

AUTÓMATAS CELULARES

CON PYTHON

Autómatas Celulares en 1 Dimensión

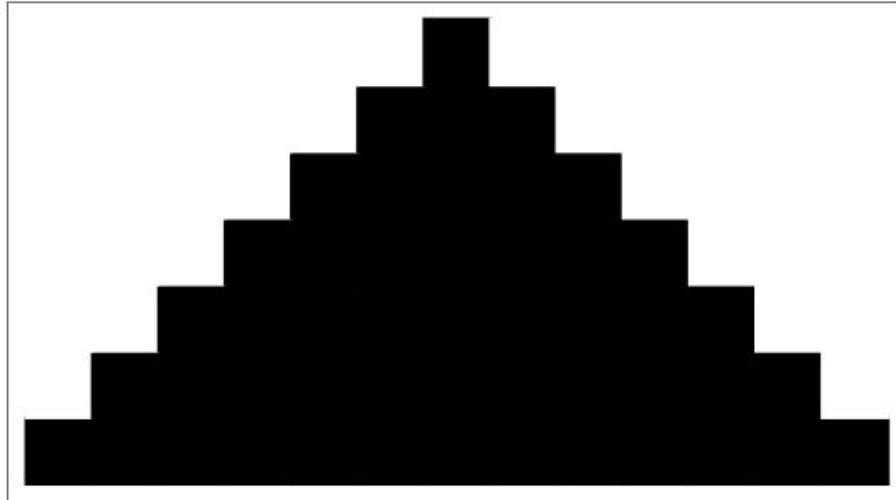
Imaginemos una red unidimensional donde cada celda j sólo puede tener dos estados; blanca o negra (0 o 1).

Esta red evolucionará en el tiempo, dando pasos discretos, a partir de una regla que define el estado de cada celda en función de su propio estado, y del estado de las celdas vecinas, un paso atrás en el tiempo.

Podemos representar el paso del tiempo en el eje vertical, donde la primera fila representa las condiciones iniciales del sistema.

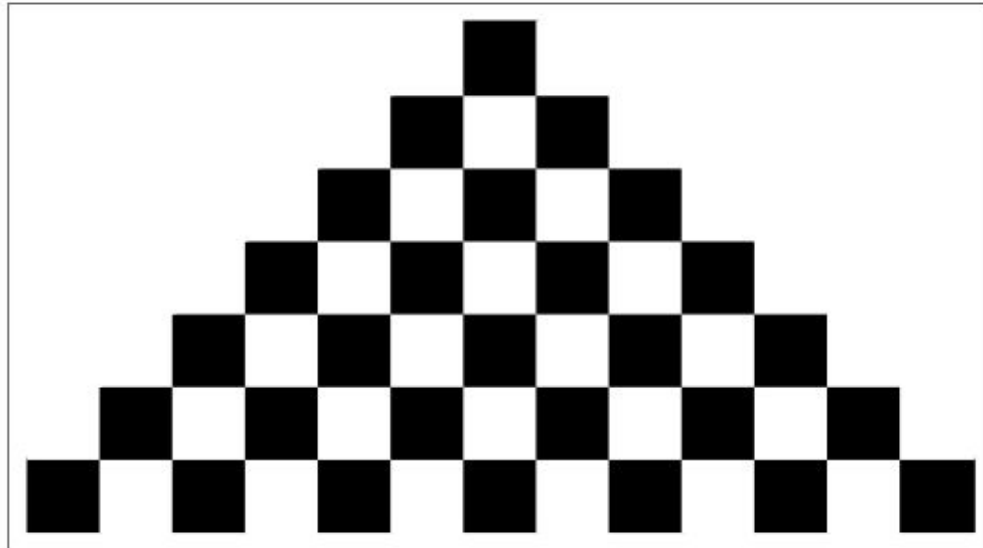
Autómatas Celulares en 1 Dimensión

Regla A: La celda j se activa si una o más del vecindario $[j - 1, j, j + 1]$ está activa. Si no, se inactiva (o queda inactiva). Veamos la evolución partiendo de una celda activa aislada:



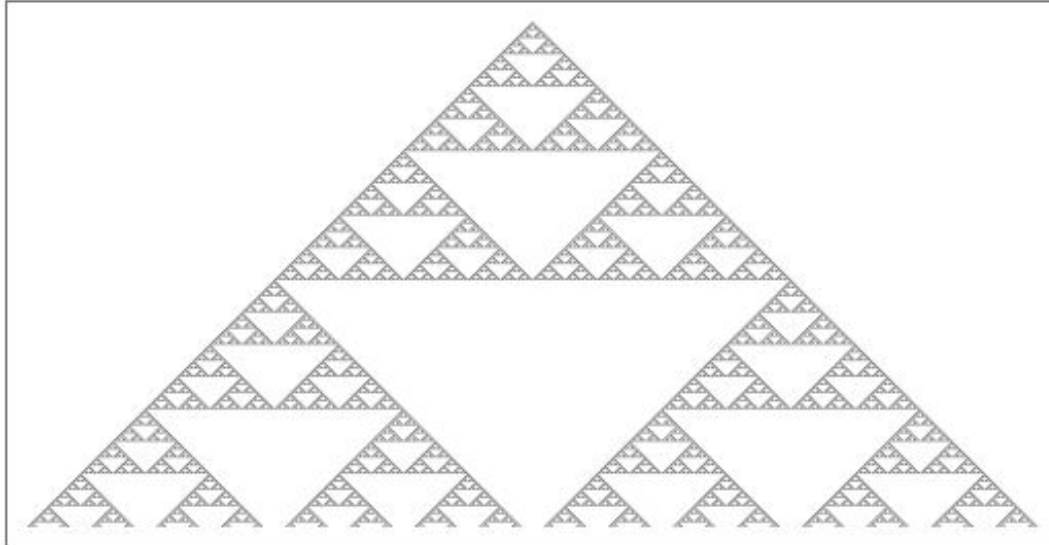
Autómatas Celulares en 1 Dimensión

Regla B. La celda j se activa si una o ambas vecinas $j - 1$ y $j + 1$ están activas. Si no, se desactiva. Partiendo de la misma celda activa aislada:



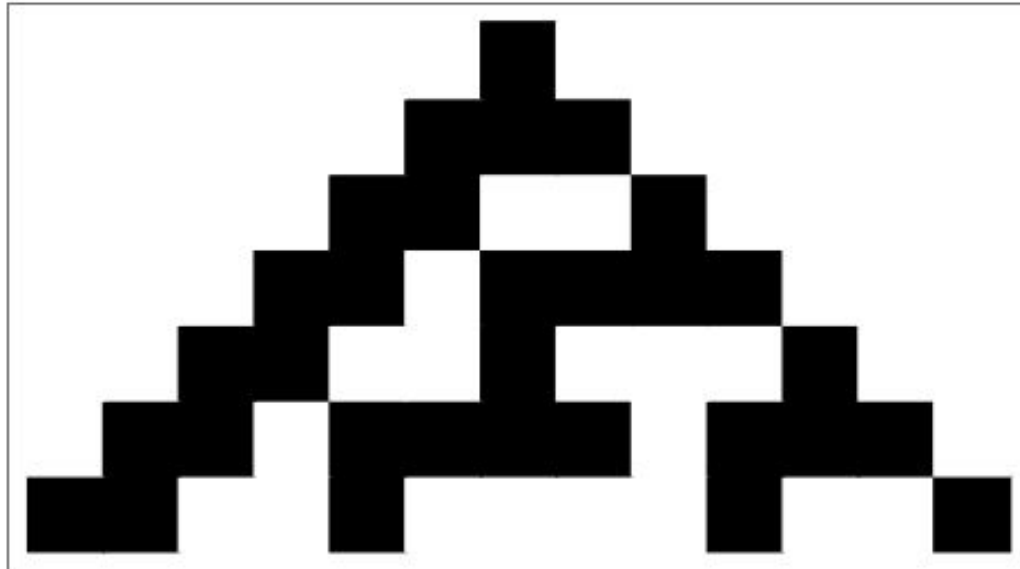
Autómatas Celulares en 1 Dimensión

Regla C. La celda j se activa si una y sólo una de sus vecinas $j - 1$ o $j + 1$ están activas. Si no, se desactiva.



Autómatas Celulares en 1 Dimensión

Regla D. La celda j se activa si una y sólo una del vecindario $[j - 1, j, j + 1]$ está activa, o si sólo j y $j + 1$ están activas. Si no, se desactiva.



Código de Wolfram

Se puede codificar cada regla partiendo de que existen 8 estados posibles para el vecindario $[j-1, j, j+1]$. Como resultado existen $2^8 = 256$ reglas posibles.

Estado Actual	111	110	101	100	011	010	001	000
Nuevo Estado de la Celda	1	1	1	1	1	1	1	0

Por ejemplo la regla de arriba es la **Regla A**, que podemos codificar como la regla $11111110_2 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 254$

Código de Wolfram

Ejemplo **Regla 30**. En python:

`bin(30) -> '0b11110'`

Estado Actual	111	110	101	100	011	010	001	000
Nuevo Estado de la Celda	0	0	0	1	1	1	1	0

Esta es la **Regla D**.

