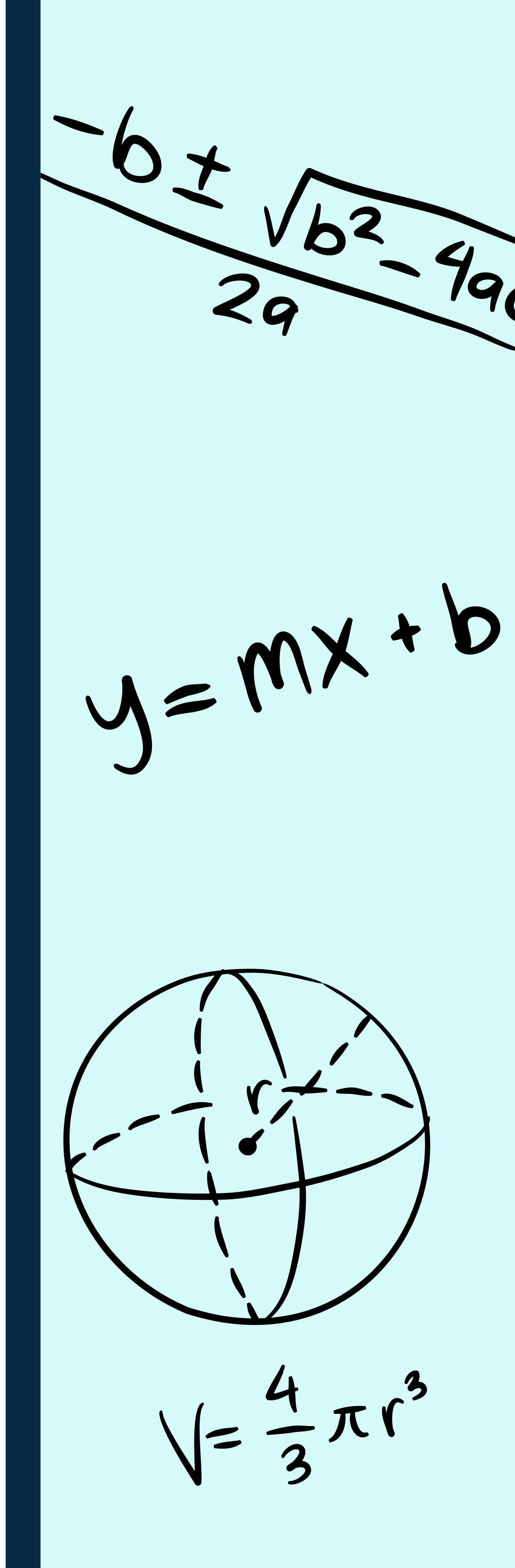
Lycée Adam de Craponne

Salon de Provence

L'AUTOMATE CELLULAIRE: RèGLE 60

PLAN DE LA PRESENTATION

1. Introduction
2. Notre chercheur, Étienne Moutot
3. Qu'est-ce qu'un automate cellulaire ?
4. Notre recherche : l'automate Règle 60



1. Introduction

Déc 2023 au labomaths

$$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

$$y = mx + b$$

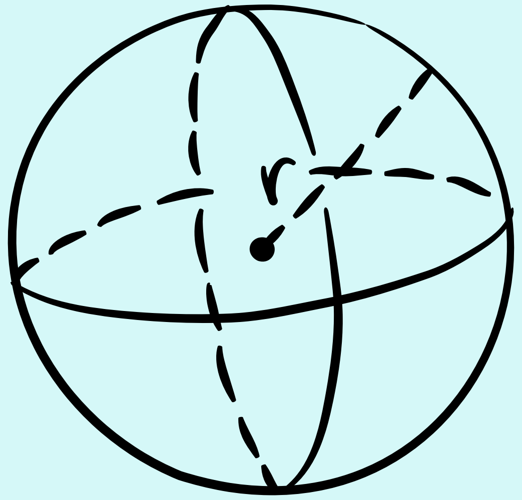
$$V = \frac{4}{3}\pi r^3$$

2. Notre chercheur: Etienne Moutot

- chargé de recherche au CNRS
- travaille à l'Institut de Mathématiques de Marseille
- s'intéresse aux pavages, à la calculabilité, à l'informatique

