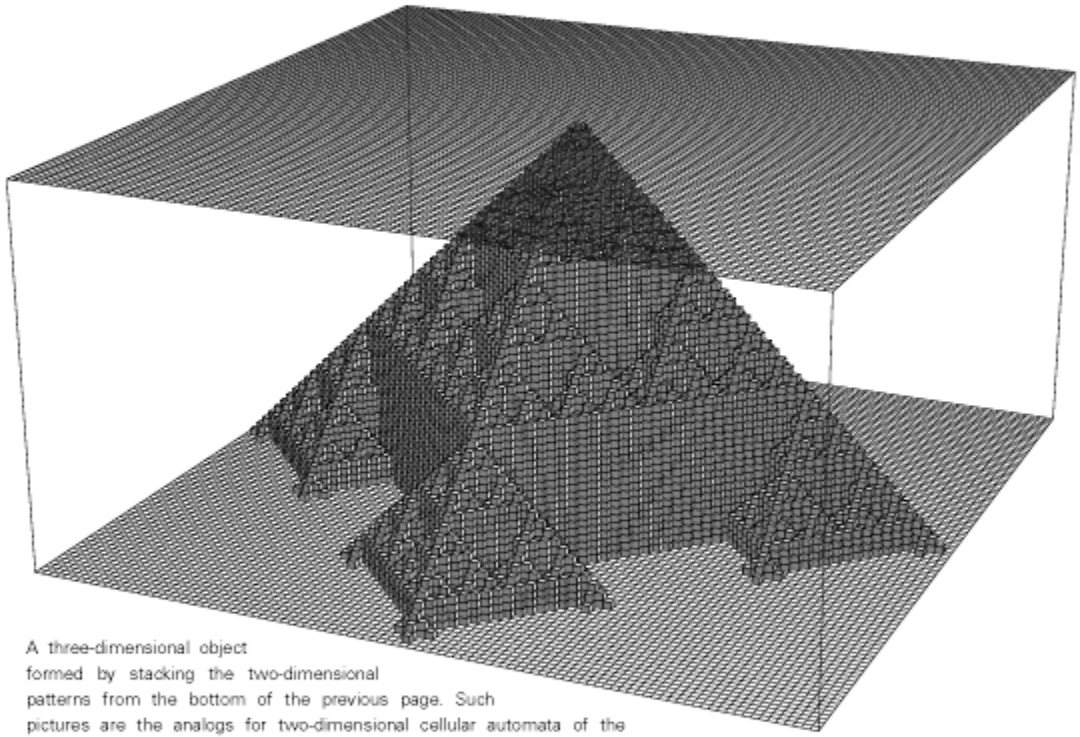
Dimensions



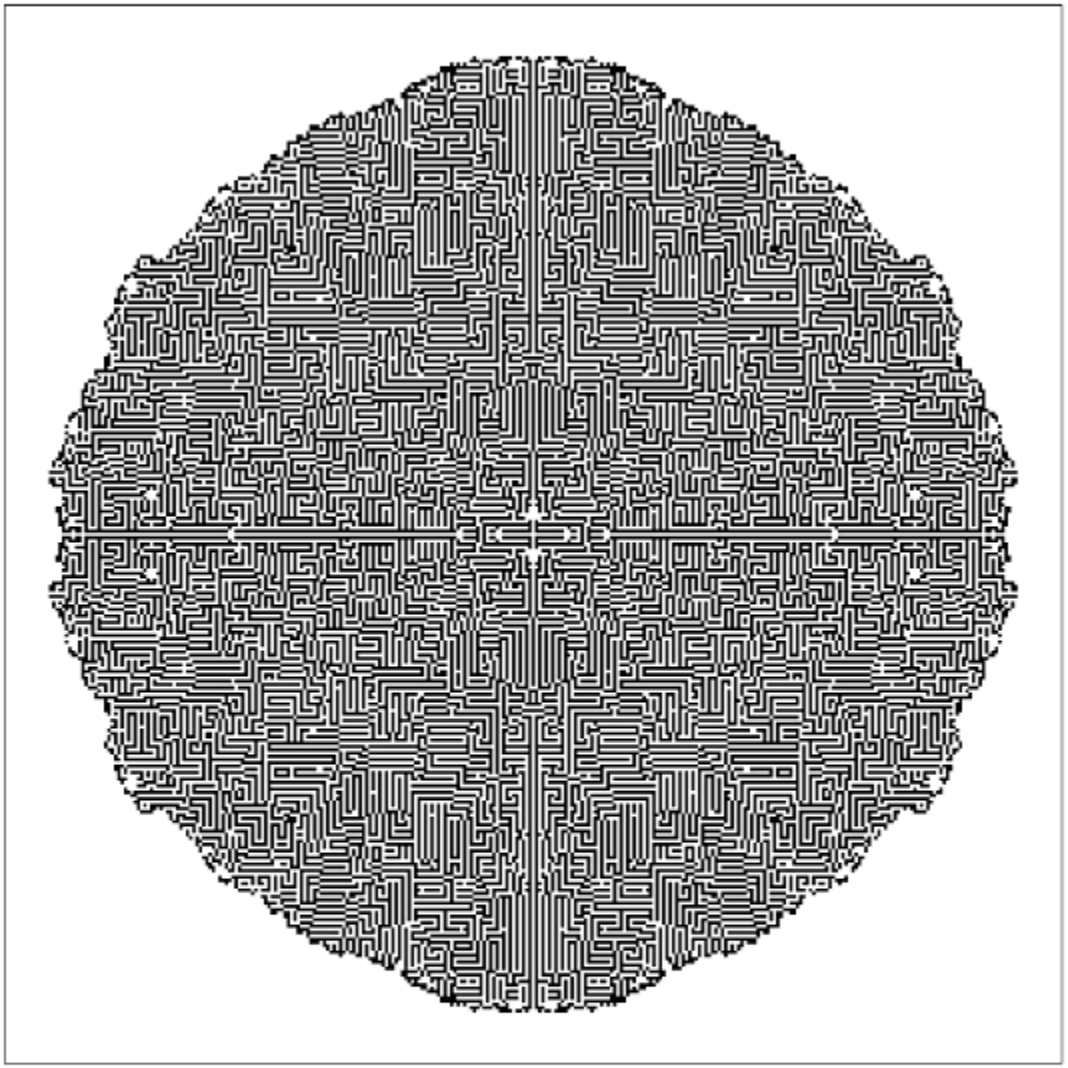




Steps in the evolution of a two-dimensional cellular automaton whose rule specifies that a particular cell should become black if exactly one or all four of its neighbors were black on the previous step, but should otherwise stay the same color. Starting with a single black cell, this rule yields an intricate, if