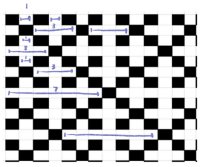
CRYSTAL CELLULAR AUTOMATA

Finding a generalized formula for the generation, or appearance, of a crystal at a given time *n*.
Crystal: the union of all crystal blocks, at a specified turn
Generation overview:

1. Start with one block of crystal at point (0, 0)
2. At every turn, or iteration, if some point is adjacent to exactly one block of crystal, it becomes crystal as well
3. At turn 1, the points (0, 0), (1, 0), (0, 1), (-1, 0), and (0, -1) will contain crystal blocks

THE CRYSTAL FORMULA

- We can use the negative—non crystal blocks—of a generated crystal to derive the crystal formula

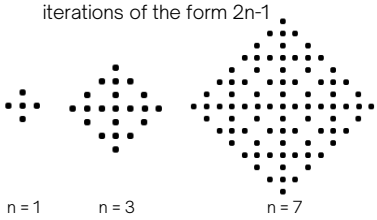


- The negative space alternates every other space in the first row, in the second row, it alternates every 3, and in the 3rd row, it alternates everyone gain, and this continues
- Notice that this is true for the columns as well
- The amount of squares for a repetition of the negative space go in order 1, 3, 1, 7, 1, 3, 1, 15, 1, 3, 1, 7, 1, 3, 1, 31...
- This gives us the following formula for whether or not a given coordinate (x, y) is in the crystal after an infinite amount of time:

$$g(x,y) = \begin{cases} \text{false} & y \equiv gcd(x, 2^x) \pmod{2 \cdot gcd(x, 2^x)} \\ \text{true} & \text{else} \end{cases}$$

FULL CRYSTALS

- **Full crystals** are crystals in the shape of diamonds
- They always form at always at iterations of the form 2n-1



- This is true because every full diamond is **4 copies** of previous full diamond
- Double the steps to get to next full crystal.

CRYSTAL COUNTING TERMINOLOGY

- We want to count the number of blocks added at each iteration — the speed at which it grows
- **c(n) = # of blocks in crystal at time n**
- **distribution sites:** where the crystal will grow

Figure 1. CRYSTAL GENERATION: 1/4 of a crystal, with red being the first generated blocks and green being the last.

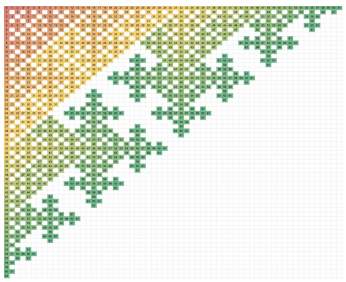


Figure 2. CRYSTAL GENERATION: the negative space of a crystal.

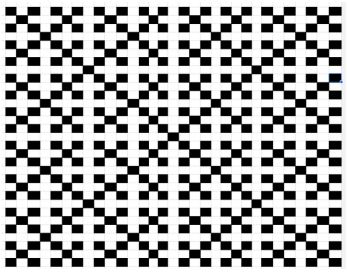


Figure 3. 3D CRYSTAL CELLULAR AUTOMATA following the same ruleset for generation:

- The starting point is (0,0,0)
- Each point checks the six neighbors around it

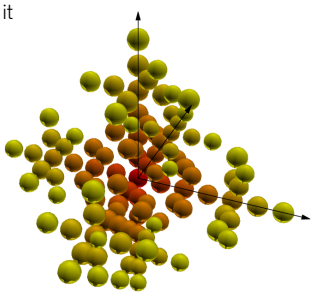


Figure 4. CELLULAR AUTOMATA COUNTING KEY, where the numbers here is the time *n*

1							
2	3						
4	5	6	7				
8	9	10	11	12	13	14	15

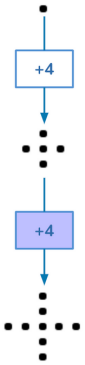
COUNTING CRYSTAL BLOCKS

- To start us off, c(0) = 1, c(1) = 5, c(2) = 9, c(3) = 21, and c(4) = 25
- We can count the distribution sites to generate a formula

4	4x3						
4	12	4x3	12x3				
4	12	12	36	4x3	12x3	12x3	36x3
4	12	12	36	12	36	36	108

- 1, 1 + 4 = 5, 1 + 4 + 4 = 9, 1 + 4 + 4 + 12 = 21, etc.
- Using this formula, C(26) = 729, and so on

- Rule to generate next row: duplicate previous row, multiply second copy by 3
- **To get C(n) (number of crystal blocks in nth iteration):** Add **1 plus first n numbers** from top-down, left-right on table



HIGHER DIMENSIONS (3D)

- The first half of each row grows the same as the previous row
- The second half of each row adds five times as many cells as the previous row

6							
6	30						
6	30	30	150				
6	30	30	150	30	30	150	750

HIGHER DIMENSIONS (N)

- If there are *n* dimensions, there are *2n* neighbors for each point
- For each perfect octahedron at iterations of the form 2x-1, there are *2n* distribution sites

2n							
2n	2n(2n-1)						
2n	2n(2n-1)	2n(2n-1)	2n(2n-1)				
2n	2n(2n-1)	2n(2n-1)	2n(2n-1)	2n(2n-1)	2n(2n-1)	2n(2n-1)	2n(2n-1)

Nico Alberti, Kailey Hua, Avery Li, Brendan Song
Proof School Math Burst 2021 - Advisor: Dr. Jason Horowitz