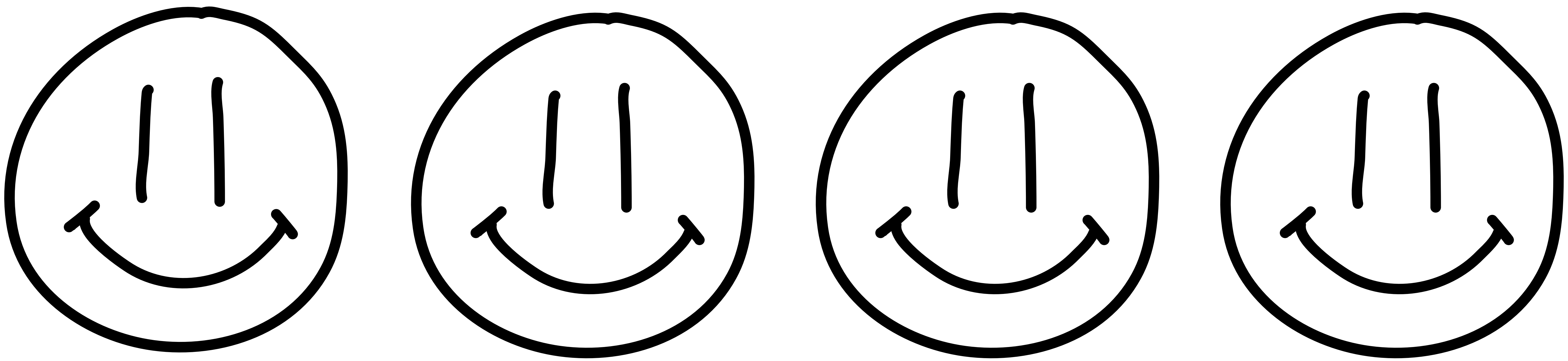
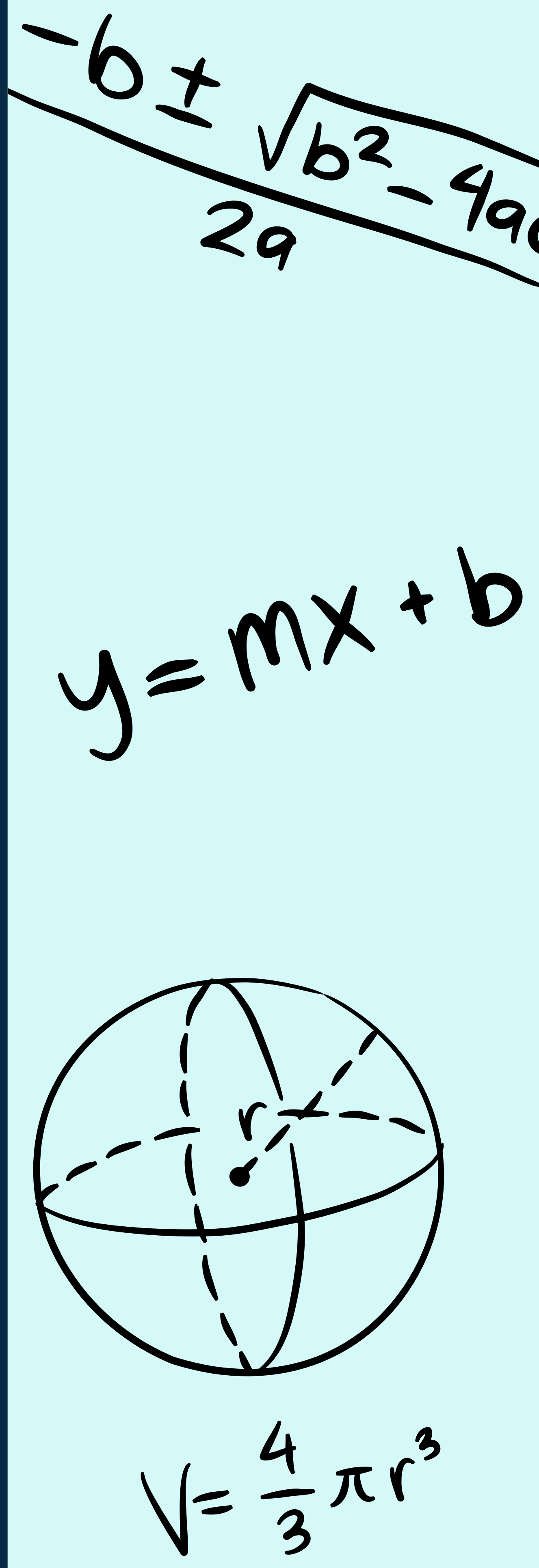
Lycée Adam de Craponne
Salon de Provence

CELLULAR AUTOMATON : RULE 60

PRESENTATION PLAN

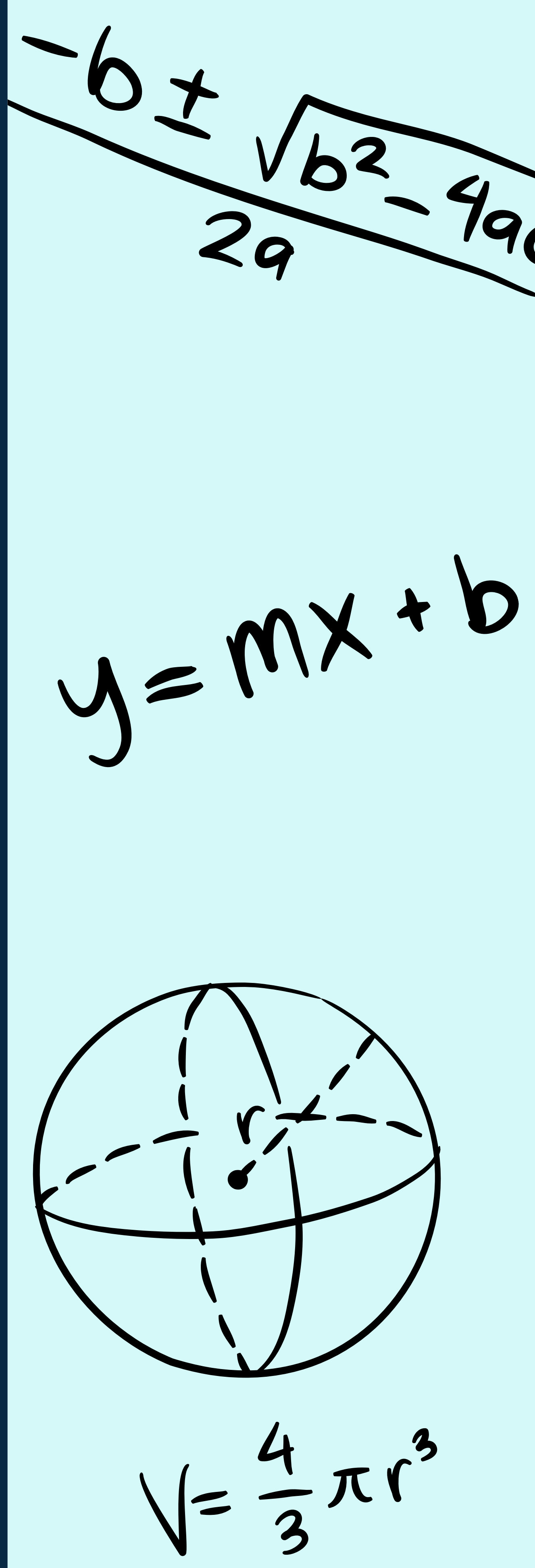
1. Introduction
2. What is a cellular automaton?
3. Our research: the Rule 60 automaton