very regular, pattern of growth. (In the numbering scheme on page 173, the rule is code 942.)



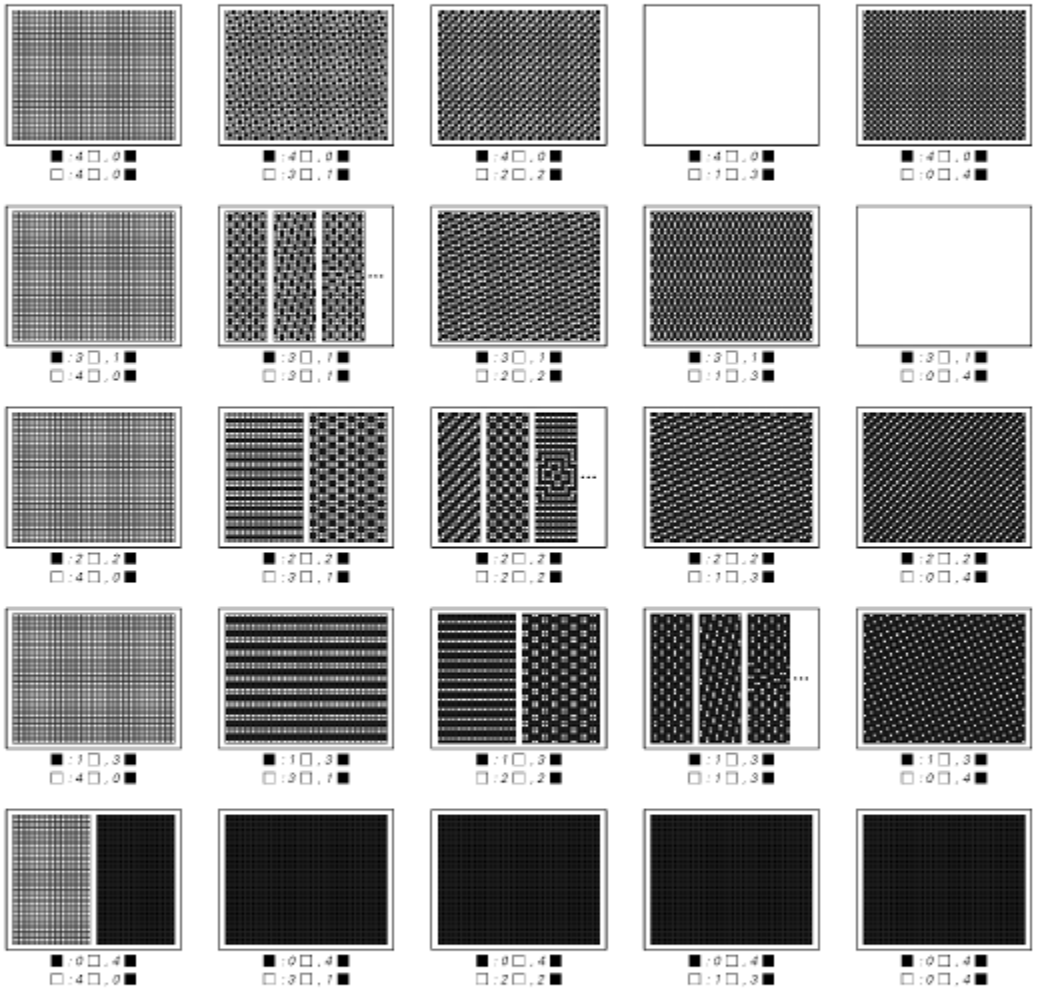
A three-dimensional object formed by stacking the two-dimensional patterns from the bottom of the previous page. Such pictures are the analogs for two-dimensional cellular automata of the two-dimensional pictures that I often generate for one-dimensional cellular automata.

