Enfoques de estudio en los Autómatas Celulares Lineales

Coordinador del verano de la Investigación:
Harold V. McIntosh
Departamento de Aplicación de Microcomputadoras
Instituto de Ciencias, Universidad Autónoma de Puebla
Email:mcintosh@servidor.unam.mx
Alumno:
José Manuel Gómez Soto
Departamento de Ingeniería Eléctrica, Sección de Computación
Centro de Estudios Avanzados del IPN
Email:jmgomez@alpha2.cs.cinvestav.mx

6 de Agosto de 1996

Resumen:

Un Autómata Celular (A.C.) es un mecanismo capaz de realizar una serie de acciones independientes. Cada parte del mecanismo puede decidir su propio accionar a partir de su entorno local. Aunque el accionar depende de una sola parte del todo, con el paso del tiempo es posible que exista comunicación entre todos los componentes: la información viaja a través del mecanismo. Este escrito discute las dos principales lineas de investigación que han surgido en el estudio de los Autómatas celulares: el enfoque "hacia adelante" y el enfoque "inverso". Para ilustrar los objetivos, métodos y perspectivas de estos dos unfoques, se ejemplifican mediante las Clases de Wolfram y la construcción de un contador binario, respectivamente. De manera adicional e implícita se contesta a la interrogante de ¿Cómo es que puede viajar la información a través de los autómatas celulares?

1 Introducción

Conforme se ha desarrollado la investigación en los A. C., han surgido en forma general dos grandes lineas de investigación sobre los mismos.

Aquí se hace énfasis en los objetivos y métodos de estudio de cada uno de ellos, mediante un ejemplo que representa la escencia de cada enfoque.

Los ejemplos elegidos son las Clases de Wolfram y la construcción de un contador binario, el primero representando al enfoque "hacia adelante" y el segundo a la línea de estudio conocida como enfoque "inverso".

La presentación que se hace se realiza en cinco partes: En la primera parte "Antecedentes" se discute el origen de los A.C. y su importancia en la actualidad.

En la segunda parte se hace una definición de los A.C. de una manera muy sencilla, sin involucrar ningún formalismo matemático.

En la tercer parte se explica en que consiste cada una de las dos lineas de Investigación de los A.C. y se mencionan ejemplos para cada una de ellas.

Finalmente en la cuarta y quinta partes se lleva a cabo la construcción de un contador binario con un A.C.

Todas las gráficas sobre evoluciones de autómatas celulares que se muestran, fueron generadas con NXLCAU[1].

[1]NXLCAU es un sistema que simula A.C. Lineales, hecho para la plataforma de NeXTSTEP.

2 Antecedentes

En 1956 John von Neumann jugaba con la idea de que una máquina pudiera autoreproducirse, el interés estaba encaminado en poder tener fábricas automáticas; sin embargo en aquella época no había (actualmente tampoco) tecnología para poder diseñar un modelo mecánico que cumpliera con dicho propósito. Una manera de solventar este obstáculo le fue sugerida por su amigo Stanislaw Ulam, la cual consististía en crear un modelo matemático para demostrar esto. El modelo matemático con el que Neumann demostró que la autoreproducción era un proceso totalmente mecánico se le conoce como Autómata Celular. En la actualidad los A.C. ocupan una posición medular en las Ciencias de la Complejidad. Una de las principales razones de esto, se debe a que los Autómatas Celulares son capaces de generar comportamiento complejo global a partir de la interacción local de múltiples de sus componentes individuales.

Este fenómeno ocurre en muchos campos y en diversos niveles de descripción; en la Física tal fenómeno se dá cuando interactúan las moléculas de agua para formar fluidos o en la Química la interacción de tres estados: quieto, activo y refractario en los sistemas de reacción-difusión; por citar sólo dos ejemplos [6]. En los A.C. los componentes individuales son entidades que tienen un número finito de estados. Dichos estados ocupan casillas (células) dentro de una malla regular. En matemáticas se piensa en estos estados de manera abstracta, etiquetándolos como 0,1,2,... Sin embargo en la aplicación del modelo se requiere asignarle significados más concretos tanto a las células como a sus estados. Por otro lado también el significado de la dimensión espacial en que se llevan a cabo las interacciones depende de la aplicación, y su importancia radica en limitar las interacciones entre los componentes individuales a una vecindad. Tal como el espacio el tiempo también es discreto. En cada unidad de tiempo las células individuales cambian a sus estados de acuerdo a una regla que determina el nuevo estado como una función de los estados previos de las células de la vecindad.

Una de las razones por las que el A.C. se ha convertido en uno de los modelos preferidos por los investigadores de los fenómenos complejos, radica en su sencillez, pues esto permite realizar análisis con un alto rigor matemático. Pero no sólo esto, gracias a que es un modelo que cuenta con espacio, tiempo y estados discretos y que realiza una actualización en paralelo de todas las células de su espacio bajo la misma regla de evolución, nos encontramos ante un modelo apropiado de Computación en Paralelo. Entonces el A.C. además de permitir hacer prediciones de los procesos naturales en términos computacionales, permite estudiar la computación en términos Biológicos y Físicos.

3 Autómata Celular Lineal

Un Autómata Celular Lineal es un mecanismo capaz de realizar una serie de acciones independientes. Cada parte del mecanismo puede decidir su propio accionar a partir de su entorno local.

Este mecanismo consiste en una serie de celdas (células) conectadas en linea, como lo muestra la figura 1.

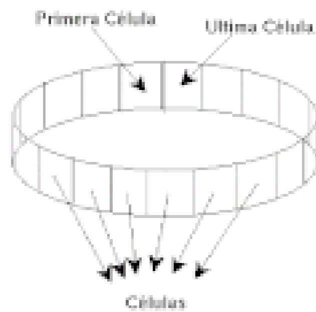

figura 1: Espacio de un A.C.

La línea de células puede considerarse como cerrada por la frontera, donde la última célula se conecta con la primera, dando la forma de un anillo.