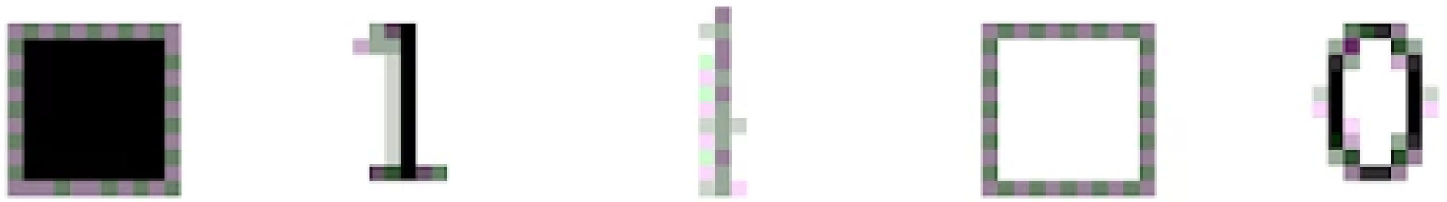
1. Introduction

December 2023

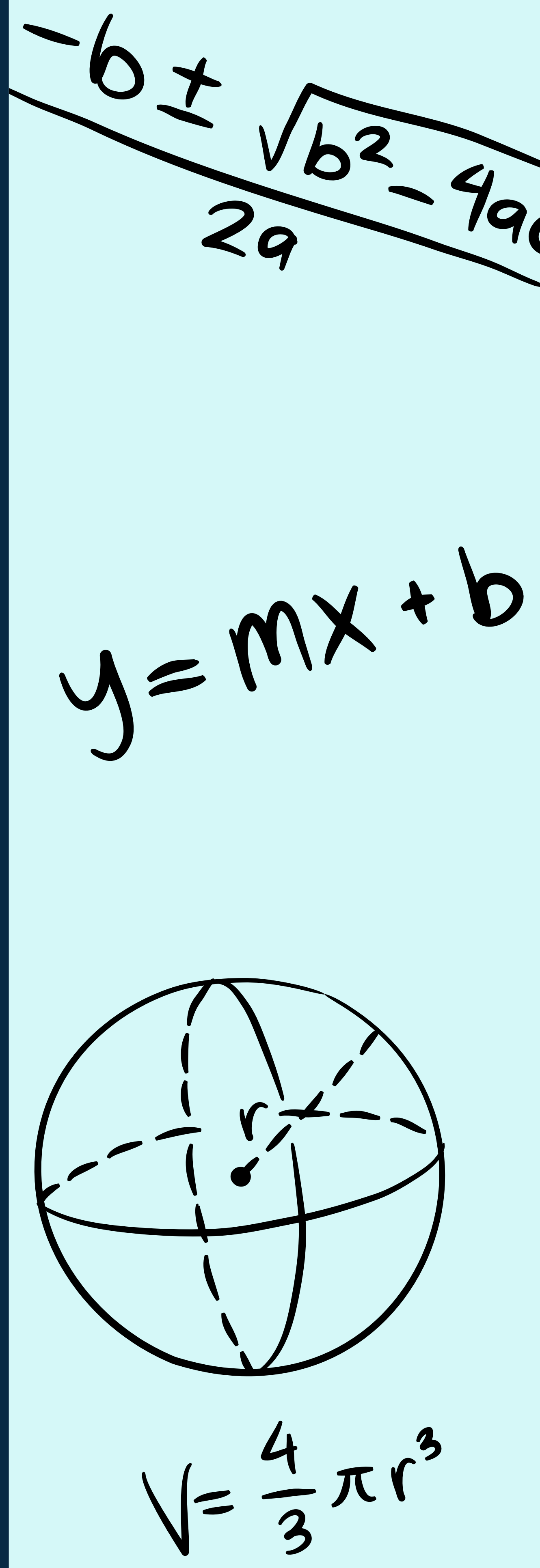


2. What is a cellular automaton?

A cellular automaton can be seen as a series of cells evolving according to a set of defined rules, thus giving rise to a new generation of cells.

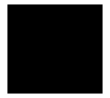


Cell evolution rules are often defined in terms of the cell's neighbors.

