Universidad
Nacional Autónoma de México

Facultad de Ciencias

Seminario de Ciencias de la Computación A.
Modelación y simulación computacional basada en agentes

Practica 2: "Análisis de modelos basados en agentes."

Integrante:
Alumno David Pérez Jacome
Numero de cuenta: 316330420

Profesor: Gustavo Carreón Vázquez
Ayudante: Marco Antonio Jiménez Limas

16 Abril, 2023

Practica 2: Análisis de modelos basados en agentes.

Parte 1. Modelo de segregación de Schelling.

El modelo propuesto originalmente por Thomas Schelling consiste en dos grupos de agentes, por ejemplo rojos y verdes, que localmente tratan de satisfacer la necesidad de estar con los de su mismo grupo. De manera general este comportamiento es establecido por un parámetro conocido como porcentaje de **similitud-requerida** o nivel de tolerancia.

Los agentes toman una desición apartir de la información que tienen en su vecindad. Si el agente satisface las condiciones del entorno entonces se queda en su posición actual, de lo contrario se mueve a otra posición vacia. Esta dinamica local genera como resultado la formación de cúmmulos de agentes del mismo tipo, hay segregación.

Definición del Sistema: El sistema se compone de una reticula de n x n donde n se establece entre 50 y 100. Cada celda con posición (i, j) alberga a un agente rojo o verde. El sistema tiene un parámetro de **densidad poblacional** usualmente se establece en 90 % (es decir, 10 % de las celdas quedan vacias). La mitad de la población es color rojo y la otra mitad, color verde. Cada agente toma una celda aleatoriamente.

Dinámica: Cada agente en la posición (i, j) se "muda.ª un lugar vacio si su vecindad de Moore no cumple con el porcentaje de similitud requerida.

Ejercicios:

1. **Implmente el modelo** de segregación de Schelling original, pueden usar de base el codigo visto en clase.
2.

Parte 2. Automata Celular Elemental (ACE)

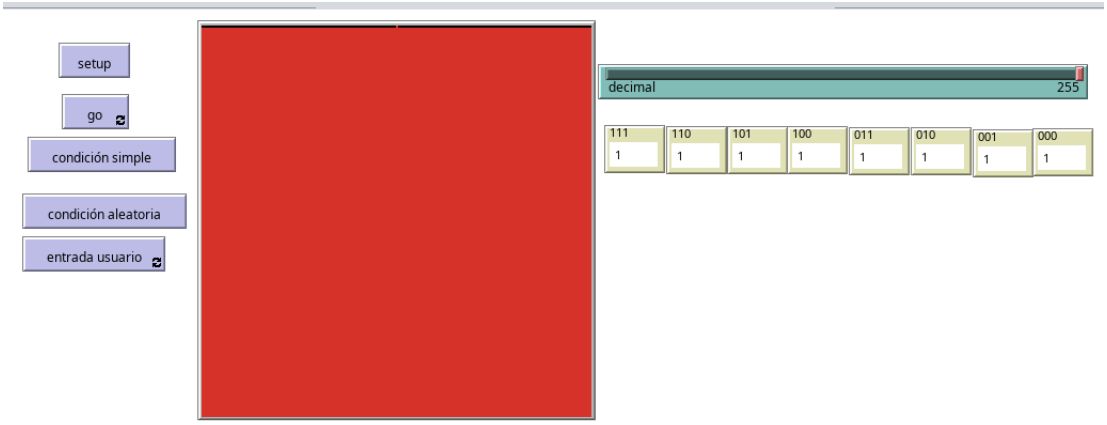
Desarrollar un autómata celular con el lenguaje de programación de su elección o usar el codigo base de NetLogo. Implmentar las caracteristicas: (especificaciones pdf original)

1. Función de transición
2. Fronteras
3. Tamaño
4. Condiciones iniciales
5. Salida gráfica

Ejercicios:

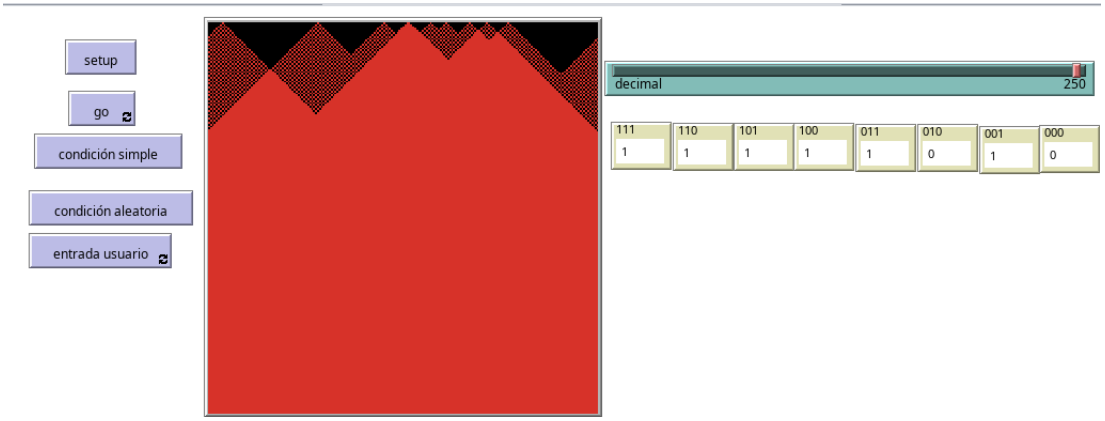
1. **Exploración:** Ejecute algunas de las 256 reglas del autómata celular a partir de la implementación de su programa con condiciones de frontera periodicas, minimo 100 tiempos de evolución y condiciones iniciales aleatorias, Identifique el tipo de dinamica que presenta y a que case pertenece según las cuatro categorías de la Clasificación de Wolfram. Muestre 1 representante de cada clase y adjunte la captura de pantalla.
   RESPUESTA:
   Después de una serie de ejecuciones del programa que implemente en NetLogo para ACE, pude deducir con base a lo visto en clase 4 ejecuciones cada una correspondiente a una clase de las 4 de Wolfram.

   a) CLASE I:



   Regla 255

   b) CLASE II:

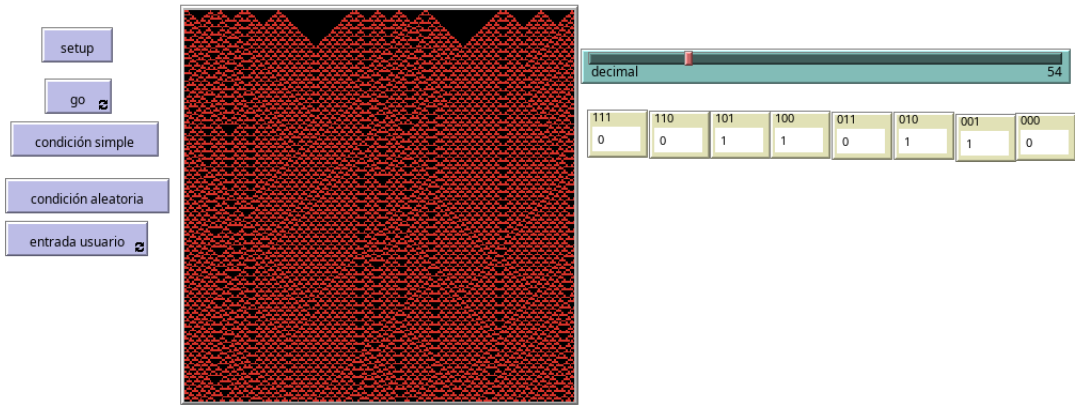


   Regla 250

   c) CLASE III:

Regla 22

d) CLASE IV:



Regla 54

2. Después de explorar todas o algunas de las 256 reglas. ¿Cuál crees que sea la clase más frecuente?
   RESPUESTA:
   A mi parecer y con base a lo visto en clase y lo ejecutado, puedo concluir que pude apreciar una mayor parte de ACE relacionados con la **Clase II** e incuso en clase vimos que tiene un aproximado de 194 reglas asociadas, aproximado porque puede que algunas esten en la frontera de I y III.

3. **Sensibilidad a condiciones iniciales.** Encuentra un autómata de clase III, perturbe la condición inicial aleatoria por un bit, evolucione y vea las diferencias. ¿La sensibilidad a las condiciones iniciales es una propiedad suficiente para catalogar la dinámica como caotica? (Adjunte imagenes del ACE perturbado, sin perturbar).
   RESPUESTA:
   Para esta parte vamos a ejecutar la regla 90 de Clase III en la **Fig.1** con una condición inicial simple (patch de en medio) y continuamnete en la **Fig.2** con un patch más encendido.