$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$y = mx + b$



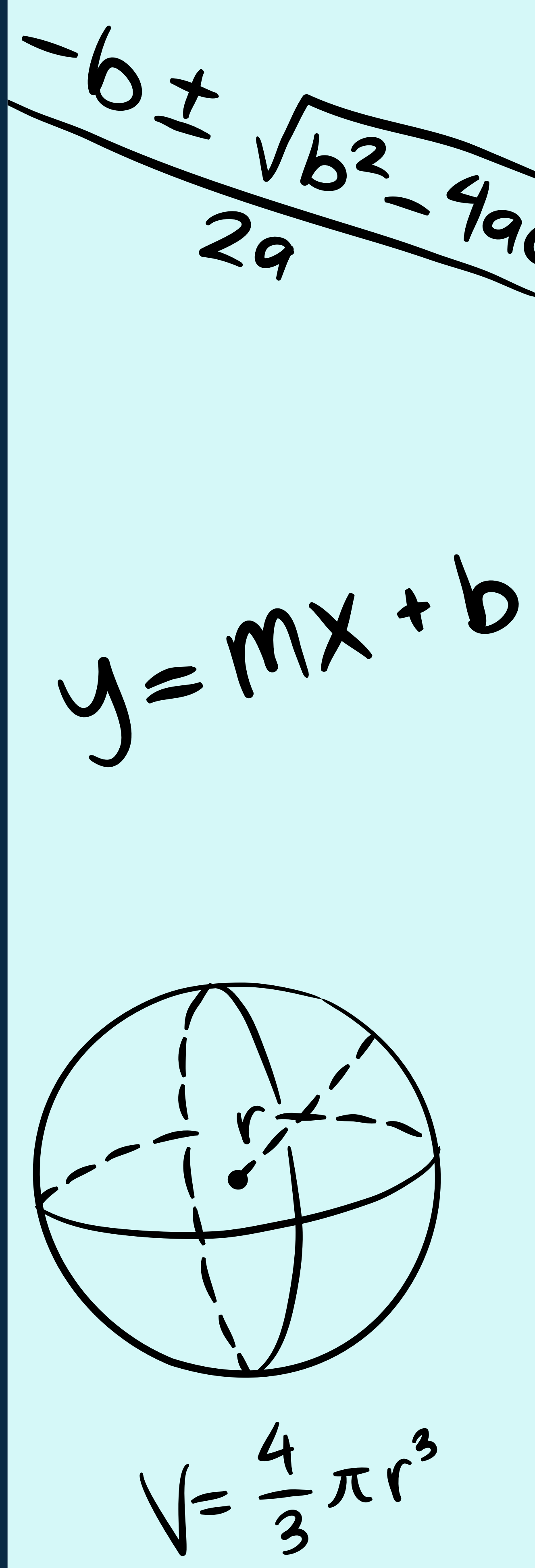
$V = \frac{4}{3}\pi r^3$

3. QU'EST-CE QU'UN AUTOMATE CELLULAIRE?

Origine des automates cellulaires

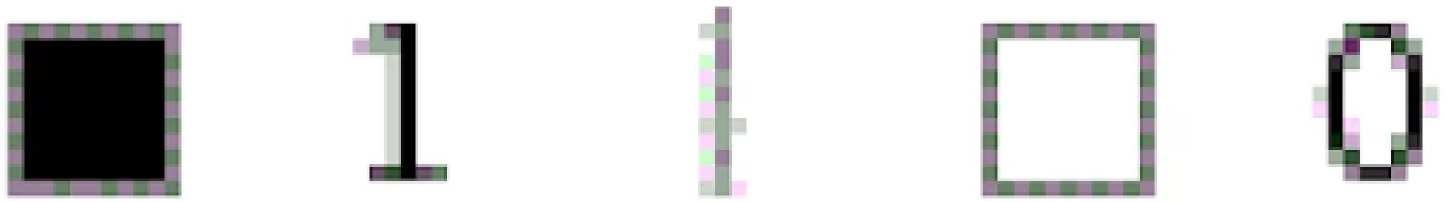
Stanislaw Ulam

John Von Neuman

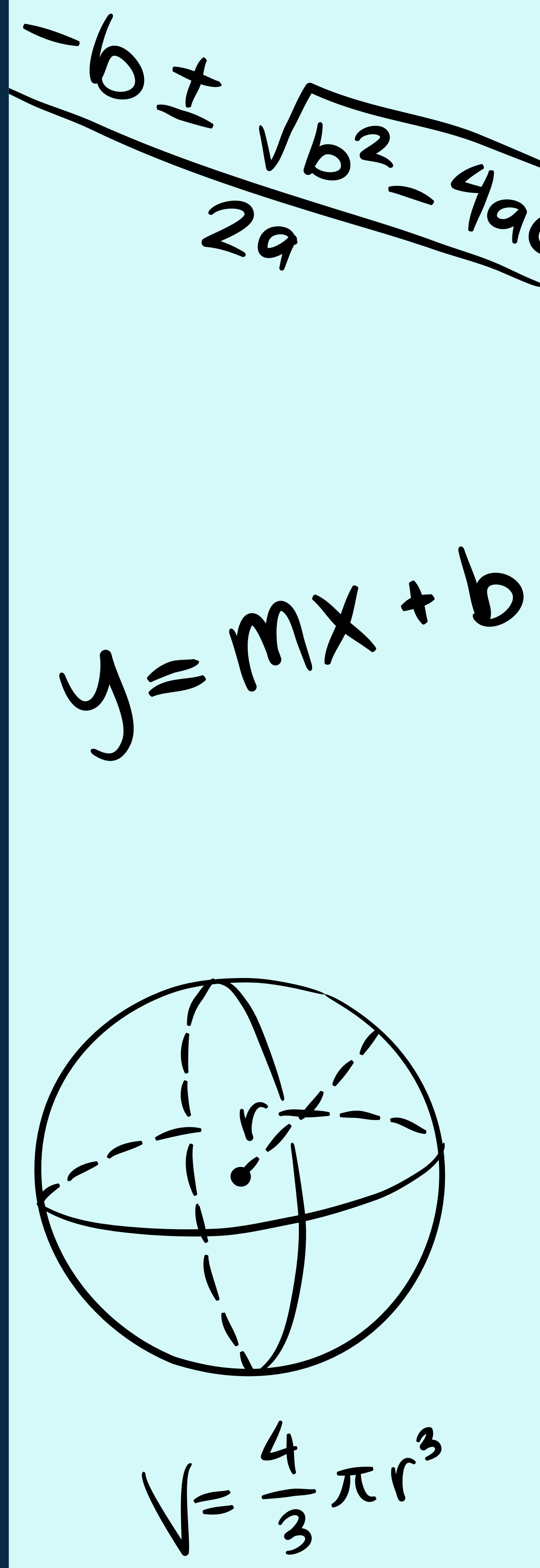


3. Qu'est-ce qu'un automate cellulaire?

Un automate cellulaire peut être vu comme une série de cellules évoluant selon un ensemble de règles définies, donnant donc lieu à une nouvelle génération de cellules.



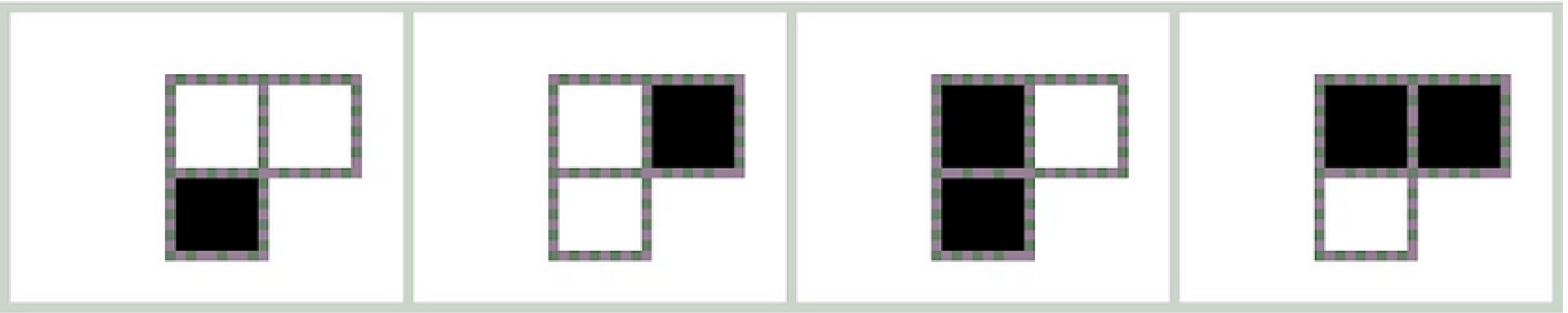
On définit souvent les règles d'évolution d'une cellule en fonction des voisines de celle-ci.