Cada parte del mecanismo puede tomar diferentes valores, tales valores pueden representar una acción o una característica en especial, para fines de generalidad estos se representan con los primeros enteros: **0,1,2,3..**

Cada una de las células en toda la línea cambia su estado a través del tiempo. Dicho cambio depende de su entorno local, es decir, de los valores que tenga la célula en cuestión y los vecinos más cercanos que hayan sido convenidos para formar el entorno o vecindad (fig. 2).

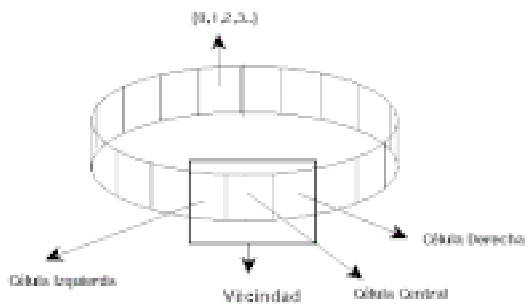

figura 2: Vecindades y estados de un A.C.

Las configuraciones que van resultando del cambio de valores de las células en toda la línea a través del tiempo se les conoce como generaciones y al conjunto de todas estas generaciones es llamado la evolución de un A.C. (fig. 3).

Suponga que se tiene una vecindad de tres elementos, es decir un entorno local donde la interacción de tres elementos decide el valor de las células, en la siguiente generación. Y suponga que los valores que puede tener cada célula son dos:0,1. Esto dá un total de 8 posibles vecindades que se pueden presentar con la combinación de estos estados en las tres células.

Si a cada vecindad se le asigna el valor resultado de la interación de los elementos que lo componen se tiene lo que se denomina como la regla de evolución del A.C..

$$\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\hline
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
 & & & & & & & \\
2^0 & 2^1 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 & 2^7 \; \rightarrow 22
\end{array} \tag{1}$$

Cada regla puede tener un nombre con la nomenclatura de Wolfram que consiste en tomar el número de la regla como binario haciendo la lectura de izquierda derecha y calcularle el decimal correspondiente. Entonces el número binario de nuestra regla es 0110100 y su decimal 22.

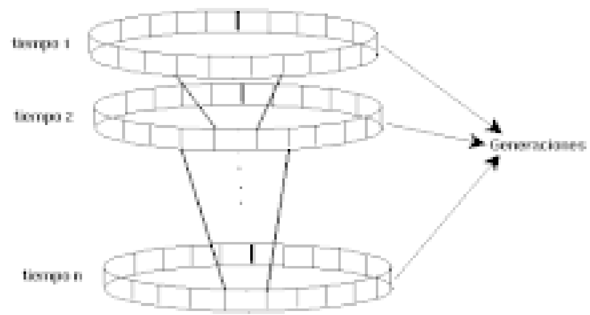

figura 3: Evolución de un A.C.

Para ver la evolución de un A.C. se necesitan dos cosas:

- La configuración inicial a partir de la cual evolucionará y
- La regla de evolución

Un A.C. es un modelo que cambia con respecto al tiempo de manera discreta, por lo que se le reconoce como un sistema dinámico discreto.

De manera general se pueden reconocer las siguientes características de un A.C.:

- El espacio en que se desarrolla es finito.
- El tiempo en que se dan las generaciones que conforman la evolución es discreto.
- El valor de los estados que puede tener una célula es finito.
- La misma regla de evolución se aplica para todo el espacio y lo hace en entornos locales.
- La regla de evolución es determinística.

Una vez determinado en número de estados y el tamaño de la vecindad, el siguiente paso es estudiar la evolución del A.C..

La evolución representa en primer término una descripción visual del comportamiento del A.C.

Sin embargo ver la evolución de un A.C. no es suficiente para poder determinar de una manera rigurosa las características de su comportamiento. Existe mucha ambigüedad en la apreciación de patrones: las semejanzas son tan aparentes como las diferencias.

Debido a lo anterior existe un esfuerzo por caracterizar a un A.C. por medio de herramienta matemática que ayude a determinar con mayor exactitud su comportamiento. Este esfuerzo se ha encaminado sobre dos grandes lineas de investigación las cuales se describen en la siguiente sección.

4 Enfoques de estudio sobre Autómatas Celulares

Según Gutowitz[3] se pueden describir en términos generales dos grandes lineas de investigación en Autómatas Celulares:

- El enfoque "hacia adelante"

- El enfoque ”inverso”

4.1 El enfoque ”hacia adelante”

El enfoque ”hacia adelante” trata de determinar las propiedades características de un A.C. Es decir, dada la regla de un A.C. determinar sus propiedades. Ejemplos de estudios del primer enfoque son:

- Determinación de ciclos o puntos fijos.
- Análisis estadísticos de los patrones generados en un conjunto de evoluciones (tiempo-espacio).
- La creación de herramientas y técnicas matemáticas en el estudio de A.C.

4.2 El enfoque ”inverso”

El enfoque ”inverso” trata de describir algunas propiedades y encontrar una regla o un conjunto de reglas que cumplan con dichas propiedades.

Ejemplos de estudios en el segundo enfoque:

- Construcción del Autómata de reproducción por John von Neumann.
- Creación del Juego de Vida por Conway.
- El autómata de Chaté y Manneville.

A continuación se ejemplificará cada uno de estos enfoques, de manera que podremos ver de una forma más precisa cual es la visión de estudio de estas dos lineas. El enfoque ”hacia adelante” se muestra mediante la clasificación de Wolfram y el enfoque ”inverso” con la construcción de un contador binario.

5 Clases de Wolfram

Una vez que se pueden ver las evoluciones de los diferentes tipos de autómatas celulares, la pregunta es: ¿Qué comportamiento representa la evolución? ¿Existe una manera de caracterizarlos? ¿La evolución depende de la configuracion inicial? ¿Qué tipo de autómatas celulares existen y cuales son sus caracteríticas? ¿Existe una forma general de describirlos? El tratar de dar respuesta a estas interrogantes motiva a los investigadores de esta línea de estudio, a buscar sobre métodos que auxilien a contestarlas.

Stephen Wolfram fue de los primeros investigadores que comenzaron a hacer estudios más sistemáticos sobre los A.C., concretamente fue quien por primera vez estudió todos los Autómatas Celulares Lineales (2,1)[2] : que daba un total de 256 reglas.

