

Seminario de Ciencias de la Computación “A”

Modelación y simulación computacional basada en Agentes 2023-II

Práctica 1: “Implementación y análisis de autómatas celulares”

Parte 1. Preguntas

- ¿Qué es modelación basada en agentes?
- ¿Qué es el enfoque bottom-up?
- ¿Cuándo se usa el concepto de autoorganización en sistemas?
- ¿Qué es una propiedad emergente?

Parte 2. Autómata Celular Elemental (ACE)

Desarrollar un autómata celular elemental con el lenguaje de programación de su elección o utilizar como base el código en NetLogo desarrollado en las ayudantías. Implementar las siguientes características:

- Función de transición.** El ACE está definido por una vecindad de radio=1 y un alfabeto $k = \{0,1\}$ generando 256 posibles reglas (o tablas de transición). Implementar la función de transición la cual recibe el número de regla en decimal para establecer la tabla de transiciones en binario.
- Fronteras.** Implementar los dos tipos de fronteras, periódicas (toro) y fijas.
- Tamaño.** Definir la longitud y el tiempo de evolución como parámetros para la construcción del ACE. Para una correcta visualización se recomienda un tamaño mínimo de 100 celdas por 100 tiempos de evolución.
- Condiciones iniciales.** Definir la posibilidad de iniciar el autómata con condiciones aleatorias, simples (fija) o introducida por el usuario.
- Salida gráfica.** Como *input* del programa se introduce el número de regla en el intervalo $[0,255]$ y como *output* la visualización gráfica.

Ejercicios:

- Exploración.** Ejecute algunas de las 256 reglas del autómata celular a partir de la implementación de su programa con condiciones de frontera periódicas (cilindro), mínimo 100 tiempos de evolución y condiciones iniciales aleatorias. Identifique el tipo de dinámica que presenta y a que clase pertenece según las cuatro categorías de la Clasificación de Wolfram. Muestre 1 representante de cada clase y adjunte la “captura de pantalla” en el documento.
- Después de explorar todas o algunas de las 256 reglas con el programa que implementaste ¿Cuál crees que sea la clase más frecuente?
- Sensibilidad a las condiciones iniciales.** Encuentre un autómata de clase III, perturbe la condición inicial aleatoria por un bit, evolucione y trate de ver las diferencias entre el autómata perturbado y el normal. **Hint:** pueden tomar las diferencias entre los estados de las celdas y pintarlos en la posición correspondiente. ¿La sensibilidad a las condiciones iniciales es una propiedad suficiente para catalogar la dinámica como caótica? Explique. Adjunte las imágenes del ACE perturbado, sin perturbar y la diferencia de estados.
- ¿Qué poder de cómputo tiene la regla 132? Explique y adjunte captura de pantalla. ¿Qué propiedades tiene la regla 30? Explique y adjunte capturas de la simulación.
- En las siguientes imágenes se muestra la evolución de la regla 22 con condición inicial simple (una celda negra en el centro, Figura A) y condición inicial aleatoria (Figura B). Explique lo que sucede con la dinámica.

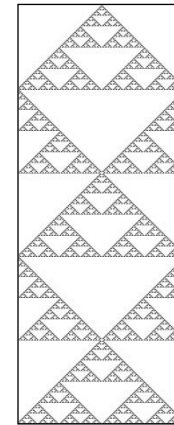


Figura A

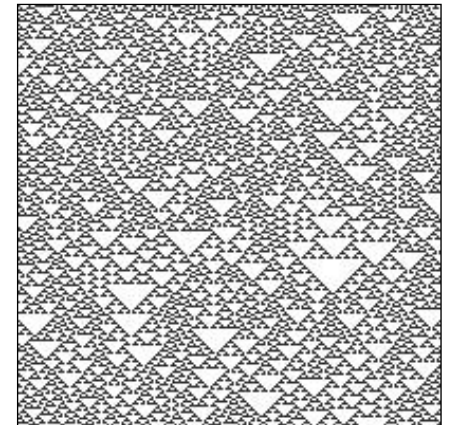


Figura B

- Si el autómata estuviera definido en un alfabeto de 3 estados considerando sus primeros vecinos, ¿cuántas posibles reglas hay?

Parte 3. Autómata Celular Bidimensional “LIFE”

El juego de la vida (LIFE), propuesto John Horton Conway, es uno de los ejemplos más representativos de un autómata celular bidimensional. Su conjunto de reglas genera una gran cantidad de patrones, que aún hoy en día, se siguen encontrando.

