2. Autómatas Celulares

José J. Martínez P.
Jmcursosingenieria111@gmail.com
Febrero 2024

2.1 Introducción.
Los sistemas dinámicos, son sistemas o fenómenos que dependen del tiempo. En estos sistemas, su estado en un momento dado se puede calcular a partir del estado anterior del sistema y de sus interacciones con el ambiente. Los sistemas dinámicos pueden ser continuos o discretos, dependiendo de la manera en que se tome la coordenada tiempo.

Vamos entonces a hablar de los Autómatas Celulares, A.C., un sistema dinámico que es discreto en el tiempo y también es discreto en el espacio. De esta manera decimos que los A.C. son sistemas dinámicos discretos con un número enumerable de estados. Habíamos visto que el nicho de aplicación de la IA son los sistemas complejos, en este caso los A.C. crean la complejidad a partir de reglas de interacción elementales entre las celdas. Los A.C. se preocupan por la pregunta de cómo surge el comportamiento emergente en un sistema partir de su actualización por la aplicación de reglas que relacionan una célula con sus vecinos cercanos.

En A.C. la idea básica es describir cualquier sistema complejo, a partir del comportamiento de sus componentes elementales homogéneas o células, que en general siguen reglas simples. Es un enfoque ascendente (bottom_up), que busca el comportamiento emergente del sistema complejo. En otras palabras, se deja que la complejidad del sistema emerja a partir de la interacción de individuos, agentes, homogéneos, que siguen, en principio, reglas simples. En inglés, un autómata es *automato*n y un grupo de autómatas es *automata*. De manera que el término en inglés *Cellular Automata*, da más claridad, sobre el hecho de que la complejidad emerge a partir de un grupo de autómatas elementales.

2.2 Vecindad

"Dime con quién andas y te diré quién eres"
Refrán popular

El concepto fundamental en los AC es el concepto de vecindad. Se trata de que en el tiempo, el comportamiento o estado de una célula está determinado por el comportamiento de la misma célula y el de sus células vecinas, son sistemas complejos. Es por esto importante hacer algunas definiciones:

Vecindad, es el conjunto de vecinos de una célula que influye en su comportamiento. En su definición común, es el conjunto de los vecinos de una población, barrio, calle o casa. Ej. A la fiesta decembrina asiste toda la vecindad. Pero además, en general, una determinada vecindad tiene costumbres comunes, sigue reglas comunes con algunas excepciones.

Como se puede observar, son muchos los comportamientos que se generan a partir de una vecindad, tantos que en general definen el comportamiento de muchas personas. Es así que, en nuestra casa, nuestro trabajo, nuestras clases, nuestro desplazamiento, nuestras reuniones, siempre hay una serie de reglas implícitas, de las cuales generalmente no somos conscientes, que si bien no están formalmente establecidas, definen nuestro comportamiento. Otros ya los definen claramente las leyes y las normas aprobadas. En otras palabras, nosotros, las personas, también somos autómatas celulares. Alguna relación con Matrix?

2.3 Definición Formal

Un autómata celular D-dimensional (o AC-D), es una secuencia C_t, definida por una 5-tupla, $(L, w, U, f,\ C_0)$ donde:

> L: es un retículo,
> w: es un alfabeto, que representa los estados de la celda.
> $U = (u_1,\ u_2, \dots, u_n)$ es una secuencia finita de elementos del retículo
> $f: w^U \rightarrow\ w$, es la función de transición o regla local
> $C_0 : L\ \rightarrow w$, es alguna configuración inicial.

Ejemplo 1. AC 1D, autómatas celulares de una dimensión.

Retículo: L

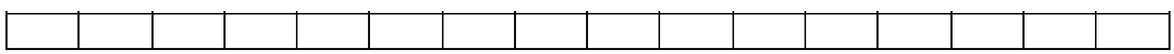

Figura 2.1, retículo o universo para los autómatas celulares de una dimensión

Alfabeto: $w = \{0, 1\}$

Vecindad: U

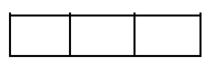

Figura 2.2. Vecindad, radio r=1 para autómatas AC-1D. Se Pueden definir otras vecindades

Configuración inicial del autómata, aleatoria: $C_0 : L\ \rightarrow w$

1	1	1	1	0	1	1	0	0	1	0	1	0	0	1	0

Figura 2.3. Estado inicial de un AC-1D, r=1, generalmente es aleatorio

En la gran mayoría de casos, el AC-1D funciona como un collar, es decir el autómata del extremo izquierdo se une con el autómata del extremo derecho. Cuando se tiene otro tipo de retículo esta configuración se denomina toroidal.

