Is Pascal's Triangle a Fractal ?

Ankith A Das

The University of Sydney

May 17, 2019

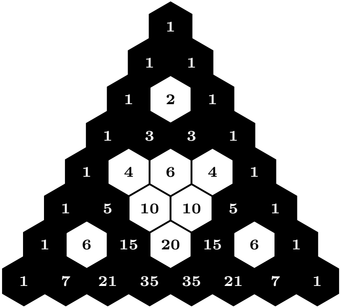
Pascal's Triangle

$$
\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1
\end{array}
$$

- One of the earliest mentions was in a Chinese document at around 1303 AD
- Looks pretty innocent right?

Pascal's Triangle

- It is observed by coloring
    1. all odd numbers black
    2. all even numbers white



Modified code:https://tex.stackexchange.com/questions/198887/how-can-i-draw-pascals-triangle-with-some-its-properties

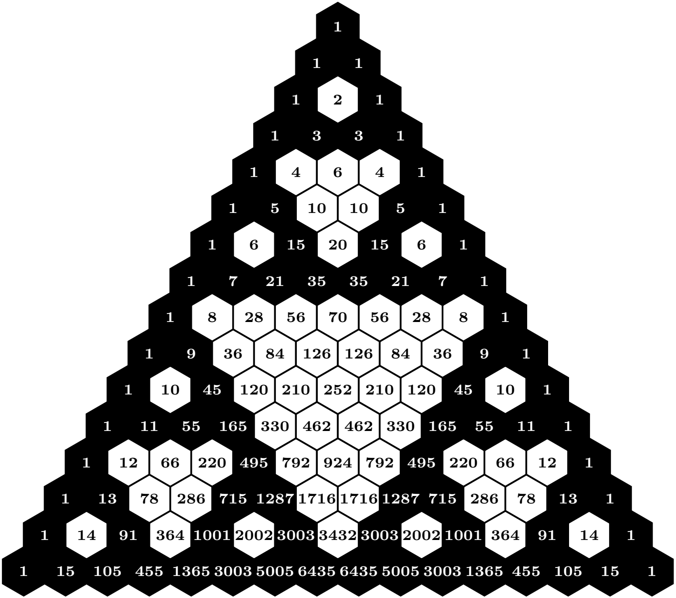


Figure: A bigger picture

Pascal's Triangle

- So how can we formulate this pattern ? We can begin by looking at binomial coefficients

$$\begin{aligned}
(1+x)^0 &= \quad\quad 1 \\
(1+x)^1 &= \quad 1 + 1x \\
(1+x)^2 &= 1 + 2x + 1x^2 \\
&\vdots \\
(1+x)^n &= a_0 + a_1 x + \cdots + a_n x^n
\end{aligned}$$

where coefficients are given by

$$a_k = \binom{n}{k} = \frac{n!}{(n-k)!k!}, \quad 0 \leq k \leq n$$

Pascal's Triangle

- Checking if a binomial coefficient is odd or even by computing $\frac{n!}{(n-k)!k!}$ is a bad idea

$$\begin{aligned}50! =&30414093201713378043612608166064768844377\\&64156896051200000000000\end{aligned}$$

- Even if we use the recursive formula

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$$

$$\binom{40}{20} = 137846528820 > 2^{32}$$

which is a big number (computer has fixed precision)

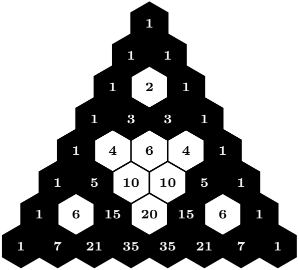
- Fortunately, we don't need to compute these large numbers

The Macroscopic view

Pascal's Triangle

- In order to color a row in the triangle, we only need to know the color of the previous row

$\binom{n+1}{k} =$	$\binom{n}{k-1} +$	$\binom{n}{k}$
even	even	even
odd	odd	even
odd	even	odd
even	odd	odd

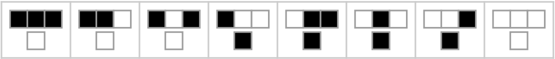


- The idea of looking at neighbor cells to determine the state of a new cell is the core idea of cellular automata

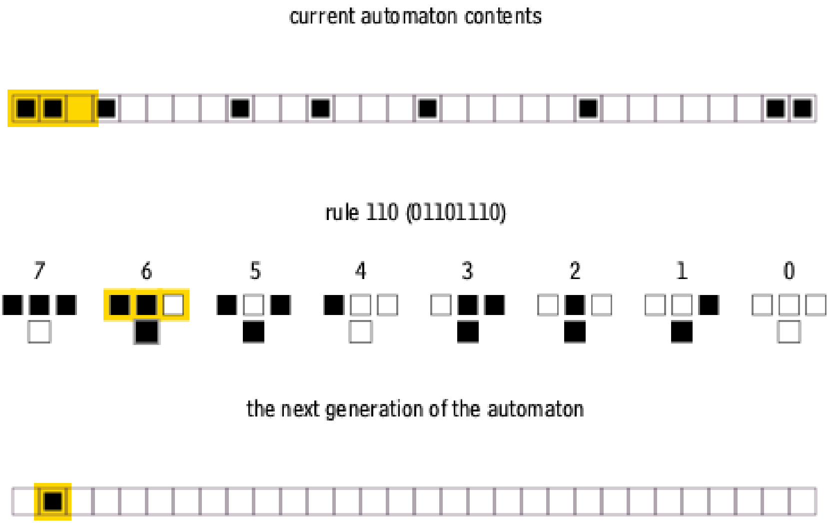
Cellular Automata

- Consists of a grid of cells, each one with a finite number of states, like 1 or 0.

- A set of rules to describe new cell state from the states of a group of cells from the previous layer
- Next generation depends on itself,cell to the left and, cell to the right (Neighbors).