Código de Wolfram

current automaton contents



rule 30 (00011110)

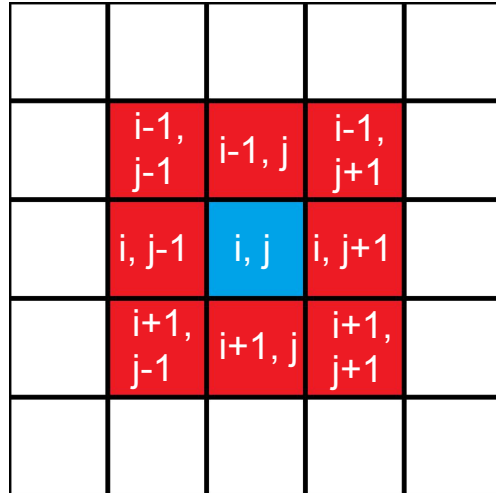


the next generation of the automaton



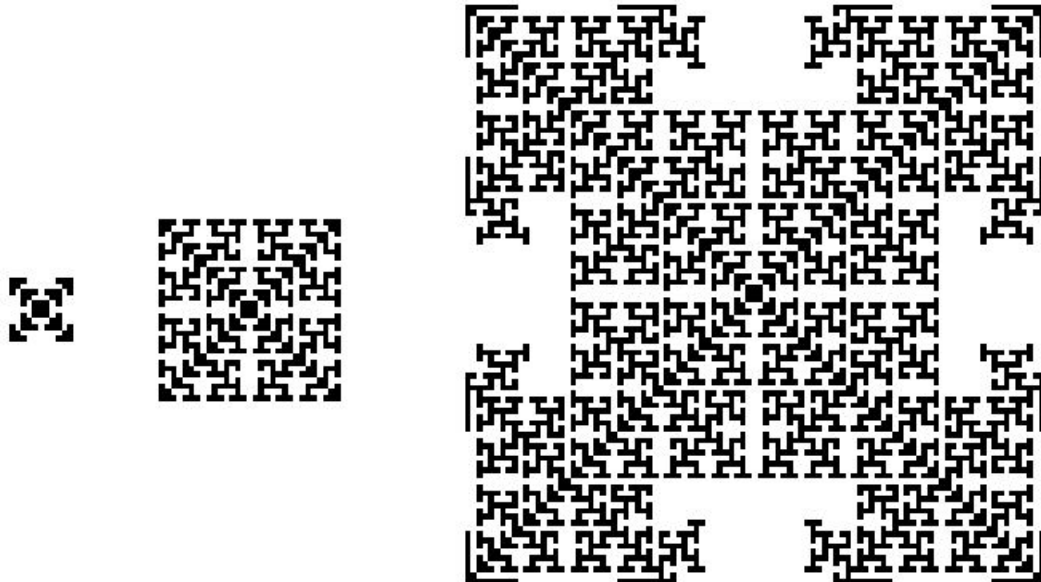
Autómatas Celulares 2D

Existen varios tipos de autómatas celulares en 2D. Nos vamos a restringir al llamado ***vecindario de Moore***. Vemos en rojo a los vecinos de la celda celeste:



Autómatas Celulares 2D

Una regla sencilla: Un nodo se activa si uno y sólo uno de sus vecinos está activo. Una vez activo, el nodo se queda así.