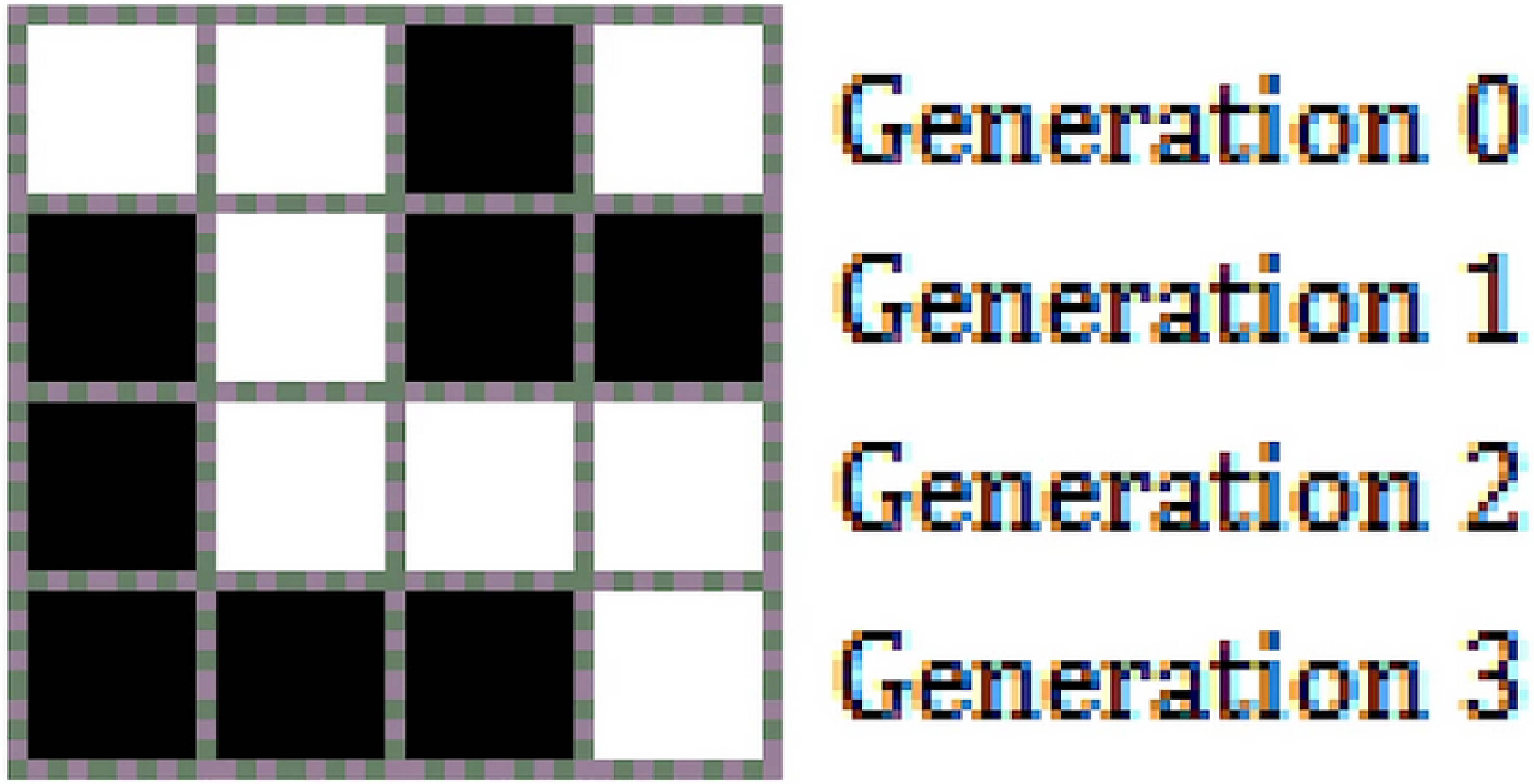


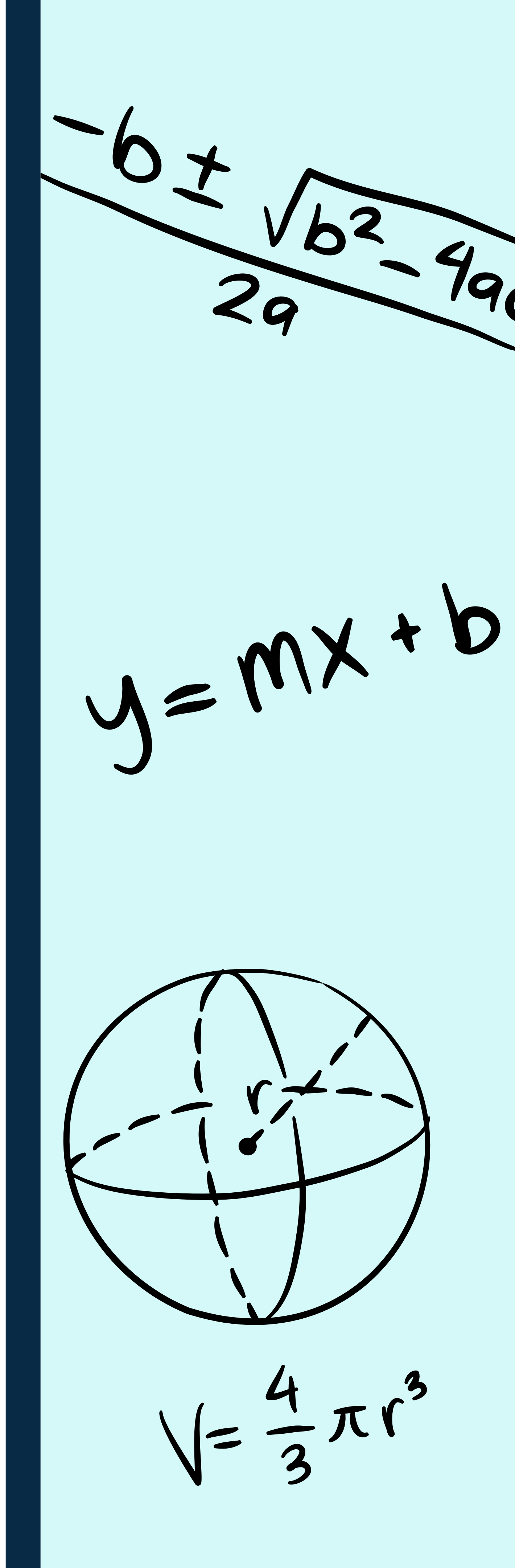
3. QU'EST-CE QU'UN AUTOMATE CELLULAIRE?

Un premier exemple

Règles du modèle



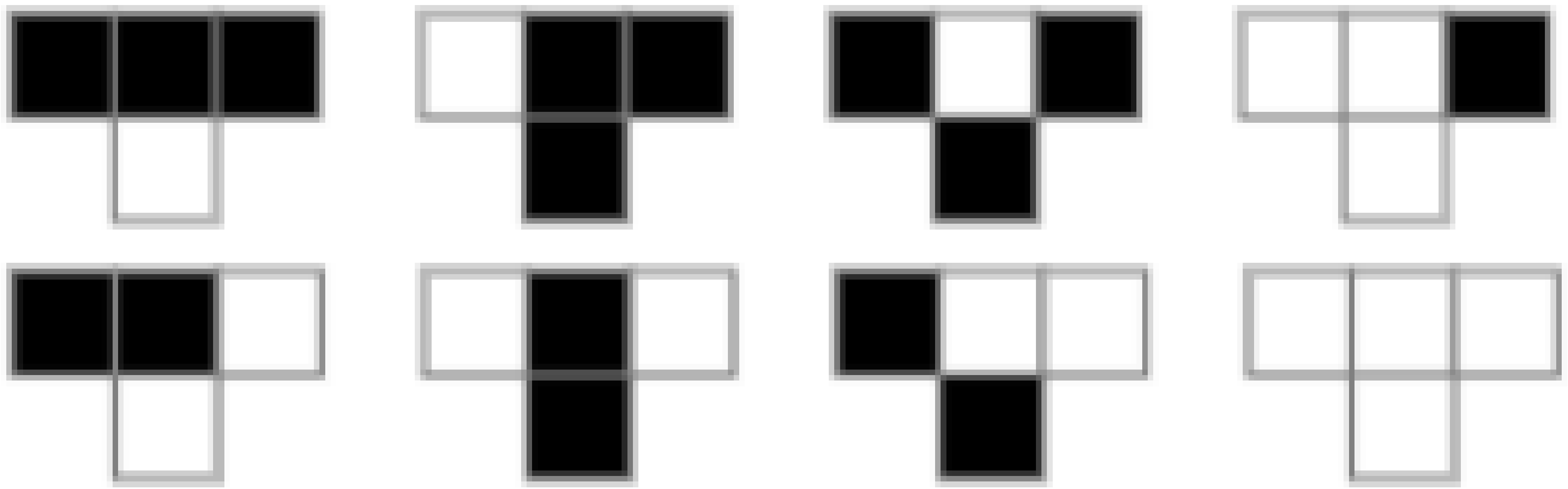




4. Notre travail de recherche: l'automate cellulaire 60

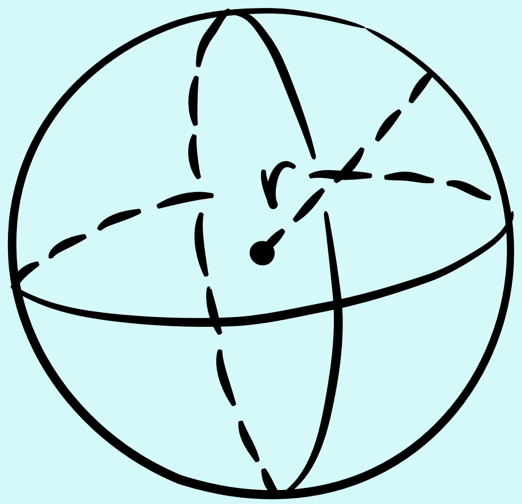
La ligne de départ comprend une case noire  et les autres sont blanches.

Les 8 règles de base



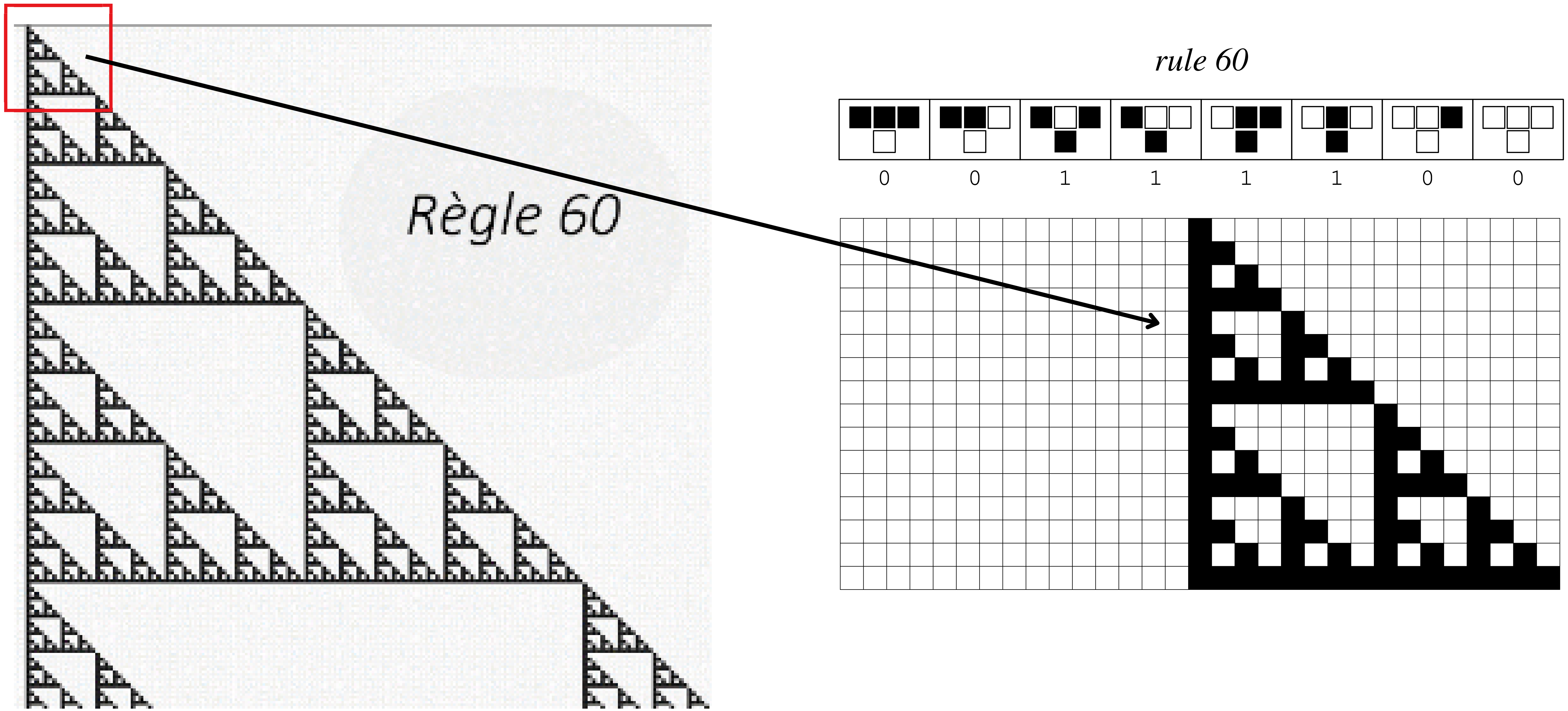
$$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

