

Introducción a la simulación de procesos con autómata celular

Genaro Juárez Martínez

genarojm@correo.unam.mx

<http://uncomp.uwe.ac.uk/genaro/>

Departamento de Posgrado, Escuela Superior de Cómputo
Instituto Politécnico Nacional

iGEM-México (IPN-UNAM)

http://parts.mit.edu/wiki/index.php/IPN_UNAM_2006

Agosto-Octubre, 2006
México DF

Part I

Antecedentes

Precursor – John von Neumann

Históricamente la teoría de autómatas celulares puede clasificarse en tres etapas con un alto grado de estudio.

- ▶ La primera etapa es con el precursor John von Neumann. Matemático de origen húngaro quien tiene una participación decisiva en la construcción de la primera computadora (la ENIAC en 1946). Von Neumann a principios de los años 50's desarrolla el análisis para describir un sistema con la capacidad de soportar dos principales características: sistemas complejos y sistemas con la capacidad de auto-reproducción. El desarrollo de esta nueva teoría además es influenciada por Stanislaw Ullam, proponiendo a von Neumann implementar su nueva teoría en un espacio celular discreto. Un hecho importante es que von Neumann no concluye la escritura de su obra "Theory of Self-reproducing Automata" porque muere antes, en 1957. La publicación de su libro es editada y completada por A. W. Burks hasta 1966.

Segunda etapa – John Horton Conway

- ▶ La segunda etapa llega a finales de los años 60's cuando Martin Gardner divulga en su columna científica *Mathematical Games* del *Scientific American* en 1969, el autómata celular propuesto por John Horton Conway, mejor conocido como “El Juego de la Vida.” Este autómata celular sería presentado como un juego ecológico durante una serie de resultados establecidos a principios de los años 70's. Varios de los resultados más interesantes serían obtenidos en el Laboratorio de Inteligencia Artificial del MIT con las facilidades gráficas de una computadora PDP-6, sobresaliendo los resultados obtenidos por William Gosper y sus estudiantes. El autómata celular de Conway había logrado captar la atención de varios investigadores, principalmente por su sistema de estructuras complejas. Estas estructuras complejas fueron conocidas como objetos Life, donde destacan los gliders y los generadores de gliders (también referidos como partículas o localidades móviles).

Tercera etapa – Stephen Wolfram

- ▶ La tercera etapa llega a mediados de los años 80's con Stephen Wolfram precursor del estudio de autómatas celulares en una dimensión. Wolfram inicia sus investigaciones aplicando varios conceptos de dinámica no-lineal y mecánica estadística, explotando muchas de las facilidades gráficas que en ese tiempo ya se tenían, por lo que se podían realizar muchos estudios experimentales. Un dato importante, es que a diferencia de los modelos propuestos por von Neumann y Conway, donde las funciones representaban números muy grandes y cada autómata tiene un propósito específico para resolver un problema. Wolfram realiza un estudio sistemático tomando un conjunto de reglas y estudia sus evoluciones. Esta idea lo lleva a una búsqueda por establecer una clasificación dentro de las evoluciones de todos los autómatas de cierto grado. Dicha clasificación es conocida como las "clases de Wolfram," que debemos discutir más adelante.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

Part II

Dinámica en autómata celular

Autómata celular

Definición

Un *autómata celular* es un sistema dinámico discreto evolucionando en un arreglo regular infinito.

La dinámica es la siguiente. Tenemos un conjunto finito de estados (nuestro alfabeto) y una función local afectando una cantidad de estados dentro del arreglo regular, donde cada elemento del arreglo toma un valor del conjunto de estados y es llamada una *célula*. El arreglo regular es mejor conocido como el *espacio de evoluciones* dentro de la literatura de autómata celular (AC). La función local es importante porque determina el comportamiento del AC en estudio. La *función local* es determinada por una *célula central* y sus células vecinas, formando una *vecindad*. La vecindad es el número de argumentos que la función local recibe. Finalmente las transformaciones determinadas para cada vecindad diferente corresponden a un elemento del conjunto de estados. Como ingrediente final necesitamos una asignación de estados en el espacio de evoluciones que representa la *configuración inicial* del sistema.

Genaro

Estructura general

[AC en una dimensión](#)

[AC en dos dimensión](#)

[AC en tres dimensión](#)

[El autómata de von Neumann](#)

[El autómata de Conway](#)

[Los autómatas de Wolfram](#)

[Morfología en AC](#)

[AC complejo](#)

Conclusiones

Referencias

[Software libre en AC](#)

Entonces podemos definir formalmente un AC de la siguiente manera:

Definición

Un *autómata celular* es una 4-tupla $A = \langle \Sigma, u, \varphi, c_0 \rangle$ evolucionando en $d \in \mathbb{Z}^+$ dimensión, con un conjunto finito de estados Σ , una conexión local u tal que $u = \{x_{i,j}, \dots, n:d\}$, la función de transición $\varphi : u \rightarrow \Sigma$ y la condición inicial del sistema c_0 .

Algunos espacios de evoluciones

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

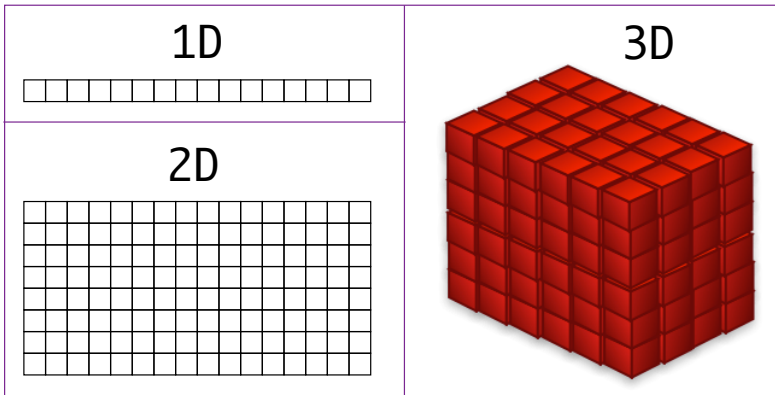


Figure: AC puede evolucionar en alguna d -dimensión.

AC en una dimensión

En una dimensión tenemos un arreglo lineal, donde cada una de las células toma un elemento del conjunto de estados $\Sigma = \{0, 1\}$:

$$\dots | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | \dots$$

En este ejemplo analizamos un AC con un radio de vecindad igual a uno. De esta manera, tenemos que la función local debe tomar tres células para evaluar una vecindad, de la siguiente manera:

$$\varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \rightarrow x_i^{t+1} \quad \forall x_i \in \Sigma$$

la evaluación de cada vecindad en la t configuración es realizada de manera simultánea produciendo la $t + 1$ configuración.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

AC en una dimensión

Wolfram representa el orden de los AC en una dimensión con dos parámetros: (k, r) . Donde k representa el número de estados de Σ (su cardinalidad) y r representa el radio de vecindad.