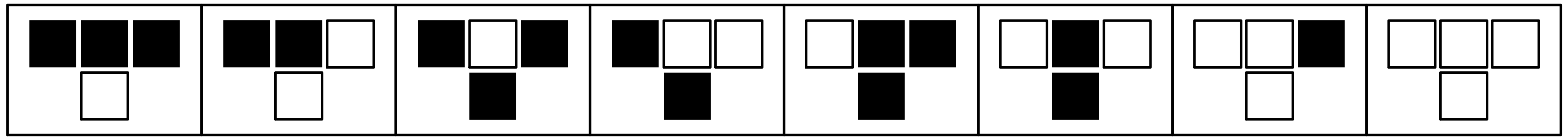


2. What is a cellular automaton?

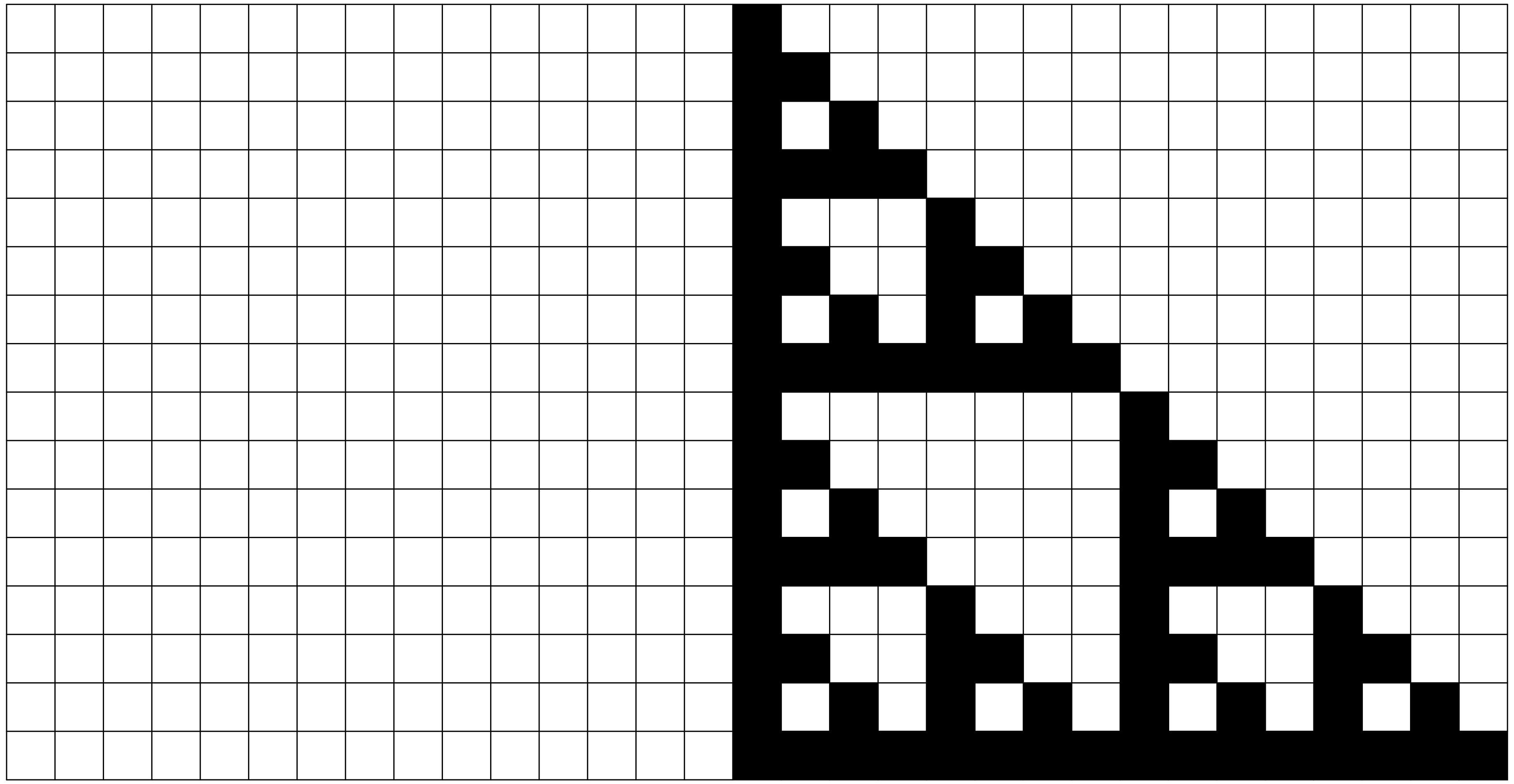
The starting line consists of one black square and the others are white.