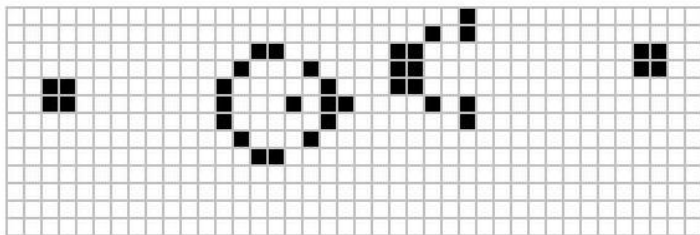
Implemente LIFE utilizando el código visto en clase o con el lenguaje de su elección, con las siguientes reglas de actualización, considerando una vecindad de Moore

1. Para una celda que esta “viva”
 - a) Si tiene uno o cero vecinos “muere” (por soledad).
 - b) Si tiene cuatro o más vecinos “muere” (por sobrepoblación).
 - c) Si tiene dos o tres vecinos “sobrevive”
2. Para una celda que esta “vacía” o “muerta”
 - a) Si tiene tres vecinos la celda “vive”.

El mínimo del tamaño de la retícula debe ser de 100x100. Implementar condición de frontera periódica y no periódica.

Ejercicios:

1. En NetLogo construya una gráfica de conteo de celdas vivas a través del tiempo. Ejecute varias veces su programa con una retícula de 100x100 celdas y condición aleatoria con 50% de celdas ocupadas, ¿en cuánto tiempo (ticks) se "estabilizó" la dinámica?, ¿para cualquier configuración inicial aleatoria pasará lo mismo? Explique y adjunte una captura de pantalla del “view” de agentes y la gráfica de conteo.
2. Realice el mismo experimento aumentando la densidad de la condición aleatoria a 85%, ¿sucede lo mismo? Explique.
3. Implemente la “pistola de planeadores” o “Gosper’s gun” con ayuda de la siguiente condición inicial:



Explique la dinámica que se genera, ¿es periódica? ¿qué implicaciones tiene controlar de alguna manera ciertas estructuras en función del tiempo?

4. ¿El juego de la vida es reversible? Explique.

Parte 4. Loop de Langton.

Christopher Langton mostró la capacidad de autoreproducción de los autómatas celulares a partir de los modelos de Von Neumann y Codd. El loop de Langton es un modelo simple el cual satisface los criterios de autoreproducción. Esta estructura logra su simplicidad almacenando su descripción en un “loop” dinámico.

El modelo se desarrolla en un autómata celular bidimensional con 8 posibles estados en sus celdas y una vecindad de Neumann

Condición inicial del sistema

```
2 2 2 2 2 2 2 2
2 1 7 0 1 4 0 1 4 2
2 0 2 2 2 2 2 2 0 2
2 7 2          2 1 2
2 1 2          2 1 2
2 0 2          2 1 2
2 7 2          2 1 2
2 1 2 2 2 2 2 2 1 2 2 2 2 2
2 0 7 1 0 7 1 0 7 1 1 1 1 2
2 2 2 2 2 2 2 2 2 2 2 2 2
```

Descripción general de los estados

Cada estado identifica de forma general alguna parte del loop de Langton.

- a) El estado “1” indican las celdas del “núcleo” o canal de datos.
- b) El estado “2” son las celdas “coraza”, el recubrimiento del canal de datos.
- c) Los estados “4”, “5”, “6” y “7” seguidas por un “0” son paquetes de información que viajan por el canal de datos, dependiendo de los

valores asignados en la tabla de transiciones será la función que realicen.

- d) El estado “3” en combinación con sus vecinos indican la apertura o cierre del canal de datos (no se utiliza en la condición inicial).

Tabla de transiciones

En la siguiente tabla se muestra los valores de la transición y la posición de cada valor de la celda dependiendo de su ubicación, “C” (center), “T” (top), “R” (right), “B” (bottom), “L” (left). Hay que considerar las posibles rotaciones sin tomar en cuenta “C”, es decir, CTRBL = CRBLT = CBLTR = CLTRB.

CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I	CTRBL->I
0000->0	02527->1	11322->1	20242->2	30102->1
00001->2	10001->1	12224->4	20245->2	30122->0
00002->0	10006->1	12227->7	20252->0	30251->1
00003->0	10007->7	12243->4	20255->2	40112->0
00005->0	10011->1	12254->7	20262->2	40122->0
00006->3	10012->1	12324->4	20272->2	40125->0
00007->1	10021->1	12327->7	20312->2	40212->0
00011->2	10024->4	12425->5	20321->6	40222->1
00012->2	10027->7	12426->7	20322->6	40232->6
00013->2	10051->1	12527->5	20342->2	40252->0
00021->2	10101->1	20001->2	20422->2	40322->1
00022->0	10111->1	20002->2	20512->2	50002->2
00023->0	10124->4	20004->2	20521->2	50021->5
00026->2	10127->7	20007->1	20522->2	50022->5
00027->2	10202->6	20012->2	20552->1	50023->2
00032->0	10212->1	20015->2	20572->5	50027->2
00052->5	10221->1	20021->2	20622->2	50052->0
00062->2	10224->4	20022->2	20672->2	50202->2
00072->2	10226->3	20023->2	20712->2	50212->2
00102->2	10227->7	20024->2	20722->2	50215->2
00112->0	10232->7	20025->0	20742->2	50222->0
00202->0	10242->4	20026->2	20772->2	50224->4
00203->0	10262->6	20027->2	21122->2	50272->2
00205->0	10264->4	20032->6	21126->1	51212->2
00212->5	10267->7	20042->3	21222->2	51222->0
00222->0	10271->0	20051->7	21224->2	51242->2
00232->2	10272->7	20052->2	21226->2	51272->2
00522->2	10542->7	20057->5	21227->2	60001->1
01232->1	11112->1	20072->2	21422->2	60002->1
01242->1	11122->1	20102->2	21522->2	60212->0
01252->5	11124->4	20112->2	21622->2	61212->5
01262->1	11125->1	20122->2	21722->2	61213->1
01272->1	11126->1	20142->2	22227->2	61222->5
01275->1	11127->7	20172->2	22244->2	70007->7
01422->1	11152->2	20202->2	22246->2	70112->0
01432->1	11212->1	20203->2	22276->2	70122->0
01442->1	11222->1	20205->2	22277->2	70125->0
01472->1	11224->4	20207->3	30001->3	70212->0
01625->1	11225->1	20212->2	30002->2	70222->1
01722->1	11227->7	20215->2	30004->1	70225->1
01725->5	11232->1	20221->2	30007->6	70232->1
01752->1	11242->4	20222->2	30012->3	70252->5
01762->1	11262->1	20227->2	30042->1	70272->0
01772->1	11272->7	20232->1	30062->2	

T
L C R
B ==> I

Ejercicios

- 1) Implemente en el lenguaje de programación de su preferencia el Loop de Langton. Utilice colores para representar los valores de los estados. El autómata debe tener por lo menos una longitud de 150x150 para observar adecuadamente el crecimiento de una colonia de loops.
- 2) Muestre la captura de pantalla en el tiempo 151. Aquí se cumplen dos generaciones del loop de Langton.
- 3) Muestre la captura de pantalla en la generación 7, ¿cuántos loops hay “muertos”? ¿cuántos “moribundos”? y ¿cuántos en fase de reproducción?
- 4) Puede deducir una fórmula para el crecimiento de la colonia de los loops.

Hormiga de Langton (punto extra)

Implemente un autómata bidimensional de n x n celdas (n > 100) que modele el comportamiento de la hormiga de Langton:

1. Si la hormiga está parada en una celda blanca, pinta la celda negra, gira 90 grados a la izquierda y avanza una celda.
2. Si la hormiga está en una celda negra, pinta la celda blanca, gira 90 grados a la derecha y avanza una celda.

a) Evolucione el sistema para observar como la hormiga artificial construye la carretera, muestre la captura de pantalla. ¿En qué tiempo exactamente inicia la construcción?

b) Grafique como cambia la cantidad de celdas en negro a través del tiempo. Para cada tiempo de evolución realice el conteo de celdas negras sobre el sistema, y grafique, ¿Qué sucede cuando la hormiga construye la carretera?

c) ¿Es reversible el sistema?, es decir, ¿existe algún mecanismo para hacer que la hormiga regrese por la trayectoria recorrida hasta el estado inicial? Explique, indique cuáles serían estas reglas.

d) Encuentra una configuración mínima o condición inicial mínima para que la hormiga construya la carretera casi inmediatamente. En su simulador implemente este caso para ver el desempeño de la hormiga.

e) Dada cualquier condición inicial, ¿la hormiga construirá la carretera?

Lineamientos para la entrega

La práctica se realiza de manera individual.

Entregar un empaquetado (.tgz, .zip, etc.) con la siguiente estructura:

```
ApellidoPaterno_ApellidoMaterno
|----- Practica1
|          |----- fuentes
|          |----- readme.txt
|          |----- solución a ejercicios.pdf
```

Fuentes: código del programa, ejecutables, etc.

Readme.txt: explicar brevemente como se ejecuta el programa, y la forma en cómo podemos cambiar los valores de los parámetros.

SolucionEjercicios.pdf: En este archivo incluir todas las respuestas de la práctica junto con las gráficas y comentarios extras (si los hay). Este es el documento que se evaluará principalmente.

Subir la práctica en el Google Classroom del grupo:

<https://classroom.google.com/u/0/c/NTq2NjcyNjA4MjY40>

Fecha límite de entrega: 12 de marzo de 2023.

Las prácticas que se entreguen después de la fecha de entrega tendrán una penalización en la calificación.

Contacto

Gustavo Carreón gcarreon@unam.mx

Marco Antonio Jiménez Limas marcojimenez@ciencias.unam.mx