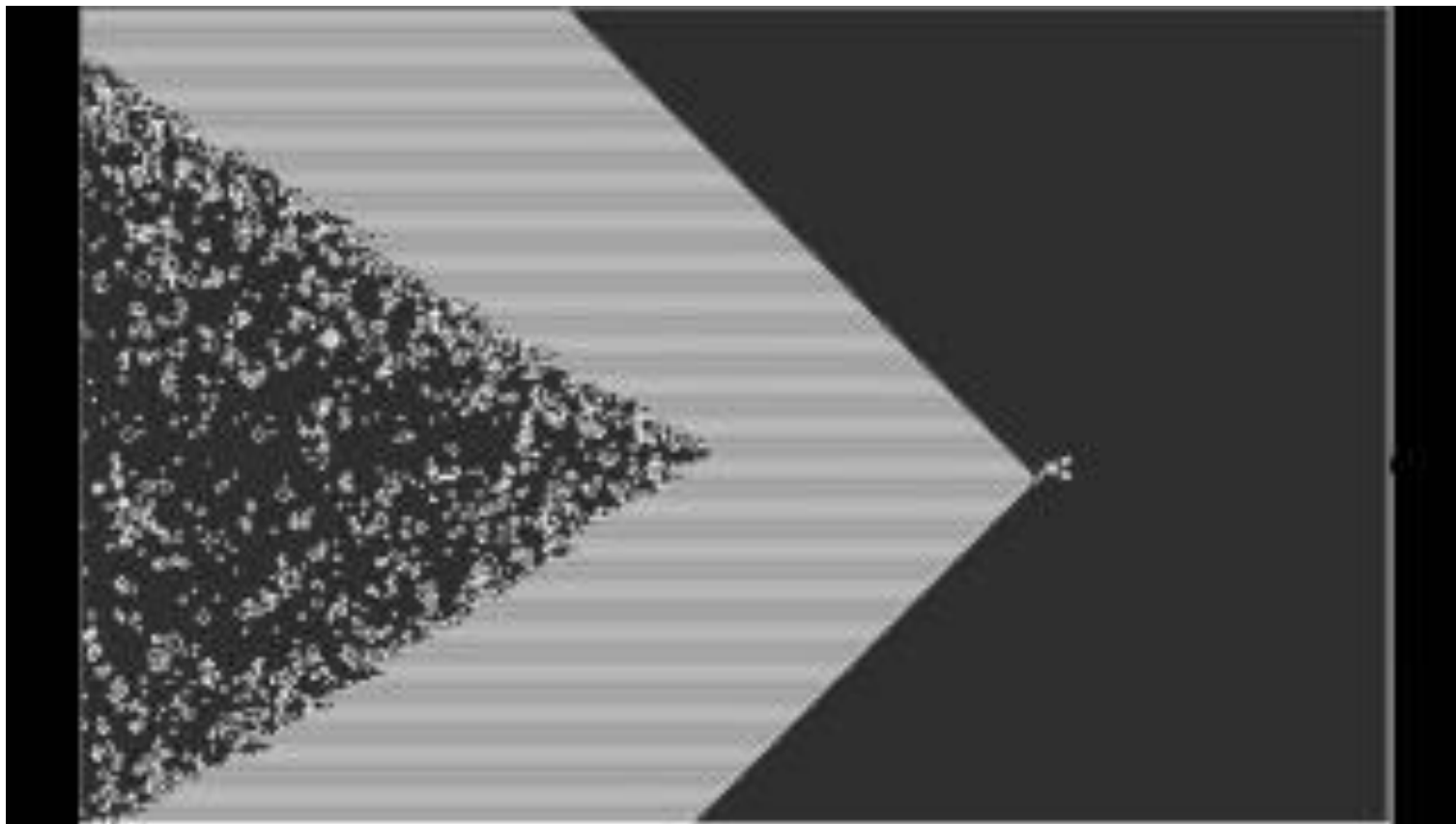


El Juego de la Vida de Conway

Tenemos las siguientes reglas:

1. Si un nodo activo tiene menos de dos vecinos activos, se inactiva.
2. Si un nodo activo tiene más de tres vecinos activos, se inactiva.
3. Si un nodo inactivo tiene tres vecinos activos, se activa.
4. Si un nodo tiene dos vecinos activos, se queda igual.

El Juego de la Vida de Conway



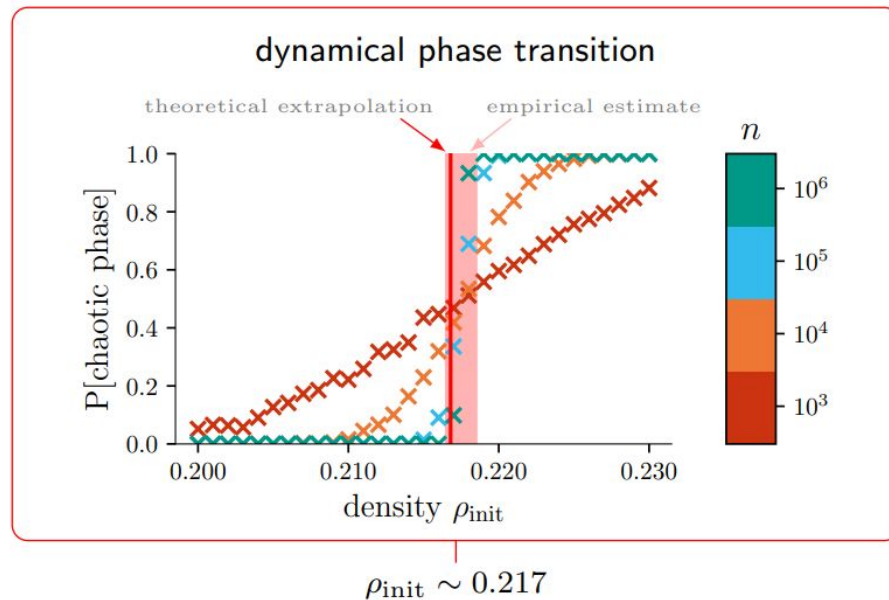
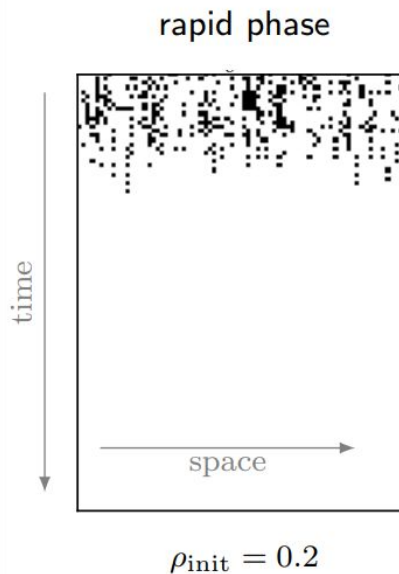
Aplicaciones