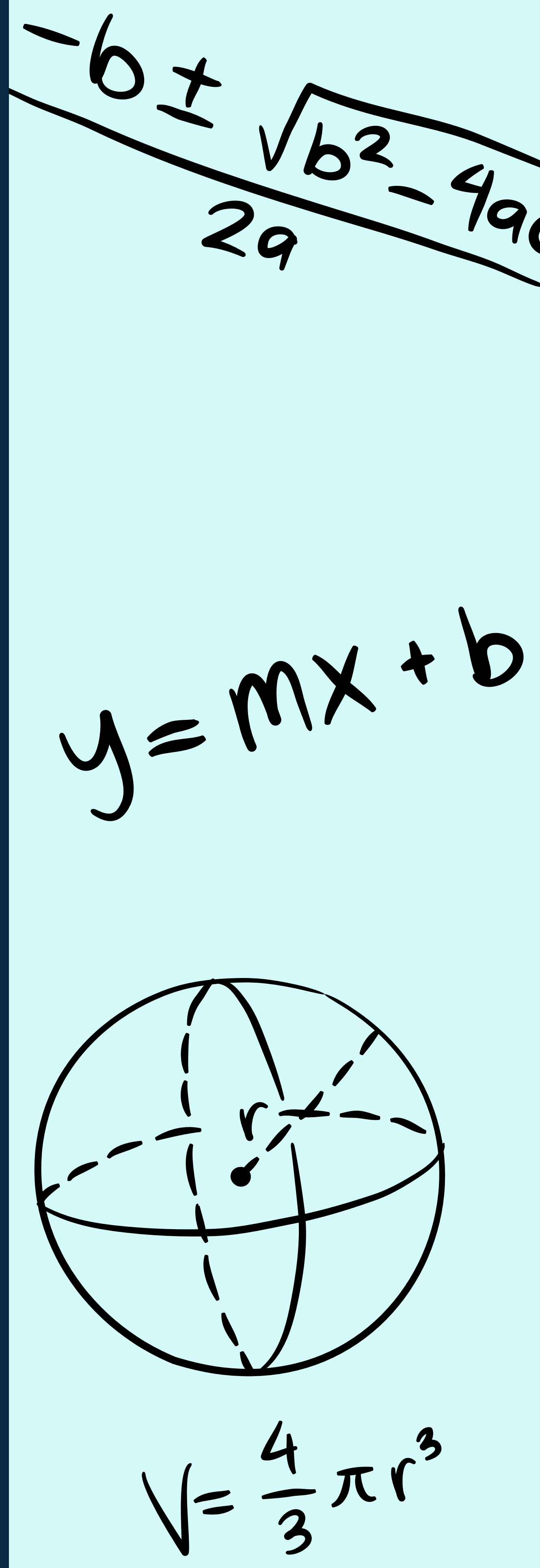


rule 60

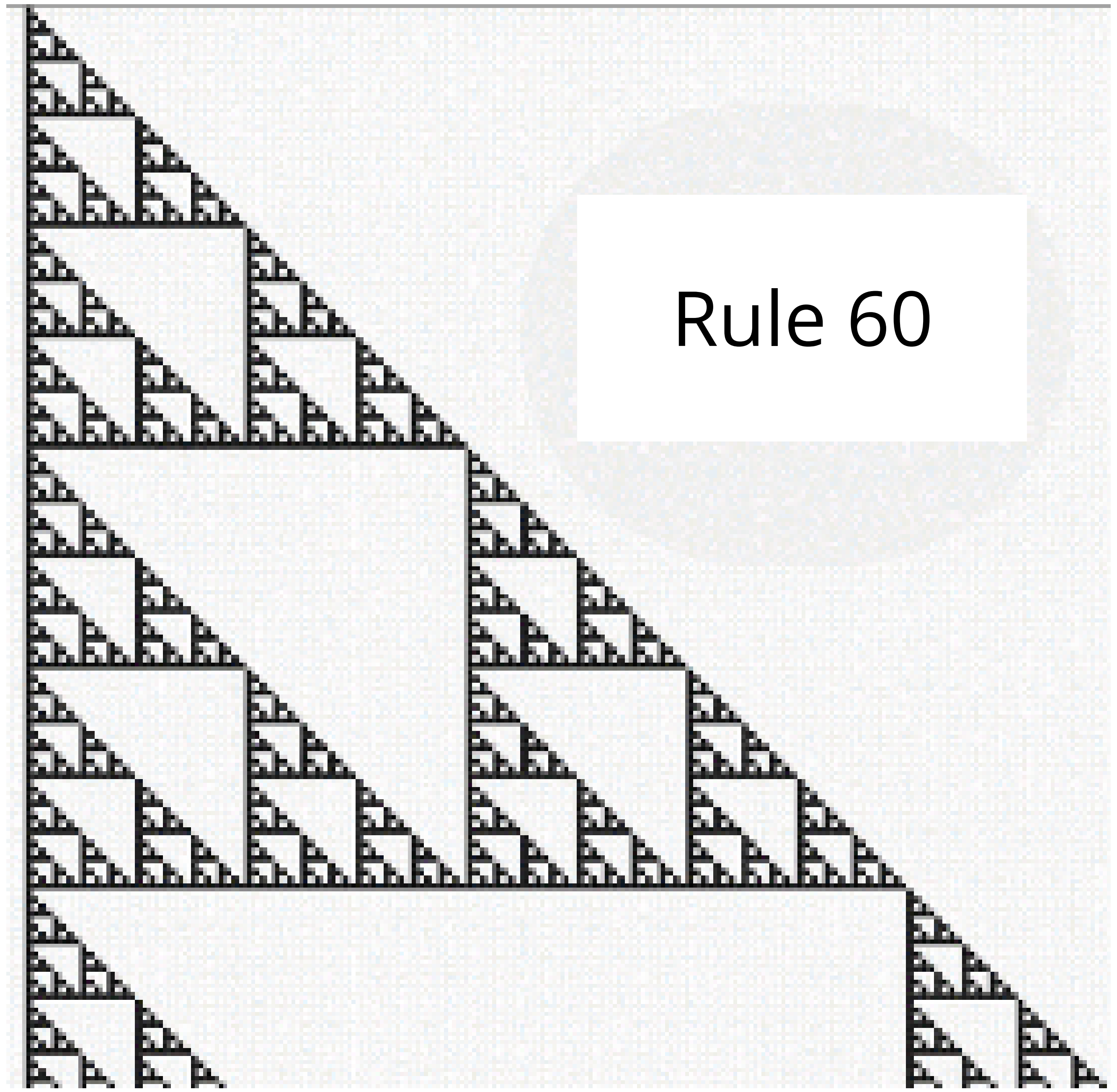


0 0 1 1 1 1 0 0



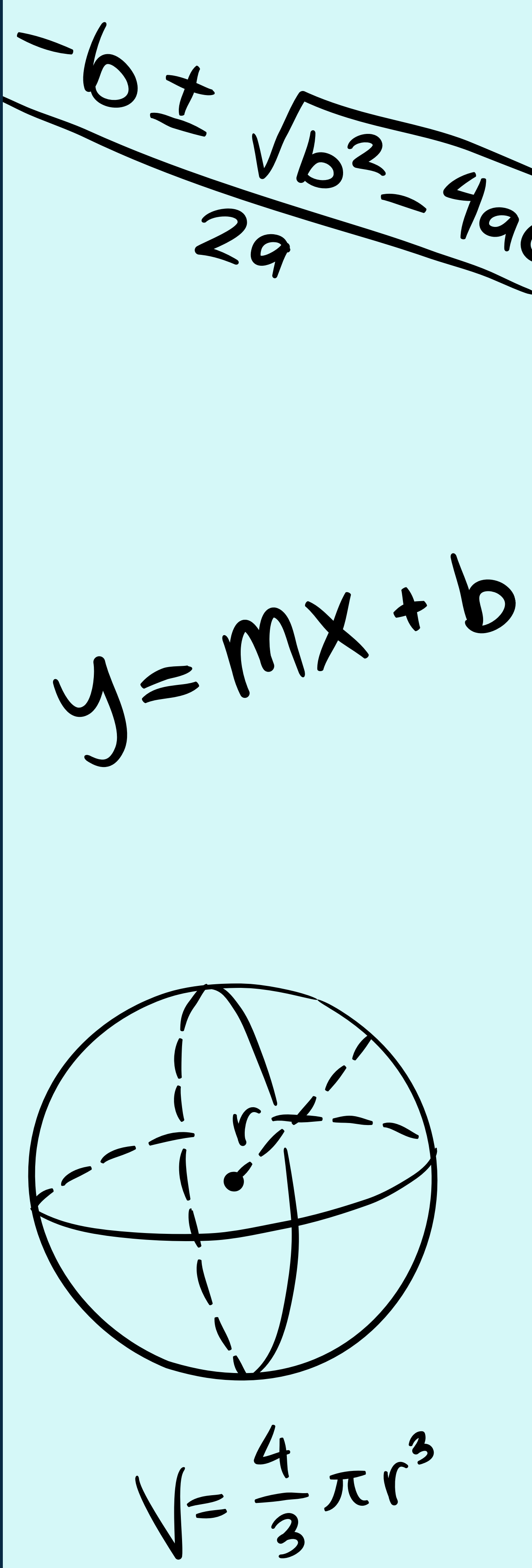


3. Our research: the Rule 60 automaton



Conjectures:

- A black column is formed below the initial black square To its left, all the squares are white
- Lines 2, 4, 8, 16, etc. have "black lines".
- The initial triangle is copied in duplicate below, then the new pattern is itself copied in duplicate and so on to infinity.