- Ex:



https://en.wikipedia.org/wiki/File:One-d-cellular-automaton-rule-110.gif

Cellular Automata

So lets run some

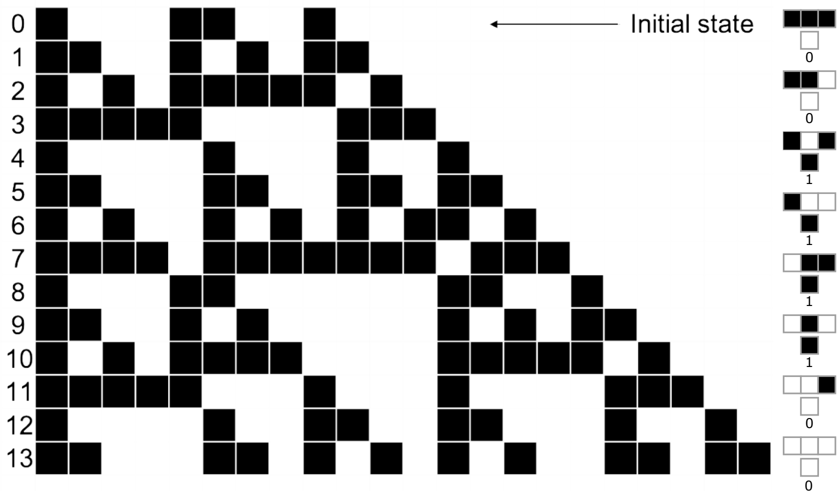


Figure: 13 Generations of a 1-D Cellular Automata

- There $2 \times 2 \times 2 = 8$ binary states for each 3 cells.
- So, there are a total of $2^8 = 256$ possible rule sets.
- Each rule can be index with 8-bit binary

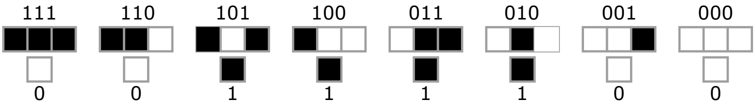


Figure: Rule $(00111100)_2 = 60$

- For the following examples, the initial state is

Cellular Automata

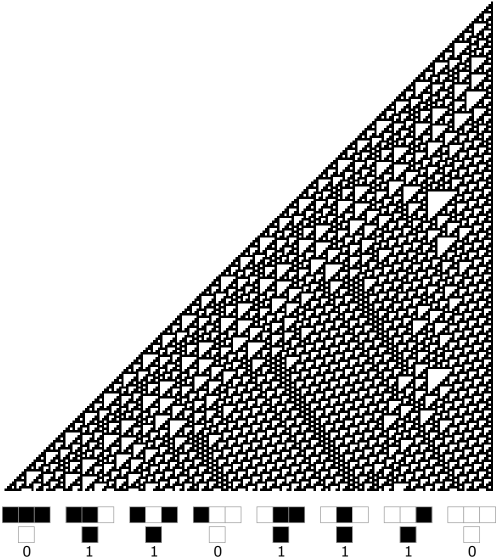


Figure: Rule 110, this might be special

Cellular Automata

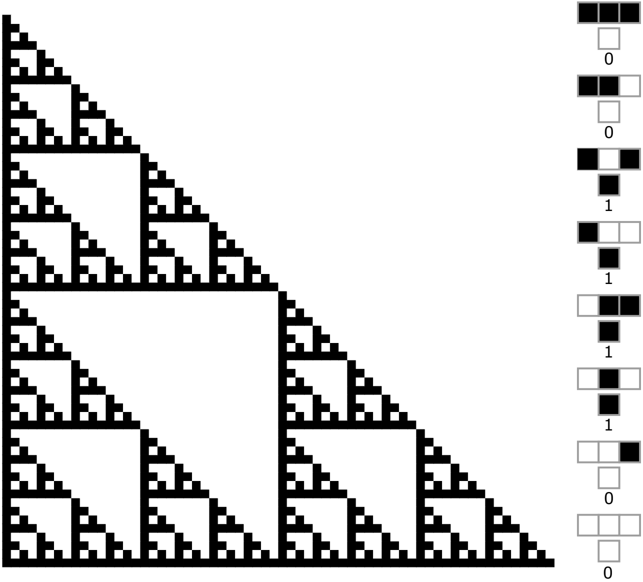


Figure: Rule 60. Sierpinski Triangle*

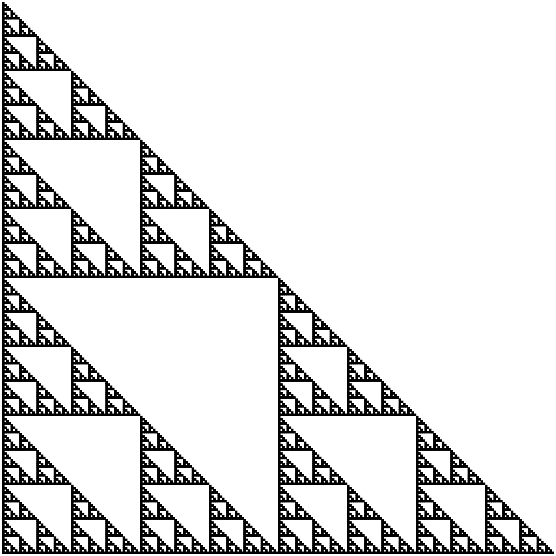


Figure: Sierpinski Triangle

Cellular Automata

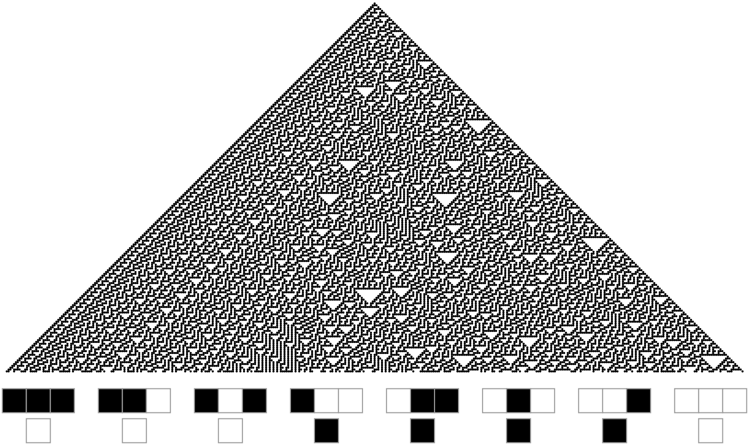


Figure: Rule 30

- Has been used as a random number generator in Mathematica
- Sensitive to initial conditons