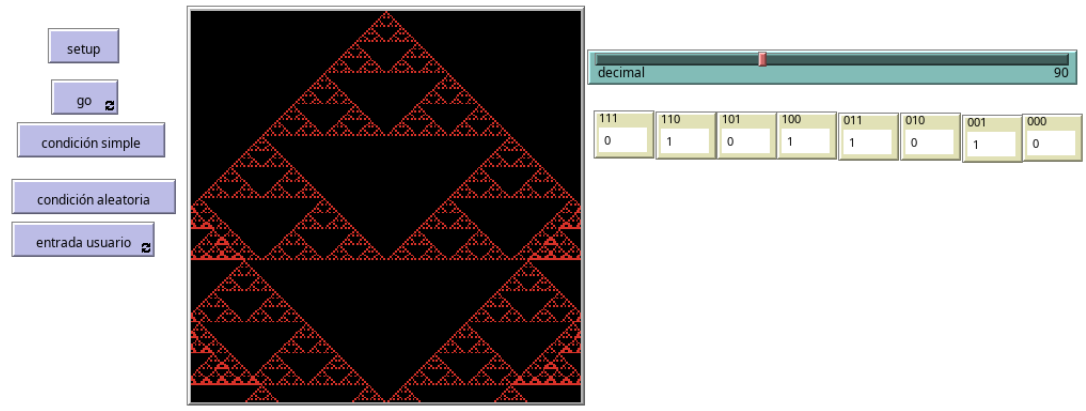
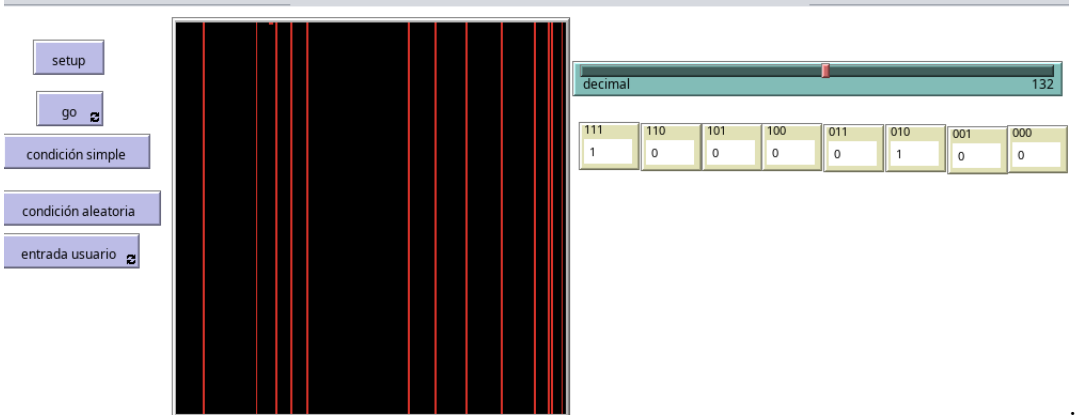


FIG.1

FIG.2

Podemos observar que no es necesaria una perturbación para considerar que es una dinamica caotica, ya que la estructura sigue persistiendo y no es tan notorio el cambio.

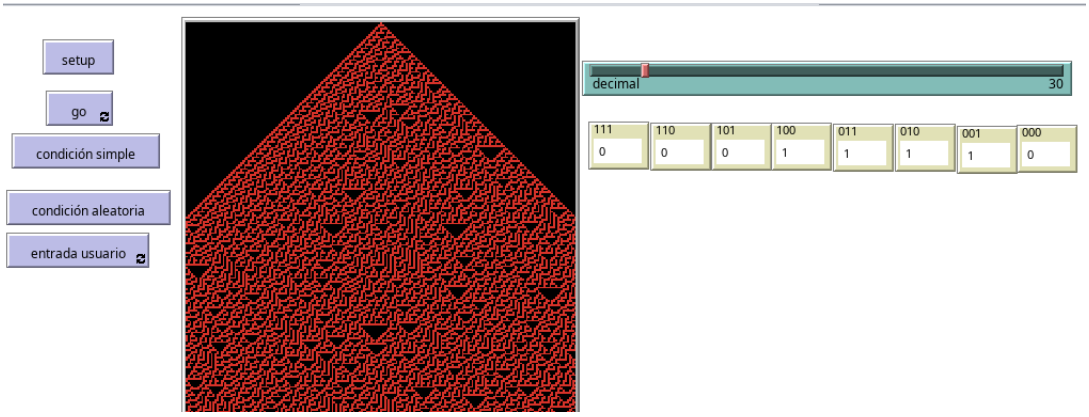
4. ¿Qué poder de cómputo tiene la regla 132? y ¿Qué propiedades tiene la regla 30? Explique y adjunte capturas.
   RESPUESTA:
   La regla 132 además de su comportamiento caótico y complejo, su poder de cómputo es limitado debido a la simplicidad de los autómatas celulares elementales en general.