Transición local o regla de comportamiento local:

$$c_c(t+1) =\ f(c_{izq}(t), c_{cc}(t), c_{der}(t))$$

Donde c_c es la celda central, c_{izq} es la celda de la izquierda y c_{der} es la celda de la derecha.

Nota. La $\boldsymbol{C}$ mayúscula hace referencia a todo el Autómata Celular, el retículo; la c hace referencia a uno de los autómatas, el cuadrado pequeño.

Ejemplo 2. AC-2D, autómatas celulares de 2D:

Retículo: L

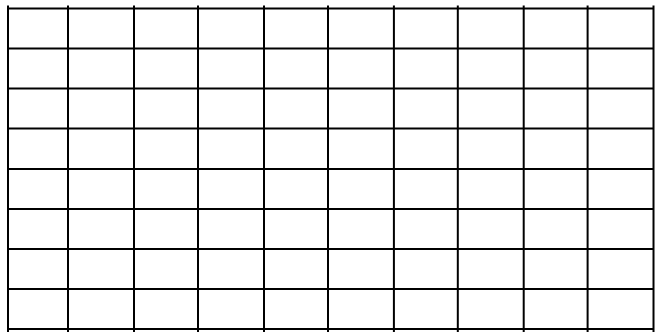

Figura 2.4. Retículo o universo para los AC 2D. Pueden conformarse otros tipos de retículos.

Alfabeto: $w = \{0, 1\}$

Vecindad: U**,** en AC-2D las vecindades comunes son

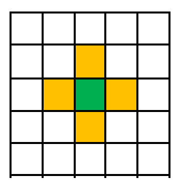

Figura 2.5, vecindad de von Newmann, tiene 5 vecinos

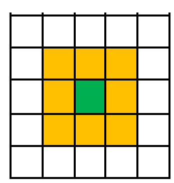

Figura 2.6, vecindad de Moore 9 vecinos, es la más utilizada.

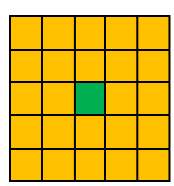

Figura 2.7, vecindad de Moore extendida 25 vecinos.

Configuración inicial del autómata, aleatoria: $C_0 : L \rightarrow w$

1	0	1	1	0	0	1	1	1	1
1	1	0	1	0	1	1	1	0	1
1	1	0	1	1	1	0	1	0	0
0	0	1	0	1	0	1	0	1	0
1	1	0	1	0	1	0	1	1	0
0	0	1	0	1	0	1	0	1	0
1	0	1	1	1	0	0	1	0	0
0	1	0	0	0	1	1	0	1	1

Figura 2.8, configuración aleatoria inicial de un AC 2D.

Transición local o regla de comportamiento local, para un AC 2D con vecindad de Moore:

$$c_c(t+1) = f(c_N(t), c_{NE}(t), c_E(t), c_{cc}(t), c_{SE}(t), c_S(t), c_{SO}(t), c_O(t), c_{NO}(t))$$

Donde c_c es la celda central, c_N es la celda norte, c_{NE} es la celda noreste, c_E es la celda este, c_{SE} es la celda sureste, c_S es la celda sur, c_{SO} es la celda suroeste, c_O es la celda oeste y c_{NO} es la celda noroeste.

2.4 Propiedades Esenciales de un Autómata Celular

1. Universo: Un *retículo* o arreglo *regular*, también se le conoce como teselación, donde cada *célula* tiene un estado discreto.
2. Dinámico: Un comportamiento dinámico que describe la regla local.
3. Homogeneidad: Todas las células se actualizan con la misma regla local
4. Paralelismo: El estado de todas las células se actualiza simultáneamente. La regla define el estado de cada célula central en el siguiente paso de tiempo, dependiendo de los estados de las células vecinas y de su propio estado.
5. Localidad: las reglas son locales por naturaleza.

Para construir un Autómata Celular, se deben definir: la célula; el retículo; la regla; y la vecindad

1. La Célula.

- Es el elemento básico de un AC, se comporta como un autómata de estado finito, sus entradas son el estado de las células vecinas y el suyo propio; su salida, su nuevo estado.
- Una **célula** es una clase de elemento de memoria que almacena **estados**. El número de estados o conjunto de estados se representa por *ω.*
- En el caso más simple, *ω* = {0, 1}. En casos más complejos las células pueden tener más estados diferentes.

2. El retículo.

- Es el espacio donde se conforma el AC. El retículo más simple es el de *una dimensión* con una configuración *cuadrada, 1D.* Esto significa que el AC es como un collar de perlas.
- Los AC más comunes son de 1D o 2D, aunque en ciertas aplicaciones se trabajan AC 3D. También se conoce como teselación.

3. ***Las reglas***

- Para introducir la *dinámica* al sistema, se le deben definir las **reglas** al AC.
- En algunas aplicaciones son ecuaciones en diferencia.