Como una parte de este esfuerzo e inspirado probablemente en el trabajo de Stephen Smale, Wolfram relacionó las clasificaciones hechas en Sistemas Dinámicos e hizo una analogía entre esta clasificación y los A.C. El resultado de esto fueron las conocidas cuatro clases de Wolfram[8].

Las clases de Wolfram consisten en lo siguiente:

5.1 Clase I: Autómatas Celulares que evolucionan a un sólo estado estable.

Como se puede ver, a partir de una configuración inicial aleatoria la regla hace que todo evolucione en el estado cero (rojo). Esta evolución se considera que es de orden o quietud.

Este autómata tiende a representar a todos aquellos sistemas que tienden a estabilizarse en un sólo punto. La analogía que hace Wolfram es con los Sistemas Dinámicos que evolucionan en un punto fijo.

[2]Se dice que un autómata es **(k,r)** , donde **k** es el número de estados y **r** es el radio de vecindad; (2,1) significa un A.C. de **2** estados y un vecino a la derecha e izquierda de la célula central: una vecindad de **3** elementos.

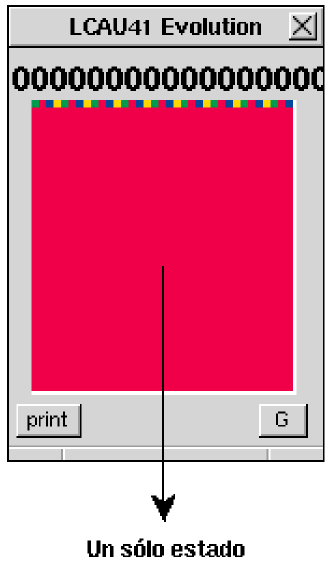

figura 4: Clase I

5.2 Clase II: Autómatas Celulares que evolucionan en ciclos de estados aislados.

La clase II tiene como característica llenar el espacio con una serie de ciclos (repetición de un estado o color) aislados.

A esta clase se le asocia con los sistemas dinámicos que evolucionan en ciclos límite que contienen escencialmente configuraciones periódicas.

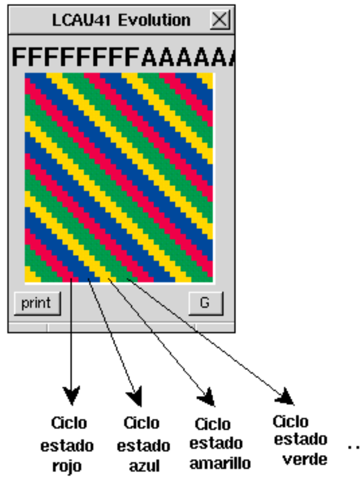

figura 5: Clase II

5.3 Clase III: Autómatas Celulares que evolucionan en una distribución caótica de estados

En esta clase la distribución de los estados sobre el espacio se dá de una manera muy aleatoria (existe mucha diversidad en la secuencia de estados). A esta clase se le conoce como caótica o compleja.

A esta clase se le asocia a los sistemas dinámicos que caen en ciclos límites aperiódicos, que contienen analogías caóticas o atractores extraños.

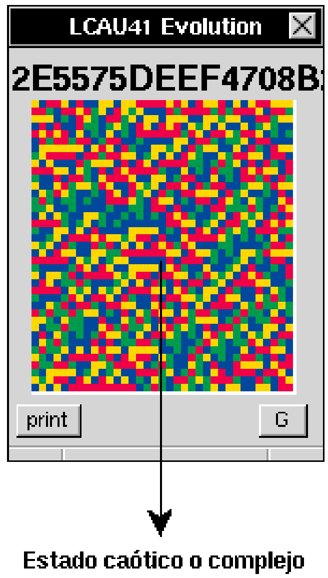

figura 6: Clase III

5.4 Clase IV: Autómatas Celulares que evolucionan en estados aislados complejos.

La clase IV resulta ser una combinación entre la clase II y la Clase III, el llenado del espacio esta dado por secuencias caóticas (complejas) y secuencias de orden.

Como se puede ver en la evolución los segmentos complejos se dán separados de segmentos de orden. (secuencias de regiones de orden y caos).

Las clases de Wolfram son el ejemplo clásico sobre el intento de dada una regla tratar de caracterizarla ubicándola en una de estas categorías. En otros términos: teniendo una regla poder determinar de alguna manera el comportamiento de ésta.

Sin embargo mediante este esquema no hay forma de determinar a que clase pertenece una regla, ó cual es la real diferencia entre una clase y la otra, en síntesis: la clasificación de Wolfram es indecidible [1].

Debido a esto se ha presentado la necesidad de encontrar herramien-

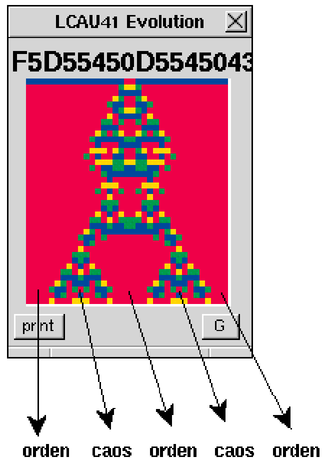

figura 7: Clase IV

tas matemáticas que permitan conocer con más precisión el comportamiento de los A.C.

Por nombrar sólo algunos podemos mencionar a:

- Diagramas de "de Bruijn"
- Diagramas de Ciclos
- Bases de atracción
- Parámetro lambda de Langton

6 Contador binario

El enfoque "inverso" es el que históricamente dió origen a los Autómatas celulares, von Neumann estaba buscando un modelo para un fin en especial y así fue como surgió el A.C., otro de los autómatas celulares célebres que fue creado bajo esta línea de estudio es el juego de la vida de Conway[2].

El ejemplo que se muestra sobre esta línea es la construcción de un A.C. para un fin muy específico: Un contador binario.

Como hacer que la información viaje a través de un A.C., a pesar de que la localidad de las interacciones es parte de lo que se va a ilustrar la construcción del contador binario.