.

La regla 30 es un ejemplo de un sistema dinámico complejo, lo que significa que exhibe patrones no triviales que emergen de interacciones simples entre sus componentes. Esta complejidad puede manifestarse en patrones simétricos, periódicos o caóticos, que pueden ser visualizados a través de las iteraciones de la regla, además comportamiento caótico, lo que significa que incluso

pequeñas variaciones en las condiciones iniciales pueden producir resultados muy diferentes a largo plazo.



5. En las imagenes del documento de la practica se muestra la evolución de la regla 22 con condición inicial simple (una celda negra en el centro) y condición inicial aleatoria. Explique lo que sucede con la dinámica.
   RESPUESTA:
   En relacion con ambas figuras podemos concluir que es caotica ya que una variación nos da un automata un tanto más complejo, claro que se puede apreciar su naturaleza, por asi decirlo si nos referimos a as estructuras que se forman, pero tambien podemos ver que la entrada al ser aleatoria y tener mas de 2 patch encendidos al ser ejecutado llega a verse cada vez más y más compleja nuestra estructura.

6. Si el autómata estuviera definido en el alfabeto de 3 estados, ¿Cuántas posibles reglas hay?
   RESPUESTA:
   Si consideramos un autómata celular de una solo dimensión con un alfabeto de 3 estados, cada celda puede tener uno de tres estados diferentes: 0, 1 o 2. Si consideramos únicamente los primeros vecinos (las celdas adyacentes a la izquierda y a la derecha), entonces cada celda tiene $3^2 = 9$ posibles combinaciones de estados que podemos utilizar para definir una regla.

   Para determinar el número total de reglas posibles, necesitamos determinar cuántas combinaciones de 9 elementos podemos formar con un alfabeto de 3 estados. Esto se puede calcular utilizando la fórmula para combinaciones con repetición, que es:

   $C(n + k - 1, k) = C(9 + 3 - 1, 3) = C(11, 3) = 165$

   Por lo tanto, hay un total de 165 posibles reglas.

Parte 3. Automata 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 usando el codigo visto en clase con las siguientes reglas de actualización, considerando una vecindad de Moore.

Para una celda que esta "viva"

1. Si tiene uno o cero vecinos "muere"(por soledad).
2. Si tiene cuatro o más vecinos "muere"(por sobrepoblación).
3. Si tiene dos o más vecinos "sobrevive".

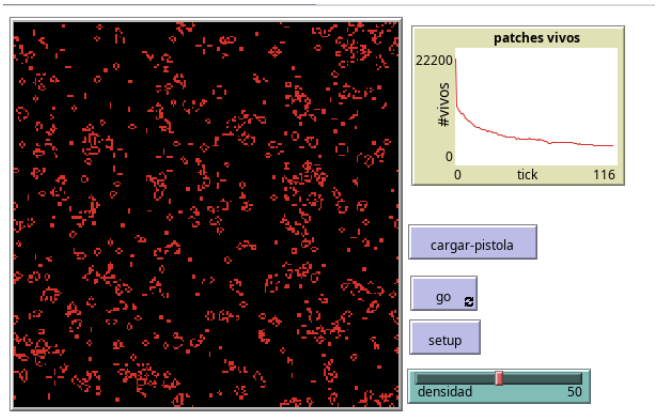
Para una celda que esta "vacia.º "muerta".

1. Si tiene tres vecinos la celda "vive".

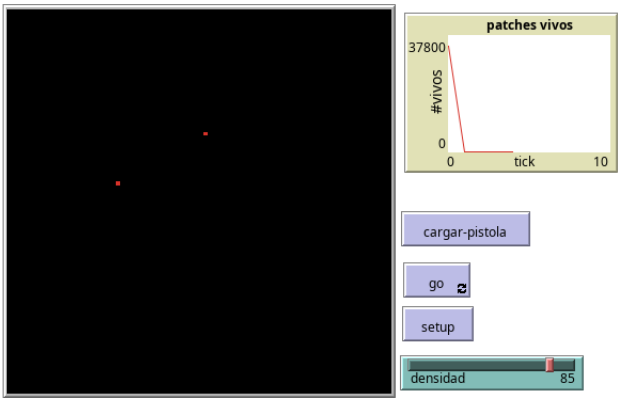
El minimo del tamaño de la reticula debe ser de $100x100$.