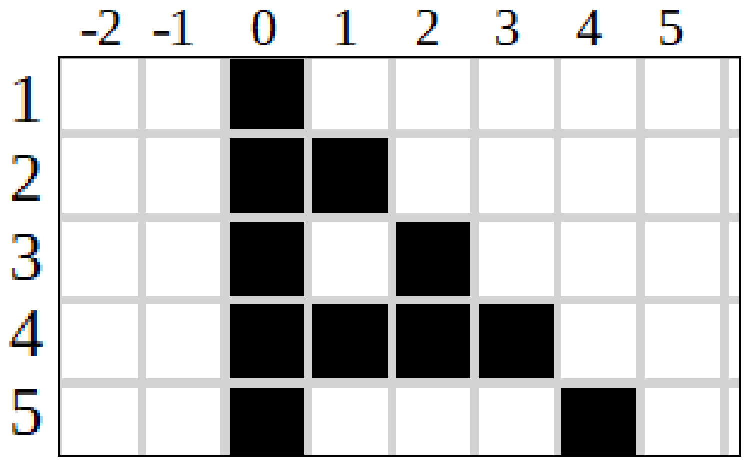


3. Our research: the Rule 60 automaton

Notations

For any automaton square, with (i ∈ℤ and j ∈ℕ*) :

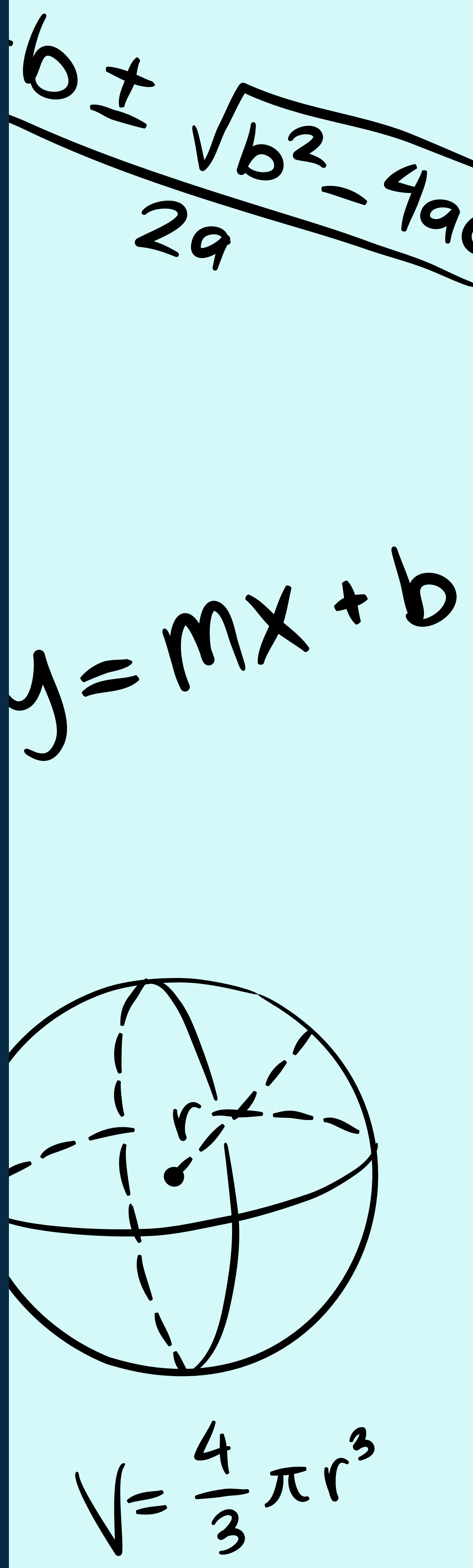
- x : designates an automaton cell
- i : column number
- j : line number



Initial black box of the automaton: column 0, line 1

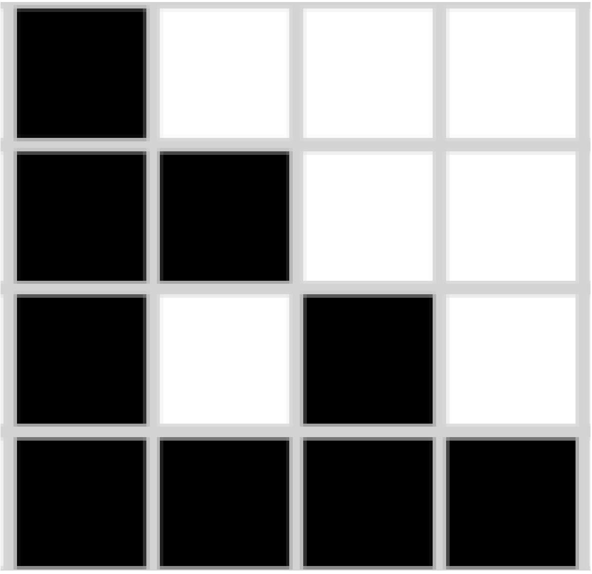
x(i;j) will be the value of the cell located in column i and line j.

- If x(i;j)=0, then the square is white.
- If x(i;j)=1, then the square is black.



