Automatas Celulares

Luis Andres Juarez Sandoval

11 Abril 2016

Contents

1	introduccion	1
	1.1 aplicaciones	1
	1.2 links de interes	1
2	Caso de estudio	2
	2.1 Automatas celulares unidimensionales	2
3	Autmatas celulares bidimensionales	3
4	Autmatas celulares tridimensionales	5
	4.1 dataset	5
5	Experimento	6
	5.1 Pruebas	9
	5.2 Graficación de los automatas celulares	24
6	repositorio	24
7	Conclusiones	24

1 introduccion

Los automatas celulares son ...

1.1 aplicaciones

Sus aplicaciones en la ingenieria y las ciencias son:

- 1. aplicación 1
- 2. aplicacion 2
- 3. aplicación n

1.2 links de interes

Modelo matematico para un sistema dinmico que evoluciona en pasos discretos. Es decir, es un modelo matematico para sistemas que soportan cambios; adems, estos cambios se suceden cada tiempos constantes, razn por la cual, se usa una escala de enteros.

La idea de la creacion de los automatas celulares nacion en la decada de 1940 en donde John Von Neumann, intentaba la creacion de un sistema que pudiera autoreplicarse o extenderse de forma autonoma, mediante un modelo matematico, aplicados en una red rectangular. Contenian celulas las cuales asemejaban al proceso de vida en donde crecian, se reproducian y morian conforme pasaba el tiempo, Tienen diferentes valores y comportamientos entre si.

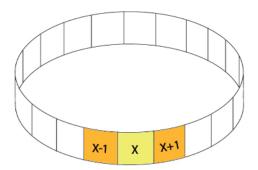
Se define Lattice, es una cuardicula de dimension N e infinitamente extendida. Dentro de cada una de las celdas en la cuadricula se encuentra una "Celula". Cada celula toma un valor de un conjunto finito de estados k, el ciclo de vida de la celula y su evolucion no dependera solo de ella, si no, tambien de su entrono, los cuales determinaran sus condiciones de vida. Para determinar su evolucion, la funcion de reticula, se aplica a cada una de la celulas dentro de la lattice, toma todos los valores de la celda y sus alrededores-, asi podra determinar los nuevos valores de cada celula.

Informacion adicional se puede encontrar en las urls : https://es.wikipedia.org/wiki/Automata_celular

2 Caso de estudio

2.1 Automatas celulares unidimensionales

Consiste en una sola fila de clulas a los que se aplica un principio de vecindad bsico de dos vecinos por clula y a los que igualmente se pueden aplicar las diversas condiciones de frontera que hemos nombrado anteriormente



Autómata celular unidimensional con un radio de vecindad r=1 y una condición de frontera periódico.

Como ejemplo, podemos tomar un autmata celular unidimensional con un radio de vecindad r=1, dos estados (0 y 1) y una condicion de frontera de tipo peridico, como el que se muestra en la imagen. Usaremos para este caso un tamao de diez clulas y unas funciones de transicin basadas en lo siguiente:

- Si ambos vecinos de la clula tienen el mismo estado, el estado de la clula a la que se aplica la funcion cambiar.
- Si ambos vecinos de la clula tienen distinto estado, el estado de la clula a la que se aplica se mantendr igual.

Stephen Wolfram clasific el comportamiento de los automatas celulares unidimensionales. Segn Wolfram, todo automata celular pertenece a una de las siguientes clases:

- Clase I. La evolucin lleva a una configuracin estable y homognea, es decir, todas las clulas terminan por llegar al mismo valor.
- Clase II. La evolucin lleva a un conjunto de estructuras simples que son estables o peridicas.
- Clase III. La evolucin lleva a un patrn catico1.
- Clase IV. La evolucin lleva a estructuras aisladas que muestran un comportamiento complejo (es decir, ni completamente catico, ni completamente ordenado, sino en la lnea entre uno y otro, este suele ser el tipo de comportamiento ms interesante que un sistema dinmico puede presentar).