De hecho se puede ver no solamente como la información viaja, sino como se construye toda una maquinaria que interactúa: la expansividad, la existencia de "gliders" (configuraciones que viajan), "still life" (configuraciones que se conservan durante la evolución) son ejemplos de todas las características que estan directamente relacionadas con la regla de transacción.

Una especificación más detallada se encuentra en[4], aquí se muestra una de las maneras de construirlo, y se recrea aprovechando las facilidades que NXLCAU ofrece.

Un contador binario es un incremento de uno sobre una base de dos números: **0** y **1** . Si uno incrementa en **1** el valor de **0** este se convertirá en **1** , y no habrá acarreo. Si por otro lado se incrementa el **1** en **1** el valor será **10** , es decir **0** y se lleva **1** en el acarreo.

Entonces para construir un contador binario mediante un A.C. se deben hacer las siguientes consideraciones:

- Se necesita un estado que sirva de fondo para llevar a cabo las operaciones (algo que haga las veces de un "pizarrón".)

- Un estado que represente el incremento en **1** y que tenga la facultad de viajar por el espacio para que interactúe con **0** y **1** . Este es el uno que incrementa en **1** a los valores **0** y **1** .

- dos estados que representen a los valores de **0** y **1** respectivamente. Estos estados son bi-estables (estables en 2 valores) para que representen el cambio en el momento de recibir el incremento, si es el valor **0** este cambia a **1** y si es el valor de **1** este cambia a **0** .

Son en total cuatro estados los que se necesitan para este tipo de contador, por lo que usaremos el A.C. (4,1).

Una característica más que tendrá nuestro contador es un lanzador de incrementos, que envia constantemente el incremento a **1** para que interactúen sobre los estados **0** y **1** .

Consideremos también la siguiente tabla de equivalencia entre estados y colores:

estado	*color*
0	*rojo*
1	*azul*
2	*amarillo*
3	*verde*

(2)

A continuación comenzamos a construir el contador binario:

6.1 Fondo quieto

Generación de un fondo quieto representado por el estado rojo.

$$\varphi(rojo, rojo, rojo) = rojo \tag{3}$$

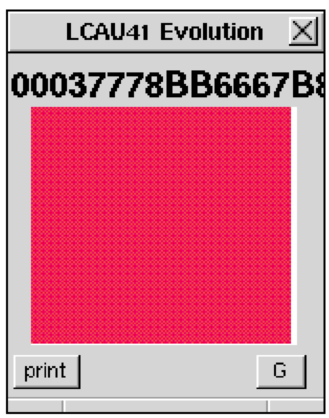

figura 8: Fondo quieto

todas las vecindades con tres elementos en rojo dán rojo. Esto genera un fondo rojo. Sobre este fondo se llevarán a cabo las operaciones.

6.2 Estado que representa al valor de ”cero”

Creación de uno de los valores: el ”cero”. Este se caracteriza por mantenerse estable sobre un medio rojo.

$$\begin{aligned} \varphi(azul, rojo, rojo) &= rojo \\ \varphi(rojo, azul, rojo) &= azul \\ \varphi(rojo, rojo, azul) &= rojo \end{aligned} \tag{4}$$

para mantener estable al valor de azul sobre un espacio rojo, se deben tomar en cuenta todas las vecindades que se tienen, es decir, el que haya un azul sobre el espacio rojo implica todo rojo [vecindades de tres elementos en rojo], [rojo, rojo, y azul], [rojo, azul, rojo], [azul, rojo, rojo]. La primer opción ya fue definida con el fondo estable. El valor que toman las siguientes vecindades permiten conservar para las siguientes generaciones el color azul en la misma célula cubierta de un espacio rojo.

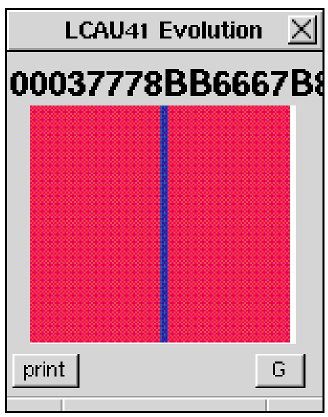

figura 9: Se genera un azul abajo de la célula con color azul y en los extremos se mantenga el color rojo del fondo.

6.3 Incremento de 1

El estado amarillo representa el incremento a **1** (acarreo). El incremento viaja por el espacio y de esta manera incrementa a los estados **0** y **1**. Este estado viaja hacia la izquierda y durante su viaje interactúa con otros componentes del mecanismo (con los que se encuentra en su trayectoria), en este caso con los estados que representan a los valores de **0** y **1**

$$\begin{aligned} \varphi(amarillo, rojo, rojo) &= rojo \\ \varphi(rojo, amarillo, rojo) &= rojo \\ \varphi(rojo, rojo, amarillo) &= amarillo \end{aligned} \tag{5}$$

Esta parte de la regla especifica el valor que deben de tomar las vecindades que representan a un amarillo sobre un fondo rojo. Cuando esto pase hay que definir que en la siguiente generación el amarillo se translade una célula a la izquierda con respecto a la célula en que se encontraba, con el paso de las evoluciones se tiene a un estado que viaja diagonalmente una célula a la izquierda por generación. Este "artefacto" en conocido en la jerga de A.C. como "glider".

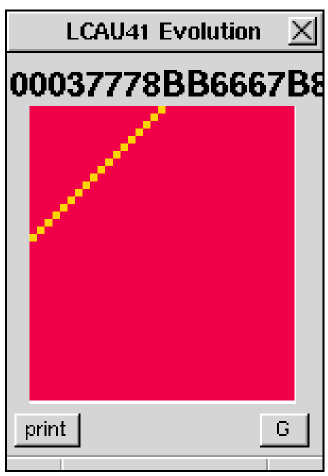

figura 10: Estado que representa el incremento. Viaja diagonalmente

6.4 Estado que representa al valor de "uno"

Creación de uno de los valores: el "uno". Este se caracteriza por mantenerse estable sobre un medio rojo.

$$\begin{aligned} \varphi(verde, rojo, rojo) &= rojo \\ \varphi(rojo, verde, rojo) &= verde \\ \varphi(rojo, rojo, verde) &= rojo \end{aligned} \tag{6}$$

para mantener estable al valor de verde sobre un espacio rojo, se deben tomar en cuenta todas las vecindades que se tienen, es decir, el que haya un verde sobre el espacio rojo implica, [rojo, rojo, y verde], [rojo, verde, rojo], [verde, rojo, rojo]. El valor que toman estas vecindades permiten conservar para las siguientes generaciones el color verde en la misma célula rodeada de un espacio rojo.