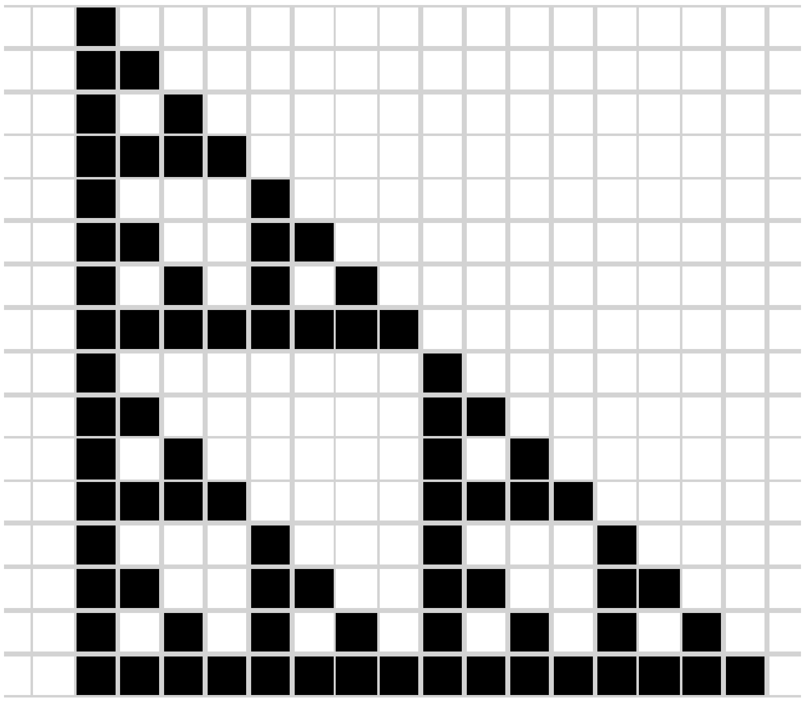
3. Our research: the Rule 60 automaton

A few lemmas



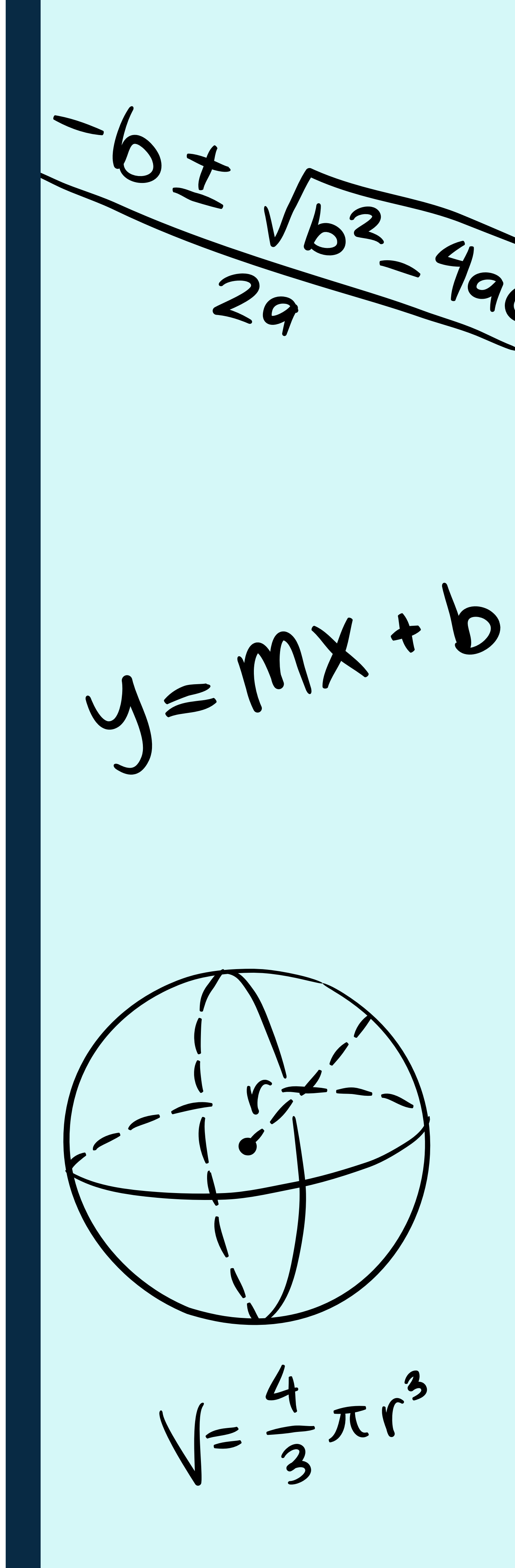
<u>Lemma 1:</u> There is a basic pattern that repeats itself regularly.

<u>Lemma 2:</u> For each new line, the number of squares in the figure increases by one.



<u>Lemma 3:</u> if i≥j then x(i;j) = 0

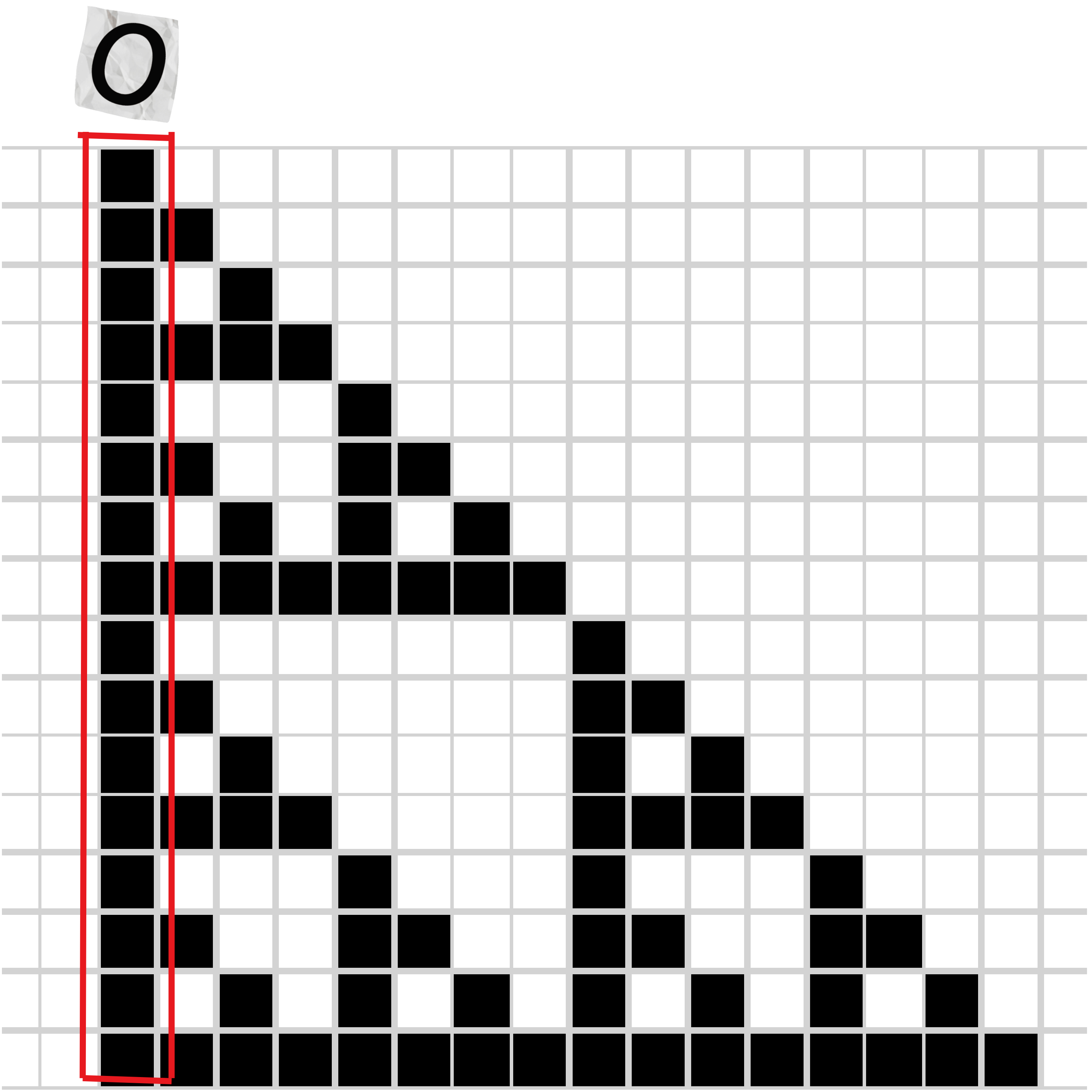
<u>Lemma 4:</u> if j= 2^n, n ∈ ℕ, Nj = j.