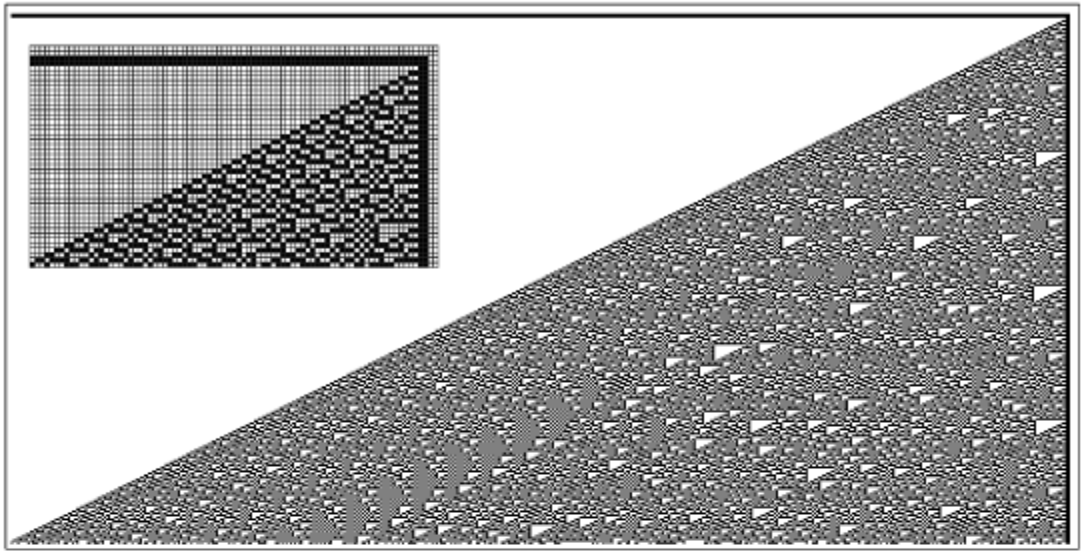


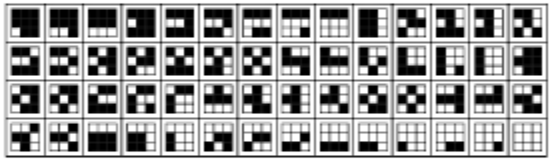
A cellular automaton that yields a pattern whose shape closely approximates a circle. The rule used is of the same kind as on the previous page, but now takes the center cell to become black only if it has exactly 3 black neighbors. If it has 1, 2 or 4 black neighbors then it stays the same color as it was before, and if it has 5 or more black neighbors, then it becomes white on the next step (code number 746). The initial condition consists of a row of 7 black cells, just as in the picture on the previous page. The pattern shown here is the result of 400 steps in the evolution of the system. After t steps, the radius of the approximate circle is about $0.37\,t$.

Constraints

Every cell must have a black and white neighbor


■:4 □:0 ■
□:4 □:0 ■
■:4 □:0 ■
□:3 □:1 ■
■:4 □:0 ■
□:2 □:2 ■
■:4 □:0 ■
□:1 □:3 ■
■:4 □:0 ■
□:0 □:4 ■
■:3 □:1 ■
□:4 □:0 ■
■:3 □:1 ■
□:3 □:1 ■
...
■:3 □:1 ■
□:2 □:2 ■
■:3 □:1 ■
□:1 □:3 ■
■:3 □:1 ■
□:0 □:4 ■
■:2 □:2 ■
□:4 □:0 ■
■:2 □:2 ■
□:3 □:1 ■
■:2 □:2 ■
□:2 □:2 ■
...
■:2 □:2 ■
□:1 □:3 ■
■:2 □:2 ■
□:0 □:4 ■
■:1 □:3 ■
□:4 □:0 ■
■:1 □:3 ■
□:3 □:1 ■
■:1 □:3 ■
□:2 □:2 ■
■:1 □:3 ■
□:1 □:3 ■
...
■:1 □:3 ■
□:0 □:4 ■
■:0 □:4 ■
□:4 □:0 ■
■:0 □:4 ■
□:3 □:1 ■
■:0 □:4 ■
□:2 □:2 ■
■:0 □:4 ■
□:1 □:3 ■
■:0 □:4 ■
□:0 □:4 ■





A system based on a constraint, in which a complex and largely random pattern is forced to occur. The constraint specifies that only the 56 3 × 3 templates shown at left can occur anywhere in the pattern, with the first template appearing at least once. The pattern required to satisfy this constraint corresponds to a shifted version of the one generated by the evolution of the rule 30 elementary one-dimensional cellular automaton.

Constraints

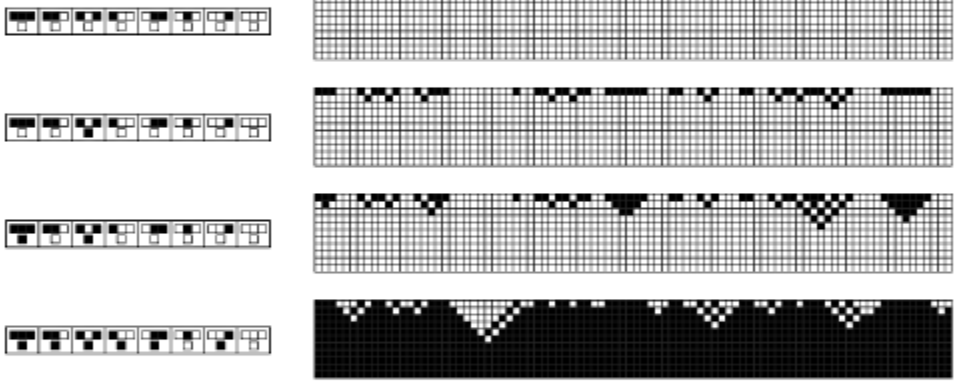
Only complicated constraints yield complicated output!