CA and Polynomials

- Let's look at the powers of $r(x) = 1 + x$:

$$\begin{aligned}
(r(x))^0 &= 1 \\
(r(x))^1 &= 1 + x \\
(r(x))^2 &= 1 + 2x + x^2 \\
&\vdots \\
(r(x))^n &= a_0(n) + a_1(n)x + a_2(n)x^2 + \cdots + a_n(n)x^n
\end{aligned}$$

- By the addition rule of binomial coefficients

$$a_k(n) = a_{k-1}(n-1) + a_k(n-1)$$

- This is a Cellular Automata kind of behavior

CA and polynomials

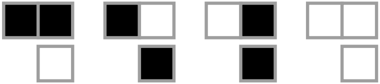
- To color the triangle, we could look at divisibility of $a_k(n)$ with respect to 2
- Now, there are 2 possibilities

$$a_k(n) \equiv 0 \pmod 2 \qquad \textit{or} \qquad a_k(n) \equiv 1 \pmod 2$$

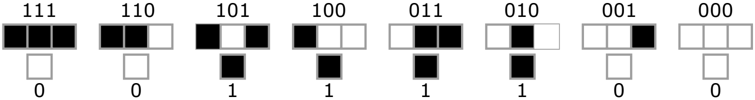
- The recursive addition rules in mod 2 arithmetic simplifies to

$\binom{n+1}{k} =$	$\binom{n}{k-1}$	$+\ \binom{n}{k}$
0	0	0
1	1	0
1	0	1
0	1	1

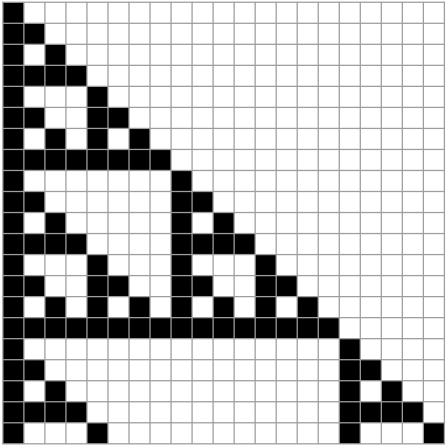
- If we set 1 = Black, 0 = White



- If we add a right neighbor to the above rule such that it doesn't affect the output



- This is rule $00111100_2 = 60$ Cellular Automata



- Thus, this figure shows the coefficients of the powers of $r(x) = 1 + x \mod 2$

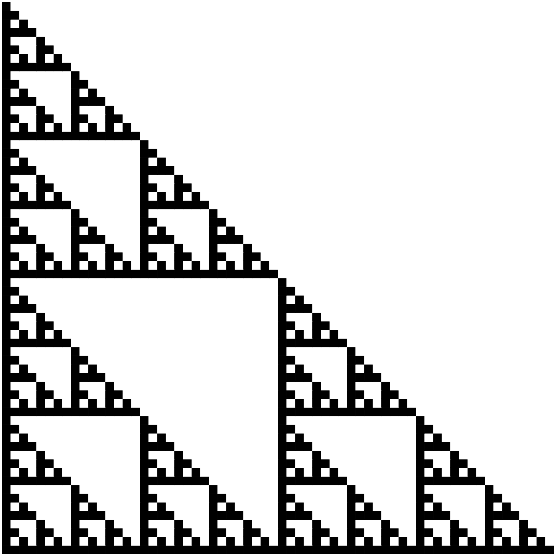


Figure: A Bigger picture

Generalizations

- There are 2 ways to generalize,
  1. A different integer modulus
  2. A different polynomial
- Ex: Let $r(x) = 1 + 2x$

$$
\begin{aligned}
(r(x))^0 &= 1 \\
(r(x))^1 &= 1 + 2x \\
(r(x))^2 &= 1 + 4x + 4x^2 \\
(r(x))^3 &= 1 + 6x + 12x^2 + 8x^3 \\
&\vdots \\
(r(x))^n &= a_0(n) + a_1(n)x + \cdots + a_n(n)x^n
\end{aligned}
$$

- By observation,

$$a_k(n) = a_k(n-1) + 2a_{k-1}(n-1)$$

- By looking at the remainder with p $= 3$,i.e $a_k(n) \mod 3$

$$(r(x))^0 = 1$$
$$(r(x))^1 = 1\ 2$$
$$(r(x))^2 = 1\ 1\ 1$$
$$(r(x))^3 = 1\ 0\ 0\ 2$$