3. Our research: the Rule 60 automaton

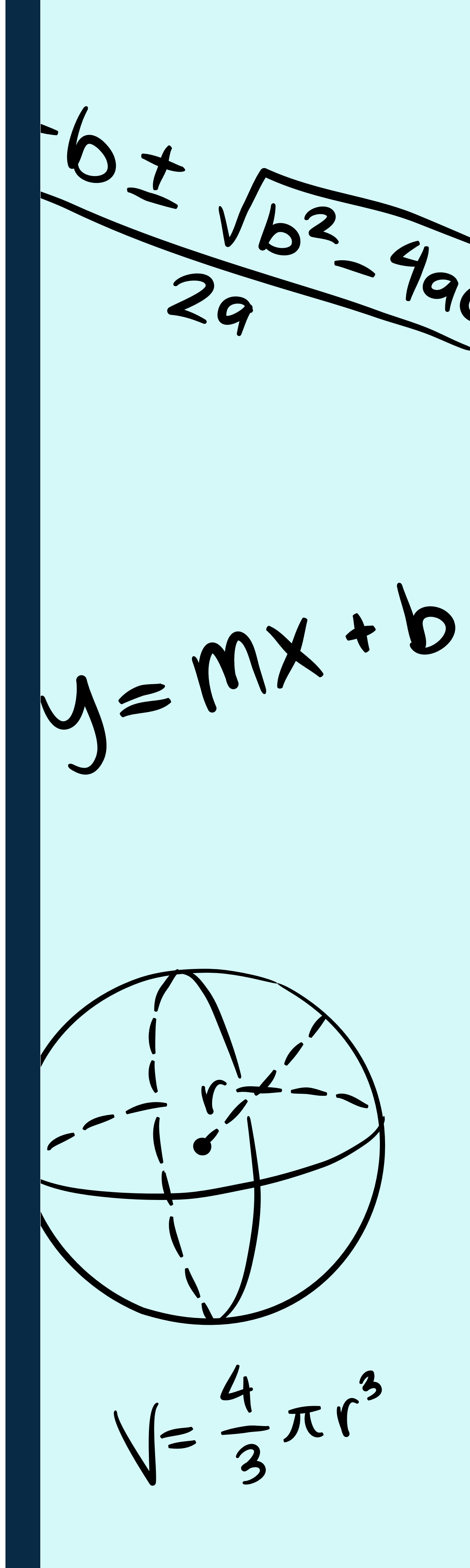
First proof

Column i=0 is always black



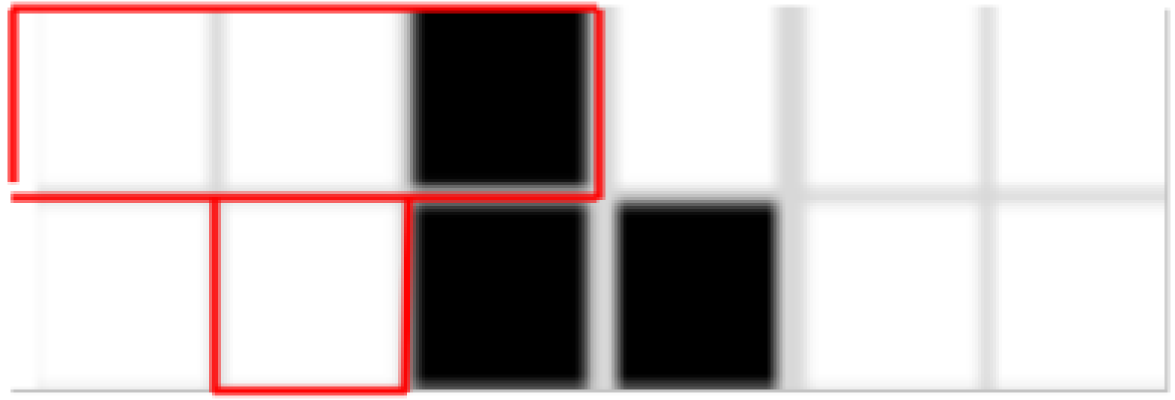
Proof by induction

- base case
- induction step
- conclusion



3. Our research: the Rule 60 automaton

A second proof



From the first to the second line:

3 white squares give a white square (000 => 0)

2 white squares and a black square on the right give a white square (001 => 0)

For the following lines :