3. ***La vecindad.***

- La vecindad se establece dentro del retículo, casi siempre con vecindades regulares.
- La vecindad, en vecindades regulares, se conforma con base en el radio ***r***, que define el horizonte del autómata.
- En ciertas aplicaciones puede uno definir su propia vecindad. Por ejemplo, en sistemas de criptografía se toman vecindades completamente irregulares, o en los diagramas de Voronoi

Un ejemplo de dinámica "microscópica" que resulta de la interacción local es la "ola" en un estadio de fútbol. Cada persona reacciona dependiendo del "estado" de sus vecinos. Si se ponen de pie y alzan los brazos, usted también lo hace, después de un corto tiempo; sus vecinos se vuelven a sentar, y usted se sienta siguiendo lo que hacen sus vecinos.

2.5. Autómatas Celulares de 1D

2.5.1 Autómatas Celulares Simples

Vamos a estudiar AC 1D con r=1, es decir una vecindad de tres células, w = {0, 1}. En estos autómatas, hay 2^8 = 256 reglas que han sido estudiadas con mucho detalle por Stephen Wolfram.

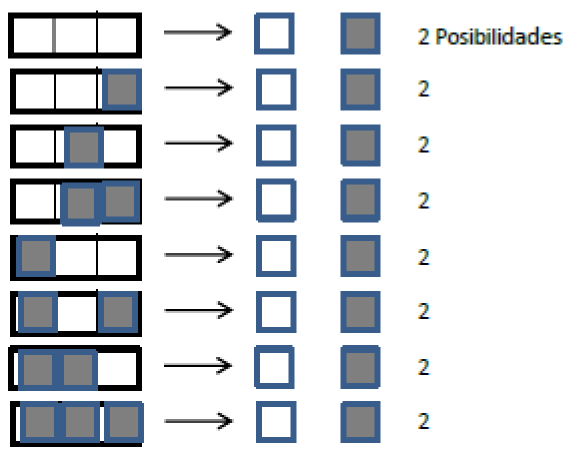

Figura 2.9, Las 256 reglas de un AC 2D.

Aunque algunas reglas generan comportamientos globales similares, en general el comportamiento del A.C. depende de sus condiciones iniciales y de su regla de comportamiento local. Cada regla se identifica con base en un número entre 0 y 255.

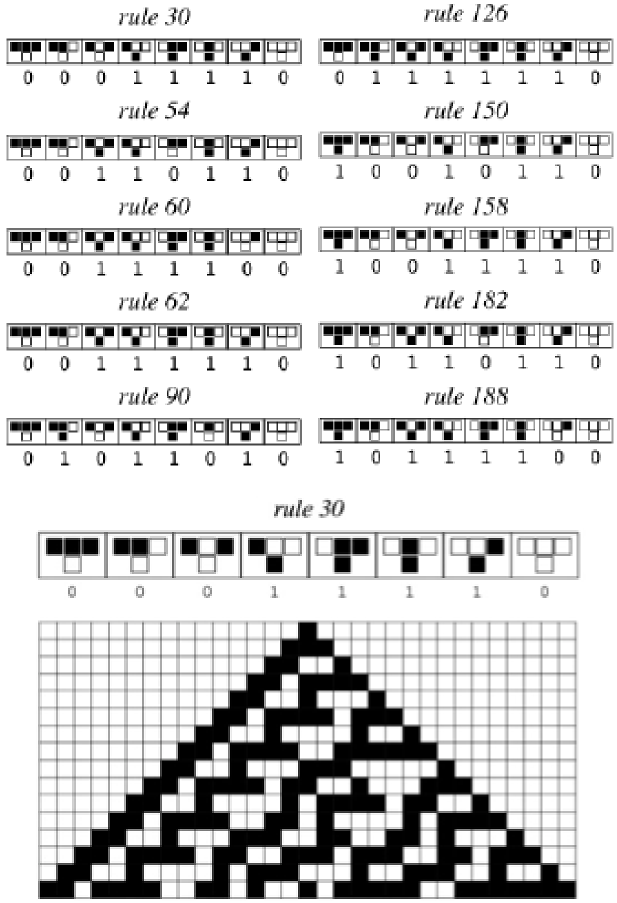

Figura 2.10, Algunas reglas de un AC 1D.

De la figura 2.10, el número de la regla se encuentra tomando la cadena de ceros y unos de la regla, así la regla 30 tiene una cadena 00011110, que se toma como un polinomio en base 2 que al evaluarse nos da el 30.