Para el ejemplo anterior tenemos un AC de orden $(2,1)$, es decir, dos estados y un radio de vecindad de una célula con respecto a la célula central (también llamado AC elemental por Wolfram). De esta manera, tenemos que el número de vecinos con respecto a una célula central es $2r$ y la vecindad es formada por $2r + 1$ células. Por lo tanto, la función local (también llamada *regla de evolución*) tiene k^{2r+1} vecindades diferentes y este AC tiene $k^{k^{2r+1}}$ reglas de evolución diferentes.

Para el caso $(2,1)$ tenemos: $2(1) = 2$ vecinos, $2(1) + 1 = 3$ células en una vecindad, $2^{2(1)+1} = 8$ vecindades diferentes y $2^{2^{2(1)+1}} = 256$ reglas de evolución diferentes.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC AC complejo

Conclusiones

Referencias

Software libre en AC

AC en una dimensión

Por ejemplo, para conocer la evolución de la configuración anterior debemos especificar la función local. Esta relación determina de manera general todas las transformaciones de la siguiente manera:

$$\varphi : \Sigma^{2r+1} \rightarrow \Sigma \quad (1)$$

Analizemos la siguiente regla de evolución:

$$\varphi(0, 0, 0) \rightarrow 0$$

$$\varphi(0, 0, 1) \rightarrow 1$$

$$\varphi(0, 1, 0) \rightarrow 1$$

$$\varphi(0, 1, 1) \rightarrow 0$$

$$\varphi(1, 0, 0) \rightarrow 1$$

$$\varphi(1, 0, 1) \rightarrow 0$$

$$\varphi(1, 1, 0) \rightarrow 0$$

$$\varphi(1, 1, 1) \rightarrow 0$$

Genero

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

AC en una dimensión

La regla de evolución expresada en notación binaria es: $(00010110)_2$. En notación decimal representa la regla 22. Ahora regresemos a nuestra configuración inicial anterior y apliquemos iterativamente la regla 22.

00101110011100101001	$t = 0$	$10/20 = 0.5$
11100001100011101111	$t = 1$	$12/20 = 0.6$
00010010010100000000	$t = 2$	$4/20 = 0.2$
00111111110110000000	$t = 3$	$10/20 = 0.5$
01000000000001000000	$t = 4$	$2/20 = 0.1$
11100000000011100000	$t = 5$	$6/20 = 0.3$
00010000000100010001	$t = 6$	$4/20 = 0.2$
10111000001110111011	$t = 7$	$12/20 = 0.6$
\vdots	\vdots	\vdots

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

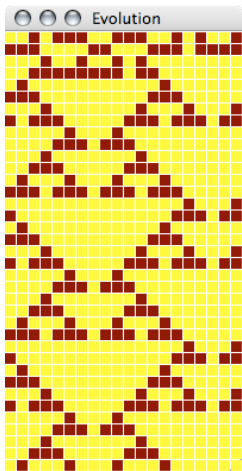
Morfología en AC
AC complejo

Conclusiones

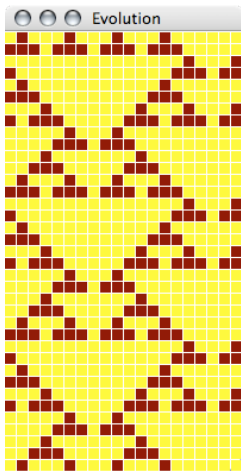
Referencias

Software libre en AC

AC en una dimensión



(a)



(b)

Figure: Regla 22 evolucionando con una configuración inicial aleatoria de 20 células: (a) ilustra las primeras 37 generaciones y (b) las últimas 74. Encontrando un período cada 8 generaciones.

AC en una dimensión

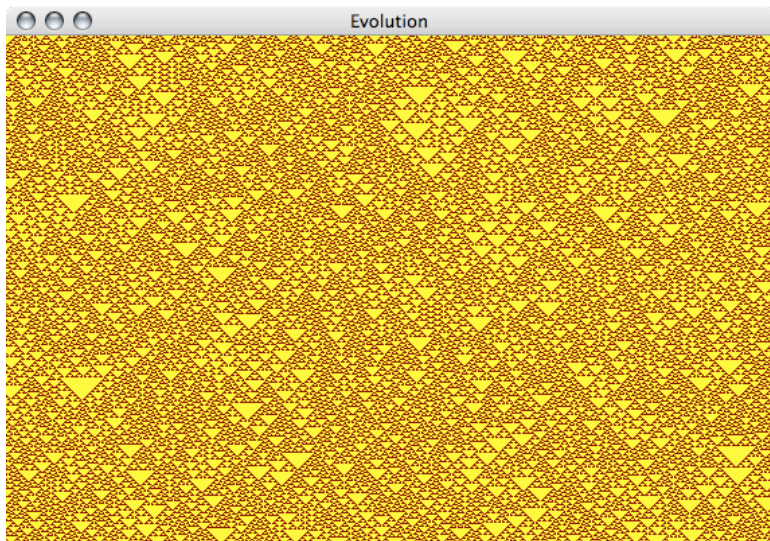


Figure: Regla 22 evolucionando con una densidad inicial aleatoria de 0.31012657, en un arreglo de 572 células, para 377 gen.

AC en una dimensión

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

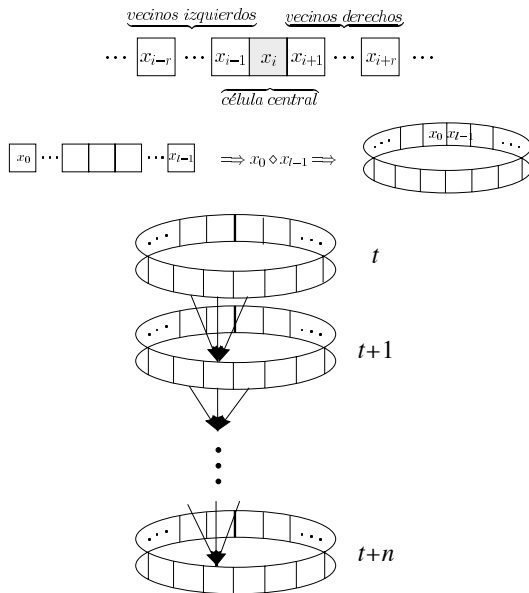
Software libre en AC

La función local establece las relaciones entre cada vecindad con un estado de Σ . Entonces una vez que hemos evaluado todo el espacio de evoluciones generamos el siguiente arreglo y a sí sucesivamente. Cada arreglo generado representa un estado del sistema en su evolución. De esta manera, la función local φ da origen a una *función global* Φ donde las relaciones se dan entre configuraciones de la siguiente manera:

$$\Phi : \Sigma^{\mathbb{Z}} \rightarrow \Sigma^{\mathbb{Z}} \quad (2)$$

donde $\Sigma^{\mathbb{Z}}$ representa el espacio de todas las configuraciones en una dimensión.