- The cellular automaton would be $a_{n,k} = a_{n-1,k} + 2a_{n-1,k-1} \mod 3$

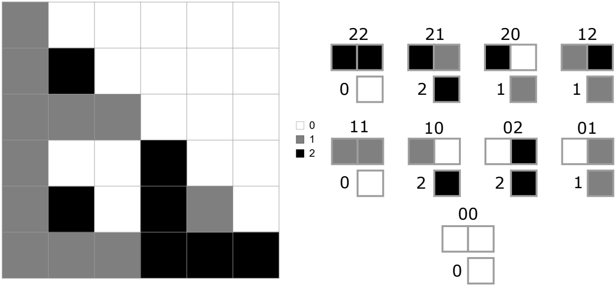


Figure: CA for $r(x) = 1 + 2x \mod 3$ and rule table

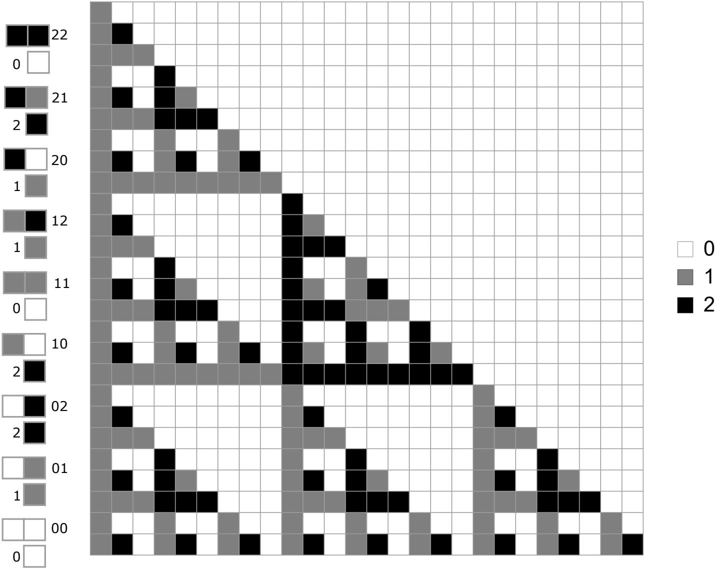
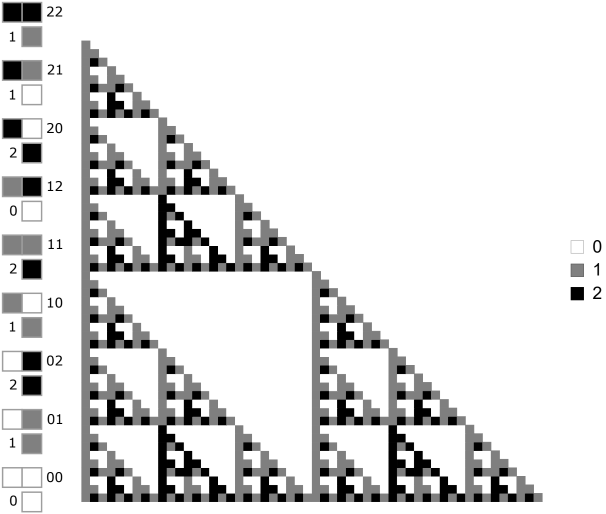


Figure: A bigger picture for rule $(02110221)_3 = 5421$

Generalizations

- Another quick example: $r(x) = 1 + x \mod 3$



- The CA rule is $a_{n,k} = a_{n-1,k} + a_{n-1,k-1} \mod 3$

The Microscopic view

Binomial Coefficients and Divisibility

- Black and white coloring of the Pascal triangle depending on divisibility with n (mod n), where n is an integer
- We will also that in order to understand the patterns formed by mod n, we should look at the patters formed by the prime factors of n
- We will look at a direct, non-recursive computation of binomial divisibility by p (prime)

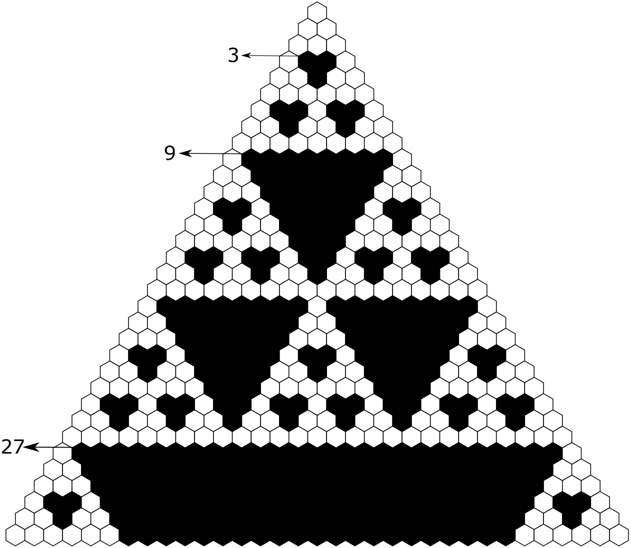


Figure: Pascal triangle Mod 3

- We define a new coordinate system such that, at position (n, k) the binomial coefficient is

$$\binom{n+k}{k} = \frac{(n+k)!}{n!k!}$$

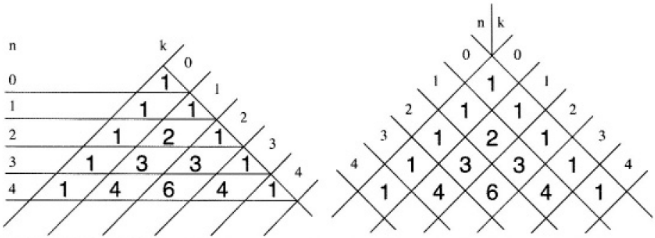


Figure: A new coordinate system

Fig. from "Chaos and Fractals" Pg 394

Divisibility Sets

- We formally define our problem

$$P(r) = \left\{ (n,k) \,\middle|\, \binom{n+k}{k} \text{ is not divisible by } r \right\}$$

- Plotting the points in this set gives the same Pascal Triangle pattern

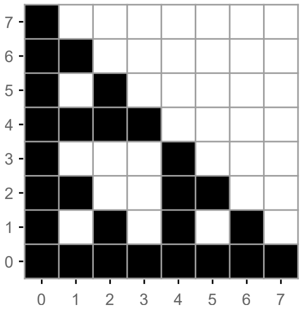
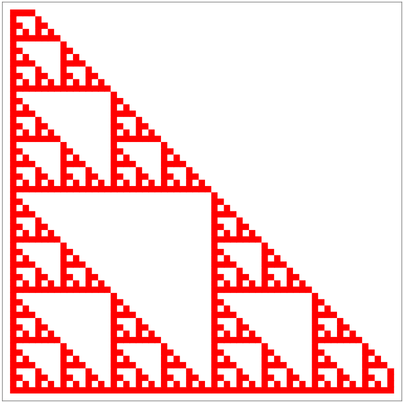


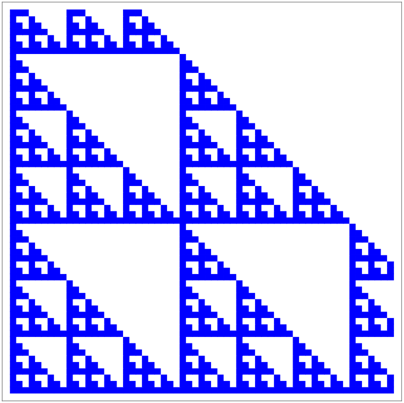
Figure: $P(2)$ or Mod 2

Divisibility Sets

- Observe that if p and q are two different prime numbers and a given integer r is **not** divisible by $p \cdot q$, then it is **not** divisible by either p or q
$$P(pq) = P(p) \cup P(q), \text{ if } p \neq q, \ p, q \text{ prime} \tag{†}$$
- Eq (†) is the negation of the statement: If pq divides r, then r is divisible by both p and q.
- Ex: $P(6) = P(2) \cup P(3)$