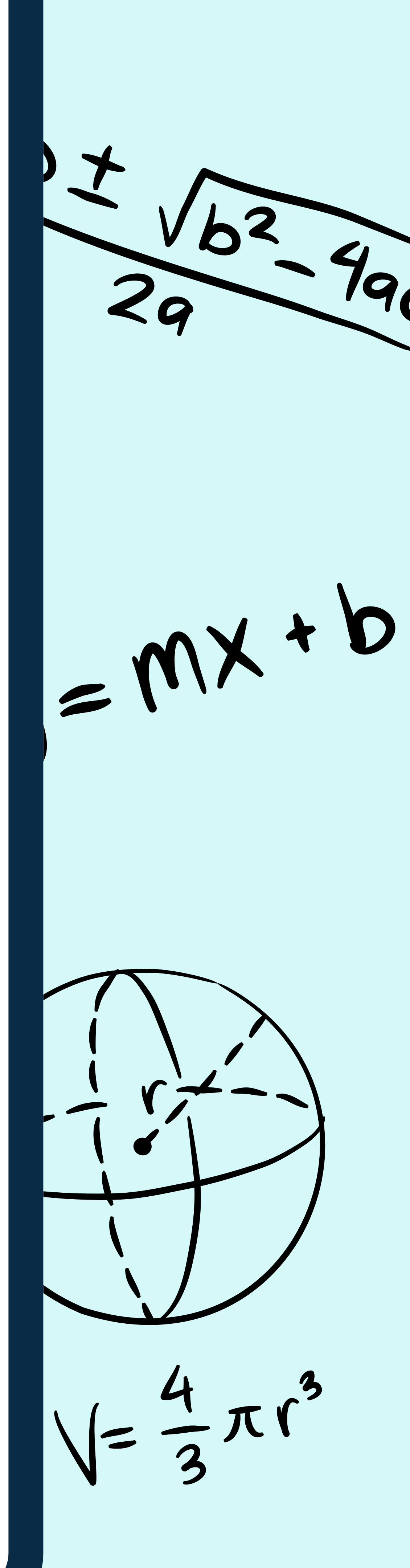
$$y = mx + b$$



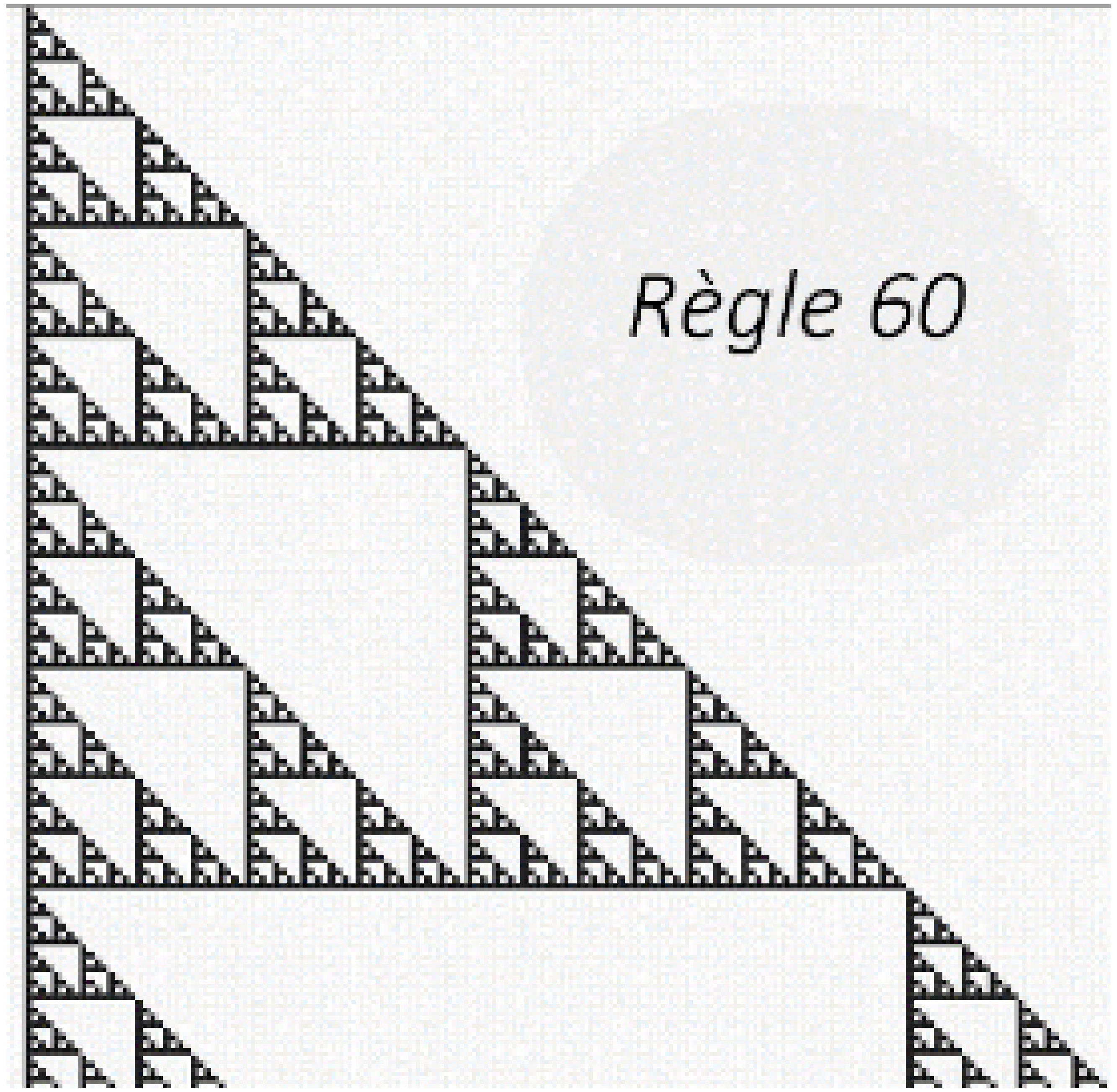
$$V = \frac{4}{3}\pi r^3$$

4. Notre travail de recherche: l’automate cellulaire 60





4. Notre travail de recherche: l'automate cellulaire 60

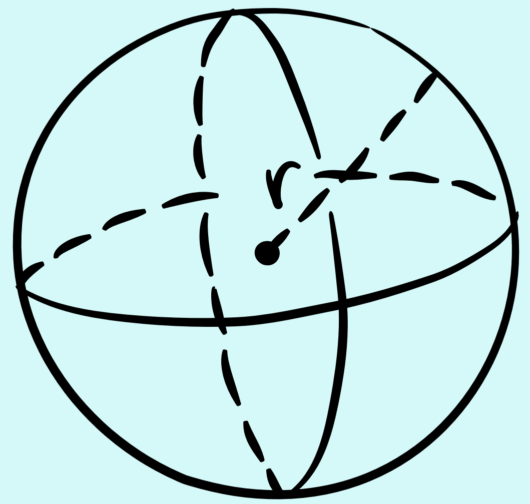


Conjectures:

- Une colonne noire se forme sous la case noire initiale
  À sa gauche, toutes les cases sont blanches
- Aux lignes 2, 4, 8, 16, etc, on a des "lignes noires"
- Le premier triangle noir est rectangle et isocèle
- Le triangle initial est copié en double en dessous, puis le nouveau motif est lui-même copié en double et ainsi de suite jusqu'à l'infini.