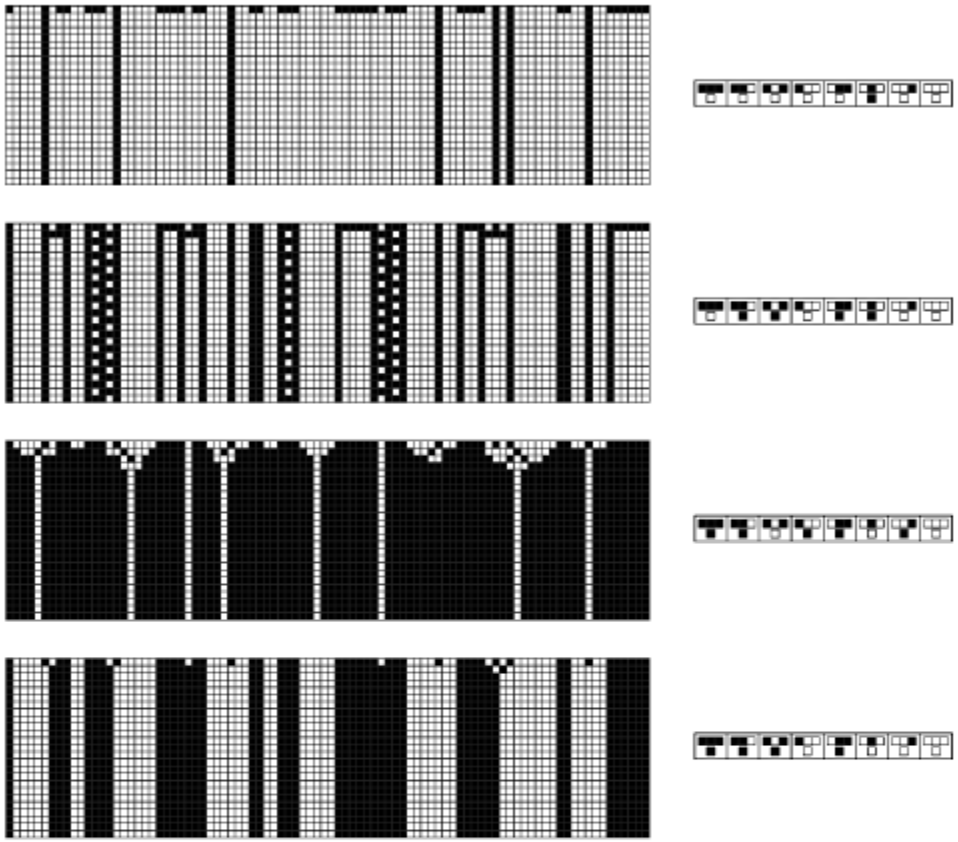
Constraint=Equation

Traditional science concentrates on equations!