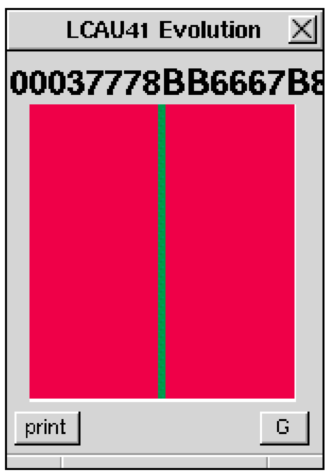

figura 11: Se genera un azul abajo de la célula con color verde y en los extremos se mantenga el color rojo del fondo.

Hasta aquí tenemos definido los elementos básicos del contador binario: los dos valores **0** y **1**, el incremento en **1** y el espacio en que operan. Sin embargo falta considerar que se quiere que suceda cuando interaccione el incremento con cualquiera de los dos valores, el espacio mínimo que se tenga entre los valores, así como otras posibilidades que genera el sistema a través de su evolución.

6.5 Incremento en 1 al valor de 0

Cuando el valor de **0** se incrementa en **1**, este toma el valor de **1** y por lo tanto el acarreo no lleva nada. En términos de A.C. esto se define mediante la interacción entre el incremento (amarillo) y el azul (**0**) donde el azul (**0**) cambia al estado verde **1** (**0+1=1**), es decir que el **0** se incrementó en **1** y como no se "lleva nada" el incremento deja de "viajar" (este es absorbido). Los mapeos en las vecindades que hay que definir para lograr esto son los siguientes:

$$\begin{gathered} \varphi(azul, rojo, amarillo) = amarillo \\ \varphi(rojo, azul, amarillo) = verde \\ \varphi(azul, amarillo, rojo) = rojo \end{gathered} \tag{7}$$

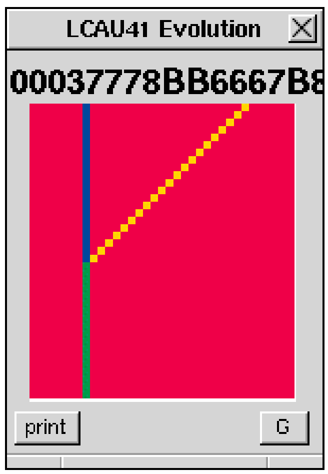

figura 12: Interactúa el color amarillo con el azul, el azul se cambia a verde y el amarillo cesa de viajar. Incrementa **1** al valor de **0**; **0** cambia a **1** y no hay acarreo.

6.6 Incremento en 1 al valor de 1

Cuando el valor de **1** se incrementa en **1**, este toma el valor de **10**, es decir **0** y se lleva **1** en el acarreo. En términos de A.C. esto se define mediante la interacción entre el incremento (amarillo) y el verde (**1**) donde el verde (**1**) se incrementa en el estado azul (**0**) dado que (**1+1=10**), es decir que el **1** se incrementó en **10** por lo cual se deja **0** llevándose a **1** en el acarreo (el incremento sigue "viajando"). Los mapeos en las vecindades que hay que definir para lograr esto son los siguientes:

$$\begin{aligned}
\varphi(verde, rojo, amarillo) &= amarillo \\
\varphi(rojo, verde, amarillo) &= amarillo \\
\varphi(verde, amarillo, rojo) &= verde \\
\varphi(rojo, amarillo, azul) &= rojo \\
\varphi(amarillo, rojo, azul) &= rojo \\
\varphi(rojo, amarillo, verde) &= azul \\
\varphi(amarillo, verde, rojo) &= rojo \\
\varphi(amarillo, azul, rojo) &= azul
\end{aligned} \tag{8}$$

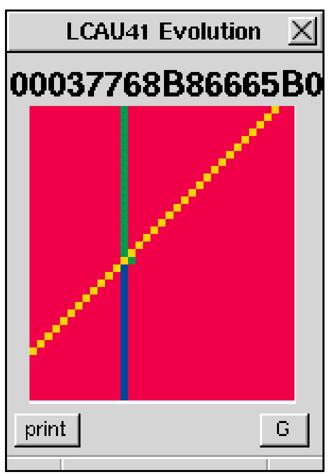

figura 13: Interactúa el color amarillo con el verde, el verde se cambia a azul y el amarillo continua viajando. Incrementa **1** al valor de **1**; **1** cambia a **0** y hay acarreo.

Ahora hay que hacer la consideración para cuando el incremento interactúa con los valores de **0** y **1** separados por un espacio. Esta observación se divide en dos situaciones: cuando estos valores son iguales o son diferentes:

6.7 Incremento en 1 sobre los valores 1 y 1 separados por un espacio

Esta situación debe cambiar a ambos valores en "0" y el incremento sigue viajando. El incremento llega al primer valor de **1** lo cambia en **0** y el incremento sigue para después toparse con otro valor de **1** que a su vez también cambiará a **0**. El hecho de que los dos valores esten separados uńicamente por un espacio involucra a otras vecindades. Los mapeos de estas nuevas vecindades deben de ser definidos de tal manera que se pueda obtener lo que se describió anterioremente. Las nuevas vecindades y sus respectivos mapeos son los siguientes:

$$\begin{aligned} \varphi(verde, rojo, verde) &= rojo \\ \varphi(verde, amarillo, azul) &= verde \\ \varphi(amarillo, verde, azul) &= rojo \\ \varphi(verde, azul, rojo) &= azul \\ \varphi(amarillo, rojo, azul) &= rojo \\ \varphi(azul, rojo, azul) &= rojo \end{aligned} \tag{9}$$

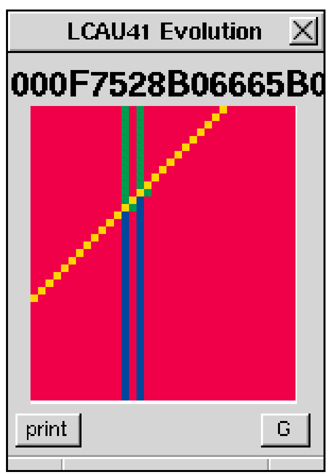

figura 14: incremento en dos valores de **1** separados por un espacio.