$\frac{-b \pm \sqrt{b^2-4ac}}{2a}$

$y = mx + b$



$V = \frac{4}{3}\pi r^3$

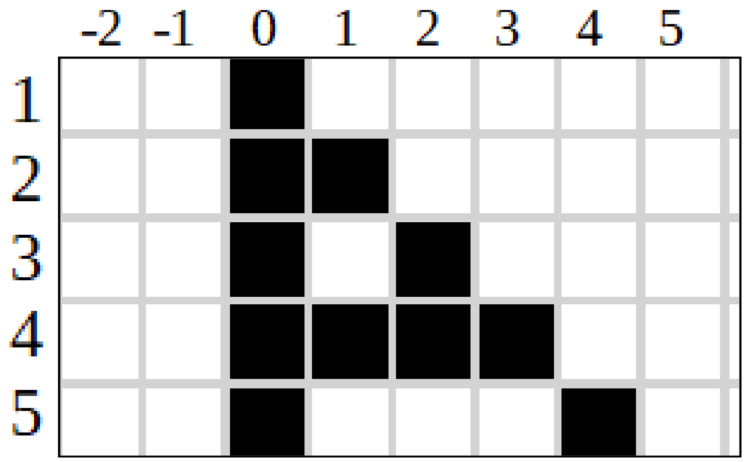
4. Notre travail de recherche: l'automate cellulaire 60

Notations

Pour toute case de l'automate, avec (i $\in \mathbb{Z}$ et j $\in \mathbb{N}^*$) :

- x : désigne une case de l'automate
- i : numéro de la colonne
- j : numéro de la ligne

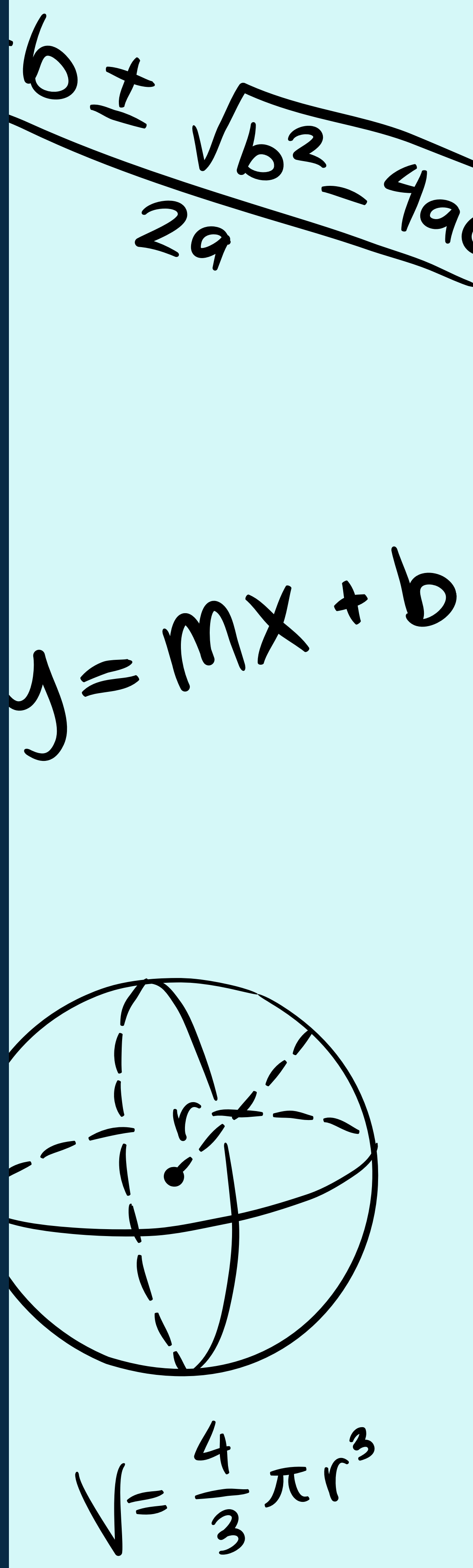
Case noire initiale de l'automate : colonne 0, ligne 1



- x(i;j) sera la valeur de la case située à la colonne i et à la ligne j.

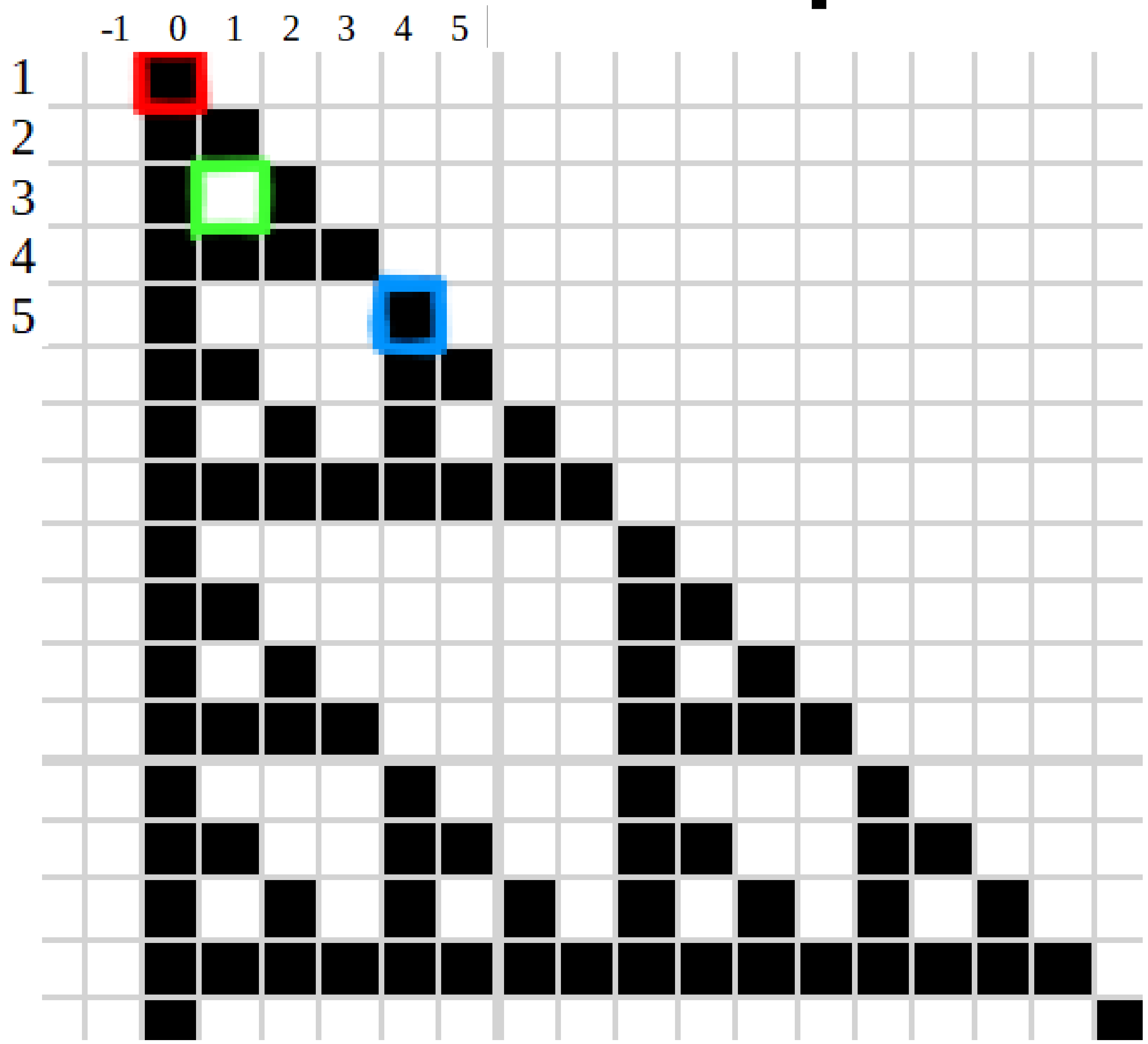
Si x(i ;j)=0, alors la case est blanche.
Si x(i ;j) =1, alors la case est noire.