A hypothesis

Starting from randomness no order can emerge

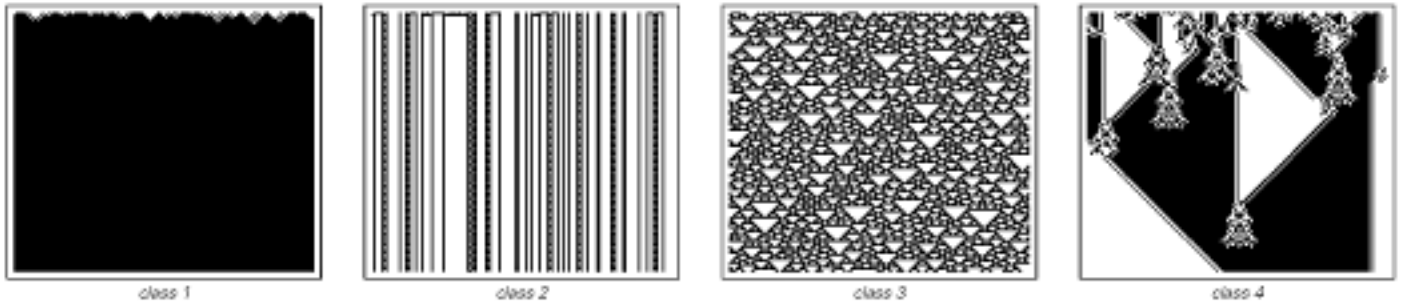




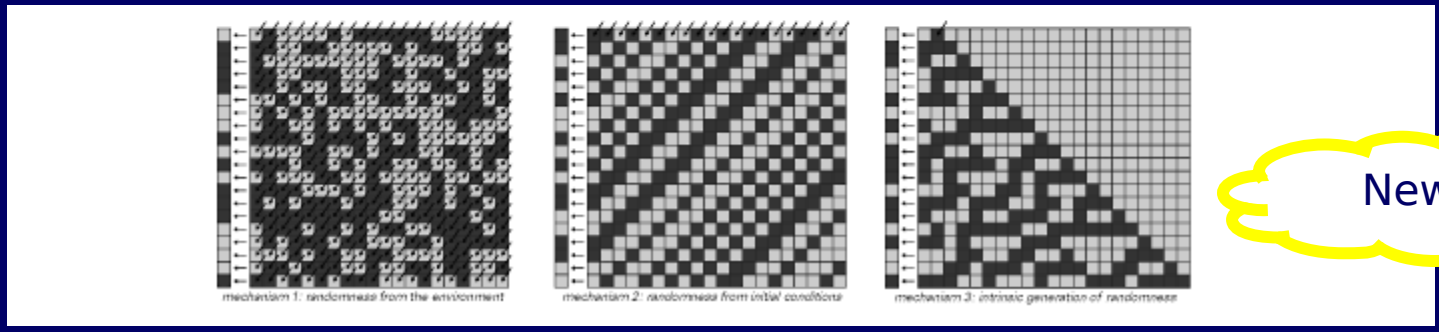
Conclusion

Order can emerge from randomness

Four classes of behavior



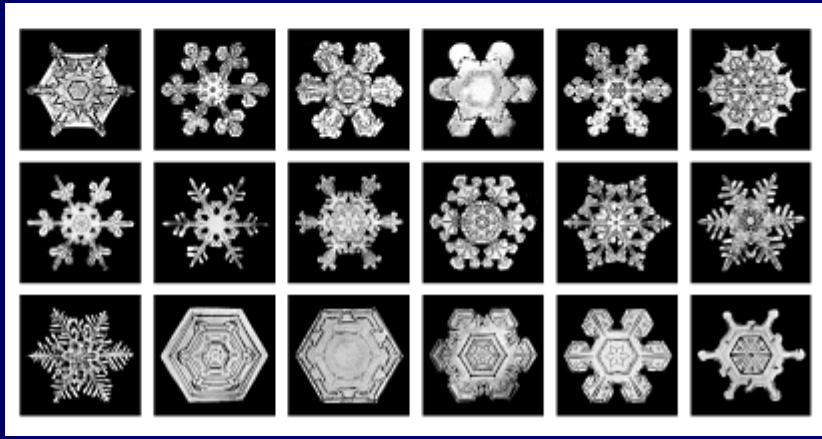
Sources of randomness

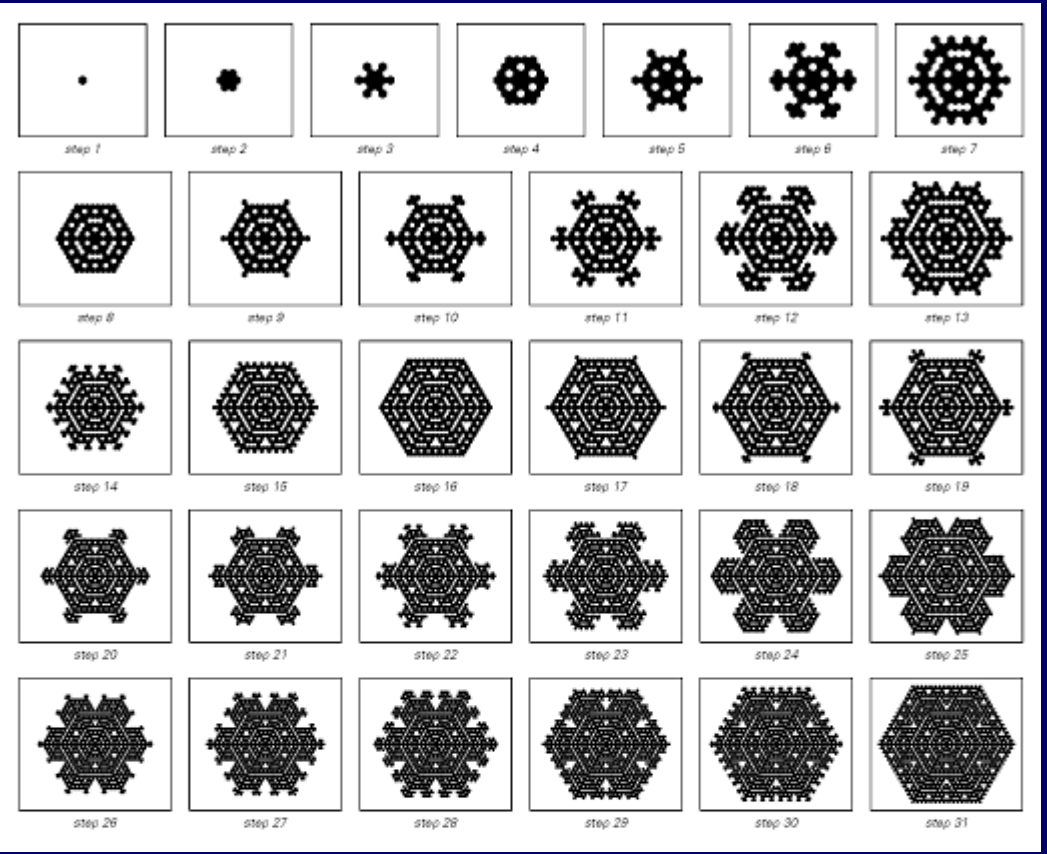


New!