6.8 Incremento en 1 sobre los valores 0 y 1 separados por un espacio

Esta situación debe cambiar a ambos valores, el primero a **0** y el segundo a **1**, despues de la primer interacción el incremento continua, y es el que incrementa después al valor de **0**, en la segunda interacción el incremento deja de viajar. El hecho de que los dos valores esten separados únicamente por un espacio involucra a otras vecindades. Los mapeos de estas nuevas vecindades deben de ser definidos de tal manera que se pueda obtener lo que se describió anterioremente. Las nuevas vecindades y sus respectivos mapeos son los siguientes:

$$\begin{array}{c} \varphi(azul, rojo, verde) = rojo \\ \varphi(azul, amarillo, azul) = rojo \\ \varphi(verde, rojo, azul) = rojo \end{array} \tag{10}$$

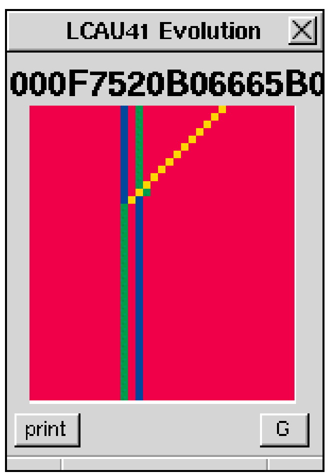

figura 15: incremento de los valores de **0** y **1** separados por un espacio

La opción de los valores **1** y **0** no se considera dado que con la primera interacción, desaparece el incremento y por lo tanto nunca hay contacto con el segundo valor. Y el que haya un incremento del valor **0** en **1** ya está contemplado.

Hasta ahora han sido contempladas 29 transacciones de mapeos en vecindades locales, mismas que son suficientes para simular un contador binario. Pero aún falta por construir el generador de incrementos:

un mecanismo que genera periódicamente incrementos que viajan e interactúan con los valores de **0** y **1**. En cuanto al generador de los valores es necesario solamente colocar en la configuración inicial a los estados que representan a los valores de manera aislada.

6.9 Mecanismo generador de incrementos periódicos

El generador de incrementos se puede definir mediante 8 transacciones de las vecindades. El generador está formado por dos columnas de estados que evolucionan de tal manera que cada 4 generaciones lanza un incremento.

Las vecindades y transacciones que forman este mecanismo son las siguientes:

$$\begin{gathered}
\varphi(rojo, amarillo, amarillo) = azul \\
\varphi(amarillo, amarillo, rojo) = azul \\
\varphi(amarillo, azul, azul) = azul \\
\varphi(azul, azul, rojo) = verde \\
\varphi(rojo, azul, verde) = verde \\
\varphi(azul, verde, rojo) = verde \\
\varphi(rojo, verde, verde) = amarillo \\
\varphi(verde, verde, rojo) = amarillo
\end{gathered} \tag{11}$$

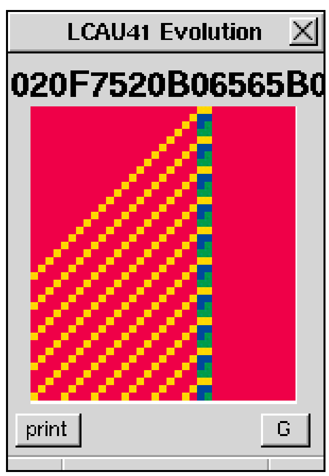

figura 16: Generador de incrementos, uno cada cuatro generaciones

6.10 Protección del mecanismo generador de incrementos

Dado que el espacio esta cerrado en forma de anillo, hay que considerar que puede haber incrementos que choquen por la derecha con la estructura generadora de incrementos. Las protecciones radican en preveer que aunque las colisiones se realicen no destruyan al generador.

Para proteger al mecanismo generador hay que considerar que la colisión puede pasar en cualquier que parte de la estructura del generador. La estructura consiste en una serie de pares de estados periódicos:

$$\begin{gathered} secuenciadeestados: \\ 2y2 \\ 1y1 \\ 1y3 \\ 3y3 \end{gathered} \tag{12}$$

De ahi que las protecciones sean en la fase 22, fase 11, fase 13 y fase 33.

En seguida se mostrarán las posibilidades en que pueden darse estas colisiones y las transacciones necesarias para evitar que estas dañen a la estructura que genera los incrementos.

Protección de la fase 22.

La protección de la fase 22 consiste en evitar que se destruya el mecanismo generador de incrementos cuando reciba el "impacto" de un incremento por la derecha y el colapso se dé en la parte del generador que esta formado por la secuencia de estados 2 y 2.

Las transacciones que se encargan de absorber el "impacto" y mantener la estructura del generador de incrementos son los siguientes:

$$\begin{gathered}\varphi(amarillo, amarillo, amarillo) = azul \\ \varphi(azul, azul, azul) = amarillo \\ \varphi(azul, amarillo, verde) = verde\end{gathered} \tag{13}$$

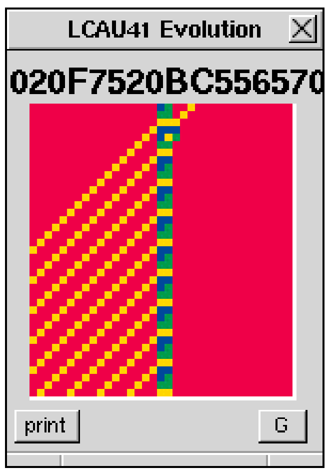

figura 17: protección de la fase 22

Protección de la fase 13.

La protección de la fase 13 consiste en evitar que se destruya el mecanismo generador de incrementos cuando reciba el "impacto" de un incremento por la derecha y el colapso se dé en la parte del generador que esta formado por la secuencia de estados 1 y 3.