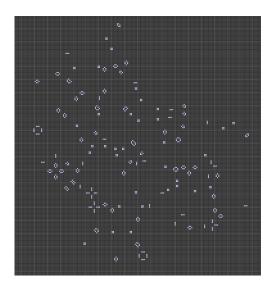
3 Autmatas celulares bidimensionales

Un autmata celular finito de 2 estados es la simulacin ms simplificada posible de una clula viva sobre un tejido. Representa una unidad de interaccin que tan solo puede presentar 2 estados (activo e inerte, encendido y apagado, vivo o muerto, o como quieran denominarse). A pesar de su simplicidad inicial permite simular la evolucin de sistemas complejos y analizar pautas tales como el comportamiento emergente. Un nuevo nivel de complejidad consiste en trabajar con un autmata celular de 2 estados, situado en un universo plano (de 2 dimensiones) ilimitado (toroidal, cerrado circularmente en sus dos direcciones).

El autmata posee solo 2 estados, representados por dos imagenes distintas. El estado evolutivo del autmata depender de su estado actual y el de su entorno (sus 8 clulas vecinas, derecha, izquierda, arriba, abajo y las cuatro esquinas). El universo es cerrado e ilimitado (toroidal), de forma que las ltimas celdas de la derecha estn unidas (interactuan) con las primeras de la izquierda y viceversa, al igual que las de la lnea inferior lo hacen con la primera y viceversa. La evolucin es discreta y simultnea para todos los componentes (clulas) del universo. Cada generacin surgir de golpe reemplazando completamente a la anterior.

Las reglas para una evolucion son las siguientes:

- Una clula se inactiva (o permanece inactiva) si posee menos de 2 o ms de 3s vecinas activas (muerte).
- Una clula mantiene su estado, sea este cual fuera, si tiene tan solo 2 clulas vecinas activas (conservacin).
- Una clula cobra actividad (o permanece activa si ya lo estaba) cuando la rodean 3 clulas activas (nacimiento).



4 Autmatas celulares tridimensionales

Ampliamente estudiados por Carter Bays, los automatas celulares de tres dimensiones se enfocan principalmente para encontrar una regla de evolucin en tres dimensiones que sea la sucesora del Juego de la Vida en el espacio tridimensional, muchos de sus resultados son de tipo cuantitativo, basados principalmente en la simulacin de varias reglas de evolucin en pequeos espacios tridimensionales para encontrar estructuras que sean similares al Juego.La aplicacin de estos autmatas celulares se centra en el estudio en campos estadsticos.

Son automatas celulares que se desarrollan en un mbito de $Z \times Z \times Z$, que usan el principio de vecindad de Moore a nivel tridimensional. Esto implica que la funcin de transicin ha de tener en cuenta el estado de veintiseis clulas vecinas adems del estado de la clula en cuestin.

Los automatas celulares tridimensionales pueden seguir los mismos principios que los anteriores, aunque dado el incremento de la complejidad de los calculos y el procesamiento, su puesta en pretica no es simple. Sin embargo, y puesto que se pueden considerar automatas celulares tridimensionales con las mismas variables de tipo frontera en el mbito terico, se debe considerar que la representacion de los automatas celulares tridimensionales considerando una frontera peridica es en forma de toroide.



4.1 dataset

Para esta situacin se realiza un proceso estocastico con una variable aleatoria que es la encargada de crear las celulas

5 Experimento

```
Para validad se creo el siguiente programa en C++
#include <iostream>
#include <stdlib.h>
using namespace std;
/*Funcion en la cual consulta la cantidad de vecinos que tiene la celula en la l
int neighbours (int a, int b, int c, int d, int e, int f, int g, int h)
 int result =a+b+c+d+e+f+g+h;
 return result;
}
int main(void)
 int i, j;
 int N = 32; // tamanio de la lattice
 //Crecion de objetos dinamicos para el arreglo de tamanio N
 int** A = new int* [N];
 for(i=0;i<N;i++) // creacion de los objetos en el arreglo dinamico</pre>
  A[i] = new int[N];
  int** B = new int* [N];
  for(i=0;i<N;i++)</pre>
   B[i] = new int[N];
  // se generan las celulas las celulas en la lattice 0 si no existe y 1 si exis
  // Estado inicial
  cout << "Initial State " << endl;</pre>
  for(i=0;i<N;i++)</pre>
   for(j=0;j<N;j++)
```