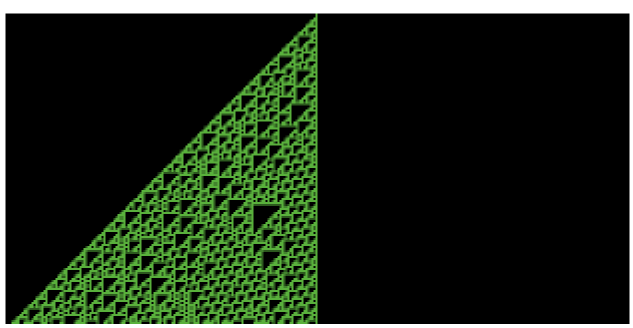

Figura 2.11, Regla 110, tiene un comportamiento complejo[1].

Por su comportamiento, también son de interés las reglas: 23, 25, 27, 30, 31 y otras, ver[2].

---

[1] Simulada en NetLOgo

[2] http://mathworld.wolfr am.com/ElementaryCellularAutomaton.html

2.5.2 Reglas Totalísticas
En este caso el estado siguiente de la célula central depende únicamente de la suma de los estados de las células vecinas, también se conocen como reglas de voto. Formalmente, la regla global de un AC totalístico 1D está definida por:

$$c_c(t+1) = f(c_{izq}(t) + c_{cc}(t), + c_{der}(t))$$

2.5.3 Comportamientos de Wolfram
En el estudio formal desarrollado por Wolfram de los AC 1D, se encontró que los comportamientos de los AC 1D, se podían clasificar en una de cuatro categorías diferentes:

Clase I. Los ACs de la Clase I siempre se resuelven en un patrón homogéneo global, donde cada célula termina quedándose en el mismo estado indefinidamente. Se pueden considerar que tienen un orden perfecto. Ejemplos son las reglas 0, 32, 160, 249.

Clase II. Los ACs de la Clase II forman estructuras que convergen rápidamente a estados periódicos que se repiten a través de un número fijo de estados. Hay orden, sin embargo, recorren rutas periódicas. Los ejemplos más simples de esta clase son los que alcanzan algún estado invariable global, como la regla 4 o la regla 12. Otros ejemplos son las reglas 4, 108 y 218. Aunque también se considera de esta clase la regla 24.

Clase III. Los ACs de la Clase III se caracterizan por su naturaleza caótica pues producen diseños aleatorios, no hay un orden. Como el ruido en una pantalla de rayos catódicos sin señal. El ejemplo clásico es la regla 30, otros ejemplos son las reglas 22, 30, 90, 126. Y 150. Aquí se puede apreciar que un pequeño cambio en las condiciones iniciales, eventualmente afecta todas las partes del sistema.

Clase IV. Los ACs en la Clase IV se encuentran en el límite el orden de la clase II y el caos de la clase III. Forman estructuras complejas, pero con estructuras localizadas que "migran" dentro del diseño principal. Dependiendo de la configuración inicial, los diseños de la Clase IV se resuelven eventualmente en diseños de Clase I o Clase II. Ejemplos son las reglas 54 y 110.

Las clases I y III no son interesantes, mientras que las clases II y IV tienen propiedades muy interesantes desde el punto de vista de la complejidad y de la vida. Se supone que la vida requiere orden para almacenar experiencia y caos para interactuar con el ambiente. Es por esto por lo que se dice que la vida está al borde del caos, debe tener un orden mínimo para almacenar experiencia y manejar el caos para adquirir esa experiencia.

Para un AC con k número de estados de la célula y n número de vecinos incluida la célula central, el número de posibles reglas se puede calcular como k^{k^n}. Si suponemos un autómata con 4 estados y nueve vecinos, el número posible de reglas seria 4^{4^9}, o 4^{262144}. Un número gigantesco. De ahí la complejidad del comportamiento de los autómatas celulares.

2.6 Autómatas Celulares 2D

Los AC 2D con un retículo cuadrado, permiten el manejo de un gran nivel de complejidad y por eso hay mucho interés en sus estudio y aplicación. Ya habíamos mencionado las vecindades y dentro de ellas la más utilizada es la vecindad de Moore.

2.6.1 LIFE de Conway

Uno de la AC de 2D más estudiado y conocido es "LIFE", desarrollado por John Conway. Se considera un buen ejemplo de emergencia y auto-organización, por la cantidad de objetos que se van produciendo en los ciclos de ejecución, dependiendo de su configuración inicial. Se ha probado que es una máquina de Turing, por lo cual es un computador universal.

LIFE comenzó a finales de los 1960s, principios de los setenta cuando Conway estableció una simplificación de las investigaciones sobre autómatas celulares de John von Neumann. Conway había querido encontrar el AC más simple que soportara la computación universal. Para esto decidió usar dos estados, muerto o vivo, y cuatro reglas simples:

- Si una célula viva tiene menos de dos vecinos vivos, se muere. Soledad.
- Si una célula viva tiene dos o tres vecinos, sigue viva. Felicidad.
- Si una célula viva tiene más de tres vecinos, se muere. Sobrepoblación.
- Si una célula muerta tiene exactamente tres vecinos, nace. Reproducción.