N(j) : nombre de cases noires à la ligne j



4. Notre travail de recherche: l'automate cellulaire 60

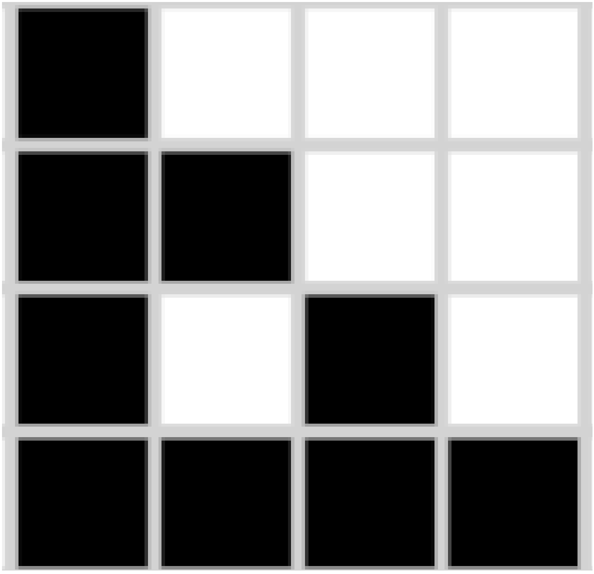
Exemples de notations



- La case rouge : x(0,1)=1
- La case verte : x(1,3)=0
- La case bleue : x(4,5)=1

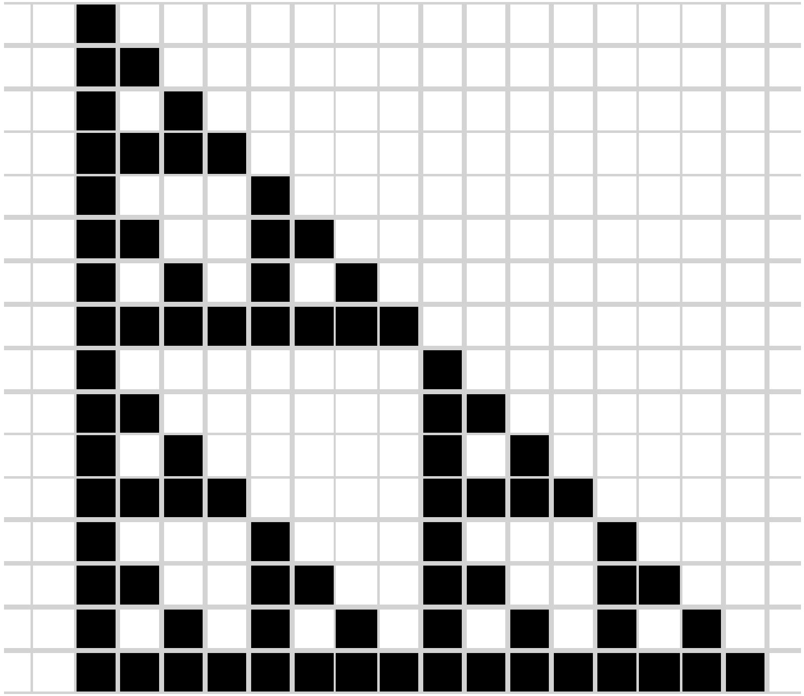
4. Notre travail de recherche: l'automate cellulaire 60

Quelques lemmes



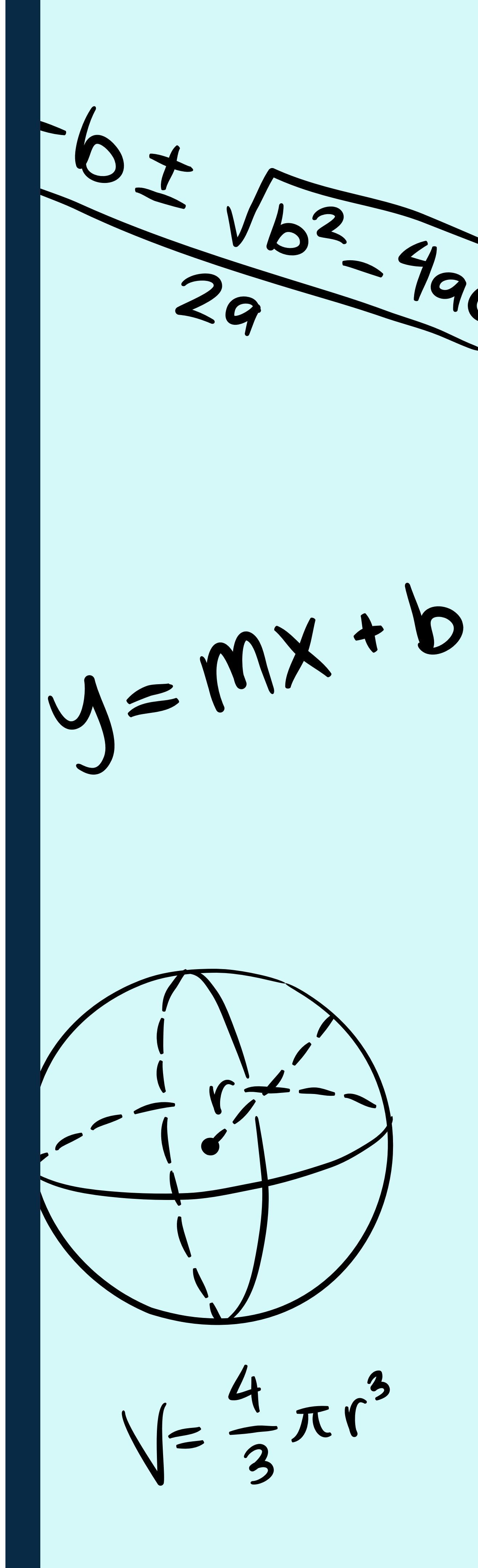
<u>Lemme 1</u>: il y a un motif de base qui se répète de façon régulière.

<u>Lemme 2</u> : À chaque nouvelle ligne, le nombre de cases de la figure augmente d'une unité.



<u>Lemme 3</u>: si $i \geq j$ alors $x(i;j) = 0$

<u>Lemme 4</u> : si $j = 2^n$, $n \in \mathbb{N}$, $N_j = j$.



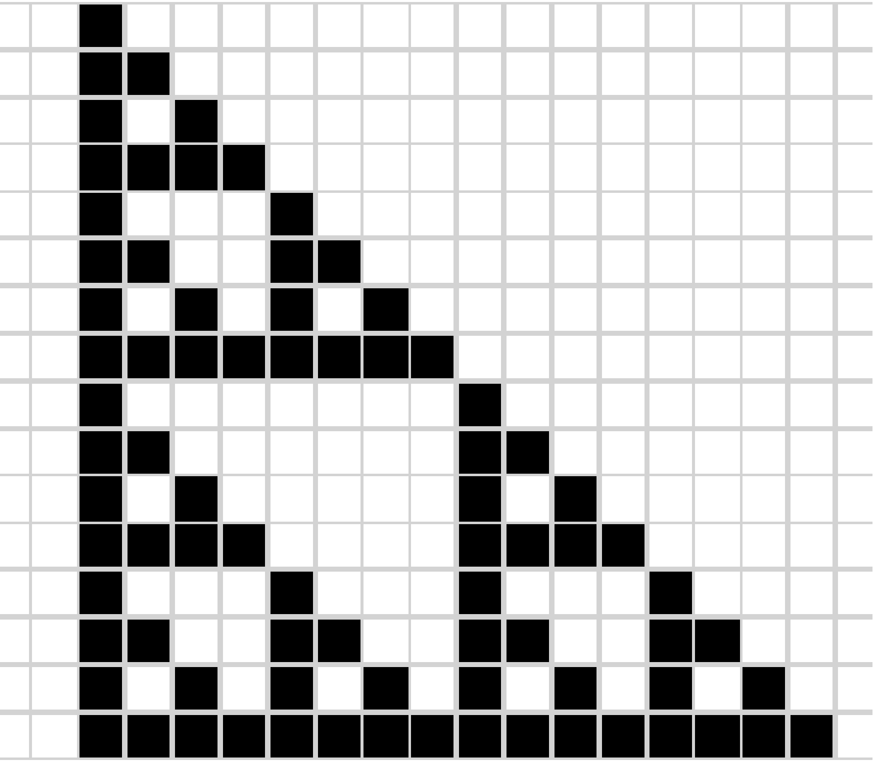
4. Notre travail de recherche: l'automate cellulaire 60

Une première démonstration

La colonne i=0 est toujours noire

Démonstration par récurrence

- Initialisation
- Hérédité
- Conclusion



Démonstration par récurrence :

-Propriété :

$\forall$ j $\in \mathbb{N}^*$, (Pn) est la proposition « x0 ; j = 1 ».

-Initialisation :

Pour j=1 : on a le motif initial qui génère une unique case noire :

X0 ; 1 = 1

Donc (P0) est vraie.

-Hérédité :

La colonne i = -1 est donc blanche $\Leftrightarrow$ x-1 ; j = 0.