AC en una dimensión



AC en dos dimensiones

En dos dimensiones tenemos un arreglo definido por las parejas ordenadas $\mathbb{Z} \times \mathbb{Z}$. De igual manera que una dimensión, cada una de las células $x_{i,j}$ toma un elemento del conjunto de estados Σ .

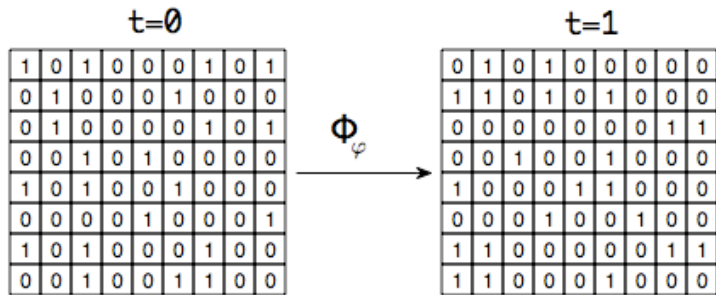


Figure: Espacio de evoluciones en dos dimensiones.

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

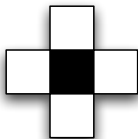
Conclusiones

Referencias

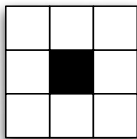
Software libre en AC

AC en dos dimensiones

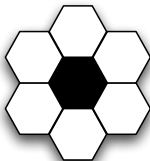
La regla de evolución puede encontrar algunas variantes a través de la historia. Sin embargo, solo consideramos las más conocidas. Iniciamos con la *vecindad de von Neumann* definida para una célula central y sus vecinos ortogonales (figura a). La *vecindad de Moore* que además de los vecinos ortogonales se consideran los vecinos diagonales (figura b). Por último la *vecindad hexagonal* (figura c).



(a)



(b)



(c)

Figure: Tipos de vecindad en dos dimensiones.

Estructura general

[AC en una dimensión](#)[AC en dos dimensión](#)[AC en tres dimensión](#)[El autómata de von Neumann](#)[El autómata de Conway](#)[Los autómatas de Wolfram](#)[Morfología en AC](#)
[AC complejo](#)

Conclusiones

Referencias

[Software libre en AC](#)

AC en dos dimensiones

De igual manera cada célula $x_{i,j}$ es evaluada por la función local de manera simultánea a todas las células para generar la siguiente configuración. Sin embargo, debemos tener en cuenta que el número de vecindades es mayor y tratar de representar la regla de evolución se convierte en un problema de simplificar el número de variables para la función φ .

Aunque este problema lo tenemos en todas las dimensiones, cuando $d > 1$ es aún más difícil.

Para ello recurrimos a dos tipos de notaciones: las *reglas totalísticas* o las *reglas semi-totalísticas*.

Reglas totalísticas:

$$\varphi(a, b, c) = \tau(a + b + c) \quad (3)$$

En las reglas totalísticas se suman los valores de los elementos que forman la vecindad y todas aquellas vecindades que correspondan a esa suma, evolucionan al mismo valor.

Reglas semi-totalísticas:

$$\varphi(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_V) = \begin{cases} 1 & \text{si} \begin{cases} \mathbf{x}_0 = 0 & \text{y} & B_{min} \leq \sum_{i=1}^V \mathbf{x}_i \leq B_{max} \\ \mathbf{x}_0 = 1 & \text{y} & S_{min} \leq \sum_{i=1}^V \mathbf{x}_i \leq S_{max} \end{cases} \\ 0 & \text{en otro caso} \end{cases} \quad (4)$$

En el caso de la reglas semi-totalísticas igualmente se realiza una suma, pero ésta solo es realizada con los vecinos sin tomar en cuenta la célula central. La célula central es considerada solo para determinar en que condición debe evaluarse la suma. En la ecuación anterior tenemos el caso binario donde \mathbf{x}_0 representa la célula central $x_{i,j}$ y las variables $\mathbf{x}_1, \dots, \mathbf{x}_V$ son el número de vecinos en dos dimensiones. La suma de estos vecinos V también es referida como la *vecindad isotrópica*.

[Estructura general](#)[AC en una dimensión](#)[AC en dos dimensión](#)[AC en tres dimensión](#)[El autómata de von Neumann](#)[El autómata de Conway](#)[Los autómatas de Wolfram](#)[Morfología en AC](#)
[AC complejo](#)[Conclusiones](#)[Referencias](#)[Software libre en AC](#)

AC en dos dimensiones

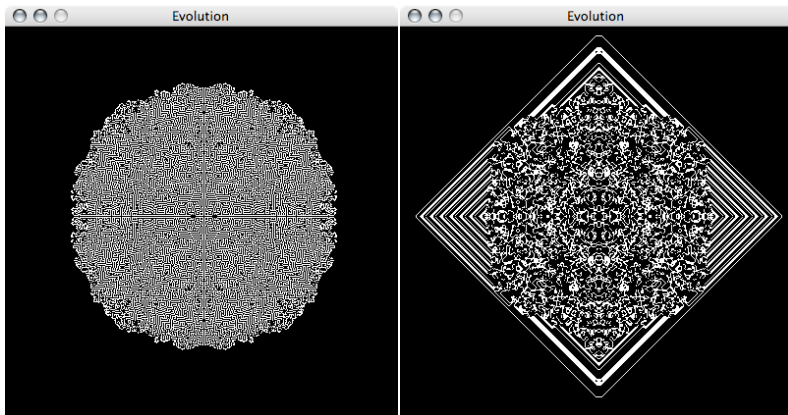
En las reglas semi-totalísticas la célula central juega un papel relevante, porque dependiendo del valor entonces la suma es condicionada a una pareja de parámetros diferentes. En la ecuación 4 los parámetros B_{min} y B_{max} son evaluados si la célula central tiene el valor de 0. Por el contrario, si tiene el valor de 1 entonces se evalúan los parámetros S_{min} y S_{max} . De esta manera podemos representar una regla de evolución semitotalística como: $R(S_{min}, S_{max}, B_{min}, B_{max})$.