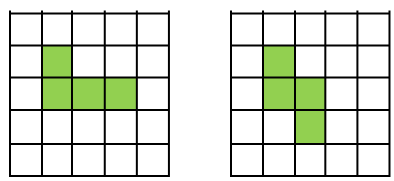

Figura 2.12. Si una célula viva tiene menos de 2 vecinos, se muere.

Como con los ACs 1D, estas reglas simples generan un comportamiento increíblemente complejo. A continuación, se presenta un "deslizador" (glider), una estructura dentro de LIFE que tiene un ciclo de 5 estados, y un movimiento abajo a la izquierda en cada ciclo.

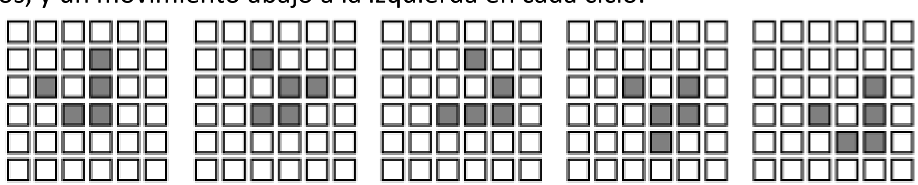

Figura 2.13. Como se desplaza un deslizador.

Life ha sido trabajado mucho en física teórica y ha habido muchos congresos internacionales, pues se comporta como un computador universal, según lo planteado por Turing.

2.7 Aplicaciones de los Autómatas Celulares

- **Epidemiología**

Se utilizan los AC para estudiar la evolución de enfermedades causadas por epidemias a través de modelado computacional. Veamos, por ejemplo, cómo se define una modelo de AC para el estudiar del comportamiento del COVID.

Suponga que tiene un AC 2D, en un retículo cuadrado de 10.000 Celdas; y que w = {Np, Ps, Cs, Cns, Ct, Pac}, donde:

- Np=No hay nada en la celda;
- Ps=Personas sanas;
- Cs=Persona con Covid-19 sintomática;
- Cns=Persona con Covid-19 asintomática;´
- Ct=Persona con Covid-19 en cuarentena o en cuidados médicos;
- Pac=Persona curada con anticuerpos.
- M=Persona muerta, no aparece en el retículo

Inicialmente el retículo se carga con 5000 Ps distribuidos uniformemente, con 5 Cs y 5 Cns también distribuidos aleatoriamente. Se usa la´vecindad de Moore.

Reglas.

- Si Ps tiene n vecinos pasa con probabilidad 0.n a ser Cns con probabilidad 0.4; o Cs con probabilidad 0.7.
- Cs después de 7 ciclos pasa a Ct
- Cns después de 30 ciclos pasa a Pa.
- Ct después de 20 ciclos con probabilidad 0.8 pasa a Pac o con probabilidad 0.2 pasa a M, que no se representa en el retículo.

Una vez se han ejecutado estas reglas en paralelo, se desplaza cada celda a una celda Norte, Sur, Este u oeste con probabilidad 0.25, si la celda está vacía. En caso contrario, se debe llevar el número de ciclos para las celdas Cs, Cns y Ct, lo mismo que el conteo de M. En cada ciclo se debe llevar el conteo de cada tipo de celda. Con la simulación se busca responder las preguntas siguientes: ¿Después de cuantos ciclos se estabiliza la población? ¿Cuantas celdas de cada estado quedan? ¿Cuántos muertos hubo?´

- **Antropología**

Se utilizan los A.C.s para hacer representaciones espacio-temporales para modelar la formación de sociedades civiles.

- **Sociología**

Consideramos que la simulación más cercana a la realidad de los sistemas sociales se logra con A.C.s, como por ejemplo para el estudio de las causas y efectos de la violencia civil.

- **Criptografía**

En el caso de los criptosistemas, se obtienen sistemas seguros frente a ataques de fuerza bruta y texto elegido. En este caso se utilizan AC **reversibles** que después de un determinado número de pasos vuelven a su configuración inicial. Es decir, el tiempo de evolución del autómata es completamente reversible. Eso significa que, en cualquier momento, en el desarrollo de las reglas, se puede ir hacia adelante o hacia atrás en el tiempo *sin perder información*.

Es un campo de desarrollo muy importante dado que la transformación del mensaje no es una transformación lineal, un cambio en un bit del mensaje genera una cascada de cambios en el mensaje codificado. Ejemplo CAFÉ, Un criptosistema de clave simétrica basado´en A.C.s.

- **Biología**

Se pueden simular varios procesos y sistemas con AC, incluyendo los patrones de las pieles de varios animales, modelos presa-predador y comportamientos de neuronas cerebrales.