Las transacciones que se encargan de absorber el "impacto" y mantener la estructura del generador de incrementos son los siguientes:

$$\begin{gathered}\varphi(azul, verde, amarillo) = verde \\ \varphi(verde, verde, verde) = amarillo\end{gathered} \tag{14}$$

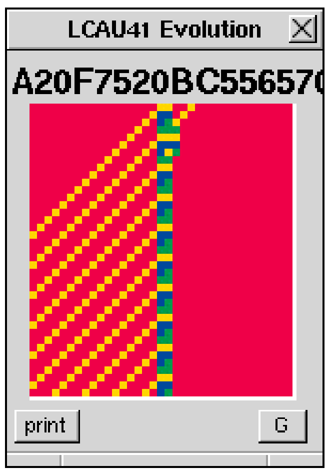

figura 18: protección de la fase 13

Protección de la fase 33.

La protección de la fase 33 consiste en evitar que se destruya el mecanismo generador de incrementos cuando reciba el ”impacto” de un incremento por la derecha y el colapso se dé en la parte del generador que esta formado por la secuencia de estados 3 y 3.

Las transacciones que se encargan de absorber el ”impacto” y mantener la estructura del generador de incrementos son los siguientes:

$$\begin{gathered}\varphi(verde, verde, amarillo) = amarillo \\ \varphi(amarillo, amarillo, verde) = azul\end{gathered} \tag{15}$$

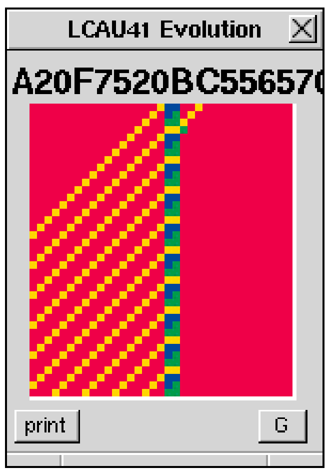

figura 19: protección de la fase 33

Protección de la fase 11.

La protección de la fase 11 consiste en evitar que se destruya el mecanismo generador de incrementos cuando reciba el ”impacto” de un incremento por la derecha y el colapso se dé en la parte del generador que esta formado por la secuencia de estados 1 y 1

Las transacciones que se encargan de absorber el ”impacto” y mantener la estructura del generador de incrementos son los siguientes:

$$\begin{gathered}\varphi(amarillo, azul, amarillo) = amarillo \\ \varphi(azul, azul, amarillo) = verde\end{gathered} \tag{16}$$

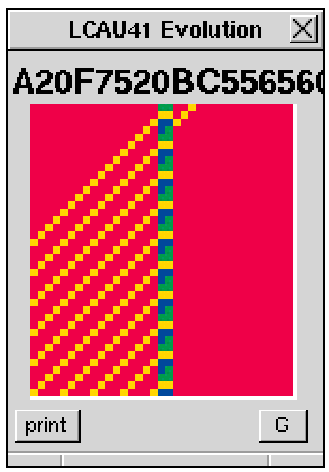

figura 20: protección de la fase 11

Dieciocho transacciones quedan por ser definidas, estas transacciones pueden usarse para hacer más complicado el mecanismo, pero tal complicación no es de utilidad para el contador binario.

protecciones adicionales

$$
\begin{gathered}
\varphi(amarillo, verde, amarillo) = azul \\
\varphi(azul, azul, verde) = amarillo \\
\varphi(azul, amarillo, amarillo) = verde \\
\varphi(verde, verde, azul) = amarillo \\
\varphi(rojo, azul, azul) = azul \\
\varphi(verde, azul, azul) = amarillo \\
\varphi(verde, amarillo, amarillo) = rojo \\
\varphi(verde, azul, verde) = verde \\
\varphi(amarillo, rojo, verde) = rojo \\
\varphi(amarillo, verde, verde) = amarillo
\end{gathered}
\tag{17}
$$

Estas protecciones adicionales no son particularmente importantes para el contador binario de manera que la discusión sobre los detalles de su definición no se tratan.

La tabla total de transacciones con que se construye el contador binario se puede apreciar con el siguiente panel de la regla del sistema NXLCAU41.

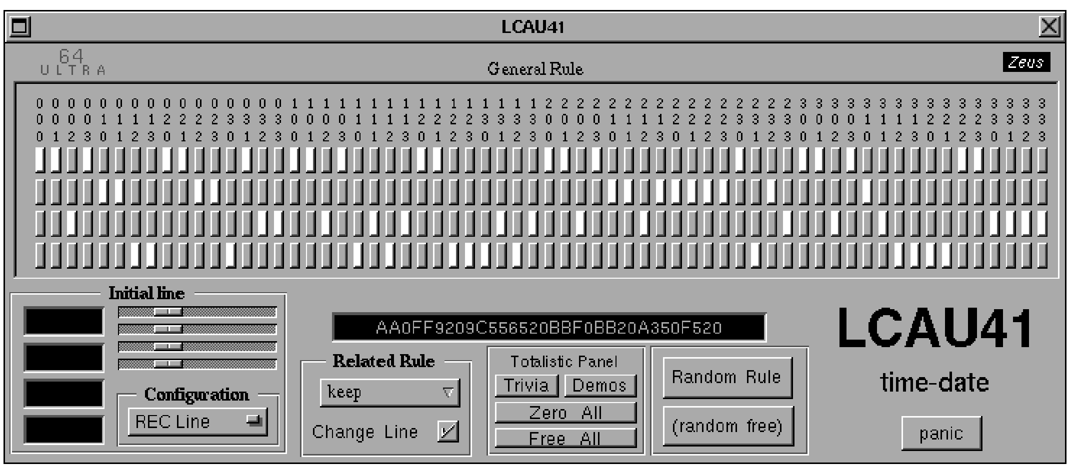

figura 21: Vecindades y mapeos del Contador Binario

Finalmente tenemos el contador binario y la siguiente evolución muestra su funcionamiento.