$P(2)$ or Mod 2



$P(3)$ Mod 3

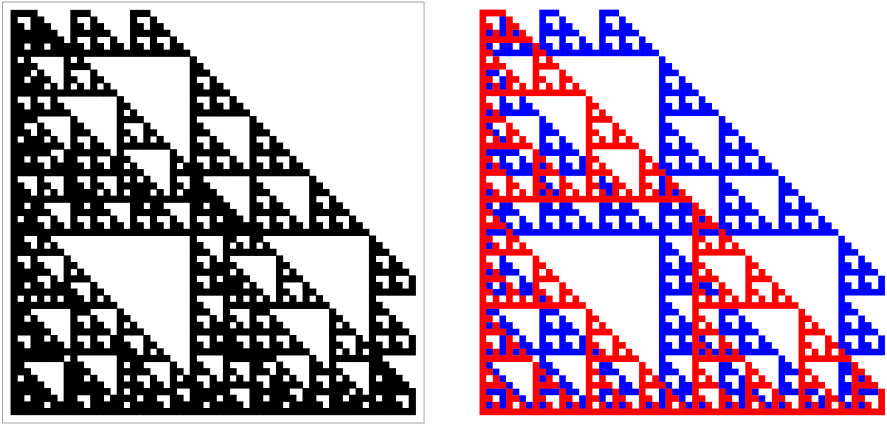


Figure: Left shows Mod 6 ($P(6)$) pattern and right shows the union of mod 2 (Red) and mod 3 (Blue)

- For an integer $r = p_1^{n_1} \dots p_s^{n_s}$ where $p_1, \dots, p_s$ are primes

$$P(r) = P(p_1^{n_1}) \cup ... \cup P(p_s^{n_s})$$

- So to understand the pattern formed by $P(r)$, we just need to understand the pattern of $P(p^n)$
- We can construct $P(p)$ in a non-recursive direct method using Kummer's result

Kummer's Result and p-adic numbers

- To understand Kummer's result, we need to consider numbers with base p, where p is prime
- Like the decimal system, p-adic expansion of an integer is given by

$$n = a_0 + a_1 p + \cdots + a_m p^m$$

  where $a_i \in \{0, 1, \ldots, p-1\}$
- The p-adic representation would be

$$n = (a_m a_{m-1} \ldots a_0)_p$$

- Ex: $15_{10} = (1111)_2 = (120)_3 = (30)_5 = (21)_7 = (14)_{11}$

- We define carry function c_p

$$c_p(n,k) = \text{number of carries in the p-adic addition of n and k}$$

- Ex: For $n = 15$ and $k = 8$

$$k = (08)_{10} = (1000)_2 = (022)_3 = (13)_5 = (11)_7 = (08)_{11}$$

- If we take the binary addition $\begin{array}{cccccc} & 0^1 & 1 & 1 & 1 & 1 \\ + & 0 & 1 & 0 & 0 & 0 \\ \hline & 1 & 0 & 1 & 1 & 1 \end{array}$

$$c_2(15,8) = 1$$

- Similarly, we can do the same for $p = 3, 5, 7 \ldots$

Kummer's result

- Kummer's Statement is:

  Let $r = c_p(n, k)$, *then* $\binom{n+k}{k}$ *is divisible by* p^r *but not* p^{r+1}

- So, prime factorization of $\binom{n+k}{k}$ contains exactly $c_p(n, k)$ factors of p
- This result is pretty amazing, since it gives a direct method to check if a binomial coefficient is divisible by p or not.

- For $n = 5$ and $k = 4$, Kummer's result says

$$n = (101)_2 = (12)_3 = (10)_5 = (5)_7$$

$$k = (100)_2 = (11)_3 = (4)_5 = (4)_7$$

- In Base 2 addition $\begin{array}{ccccc} & 0^1 & 1 & 0 & 1 \\ + & 0 & 1 & 0 & 0 \\ \hline & 1 & 0 & 0 & 1 \end{array} \Rightarrow c_2(5,4) = 1$

- In Base 3 addition $\begin{array}{cccc} & 0^1 & 1^1 & 2 \\ + & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 \end{array} \Rightarrow c_3(5,4) = 2$

- Similar additions can be done for $p = 5, 7, 11 \ldots$

- Kummer's result says:

$$c_p(5,4) = \begin{cases} 1, \text{ for } p = 2,7 \\ 2, \text{ for } p = 3 \\ 0 \text{ otherwise} \end{cases}$$

- This implies

$$\binom{5+4}{4} = \binom{9}{4} = 2^1 \cdot 3^2 \cdot 7^1$$

- Applying Kummer's result to Divisibility Set gives

$$P(p) = \{(n, k) |\ c_p(n, k) = 0\}$$

- That is, the number of carries $c_p(n, k)$ is only 0 if and only if

$$a_i + b_i < p, \quad i = 0, \ldots, m$$

  where a_i and b_i are the p-adic digits of n and k.