Figura 2.14, a la izquierda un tigre con su piel, como resultado de un autómata celular; a la derecha un autómata celular 3D.

Muchos modelos dinámicos interesantes provienen de la biología, cuando interactúan dos especies predador y presa. Este es un ejemplo de modelo predador-presa muy interesante. El universo definido en este modelo se compone de tiburones, peces y vacío. Por otra parte, se introducen funciones aleatorias, es decir si además se cumplen ciertas condiciones, se sigue una regla con cierta probabilidad. Es por lo menos inquietante la piel de algunos animales que casi corresponden a autómatas celulares, en la gráfica 2.14, se puede uno preguntar ¿son ambos animales?

Reglas. Son varias las reglas que describen el comportamiento de cada célula:

- Nacimiento, longevidad y muerte de un pez o un tiburón
- Reproducción de peces y tiburones
- Sobrepoblación o subpoblación
- Interacción pez-tiburón

- **Flujo vehicular**

Se emula el desempeño de un tramo vial, considerado como un A.C., considerando el espacio entre vehículos, el desempeño de un conductor y la dinámica de su vehículo y de los vehículos que comparten el tramo,

- **Simulación de fenómenos naturales**

El modelado de sistemas físicos complejos a través de simulación por computador es una herramienta común para comprender el mundo natural. La complejidad de estos modelos a veces oscurece los principios físicos básicos subyacentes en el fenómeno modelado.

En este contexto se busca simular fenómenos naturales en donde su comportamiento se rige por ecuaciones diferenciales ED o por la interacción local de sus componentes. En el primer caso las ecuaciones ED se convierten en ecuaciones en diferencia EDF, para la discretización del sistema, y se aplican a cada autómata; permitiendo además, agregar algunas características que no permiten que haya inestabilidad estructural. De este modo se han podido modelar campos electromagnéticos, mecánica de fluidos, distribución de calor en placas, entre otros.

Veamos el caso de la transmisión de calor en una barra. Supongamos que se tiene la ecuación diferencial genérica:

$$\frac{dx}{dy} = f(x, y)$$

Lo primero es transformar la ED en EDF, como:

$$x(t+1) = x(t) + f\big(x(t), Y(t)\big)$$

Luego, se reemplazan todas las variables reales x y y por el estado discreto de las variables del AC, entonces el límite de estas variables tiene un alcance finito. De manera que las variables de estado se muestran como conjuntos pequeños finitos.

Veamos un ejemplo con la ecuación de difusión de calor, en 1D:

$$\frac{dT}{dt} = k\frac{d^2T}{dx^2}$$

Donde T es la temperatura y k es la difusividad térmica. La temperatura es una función con valores reales y es continua para cualquier valor de cualesquiera sean las condiciones iniciales. La discretización de la ecuación anterior es:

$$T_i^{n+1} - T_i^n = \frac{k\Delta t}{\Delta x^2}\,(T_{i+1}^n - 2T_i^n + T_{i-1}^n)$$

Donde i y n son los índices espaciales y temporales. Si se escogen los pasos de espacio y de tiempo tal que $\frac{k\Delta t}{\Delta x^2} = \propto$ y haciendo transformaciones nos lleva a la ecuación:

$$T_i^{t+1} = \propto\, \left(T_{i+1}^t - 2T_i^t + T_{i-1}^t\right) + T_i^n$$

Ahora, suponga que tiene una barra de longitud L cuyo extremo izquierdo se pega a una pared que está a 150^oC. Suponga que la temperatura promedio del ambiente es de 15^oC. La barra tiene L = 1m, α = 0.5, divídala en 4 segmentos y diga cuál es la temperatura en el segmento final después de 4 pasos de tiempo.

- **En robótica**

Aunque esta herramienta no se ha utilizado mucho la robótica y en los campos de la producción industrial, o por lo menos se desconoce su uso, vamos a ver un ejemplo donde se puede ver su gran potencialidad.

Se tiene un robot móvil simple equipado con sensores de proximidad y dos motores que pueden girar en ambos sentidos. En este caso el uso de los A.C.s está dirigido a que el robot evite los obstáculos que se le puedan ir presentando en el camino, eliminando las colisiones.

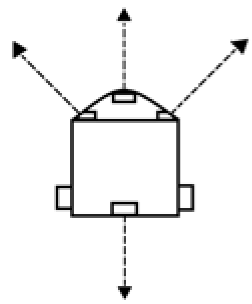

Figura 2.14. Modelo esquemático del robot.

El robot se modela como dos autómatas celulares no uniformes, con una vecindad de Von Neumann. Es un AC no uniforme porque los cambios en los estados de las cuatro células externas, dependen de la información que proveen los sensores. Mientras que la célula central, que representan los motores izquierdo y derecho, cambia de estado dependiendo de los estados de las células anteriores, siguiendo una regla.