Ejemplos

1. Regla de evolución $R(2, 4, 3, 3)$.
2. Regla de evolución $R(1, 2, 2, 3)$.

[Estructura general](#)[AC en una dimensión](#)[AC en dos dimensión](#)[AC en tres dimensión](#)[El autómata de von Neumann](#)[El autómata de Conway](#)[Los autómatas de Wolfram](#)[Morfología en AC](#)[AC complejo](#)[Conclusiones](#)[Referencias](#)[Software libre en AC](#)

AC en dos dimensiones



(a)

(b)

Figure: Iniciamos con una línea de 7 células en el estado 1. La evolución (a) con la regla $R(2, 4, 3, 3)$ construye un crecimiento circular con 29,464 células en el estado 1 en 402 generaciones. (b) muestra el resultado con la regla $R(1, 2, 2, 3)$ produciendo un crecimiento mixto entre caos y periodicidad con 24,142 células en 183 generaciones.

AC en dos dimensiones

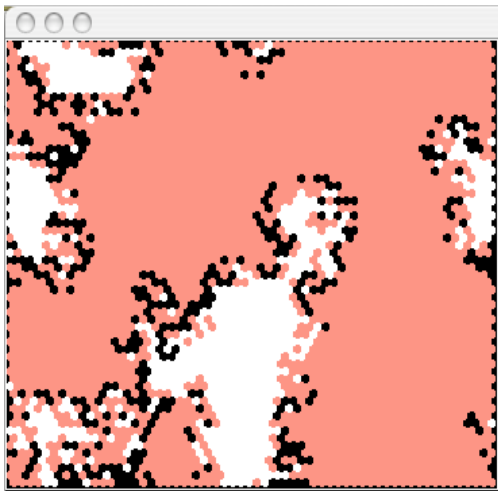


Figure: Ejemplo de un AC de arreglo hexagonal con la regla de evolución 51 54 44 65 06 a9 94, con tres estados en un espacio de 3,600 células (DDLAB software).

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

AC en tres dimensiones

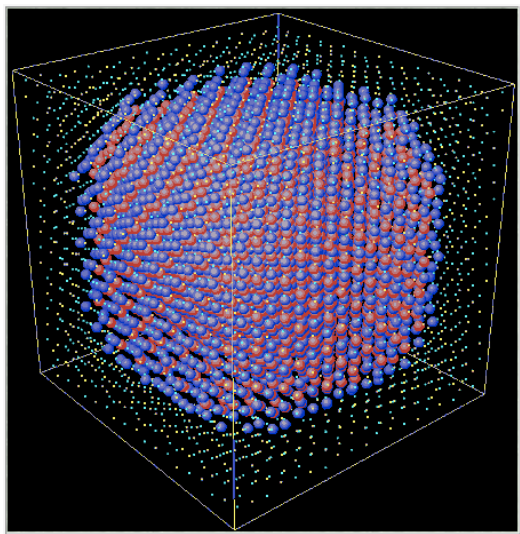


Figure: Evolución por E. Fredkin presentando un crecimiento casi-simétrico (http://finitenature.com/fredkin_essay/).

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

El autómata de von Neumann

El AC propuesto por von Neumann fue construido para resolver dos problemas fundamentales: la capacidad de resolver problemas complejos, como la implementación de una máquina universal y también con la capacidad de copiar dicha máquina, construyendo una máquina similar o de mayor complejidad, es decir, la capacidad de reproducirse a sí mismo. La complejidad del AC de von Neumann es determinada inicialmente por el conjunto de estados $\Sigma = \{0, \dots, 28\}$, evolucionando con la vecindad de von Neumann, generando $29^5 = 20,511,149$ vecindades!

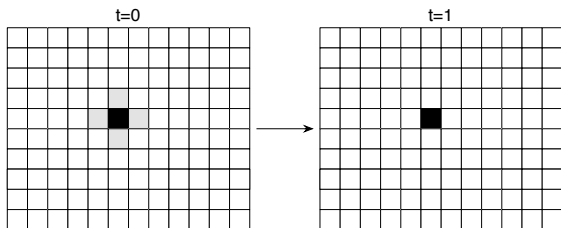


Figure: La función local evalúa la vecindad de von Neumann

$$\varphi(x_{i-1,j}^t, x_{i,j-1}^t, x_{i,j}^t, x_{i+1,j}^t, x_{i,j+1}^t) \rightarrow x_{i,j}^{t+1} \quad \forall x_{i,j} \in \Sigma.$$

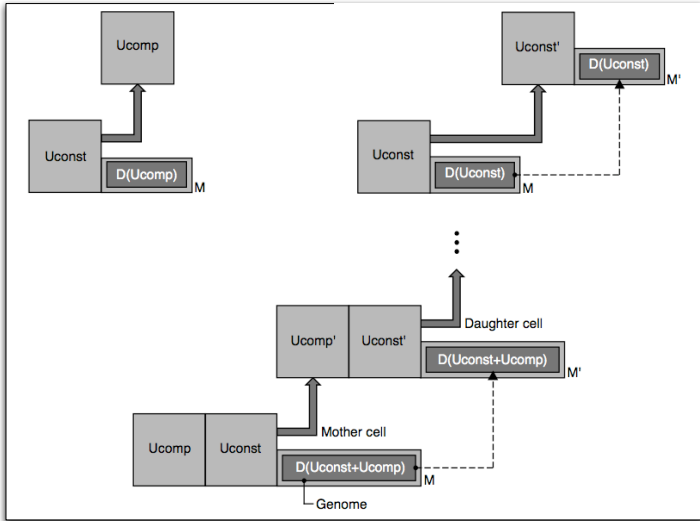


Figure: AC von Neumann (fig. por tesis de Gianluca Tempesti).

Life

El autómata celular de Conway en dos dimensiones llamado “*The Game of Life*,” es llamado así por su relación que tiene con la modelación de sistemas biológicos, tal como: ecosistemas, incendios forestales, comportamientos colectivos de seres vivos, entre otros. Entonces tenemos que $\Sigma = \{0, 1\}$ es el conjunto de estados, el estado 0 representa una célula muerta (color blanco) y el estado 1 representa una célula viva (color negro).

La función de transición φ emplea la vecindad de Moore. La regla de evolución es semi-totalística y es representada como

$R(2, 3, 3, 3)$. La primera pareja de números (2,3) son un mínimo y un máximo respectivamente, para definir la sobrevivencia de una célula que está viva. La segunda pareja de números (3,3) también son un mínimo y un máximo respectivamente, para definir el nacimiento de una célula. De esta manera, las variables de la Ecuación 4 toman los valores de $S_{min} = 2$, $S_{max} = 3$, $B_{min} = 3$ y $B_{max} = 3$. Por eso es que S representa la *sobrevivencia* en el tiempo $t + 1$ de una célula que se encuentra viva en el tiempo t y B representa el *nacimiento* de una célula en el tiempo $t + 1$ de una célula que se encuentra muerta en el tiempo t .