A[i][j] = rand() % 2;

} }

```
//impresion de la primer lattice
  for (i=0; i<N; i++)
   for(j=0; j<N; j++)</pre>
    cout << A[i][j] << " ";
   }
    cout << endl;</pre>
  int T = 100; // cantidad de iteraciones
  for(int t=0;t< T;t++)
   for(i=0;i<N;i++)</pre>
    for(j=0;j<N;j++)</pre>
    B[i][j] = A[i][j]; // paso de la lattice original a la nueva lattice
   }
   /*
        Recorre la lattice
   */
  for(i=0;i<N;i++)</pre>
   {
    for(j=0;j<N;j++)
     int temp= //GUarda la cantidad de vecinos al rededor de la celula
     neighbours(
     B[((i+1)%N)][j],
     B[((i+1)%N)][((j+1)%N)],
     B[i][((j+1)%N)],
     B[((i-1+N)%N)][((j+1)%N)],
     B[((i-1+N)%N)][j],
     B[((i-1+N)%N)][((j-1+N)%N)],
     B[i][((j-1+N)%N)],
     B[((i+1)%N)][((j-1+N)%N)]);
     // rule 1:Una celula cobra actividad (o permanece activa si ya lo estaba) c
```

```
if((B[i][j] == 0) \&\& (temp == 3))
      A[i][j] = 1;
     // rule 2: Una celula mantiene su estado, sea este cual fuera, si tiene tan
     if((B[i][j] == 1) \&\& ((temp == 2) || (temp == 3)))
      A[i][j] = 1;
     // rule 3: Una celula se inactiva (o permanece inactiva) si posee menos de
     if((B[i][j] == 1) \&\& ((temp >= 4) || (temp <= 1)))
      A[i][j] = 0 ;
}
cout << endl; //impresion de iteracion</pre>
cout << "t = " << t;
cout << endl;</pre>
//impresion de la lattice despues de cada evolucion
for(i=0;i<N;i++)</pre>
{
 for(j=0;j<N;j++)
 cout << A[i][j] << " ";
 cout << endl;</pre>
  cout << endl << endl;</pre>
}
// libera la memoria para no hace stackoverflow
for(i=0;i<N;i++)</pre>
 delete[] A[i];
delete[] A;
```

```
for(i=0;i<N;i++)
  delete[] B[i];

delete[] B;

return 0;
}</pre>
```

5.1 Pruebas

se ejecuto el programa con los siguientes comandos y se guardo en en una un archivo

```
g++ twocellular.cpp -o test
./test > datos.dat
```

y se genero e siguiente archivo de salida

```
Initial State
```

 $1\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0$ $t_{r} = 0$

 $1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0$ $0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

t = 1

t = 2 $1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0$ t = 3 $0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0$

 $1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1$

 $1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1$

t = 7

t = 8 $0\; 0\; 0\; 0\; 0\; 1\; 1\; 0\; 0\; 0\; 0\; 0\; 1\; 1\; 0\; 0\; 0\; 0\; 0\; 1\; 1\; 0\; 1\; 0\; 0\; 1\; 1\; 0\; 0\; 0$ $1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

t = 9 $1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0$

t = 10 $1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$ t = 11

 $1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\;$

t = 15

5.2 Graficación de los automatas celulares

En este caso no se pudieron graficar los datos debido al resultado que no se podia expresar en los ejes x, y

6 repositorio

se creo un repositorio en https://bitbucket.org/andresjz/automatas_cellulares y su subieron de forma publica los archivos del proyecto

- 1. proyecto.tex archivo en latex del proyecto
- 2. proyecto.pdf pdf generado con pdflatex
- 3. twocellular.cpp codigo fuente del programa en C++
- 4. datos.dat archivo de datos generado por el programa
- 5. imagenes son imagenes generadas de los resultados

7 Conclusiones

como conclusion de este experimento se obtiene que el modelo desarrollado para explicar la conformacion celular nos da como resultado una serie de elementos que nos pueden ayudar a comprender como se forman nuevas poblaciones y como interactuan con sus lados opuestos (fronteras).