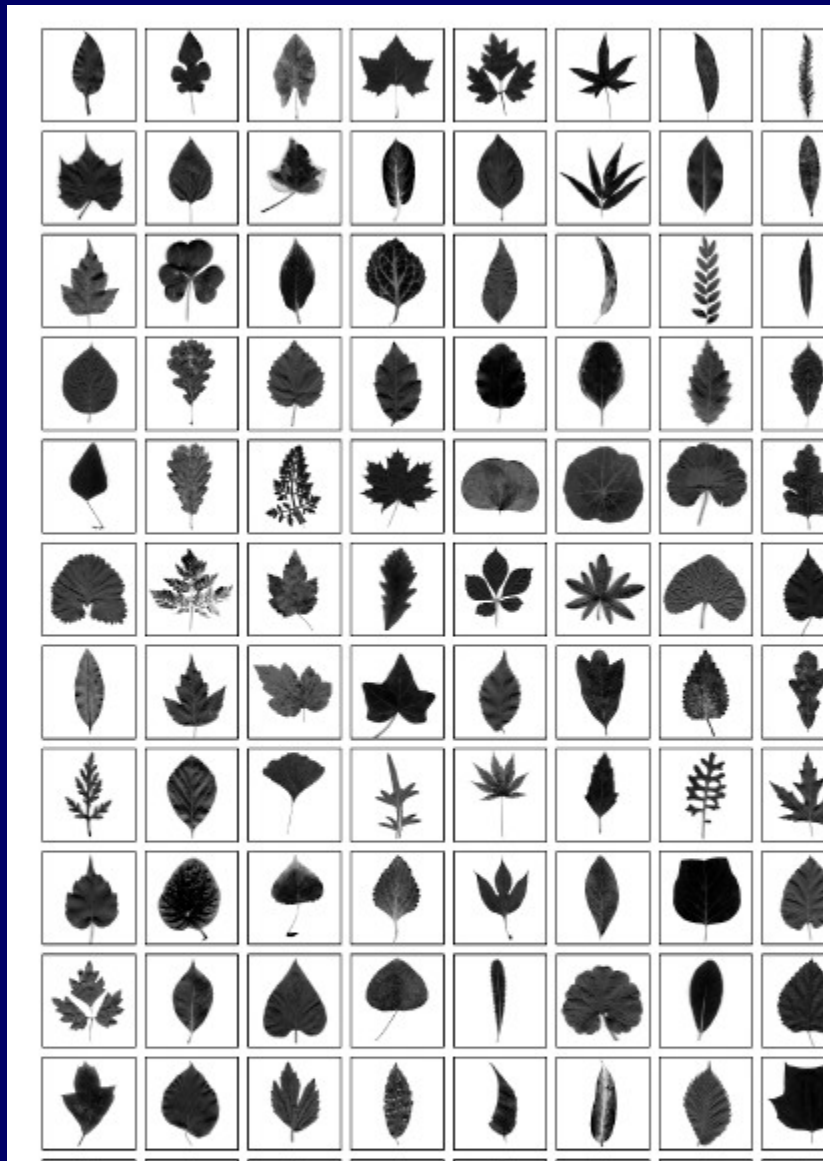
Is this useful?