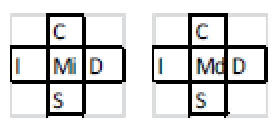

Figura 2.15. Vecindad de los autómatas.

Donde C, D, I, S, son células pasivas que simbolizan los sensores centro, izquierdo, derecho, y sur, las células Mi, Md, son células operacionales activas que representan el motor izquierdo y el motor derecho respectivamente. En este caso solo se trabajan las células C, D, I.

Las células C, D, I de los sensores del robot en el modelo, representan distancias. Haciendo $k = C, D, I$; los estados $S^k = \{0,1,2\}$ de estas células tienen un significado particular: el estado cero es un intervalo entre 0 y 8 cm; el estado 1, un intervalo entre 8 y 16 cm; y el estado 2, un intervalo mayor de 16 cm.

Con respecto al estado de las células de los motores, tienen un significado particular, así los estados de los motores izquierdo y derecho $S^{Mi} = S^{Md} = \{0,1,2\}$, el estado 0, representa el motor apagado; el estado 1, motor encendido, con giro hacia adelante; y el estado 2, motor encendido con giro hacia atrás.

Tomando η_x^t para simbolizar los estados de la vecindad de la célula activa x en el tiempo t, obedece a la regla de transición $\Phi(\eta_x^t)$, que lo lleva al estado $S_x^{t+1} = \Phi(\eta_x^t)$. Considerando que se tienen tres estados para los tres sensores considerados, se tiene un total de 27 reglas.

La tabla de las reglas para los dos autómatas es la siguiente:

M_I	C	D	I	S_{MI}^{t+1}	M_D	C	D	I	S_{MD}^{t+1}
*	0	0	0	2	*	0	0	0	2
*	0	1	0	0	*	0	1	0	2
*	0	2	0	2	*	0	2	0	0
*	0	0	1	2	*	0	0	1	0
*	0	1	1	2	*	0	1	1	0
*	0	2	1	2	*	0	2	1	0
*	0	0	2	2	*	0	0	2	0
*	0	1	2	2	*	0	1	2	0
*	0	2	2	2	*	0	2	2	0
*	1	0	0	2	*	1	0	0	0
*	1	1	0	0	*	1	1	0	2
*	1	2	0	0	*	1	2	0	2
*	1	0	1	2	*	1	0	1	0
*	1	1	1	1	*	1	1	1	0
*	1	2	1	1	*	1	2	1	0
*	1	0	2	2	*	1	0	2	0
*	1	1	2	0	*	1	1	2	2
*	1	2	2	0	*	1	2	2	2
*	2	0	0	0	*	2	0	0	2
*	2	1	0	1	*	2	1	0	0
*	2	2	0	1	*	2	2	0	0
*	2	0	1	0	*	2	0	1	1
*	2	1	1	1	*	2	1	1	0
*	2	2	1	1	*	2	2	1	1
*	2	0	2	0	*	2	0	2	1
*	2	1	2	1	*	2	1	2	1
*	2	2	2	1	*	2	2	2	1

Tabla 2.1. Reglas de transición de los autómatas. * significa que no importa el estado.

Sin embargo, se pueden obtener reglas con un comportamiento mucho mejor, usando para encontrarlas la técnica de Algoritmos Genéticos. En este caso es interesante ver la forma en que el robot va aprendiendo. Con este ejemplo, aunque simple, se permite entrever las muchas aplicaciones que se pueden desarrollar en el campo de la robótica con los ACs.

- **Estudios Teóricos**

En este campo se utilizan los AC para el estudio de temas como complejidad, computación universal, computación en paralelo, síntesis de sistemas, auto-organización, emergencia, entropía, teoría de lenguajes computacionales o estudio de fractales, biología computacional, ciencias sociales entre otros.

- **Otras aplicaciones**

Entre otras aplicaciones que se han trabajado están: la simulación de crecimiento del cáncer, los modelos de uso de la tierra, los modelos de incendios forestales, modelos de reacciones químicas, los patrones de pigmentación de piel, crecimiento de conchas marinas y corales, comportamiento de colonias de microorganismos, modelo de simulación de evacuaciones, entre otros.

El problema inverso
Como problema inverso se plantea el caso en que se conoce un comportamiento a nivel macro, a nivel del SCA, y se desea encontrar las reglas que generan ese comportamiento. Este problema tiene muchas implicaciones importantes dado que en general estamos observando dinámicas de comportamiento que no comprendemos y que queremos explicarnos, no es un problema simple de resolver y en muchos casos se considera un problema abierto.