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Life

Conway hace un amplio análisis para poder determinar una regla de evolución que tuviera dos características fundamentales:

1. Una configuración no debe desaparecer rápidamente.
2. Una configuración debe crecer ilimitadamente.

además se deben de definir tres condiciones, cuando una célula debe de *nacer*, *sobrevivir* o *morir*. Estas condiciones son muy importantes para obtener las características mencionadas.

- (a) **Nacimiento.** Una célula muerta en el tiempo t llega a vivir en el tiempo $t + 1$ si tiene exactamente tres de sus vecinos vivos en el tiempo t .
- (b) **Muerte por sobre-población.** Si una célula vive en el tiempo t y tiene cuatro o más de sus ocho vecinos vivos en el tiempo t , esta célula debe morir en el tiempo $t + 1$.
- (c) **Muerte por aislamiento.** Si una célula vive en el tiempo t y tiene un vecino vivo o ninguno en el tiempo t , esta célula debe morir en el tiempo $t + 1$.
- (d) **Sobrevivencia.** Una célula que vive en el tiempo t deberá permanecer viva en el tiempo $t + 1$, si y solo si, tiene dos o tres vecinos vivos en el tiempo t .

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Por lo tanto la Ecuación 4 para la regla $R(2, 3, 3, 3)$ en particular queda como:

$$\varphi(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_8) = \begin{cases} 1 & \text{si} \begin{cases} \mathbf{x}_0 = 0 & \text{y} & 3 \leq \sum_{i=1}^8 \mathbf{x}_i \leq 3 \\ \mathbf{x}_0 = 1 & \text{y} & 2 \leq \sum_{i=1}^8 \mathbf{x}_i \leq 3 \end{cases} \\ 0 & \text{en otro caso} \end{cases} \quad (5)$$

se tiene que $3 \leq \sum_{i=1}^8 \mathbf{x}_i \leq 3$ es el intervalo de células vivas entre B_{min} y B_{max} para un nacimiento y $2 \leq \sum_{i=1}^8 \mathbf{x}_i \leq 3$ es el intervalo de células vivas entre S_{min} y S_{max} para la sobrevivencia. Las células se encuentran en nuestro espacio de evoluciones $\mathbb{Z} \times \mathbb{Z}$ donde las transformaciones para cada una de las células $x_{i,j}$ se efectúan de manera simultánea o en paralelo.

[Estructura general](#)
[AC en una dimensión](#)
[AC en dos dimensión](#)
[AC en tres dimensión](#)
[El autómata de von Neumann](#)
[El autómata de Conway](#)
[Los autómatas de Wolfram](#)
[Morfología en AC](#)
[AC complejo](#)
[Conclusiones](#)
[Referencias](#)
[Software libre en AC](#)

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

Comportamientos en *Life*

El interés que se originó en *Life* se debe principalmente a los comportamientos complejos que la regla puede producir, llegando a resultados muy importantes como fue el hecho de demostrar que la regla *Life* puede realizar computación universal.

En el espacio de evoluciones existen cuatro tipos de comportamientos:

1. Configuraciones que desaparecen.
2. Configuraciones estáticas.
3. Configuraciones periódicas.
4. Configuraciones periódicas con desplazamiento.

evolucionando sobre un fondo estable de estados 0's. En la literatura de *Life* se les han asignado nombres a cada una de estas estructuras como:

blinker, blanker, block, beehive, loaf, pond, tub, traffic lights, ships, baker, barber, train, beacon maker, bigship, hertz, Z, rocket, honey farm, snake, barge, boat, glider, flip flop, pants, glider gun, ticks, flashes, spaceships, boat maker, cheshire cat, mega flip flop, pool table, pin wheel, catherine wheel, puffer, candelabra, mosaic kills virus, latin cross, toad, r-pentomino, clock, swastika, etc..

Comportamientos en *Life*

Por ejemplo, tenemos estructuras de células vivas que desaparecen en pocas iteraciones, es decir, todas las células vivas van muriendo como se ilustra en la figura. La cruz *swastika* desaparece por completo a partir de la sexta generación, se puede observar la simetría que existe en el espacio de evoluciones a través del tiempo. En particular el autómata celular *Life* es uniforme, determinístico y simétrico.

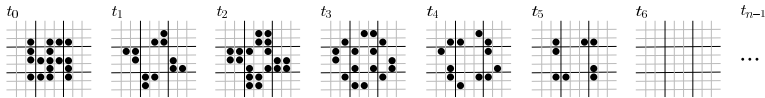


Figure: Configuración que desaparece (estructura *swastika*).

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Comportamientos en *Life*

Una configuración es estable a través del tiempo, si ésta no cambia su forma ni desaparece. A todas estas configuraciones se les conoce como *naturaleza muerta* o *still life*. A partir de la tercera generación la estructura de nombre *beehive* compuesta de seis células, se mantiene estática a través del tiempo. *Life* tiene varias formas de estas estructuras, si se evoluciona una configuración aleatoria muchas de estas estructuras se producen casi inmediatamente.

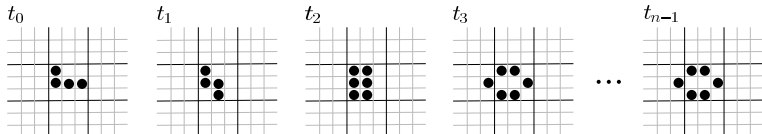


Figure: Configuración estática (estructura estable *beehive*).

Comportamientos en *Life*

También existen estructuras con un período dado y las configuraciones que producen estas estructuras son fijas, dentro de la literatura tenemos dos nombres que identifican dichas osciladores: *flip-flop* o *blinker*. En la figura se puede ver como una población compuesta de tres células cambia de su posición horizontal a una posición vertical a través del tiempo.

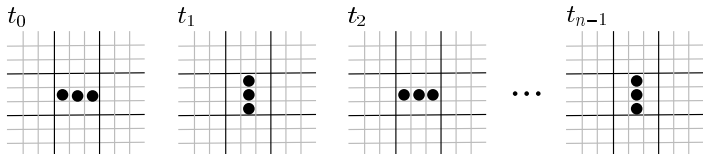


Figure: Oscilador (estructura flip-flop).

Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Glider (partícula o localidad móvil)

Este tipo de estructuras son muy interesantes e importantes en esta regla. Muchos de los descubrimientos más relevantes están relacionados a este tipo de estructuras, determinando las trayectorias que pueden tener a través del espacio y que comportamientos o estructuras tan complejas se pueden llegar a producir cuando estos gliders chocan. Además estos choques se pueden realizar con cualquier otra estructura, varias de estas estructuras complejas que se han llegado a obtener se han logrado a través de la computadora CAM (Cellular Automata Machine), desarrollada en el MIT por Tommaso Toffoli y Norman Margolus. El glider más relevante en la literatura de Life es el glider de cinco células como se ilustra en la figura, donde este glider tiene la característica de ser simétrico en un arreglo de 3×3 . Su período de desplazamiento es en cuatro iteraciones.