Ejercicios:

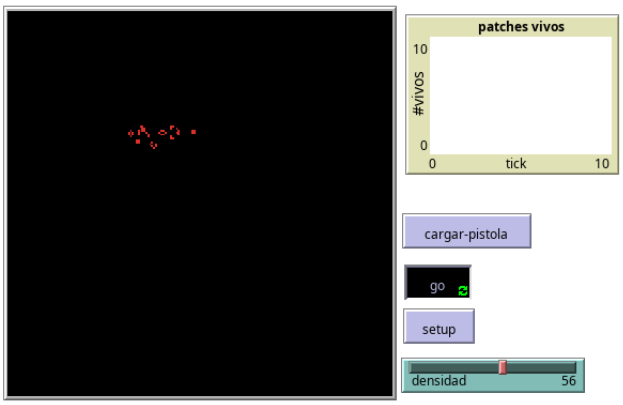
1. En NetLogo coonstruya una gráfica de conteo de celdas vivas a través del tiempo. Ejecute varias veces su programa con una reticula de $100x100$ celdas y condición aleatoria con 50 % de celdas ocupadas, ¿en cuánto tiempo (ticks) se .estabilizó"la dinámica?, ¿para cualquier configuración inicial aleatoria pasará lo mismo? Explique y adjunte una captura de pantalla del "view" de agentes y la gráfica de conteo.
   RESPUESTA:
   Al ejecutar el juego de la vida un total de 10 veces, en un promedio obtuve que emepzo en un tiempo de regulación de 231,4



2. Realice el mismo experimento aumentando la densidad de la condición aleatoria a 85 %, ¿sucede lo mismo? Explique.
   RESPUESTA:
   En este caso e sistema se nivela o acaba de una manera más rapida, el ejecutar de igual forma un total de 10 ejecuciones, pude llegar a un promedio de 11,16 en el que en tan pocos ticks acaba.



3. Implemente la "pistola de planeadores.º "Gosper's gun"dada la condición inicial del pdf original. Explique la dinamica que se genera, ¿es periódica? ¿qué implicaciones tiene controlar de alguna manera ciertas estructuras en función del tiempo?
   RESPUESTA:
   Sí, la "Gosper's gun.es un patrón periódico en el juego de la vida. Tiene un período de 30 ciclos de reloj, lo que significa que después de 30 iteraciones del juego de la vida, el patrón se repite a sí mismo en su forma original. Durante cada período, la "Gosper's gun"dispara una serie de naves espaciales que se mueven hacia afuera y luego desaparecen. Estos movimientos son determinados por las reglas del juego de la vida y por la forma específica en que se organizan las células dentro del patrón. El período de la "Gosper's gun.es una de las características que la hacen interesante desde el punto de vista matemático y computacional, ya que permite predecir y simular su comportamiento de manera precisa y eficiente.
   El control de estructuras como la "Gosper's gun"podría permitir la creación de patrones de células específicos que pueden usarse para resolver problemas complejos de manera más eficiente. Por ejemplo, se ha demostrado que ciertos patrones de células en el juego de la vida pueden usarse para computación universal, lo que significa que pueden simular cualquier proceso computacional. En la biología y la medicina, el control de estructuras en función del tiempo puede permitir la modulación de procesos celulares y fisiológicos, lo que podría tener aplicaciones en el tratamiento de enfermedades y la regeneración de tejidos. En la ingeniería y la informática, el control de estructuras en función del tiempo puede permitir la optimización de procesos y sistemas complejos, lo que podría tener aplicaciones en el diseño de algoritmos y sistemas de control.



4. ¿El juego de la vida es reversible? Explique.
   RESPUESTA:
   No, en general el juego de la vida no es reversible, lo que significa que no se puede determinar el estado anterior del juego a partir del estado actual. Esto se debe a que el juego de la vida no conserva la información necesaria para reconstruir el estado anterior del juego. En particular, la evolución del juego de la vida es determinista, lo que significa que el estado actual del juego se determina completamente a partir del estado anterior. Sin embargo, esta evolución no es invertible, ya que hay muchos estados anteriores posibles que pueden llevar al mismo estado actual, lo que se conoce como la propiedad de la no unicidad de la inversión. Hay algunas excepciones a esta regla general. Por ejemplo, se ha demostrado que ciertos patrones especiales en el juego de la vida, como los patrones estables y los osciladores, son reversibles. En estos casos, se puede determinar el estado anterior del juego a partir del estado actual, ya que se conserva información suficiente sobre la evolución del patrón.