Les cas suivants ne se produiront donc jamais car il y a une case noire en i appartenant $\mathbb{Z}^* \setminus \mathbb{N}$:

111 ; 110 ; 101 ; 100

Il nous reste donc à étudier :

011 ; 010 ; 001 ; 000

On suppose (Pk) vraie pour k $\in \mathbb{N}^*$: x0 ; k = 1

La case directement supérieure à la case étudiée est donc noire. On étudie la case x0 ; k =1.

On élimine les cas pour lesquels x0 ; k = 0. C'est-à-dire ces cas-là car il fournisse des case blanche.

001 ; 000

Il reste donc les cas :

011 ; 010

Dans les deux cas, grâce aux définitions de la règle 60, on obtient :

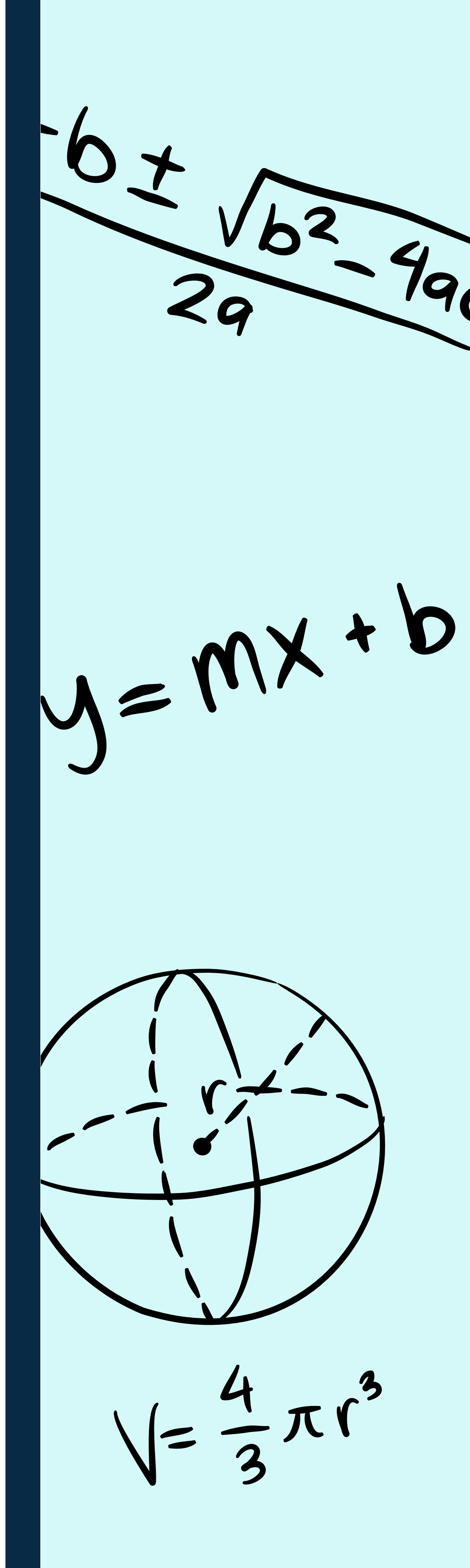
X0 ; k+1 =1

(Pk+1) est donc vraie.

-Conclusion :

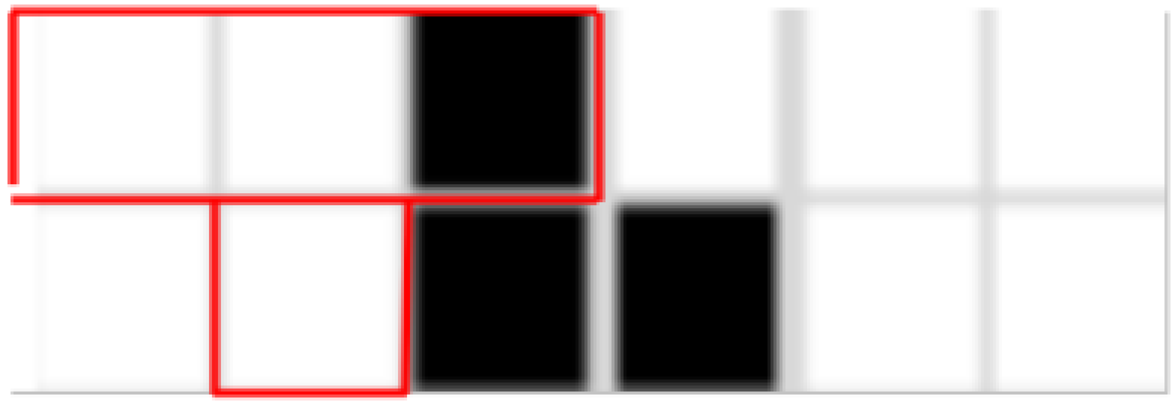
On a bien x0 ;j =1 , $\forall$j$\in \mathbb{N}^*$

La colonne 0 est noire, donc le motif s'étend à l'infini vers le bas.



4. Notre travail de recherche: l'automate cellulaire 60

Une deuxième démonstration



De la première à la seconde ligne :

3 cases blanches donnent une case blanche (000 => 0)

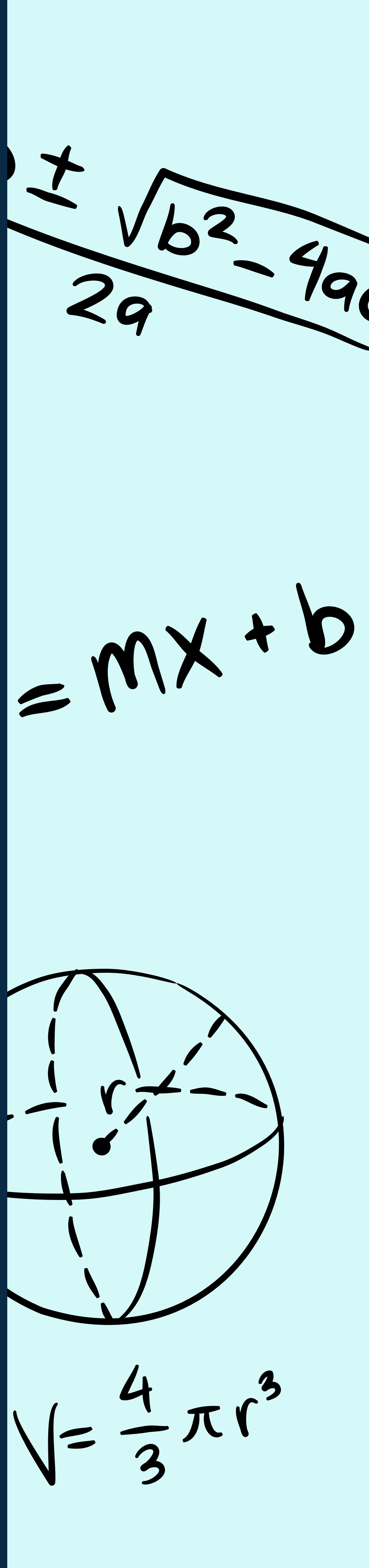
2 cases blanches et une case noire à droite donnent une case blanche (001 => 0)

Pour les lignes suivantes :

On voit qu'une colonne de cases noires se forme sous la case noire initiale, donc les mêmes règles s'appliquent :