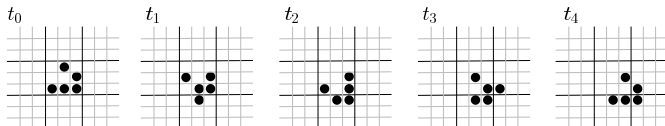


Figure: Configuración periódica con desplazamiento *glider*.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Comportamientos en *Life*

Existen otros tipos de gliders que varían de forma, tamaño, período de desplazamiento y dirección en el espacio de evoluciones, por lo tanto comportamientos de este tipo fueron los que originaron un interés más a fondo para poder explicar el origen y el alcance de estas estructuras. El *glider gun* de la figura es una estructura que produce gliders, de manera continua cada treinta iteraciones. Pero además algo verdaderamente sorprendente es que 8 gliders de cinco células pueden construir su propio glider gun! De esta manera, Life crece ilimitadamente y puede producir sus estructuras en base a sus propios partículas.

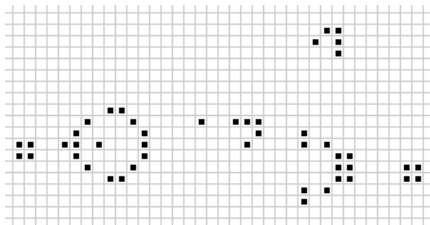


Figure: Glider gun.

Discusión y simulación de evoluciones en Life con los sistemas OSX2DCASM y Golly.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Clases de Wolfram

Wolfram realiza todo un estudio sistemático en AC de una dimensión. Principalmente en AC elemental (orden (2,1)) y de otros ordenes con representación de reglas totalísticas. Una de sus principales aportaciones son las conocidas *clases de Wolfram*.

Karel Culik II define estas clases de la siguiente manera:

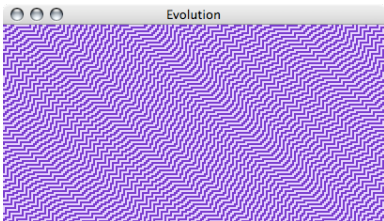
Clasificación

- ▶ Un AC es de clase I si existe un estado estable $x_i \in \Sigma$ tal que todas las configuraciones finitas x_i evolucionan a la *configuración homogénea* de x_i .
- ▶ Un AC es de clase II si existe un estado estable $x_i \in \Sigma$ tal que cualquier configuración finita tiene una evolución que llega a ser periódica.
- ▶ Un AC es de clase III si existe un estado estable $x_i \in \Sigma$ tal que para alguna pareja de configuraciones finitas x_i de c_i y c_j , es decidible si c_i evoluciona a c_j .
- ▶ Un AC es de clase IV si incluye todo AC.

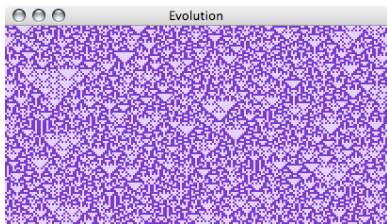
Clases de Wolfram



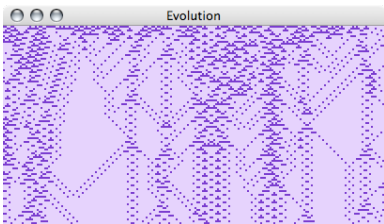
Regla 0



Regla 15



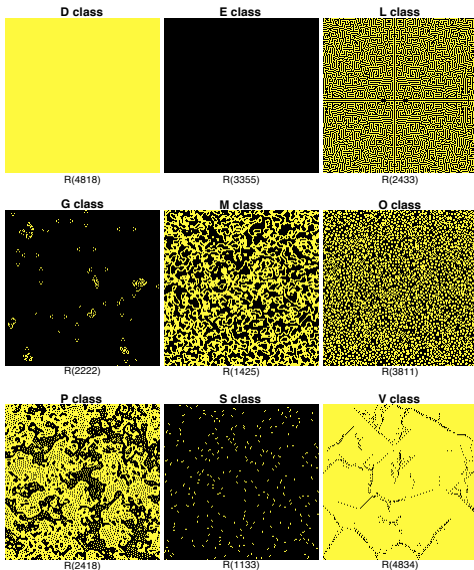
Regla 90



Regla 54

Figure: Cuatro clases en AC de una dimensión.

Clases morfológicas en AC de dos dimensiones



Genaro

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Figure: Clasificación morfológica en AC de dos dimensiones.

Clases morfológicas en AC de dos dimensiones

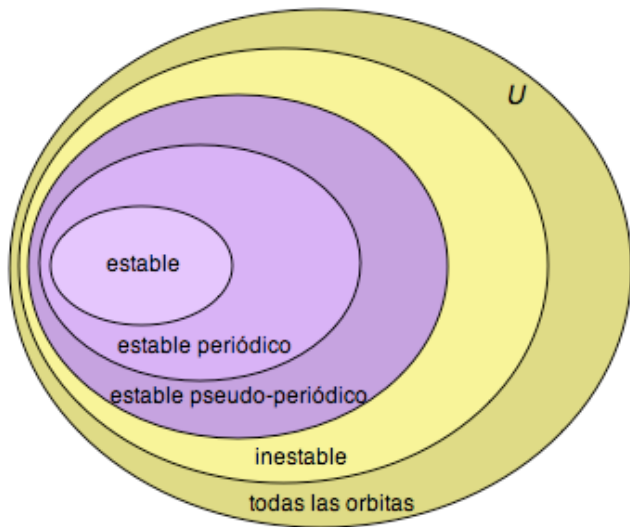


Figure: Clasificación morfológica en AC de dos dimensiones.

Computación convencional

Toda la computación realizada hasta ahora es a través de transformaciones seriales. Ellas se realizan internamente en el microprocesador o cualquier dispositivo electrónico de la computadora.

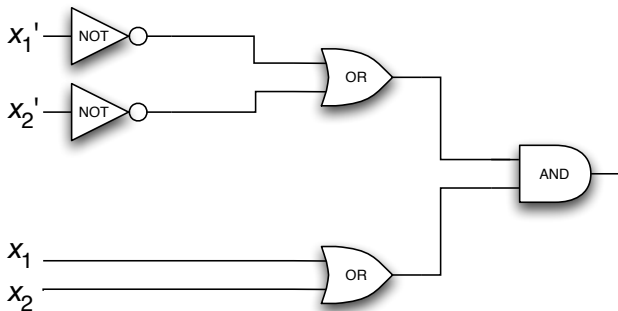


Figure: Implementación de una red derivada de la función normal disyuntiva $f_9 = x_1' \cdot x_2' + x_1 \cdot x_2$. En un circuito utilizando las compuertas lógicas OR, NOT y AND.