En “Prototipo de un Sistema Clasificador para Búsqueda de Reglas de Autómatas Celulares en 2D[3]”, se presenta un ejemplo de búsqueda de las reglas de LIFE de Conway a través de un Sistema Clasificador de Holland.

2.8 Diagramas de Voronoi
¿Pero qué pasa con la vecindad si la telesación no es regular? Vemos los planos de una ciudad o de un departamento y no encontramos una forma regular de universo como se definen los A.C.s, para poder utilizar su potencialidad en este tipo de escenarios reales.

Una forma de hacerlo es utilizar los diagramas de Voronoi. Con estos diagramas

2.8 Autómatas Celulares 3D

[3] Prototipo de un Sistema Clasificador para Búsqueda de Reglas de Autómatas Celulares en 2D, L. R. Benavides y O. I. Triana, Facultad de Ingeniería U.N., Bogotá, 2002.

Suponiendo un universo como un cubo, podemos definir nuestros AC 3D. Nos encontramos con que casi todas las reglas de transición que se usan en AC 2D se pueden usar en AC 3D, la gran diferencia está en que la vecindad de los AC 3D es mucho más grande.

Si se trabajan los AC 1D en 3D, con r=1, y = {0, 1}, cada autómata es un cuadro en un plano y la otra dimensión la da el tiempo. Se tienen ahora 4294967296 autómatas y 3283936144 reglas. En la figura 5.14, se presenta una extensión de la regla 30 AC 1D a 3D.

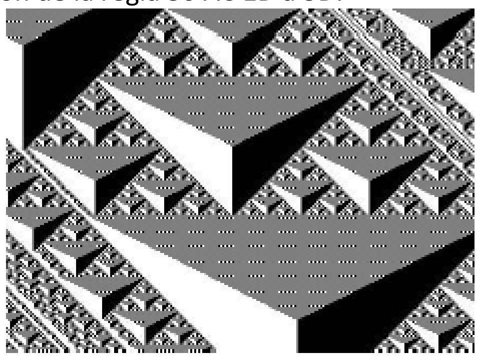

Figura 6.14. Extensión de la regla 30 AC 1D a 3D[4]*.*

Con el desarrollo de las impresoras 3D, se pueden generar patrones emergentes estocásticos. Las impresoras 3D se pueden considerar como autómatas celulares asincrónicos 3D, pues generan patrones 3D acumulando patrones 1D, el caso más simple de un AC 1D asincrónico. Esto nos permite ampliar nuestras posibilidades de utilizar los AC 3D en muchas otras aplicaciones.

2.9 Ejercicios y Problemas.

1. Observe sus comportamientos en la casa, en la universidad y en el medio de transporte que utiliza. Encuentre, para cada uno de estos escenarios sus reglas básicas.
2. En la librería de modelos de NetLogo, encuentre una aplicación de AC, describa el modelo, córralo y haga un análisis del resultado.
3. Imprima un AC 1D en una impresora 3D.
4. Tome el plano de una ciudad pequeña y localice, por ejemplo, las droguerías, ¿es posible que falte alguna en la ciudad? Utilice diagramas de Voronoi.
5. Con base en las aplicaciones, escoja una de ellas y realice la simulación. .

2.10 Bibliografía

Cortes, C., P., y Cubillos, M. G., Desarrollo de Software Adaptativo para el Aprendizaje de Robots Móviles, Basado en Autómatas Celulares, Facultad de Ingeniería, Universidad Nacional, Bogotá, 1999.
Kanada Y, 3D Printing and Simulation of Naturally-Randomized Cellular-Automata, AROB, 2014Mitchell Melanie, Introduction of Complexity, MOOC,2015
Martínez José J., Notas Vida Artificial, 2009.
Ortiz Z, Martínez J., Cifuentes J., Cortés E, Ortíz E, CAFE: Un Criptosistema de Clave Simétrica Basado en Autómatas Celulares, Congreso Colombiano de Inteligencia Computacional, 2009.
Weisbuch G, Rickebush S, Complex System Dynamics, Santa Fe Institute, 1990.
Wolfram, S. *A New Kind of Science.* Champaign, IL: Wolfram Media, 2002.

[4] https://nbickford.wordpress.com/2010/05/19/cellular-automata/

http://roman.montclair.edu/Research/Parallel/
http://mathworld.wolfram.com/ElementaryCellularAutomaton.html
https://nbickford.wordpress.com/2012/01/22/sliding-block-puzzles-part-3/
http://natureofcode.com/book/chapter-7-cellular-automata/
https://argentinainvestiga.edu.ar/noticia.php?titulo=un_modelo_de_simulacin_de_evacuaciones_que_emplea_autmatas_celulares&id=2142_ Sobre modelos de evacuación.
https://www.youtube.com/watch?v=qjEXwJScXyw Sobre diagramsa de Voronoi.