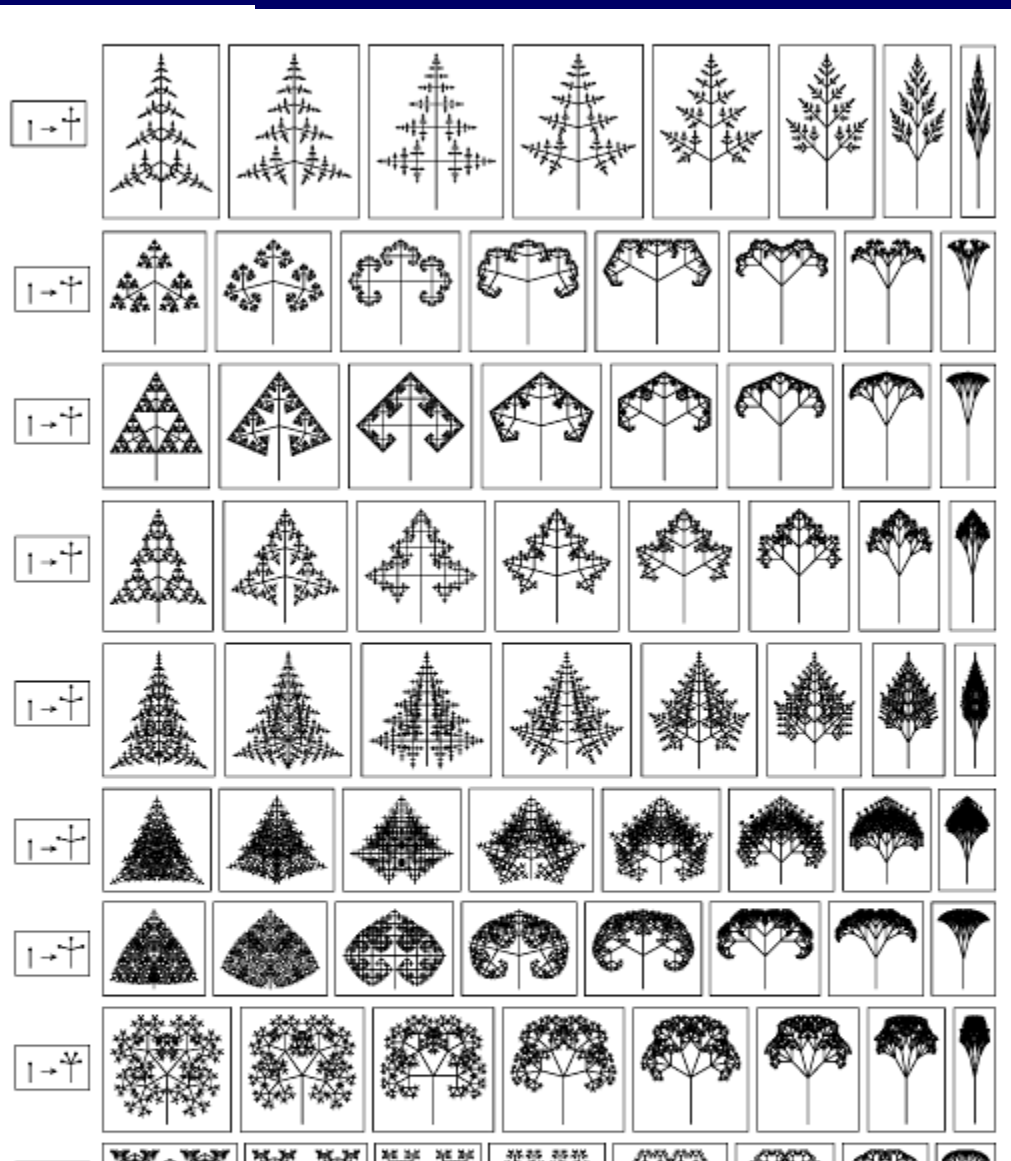
Snow flakes



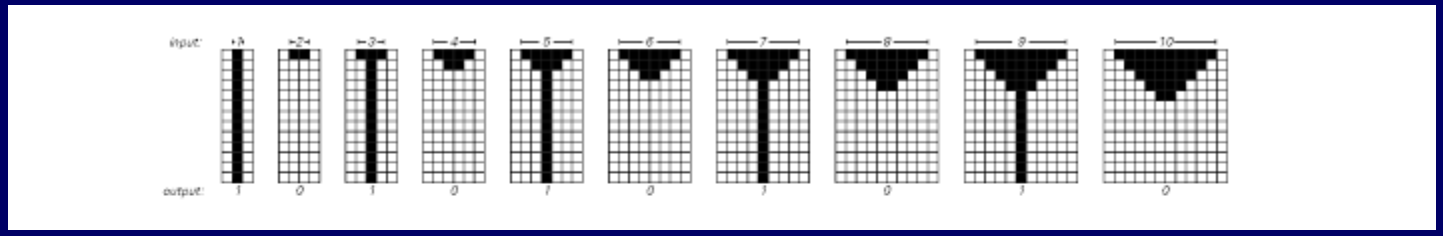

step 1
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
step 12
step 13
step 14
step 15
step 16
step 17
step 18
step 19
step 20
step 21
step 22
step 23
step 24
step 25
step 26
step 27
step 28
step 29
step 30
step 31