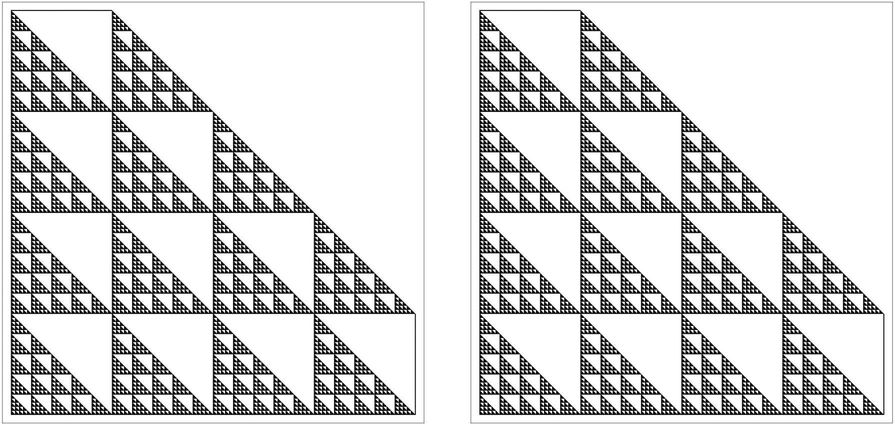


Figure: $P(5)$ generated using carry function (left) vs binomial divisibility (right)

The Big Picture

Iterated Function System

- A method of constructing fractals using a set of contraction mappings.
- A contraction mapping is an affine linear transformation

$$f(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

  where

$$\det \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) < 1$$

- Union of these contraction mappings gives the Hutchinson equation of the fractal

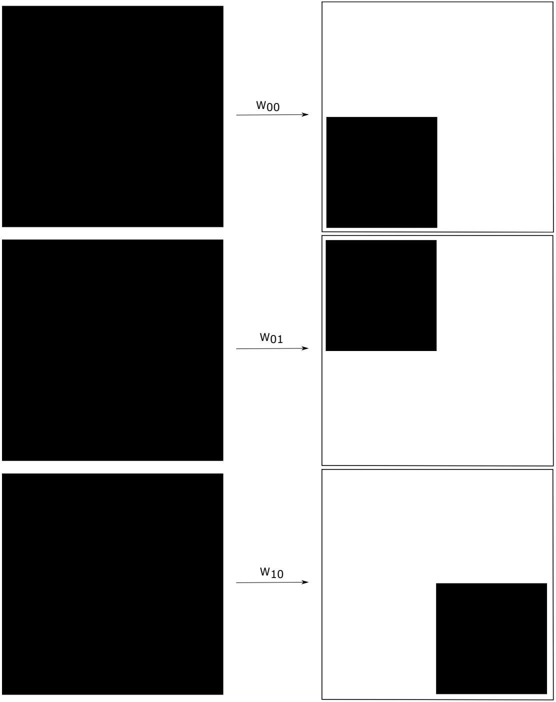
- The Hutchinson operator for Sierpinski Triangle is:

$$S = w_{00}(S) \cup w_{01}(S) \cup w_{10}(S)$$

  where w_{ij} are contraction mappings of a unit square

$$\begin{aligned}
w_{00} &= (x/2, y/2) \\
w_{01} &= (x/2, y/2 + 1/2) \\
w_{10} &= (x/2 + 1/2, y/2) \\
w_{11} &= (x/2 + 1/2 + y/2 + 1/2)
\end{aligned}$$