Computación no-convencional

Computación no-convencional es investigada y desarrollada en tres grandes ciencias. La manera más natural de encontrar un proceso que es realizado simultáneamente con miles, millones, trillones de elementos, podemos encontrarlos en la naturaleza misma, en nuestro universo.

Tres ciencias convergen

1. Computación cuántica (computadora 7-qubit)
2. Computación de reacción difusión (computadora de reacción-difusión)
3. Computación DNA (computadora auto-ensamblada)

[Estructura general](#)[AC en una dimensión](#)[AC en dos dimensión](#)[AC en tres dimensión](#)[El autómata de von Neumann](#)[El autómata de Conway](#)[Los autómatas de Wolfram](#)[Morfología en AC](#)[AC complejo](#)[Conclusiones](#)[Referencias](#)[Software libre en AC](#)

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Computación no-convencional

Toda la computación realizada hasta ahora es a través de transformaciones seriales. Ellas se realizan internamente en el microprocesador o cualquier dispositivo electrónico de la computadora.

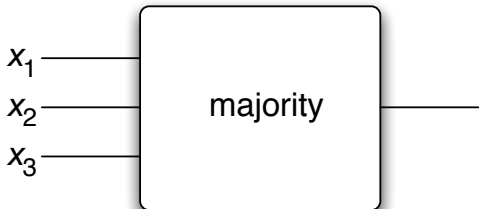


Figure: Interpretación de una compuerta cuántica MAJORITY.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfolgía en AC
AC complejo

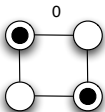
Conclusiones

Referencias

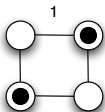
Software libre en
AC

Computación no-convencional

célula cuántica - dot



polarización -1



polarización +1

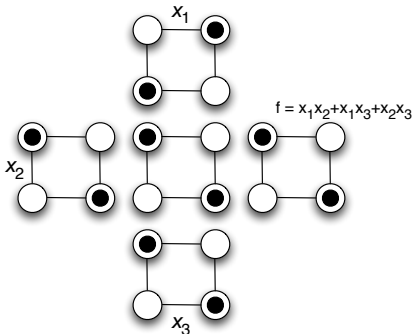


Figure: Implementación de la compuerta cuántica MAJORITY

$$f = x_1x_2 + x_1x_3 + x_2x_3.$$

Computación no-conventional

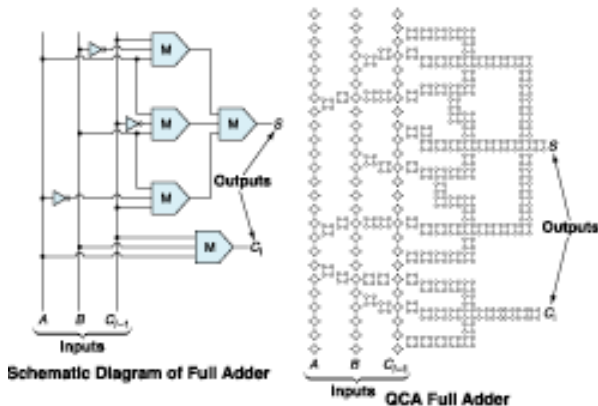


Figure: El circuito para realizar un sumador es implementado con tres compuertas inversores y cinco compuertas mayoritarias, dados dos bits de entrada (A y B) y un bit de acarreo (C_{i-1}), generando una suma en el bit S y un nuevo acarreo C_i . Donde ellos son arreglados en un AC quantum-dot (fig. de <http://www.nasatech.com/Briefs/Jan03/NP020869.html>).

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Procesos naturales

Autómata celular con comportamiento complejo es caracterizado por aquellas reglas de evolución que soportan partículas.

Sin embargo debemos considerar dos cosas. Simular procesos biológicos y procesos computables, porque ambas establecen una división muy fuerte en su implementación y en su ingeniería.

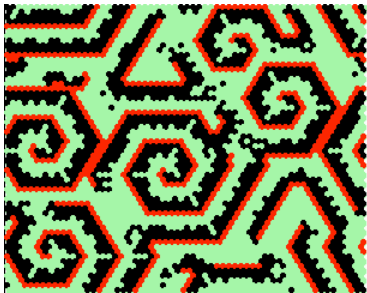


Figure: Simulando patrones espirales con tres estados (DDLAB software).

AC con comportamiento complejo (clase **G**)

AC en una dimensión

1. Rule 110

<http://uncomp.uwe.ac.uk/genaro/Rule110.html>

2. Rule 54

<http://uncomp.uwe.ac.uk/genaro/Rule54.html>

AC en dos dimensiones

1. The game of life

2. Diffusion Rule

<http://uncomp.uwe.ac.uk/genaro/diffusionLife/diffusionLife.html>

AC en dos dimensiones hexagonal

1. Beehive Rule

http://www.cogs.susx.ac.uk/users/andywu/multi_value/self_rep.html

2. Spiral Rule

http://www.cogs.susx.ac.uk/users/andywu/multi_value/spiral_rule.html

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

AC complejos (clase **G**)

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Discusión y simulación de evoluciones de las reglas 110, 54 en 1D y las reglas Diffusion, Beehive y Spiral en 2D. Con los sistemas OSXLCAU21, OSX2DCASM, DDLAB y Golly.

Regla 54

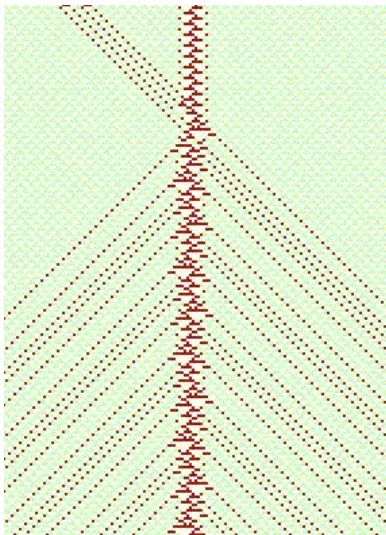


Figure: Generador de partículas por choques en Regla 54.

Regla 110

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

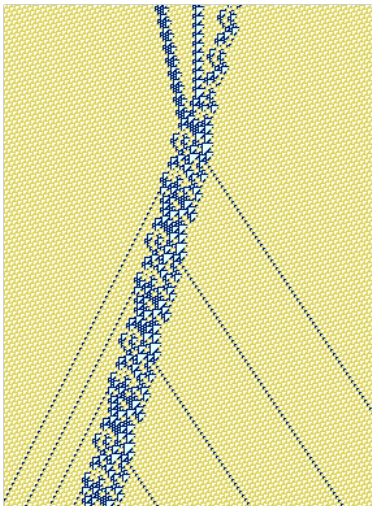


Figure: Generador de partículas por choques en Regla 110.