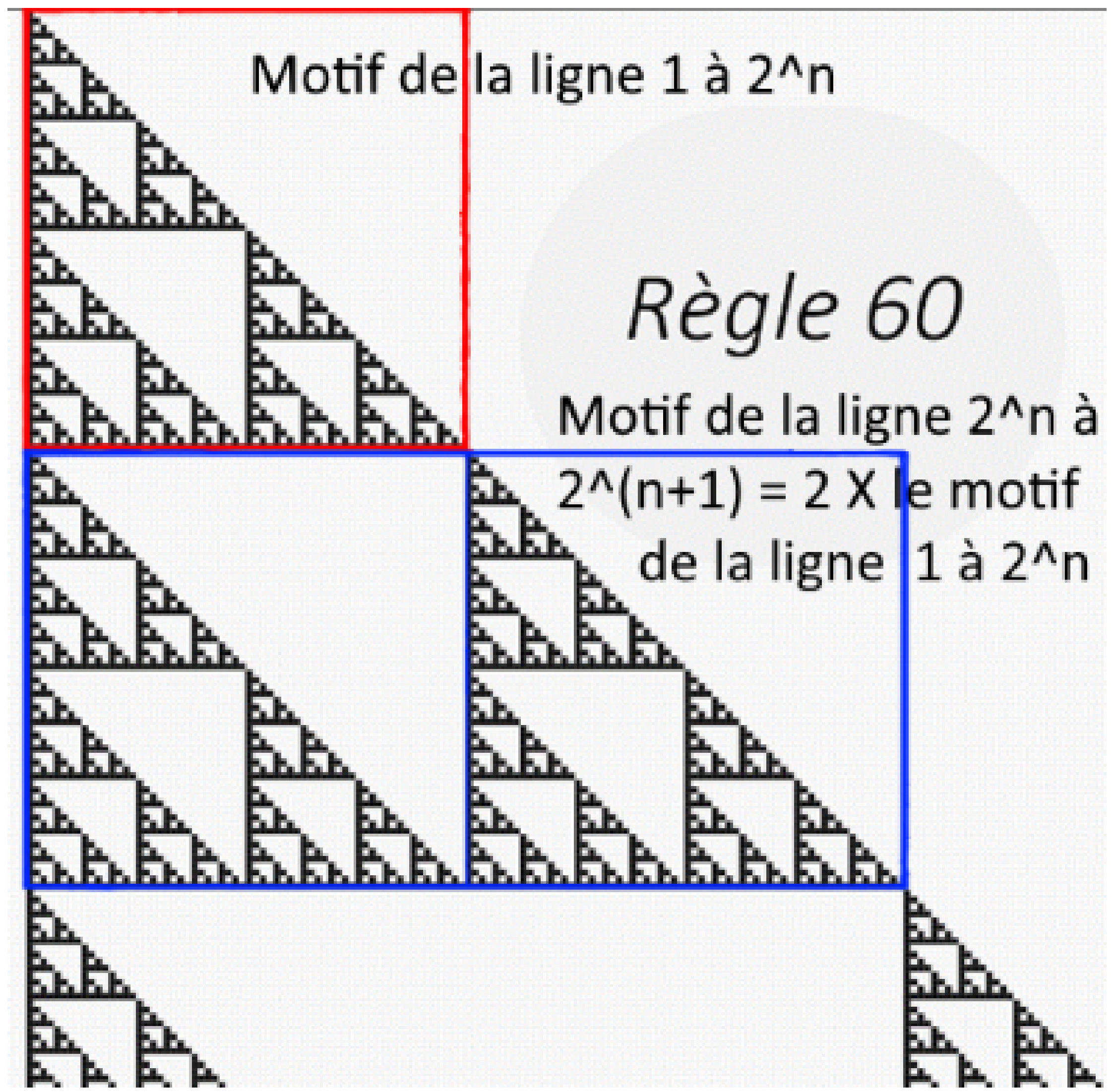
3 cases blanches donnent une case blanche (000 => 0)

2 cases blanches et une case noire à droite donnent une case blanche (001 => 0)

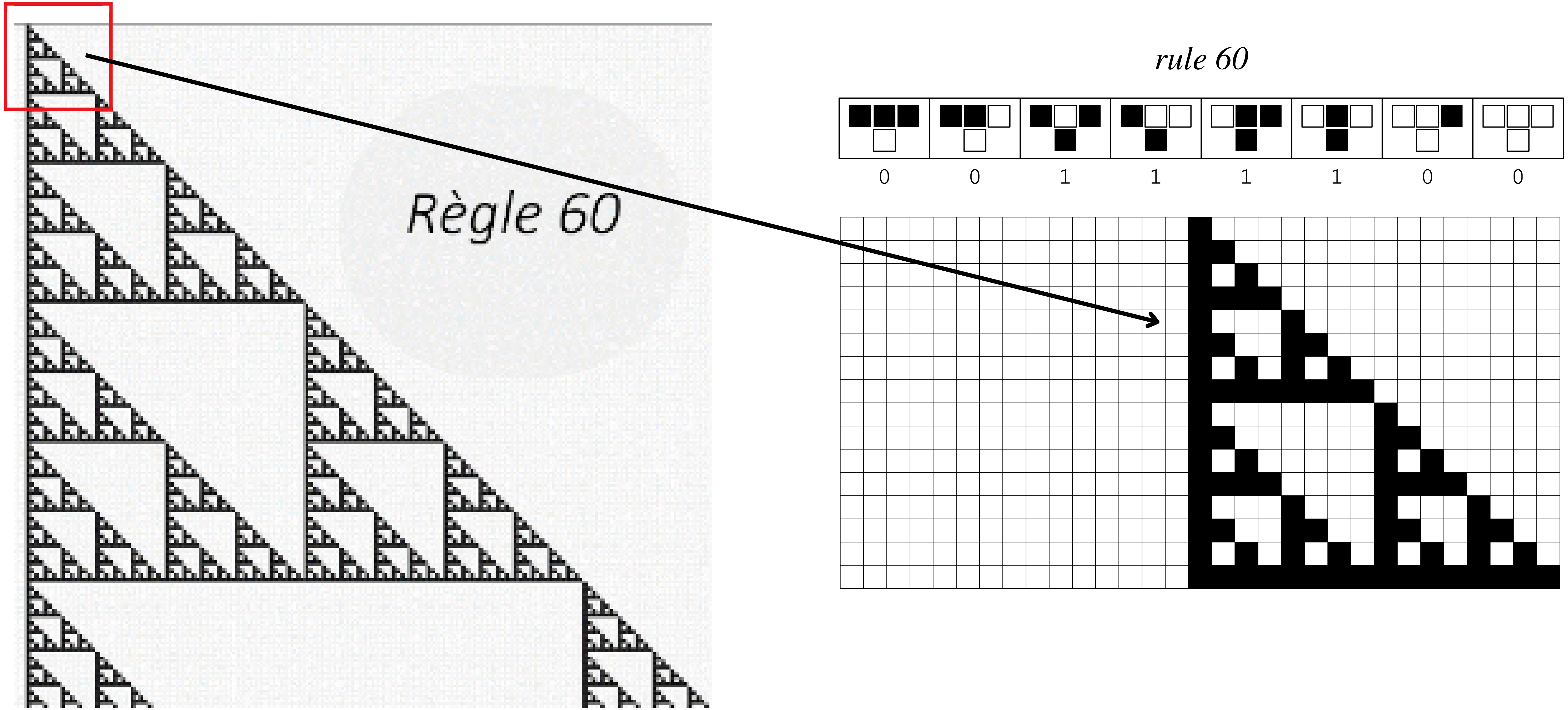


4. Notre travail de recherche: l'automate cellulaire 60

Méthode permettant de savoir la couleur d'une case en connaissant ses coordonnées



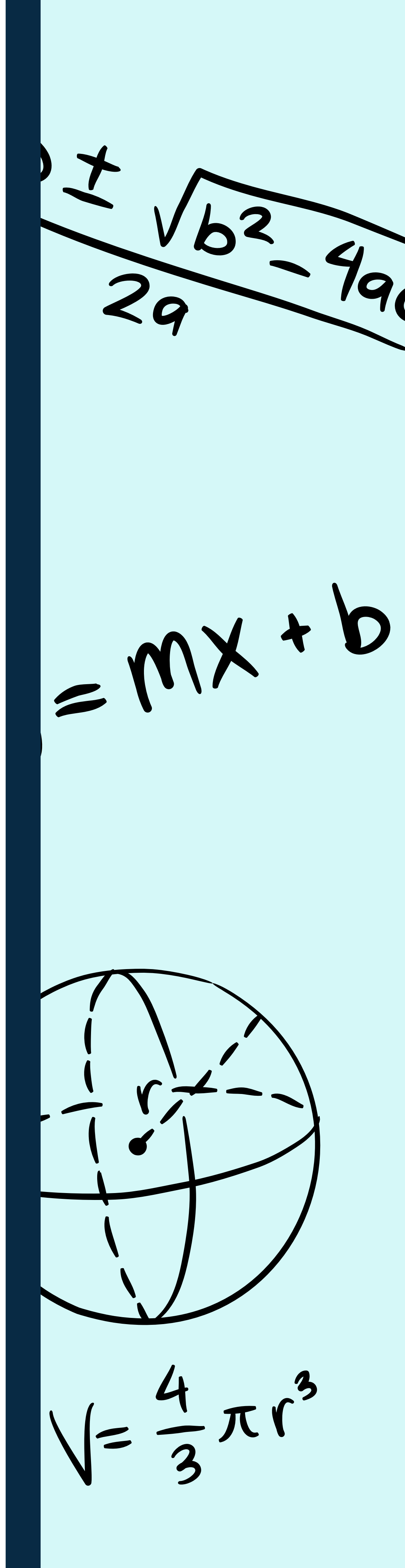
Notre code sur python™



Notre code sur python™

```
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
```



Merci pour votre attention