- Let Q be a unit square


w_{00}
w_{01}
w_{10}

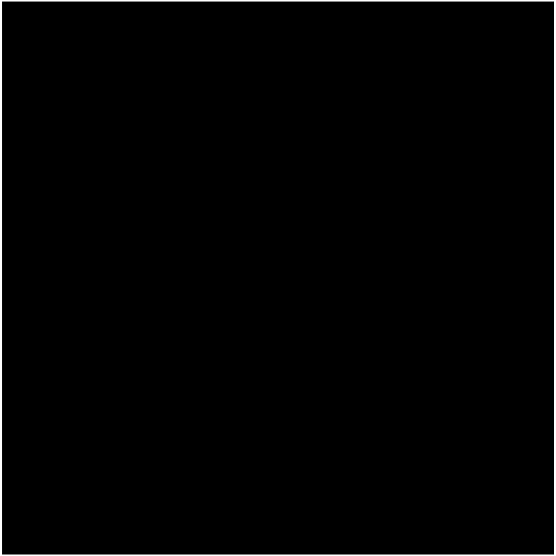


Figure: Initial region, $Q = S^0$

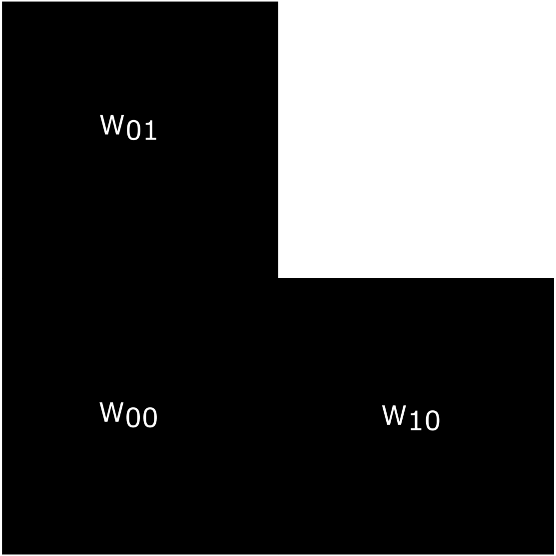


Figure: $S^1 = S(Q)$

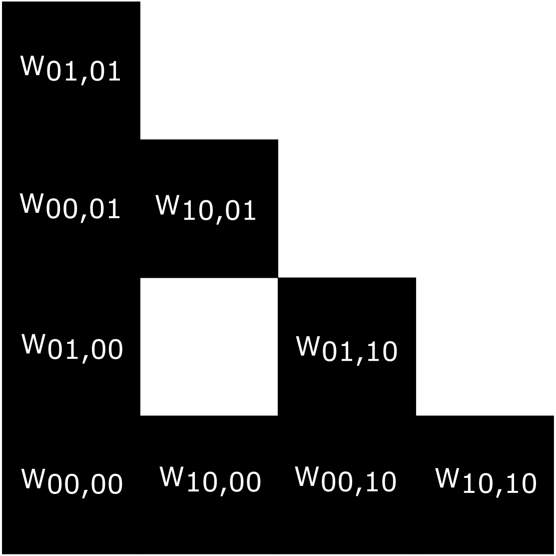


Figure: $S^2 = S(S(Q))$

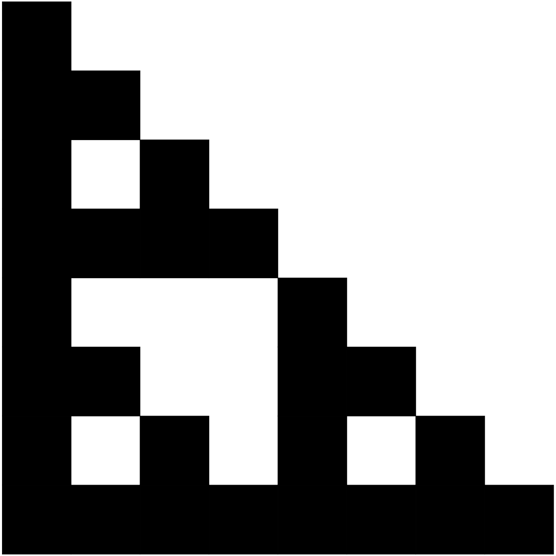


Figure: $S^3 = S(S(S(Q)))$

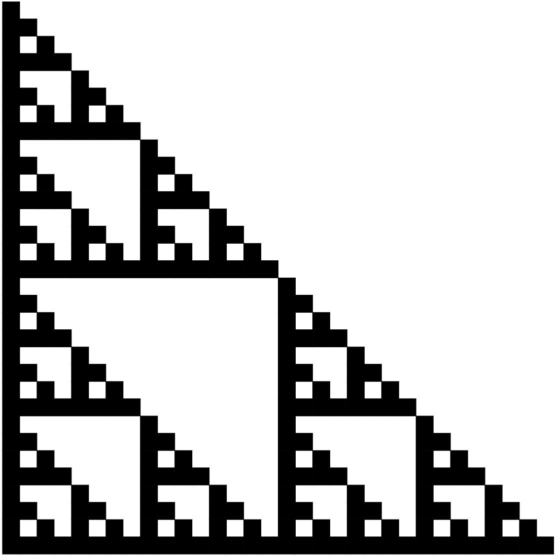


Figure: S^5

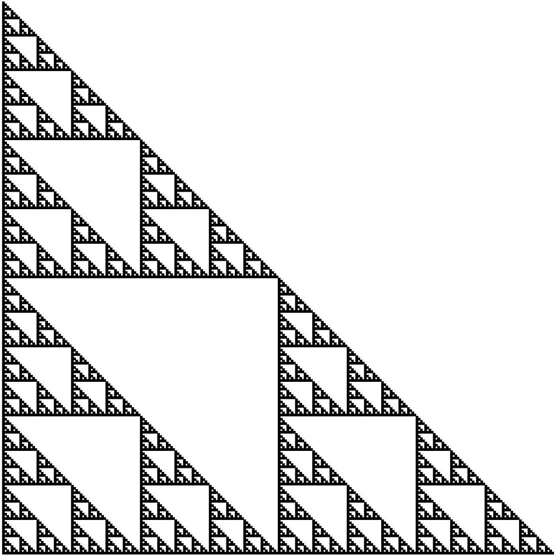


Figure: S^{∞}

So how do we know for sure that our Pascal Triangle pattern is the Sierpinski Triangle ?

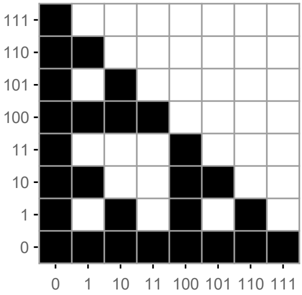
- Let's construct Mod 2 Pascal triangle in a unit square
- Let Q be the unit square

$$Q = \{(x, y) | \ (x, y) \in [0, 1] \times [0, 1]\}$$

- Expand x and y in base 2

$$x = \sum_{i=1}^{\infty} a_i 2^{-i}, \ \ a_i \in \{0, 1\}$$

$$y = \sum_{i=1}^{\infty} b_i 2^{-i}, \ \ b_i \in \{0, 1\}$$



- From Kummer's result, we know that the coordinates of the points not divisible by 2 are those which have no carries in the binary addition of the coordinates

$$S = \{(x, y) \in Q | \ a_i + b_i \leq 1 \ \forall i\}$$

Proof

- To show S is the Sierpinski Triangle, S must be invariant under the Hutchinson operator

$$w_{00}(S) \cup w_{01}(S) \cup w_{10}(S) = S$$

- Take any point $(x, y) \in S$

$$(x, y) = (0.a_1 a_2 \ldots, 0.b_1 b_2 \ldots)$$

  where $a_i + b_i \leq 1$

- To understand Hutchinson operator in action

$$\frac{x}{2} = \frac{a_1 2^{-1} + a_2 2^{-2} + a_3 2^{-3} \ldots}{2} = 0 \cdot 2^{-1} a_1 2^{-2} + a_2 2^{-3} + a_3 2^{-4} \ldots$$
$$= (0.0 a_1 a_2 a_3)_2$$

$$\left(\frac{x}{2} + \frac{1}{2}\right) = \frac{1}{2} + 0 \cdot 2^{-1} a_1 2^{-2} + a_2 2^{-3} + a_3 2^{-4} \ldots$$
$$= (0.1 a_1 a_2 a_3)_2$$

- Applying the 3 transformations gives

$$
\begin{aligned}
w_{00}(0.a_1a_2\dots, 0.b_1b_2\dots) &= (0.0a_1a_2\dots, 0.0b_1b_2\dots)\\
w_{01}(0.a_1a_2\dots, 0.b_1b_2\dots) &= (0.0a_1a_2\dots, 0.1b_1b_2\dots)\\
w_{10}(0.a_1a_2\dots, 0.b_1b_2\dots) &= (0.1a_1a_2\dots, 0.0b_1b_2\dots)
\end{aligned}
$$

- Clearly, all these points are also in S.
- Therefore, S is invariant under the Hutchinson operator
- So..., S is indeed the Sierpinski Triangle