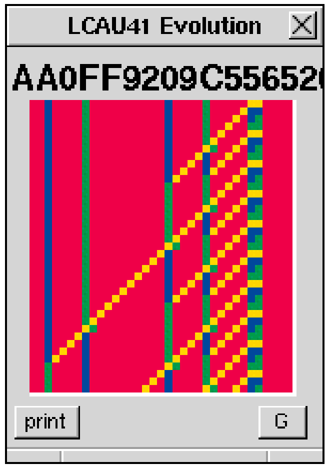

figura 22: Contador Binario: Lanzador de incrementos que afectan a un par de valores **0** y **1**

En esta evolución se parte de una configuración inicial que dá origen a cuatro valores: **0**, **1**, **0**, **1**, separados por espacios, seguidos de un generador de incrementos que lanza incrementos en periodos de 4 generaciones.

El primer incremento que lanza interactúa en primera instancia con el valor de **1** que es el valor más cercano al generador por su izquierda. El resultado de esta interacción cambia el valor de **1** (verde) a **0** (azul) y dado que hay acarreo el incremento sigue viajando hasta llegar al valor de **0** (azul) al cual incrementa cambiandolo a **1** (verde) y el incremento es absorbido dado que no hay acarreo.
Como resultado de esta interacción entre el incremento y los valores **0,1,0,1**, deja los valores como **0,1,1,0**.
El segundo incremento que se lanza con este nuevo resultado y como el primer elemento con el que interactúa es el **0**, lo incrementa en 1 y deja de "viajar", dejando como resultado a **0,1,1,1**.
El tercer incremento interactúa con los valores **0,1,1,1** por lo que incrementa al primero en **1** de la derecha en **0** y sigue "viajando" (lleva acarreo), incrementa el segundo **1** en **0** y continua el "viaje". Incrementa el tercer **1** en **0** y sigue llevando el acarreo. El último valor que

se topa es el **0**, lo incrementa en **1**; no lleva acarreo y por lo tanto es absorbido: deja de "viajar". Un análisis muy similar a los anteriores se puede realizar para los siguientes incrementos lanzados.
La siguiente evolución muestra como trabajan las protecciones del generador de incrementos cuando este recibe los "impactos" de incrementos que el mismo generó. Estos incrementos después de interactuar con los valores ubicados a la izquierda del generador siguieron manteniendo el acarreo (no fueron absorbidos), por lo que siguieron viajando por todo el anillo hasta que finalmente llegaron a encontrarse con el generador incrementos por la derecha.

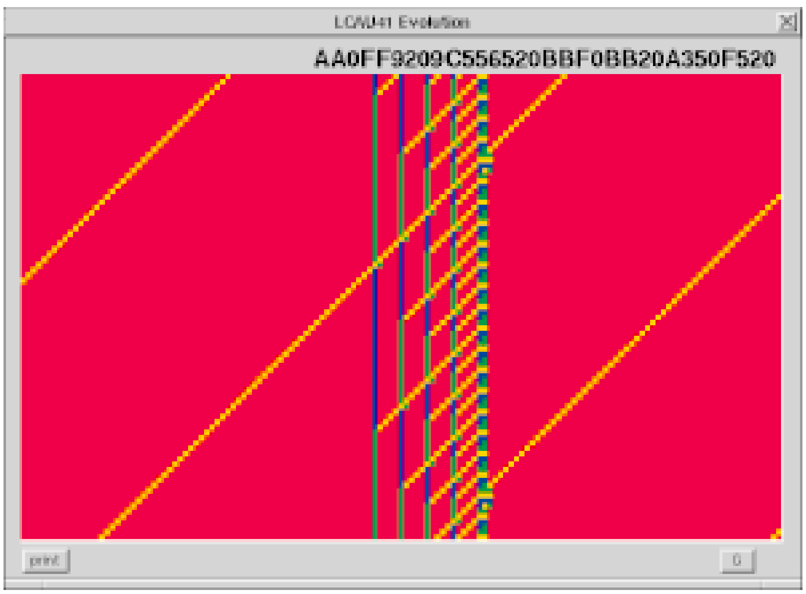

figura 23: Contador Binario y los "impactos" de sus incrementos por la derecha

7 Conclusiones

Siempre es difícil hacer una generalización sobre algo, debido a que no siempre es posible abarcar todas las peculariedades del fenómeno. La división propuesta por Gutowitz adolece del mismo problema: es difícil hacer una real separación entre estos enfoques debido a que estan estrechamente relacionados. Al tratar de llevar a cabo un enfoque siempre de alguna u otra manera está involucrado el otro.
Sin embargo para fines didácticos esta división es muy útil, ya que crea una perspectiva de hacia donde van y como se estan realizando los estudios y avances en el área de autómatas celulares.
Por otro lado, un elemento que brinda gran utilidad es el contar con el sistema NXLCAU mediante el cual se puede directamente experimentar e ilustrar los conceptos que se estan tratando. Este documento nació por estar "jugando" e interaccionado con tal sistema y aqui se puede apreciar otro punto de vista interesante: como se puede formar una idea al "jugar" con NXLCAU y por otra parte como se puede indagar sobre una idea preconcebida mediante este sistema de simulación de autómatas celulares.

Referencias

[1] Karel Culik II, Sheng Yu, *Undecidability of CA Clasification Schemes*, Complex Systems Vol. 2, April, 1988.

[2] Martin Gardner, *Wheels, Life and other mathematical amusements* W.H. Freeman and Company, New York, 1983.

[3] Howard Gutowitz, *Cellular Automata Theory and Experiment*, MIT/North-Holland, 1991.

[4] Harold V. McIntosh, *Automata Designed to Order*, Departamento de Microcomputadoras, Instituto de Ciencias, Universidad Autónoma de Puebla, April 19 , 1988.

[5] Tommaso Toffoli, Norman Margolus, *Cellular Automata Machine*, Cambridge, Massachusetts, 1991.

[6] Burton H. Voohees, *Computational Analisys of one-dimensional Cellular Automata*, World Scientific, 1996.

[7] John von Neumann, edited and completed by A.W. Burk *Theory of Self-Reproducting Automata*, University of Illinois Press, 1966.

[8] Stephen Wolfram, *Cellular Automata as models of complexity*, Nature Vol. 311, October, 1984.