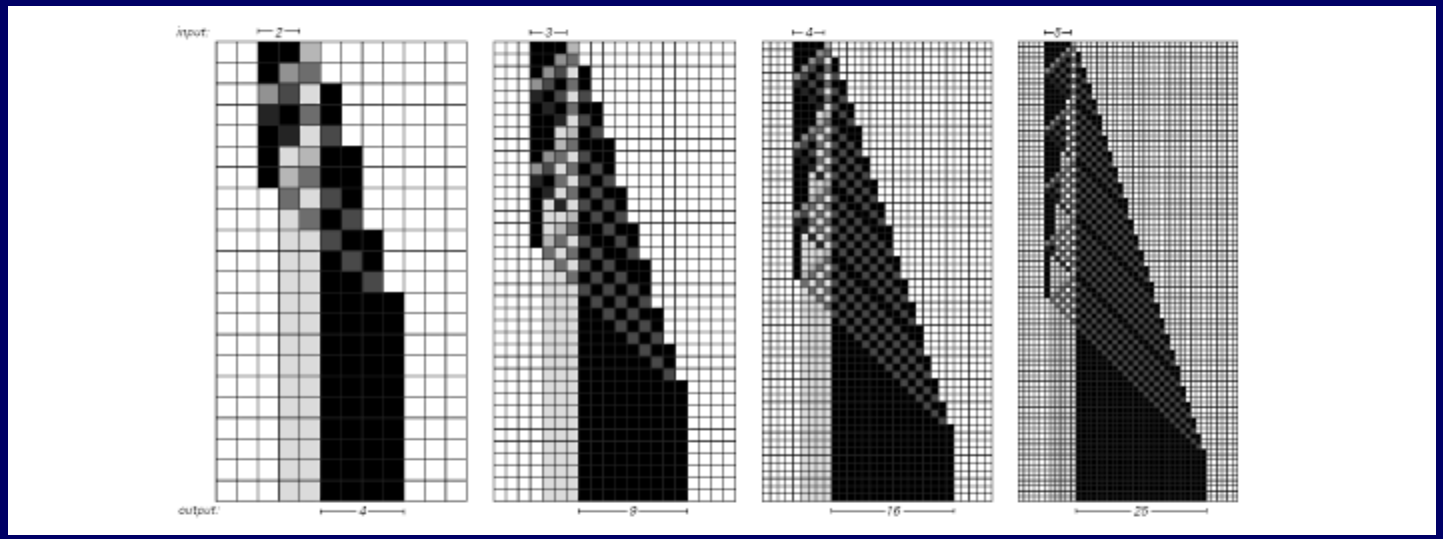
Growth of plants





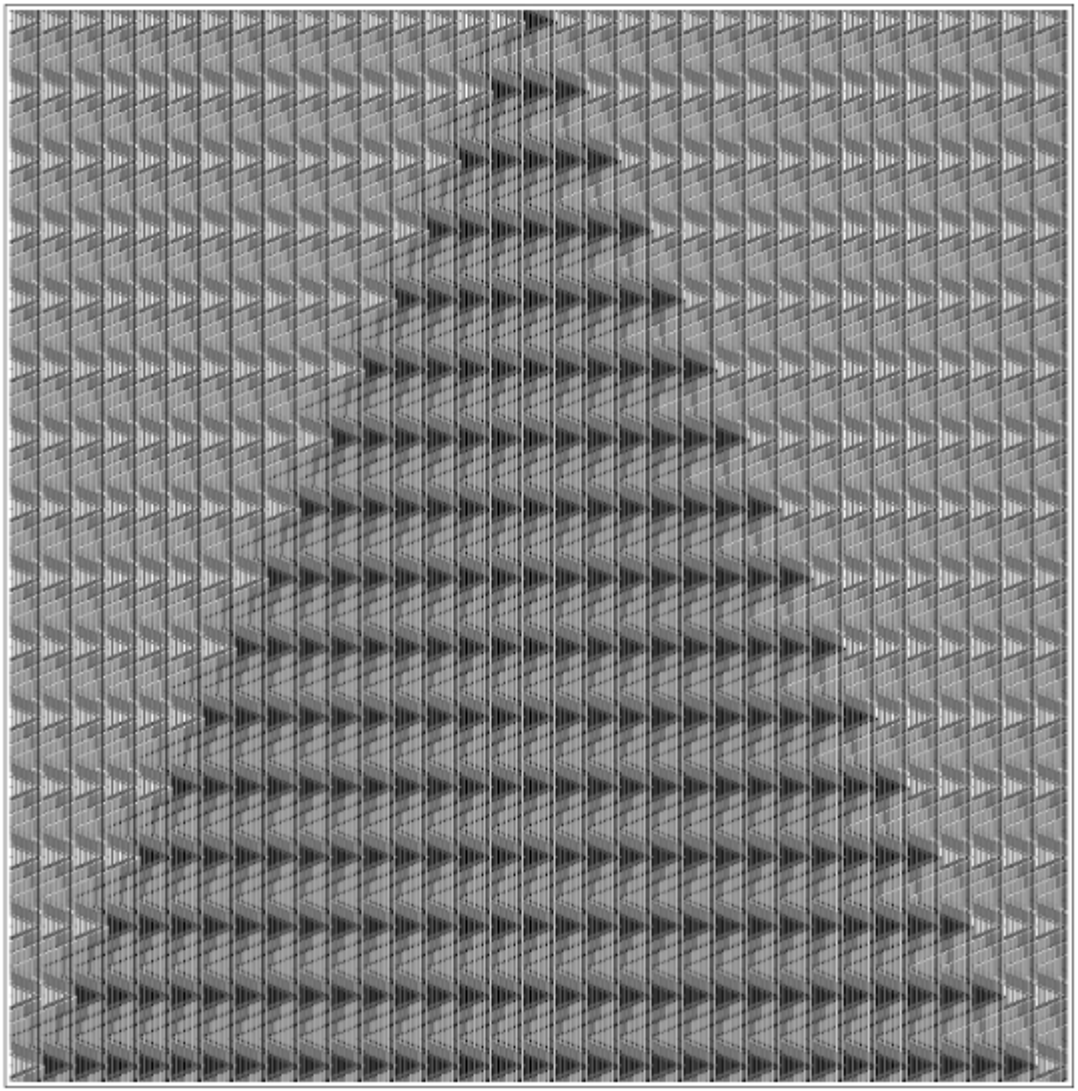
Computation

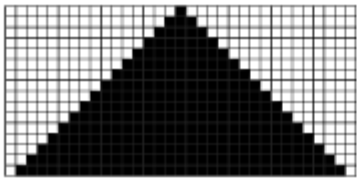




Is there a universal cellular automata?

Yes!





The universal cellular automaton emulating elementary rule 254. Each cell in rule 254 is represented by a block of 20 cells in the universal cellular automaton. Each of these blocks encodes both the color of the cell it represents, and the rule for updating this color.

rule 254

Take home message

Thinking in terms of programs instead of equations can lead to new insights

A simple program could produce all the complexity we see.

...Go and find it!

REFERENCES

Coxeter, H. S. M. ``The Golden Section, Phyllotaxis, and Wythoff's Game.'' *Scripta Mathematica* **19**, 135-143, 1953.

"Recounting Fibonacci and Lucas Identities" by Arthur T. Benjamin and Jennifer J. Quinn, College Mathematics Journal, Vol. 30(5): 359--366, 1999.

Stephen Wolfram, “A New Kind of Science”, 2002