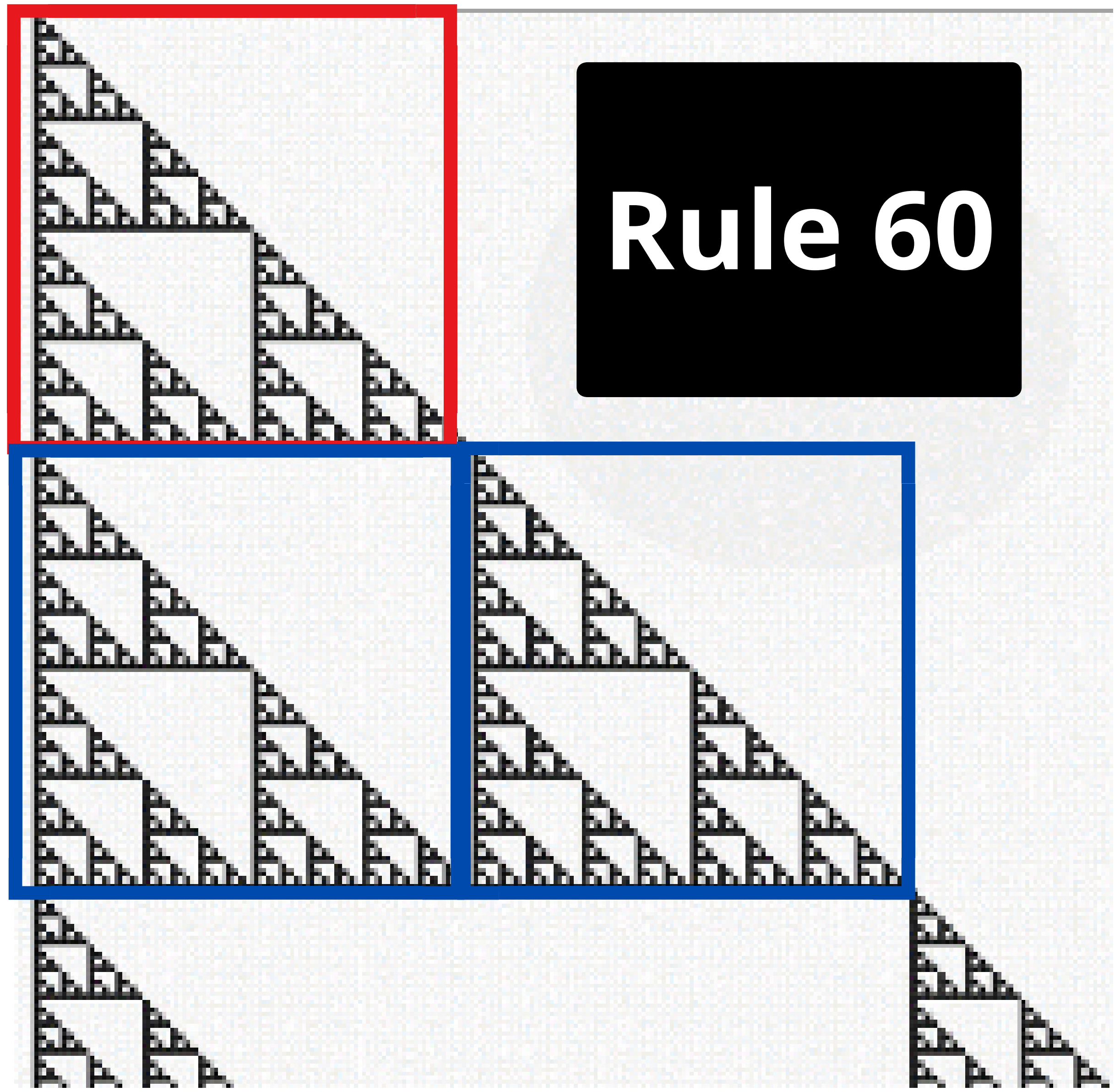
We can see that a column of black squares is formed below the initial black square, so the same rules apply:

3 white squares give a white square (000 => 0)

2 white squares and a black square on the right give a white square (001 => 0)

3. Our research: the Rule 60 automaton

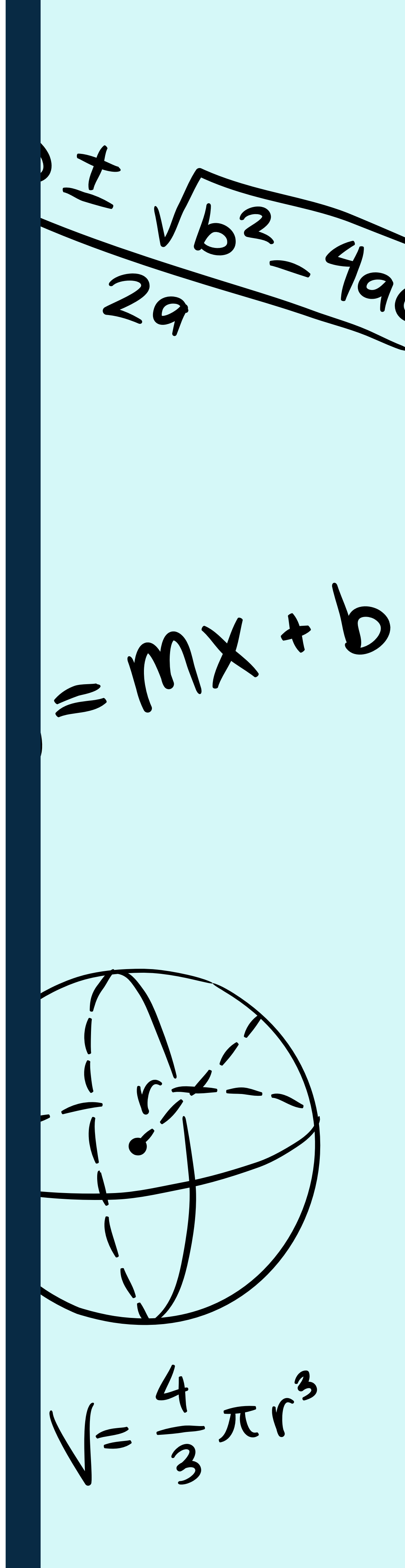
Method for determining the color of a square by knowing its coordinates



Our python™ code

```
from math import*
i= int(input("colonne :"))                                #demande de la colonne
j=int(input("ligne :"))                                   #demande de la ligne
def x(i,j):
    if i>=j or i<0 or (i==1 and j==3) :                   # définition de l'exterieur
        print("blanc")                                    # du triangle, de la colonne
    elif i==0:                                            # 0 et du motif initial
        print("noir")
    elif j==i+1 or j==4 :
        print("noir")
    else:                                                 # si on n'est ni dans le motif
        n=0                                               # ni à l'exterieur on rentre ici
        a=0
        while j > 4:                                      # tant qu'on n'est pas dans le
            if i>=j:                                      # motif initial
                print("blanc")
                return
            while 2**n<=j:                                # définition des puissances inf
                n=n+1                                     # et sup
            a=2**(n-1)
            j=j-a
            if i>=j and i<=a:                             # si dans le grand triangle
                print("blanc")                            # blanc
                return
            elif i>a:                                     # redéfinition du i
                i=i-a

        if i>=j :                                         # une fois sorti de la boucle
            print("blanc")                                # étude du motif initial
        elif i==1 and j==3:
            print("blanc")
        else:
            print("noir")
x(i,j)                                                    # résultat noir / blanc
```



WE’RE WAITING FOR YOU AT THE STAND 33 !

THANKS !!