Dynamical Phase Transitions in Graph Cellular Automata

Freya Behrens,¹ Barbora Hudcová,^{2,3} and Lenka Zdeborová¹

¹*Statistical Physics Of Computation Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*

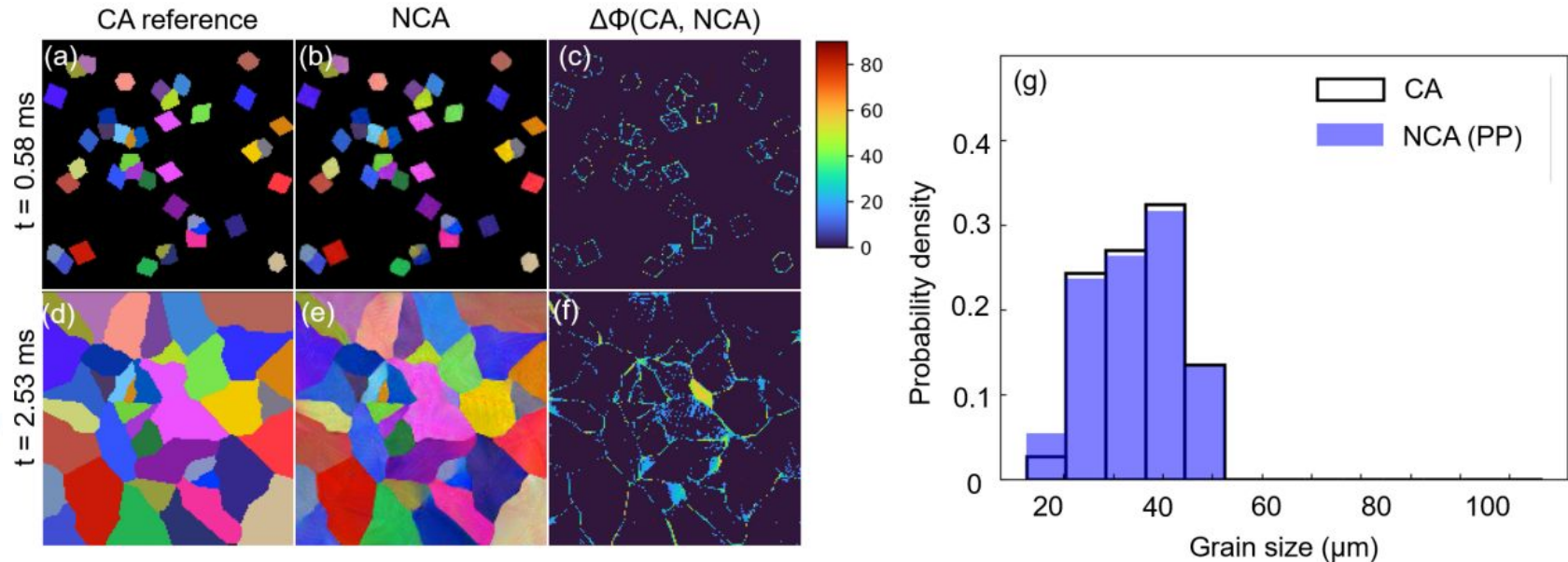
²*Algebra Department, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic*



Aplicaciones

Neural Cellular Automata for Solidification Microstructure Modelling

Ben Tanaka^{a,b}, Siddhant Kumar^c, Louis De Lorenz^{a,b,*}, Elham Hasselblad^{a,**}



VIDEOS



VIDEOS