Regla Beehive

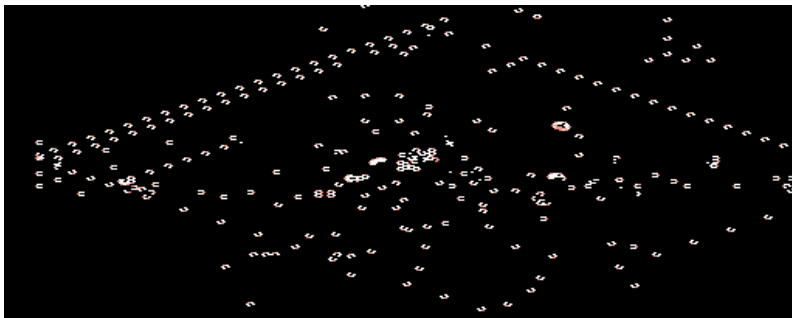


Figure: Localidades en la regla hexagonal Beehive.

Regla Spiral

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

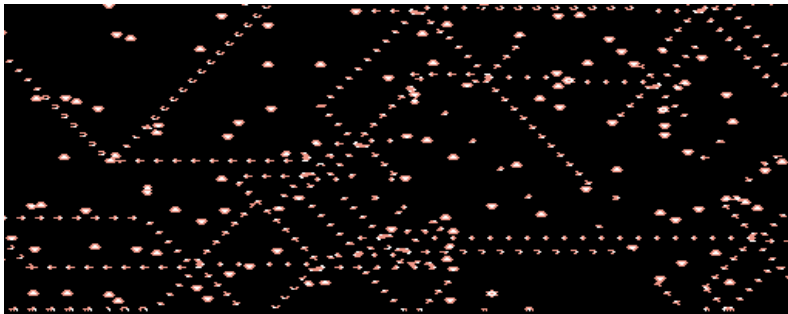


Figure: Localidades en la regla hexagonal Spiral.

Regla Diffusion

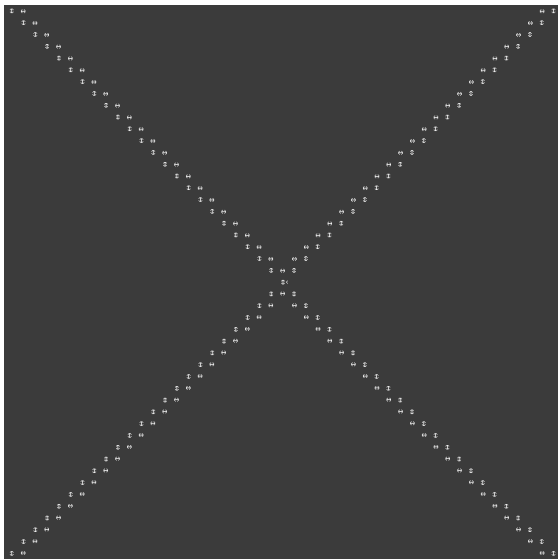


Figure: Localidades operando reflexiones múltiples utilizando compuertas lógicas FANOUT con la regla Diffusion.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC

AC complejo

Conclusiones

Referencias

Software libre en AC

Se presenta de manera general y básica la teoría de automata celular. Presentada para el grupo de investigación “iGEM México.” La idea central es el estudio de AC con comportamiento complejo, que nos permita simular procesos celulares en un camino convencional o no-convencional. Para ésto proponemos aplicar específicamente algunas reglas de evolución con dichas capacidades. Al momento se tiene construido un simple fenómeno de luminiscencia utilizando partículas estacionarias y la reflexión de partículas utilizando compuertas lógicas, ambas con el autómata celular complejo “Diffusion Rule.”

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

Referencias – Libros



Andrew Adamatzky
Collision-Based Computing
Springer-Verlag, 2002.



Harold V. McIntosh
One-Dimensional Cellular Automata
by publish, 2007.



William Poundstone
The Recursive Universe
William Morrow and Company, Inc., New York, 1985.



Tomasso Toffoli and Norman Margolus
Cellular Automata Machines
The MIT Press, Cambridge, Massachusetts, 1987.



Stephen Wolfram
A New Kind of Science
Wolfram Media, Inc., Champaign, Illinois, 2002.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC


AC complejo


Conclusiones


Referencias


Software libre en AC

Referencias – Artículos

 Andrew Adamatzky, Genaro J. Martínez, and Juan C. S. T. Mora
Phenomenology of reaction-diffusion binary-state cellular automata
Int. J. of Bifurcation and Chaos, 16/10:1–21, 2006.

 Andrew Adamatzky and Andrew Wuensche
Computing in ‘spiral rule’ reaction-diffusion cellular automaton
Complex Systems, in press, 2006.

 Martin Gardner
Mathematical Games - The fantastic combinations of John H. Conway’s new solitaire game Life
Scientific American, 223:120-123, 1970.

 Genaro J. Martínez, Andrew Adamatzky and Harold V. McIntosh
Phenomenology of glider collisions in cellular automaton Rule 54 and associated logical gates
Chaos, Fractals and Solitons, 28:100–111, 2006.

 Genaro J. Martínez, Andrew Adamatzky and Harold V. McIntosh

Localization dynamic in binary two-dimensional cellular automaton: Diffusion Rule
submitted to J. of Cellular Automata, September 2006.

 Genaro J. Martínez, Harold V. McIntosh and Juan C. S. T. Mora

Gliders in Rule 110
Int. J. of Unconventional Computing, 2/1:1–49, January 2006.

 Andrew Wuensche

Self-reproduction by glider collisions: the beehive rule
Alife9 Proceedings, pp. 286–291, MIT Press, 2004.

 Andrew Wuensche and Andrew Adamatzky

On spiral glider-guns in hexagonal cellular automata: activator-inhibitor paradigm
Int. J. Modern Phys. C 17, in press, 2006.

Estructura general

AC en una dimensión

AC en dos dimensión

AC en tres dimensión

El autómata de von Neumann

El autómata de Conway

Los autómatas de Wolfram

Morfología en AC
AC complejo

Conclusiones

Referencias

Software libre en AC

- ▶ Sistemas “OSXCA” :
<http://uncomp.uwe.ac.uk/genaro/OSXCASystems.html>
- ▶ Andrew Wuensche, sistema “Discrete Dynamics Lab” (DDLAB):
<http://www.ddlab.org/>
- ▶ Harold V. McIntosh, sistemas “NXLCAU” :
<http://cellular.ci.ulsal.mx/>
- ▶ Andrew Trevorrow y Tomas Rokicki, sistema “Golly” :
<http://golly.sourceforge.net/>