Concequences

- The binary representation allows us to see Hutchinson operator in action applied to a point inside square
- If (x, y) is an arbitrary point in Q, the applying the map w_{00}, w_{01}, w_{10} again and again yields points with leading binary decimal points satisfying $a_i + b_i \leq 1$
- In symbols,

  $$A_0 = Q$$

  Then running the IFS gives

  $$A_n = w_{00}(A_{n-1}) \cup w_{01}(A_{n-1}) \cup w_{10}(A_{n-1})$$

  where the leading n binary digits satisfy $a_i + b_i \leq 1$
- Finally, the sequence will lead to Sierpinski triangle

  $$\lim_{n \to \infty} A_n = S$$

IFS for other primes

- With similar arguments we can construct IFS for any prime p
- Divide the unit square Q in p^2 congruent square $Q_{a,b}$ with $a, b \in \{0, \ldots, p-1\}$. We introduce the contraction mappings

$$w_{a,b}(x,y) = \left(\frac{x+a}{p}, \frac{y+b}{p}\right)$$

where

$$w_{a,b}(Q) = Q_{a,b}$$

- Then we set the restriction to define the set of transformation

$$a + b \leq p - 1$$

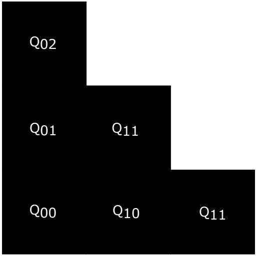


Figure: 1st Generation of Mod 3 IFS

- This restriction follows from Kummer's result,i.e $\binom{n+k}{k}$ is indivisible by p when there is no carries in the p-adic addition of n, k
- The Hutchinson operator for these contractions,

$$W_p(A) = \bigcup_{a+b<p} w_{a,b}(A)$$

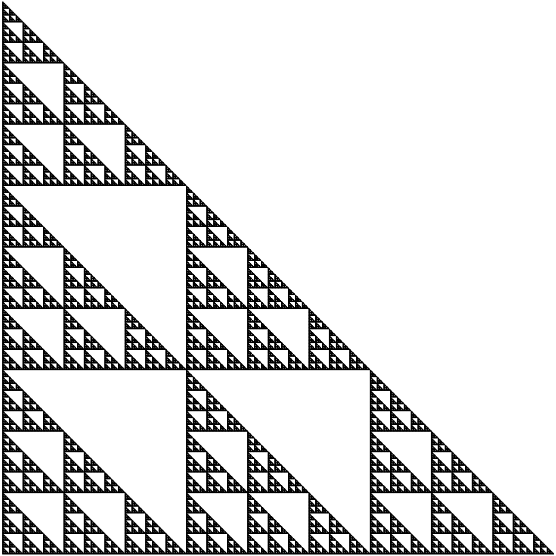


Figure: Limit of Mod 3 IFS.

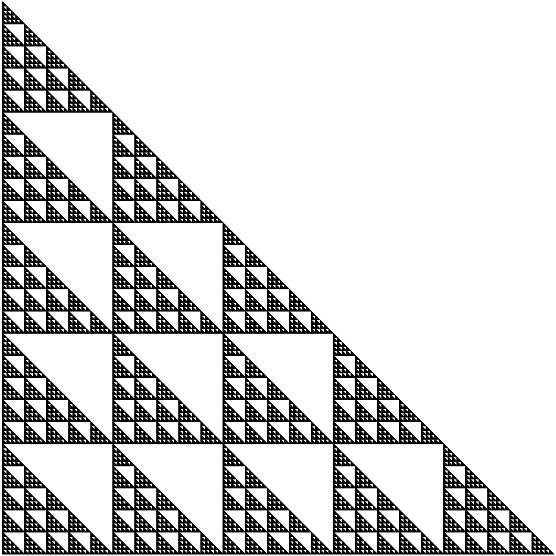


Figure: Limit of Mod 5 IFS.

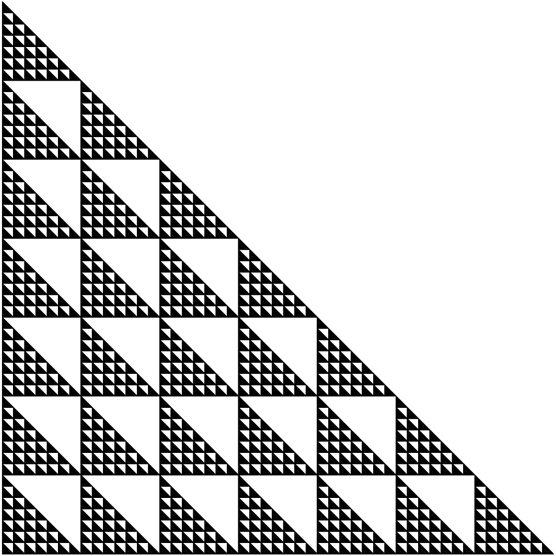


Figure: Limit of Mod 7 IFS.

Summary

- We approached the problem of coloring (or divisibility) of pascal's triangle in 3 different ways
  1. First, we looked at the macroscopic nature of divisibility, i.e how the neighbors affected the divisibility of a cell. We then explored the realm of cellular automata and how it extends this idea to general polynomials.
  2. Second, we looked at the microscopic nature of divisibility,i.e how the divisibility of a cell depends on its coordinate (Kummer's Result). We looked how p-adic numbers gave insight about binomial divisibility with primes.
  3. Finally, we looked at the global pattern using Iterated function system. We showed that the limit of Pascal Mod 2 pattern was indeed the Sierpinski triangle. We also explored the global patterns of other primes and their IFS.

- Each method gave a different perspective of the same problem.
- This entire chapter is trying to solve the jig-saw puzzle that relates fractals, Pascal's triangle, and Cellular Automata.
- Understanding that these 3 very different things are all tied up by one common thread is, honestly, amazing.

Thank You!

References

- Most of the content in this presentation is based on "Chaos and Fractals: New Frontiers of Science" by Peitgen, H et al 2003.
- Slide 29: Image was directly taken from the book "Chaos and Fractals" Pg. 394
- Slide 3,4,28: Modified Tikz code from `https://tex.stackexchange.com/questions/198887/how-can-i-draw-pascals-triangle-with-some-its-properties`
- Mathematica was used to generate most of the figures.
- Inkscape, an open source image editing software, was used to edit all vector images. `https://inkscape.org/`