

Introducción al Estudio de los Autómatas Celulares Lineales Reversibles

Investigador:

Dr. Harold V. McIntosh

e-mail:mcintosh@servidor.dgsca.unam.mx

Departamento de Aplicación de Microcomputadoras
Instituto de Ciencias, Universidad Autónoma de Puebla
Puebla, Puebla

Alumnos:

Juárez Martínez Genaro

e-mail:genaro@sparcomp.cs.cinvestav.mx

Seck Tuoh Mora Juan Carlos

e-mail:seck@delta.cs.cinvestav.mx

Lic. Matemáticas Aplicadas y Computación
Universidad Nacional Autónoma de México
Campus Acatlán

”Por mi raza hablará el espíritu”

Agosto 16, 1996

Resumen

Los Autómatas Celulares Lineales (ACL) son un arreglo de elementos (denominados células) que toman valores de un conjunto finito de estados individuales; tales elementos evolucionan a través de un tiempo discreto debido a una condición dada tomando valores del mismo conjunto finito de estados. Esta teoría matemática ha encontrado aplicaciones en varios campos de estudio tales como: criptología, simulación de epidemias, simulación de reacciones químicas, computación en paralelo, entre otras. Existen ACL en los cuales su condición de evolución es invertible, es decir, estos pueden evolucionar tanto adelante como hacia atrás a través del tiempo, a tales ACL se les denomina *Reversibles*. La representación gráfica y matricial utilizadas en los ACL son un poderoso medio para el estudio de las características de los ACL-Reversibles.

Introducción

El presente reporte muestra una introducción sobre el estudio de los *Autómatas Celulares Reversibles*, haciendo hincapié en diferentes herramientas básicas (diagrama de de Bruijn, diagrama de Subconjuntos y el diagrama de Parejas, así como sus representaciones matriciales respectivamente), ya que facilitan el análisis de estos autómatas.

Otra herramienta útil para ejemplificar e ilustrar los autómatas y sus tipos de diagramas fue el uso del software NXLCAU, este paquete ha sido programado bajo el sistema operativo NeXTSTEP (existe también versión para MS-DOS, LCAU), desarrollado por Harold V. McIntosh en el Instituto de Ciencias, Universidad Autónoma de Puebla.

La primera sección aborda los conceptos básicos de los Autómatas Celulares Lineales; las secciones dos, tres y cuatro muestran como se construyen y representan los diagramas de de Bruijn, Subconjuntos y Parejas respectivamente, la quinta sección introduce la idea básica de los *Autómatas-Reversibles* en una dimensión, las secciones seis y siete presentan métodos para poder obtener el modelo inverso de un *Autómata-Reversible*, las secciones ocho y nueve se enfocan en la búsqueda de estos autómatas y la sección diez muestra un caso "especial" de producto cartesiano y *reversibilidad*.

Contenido

1	Conceptos Básicos	5
2	Construcción del diagrama de de Bruijn	8
2.1	Diagrama de de Bruijn autómatas (2,1), regla 15	9
2.2	Diagrama de de Bruijn autómatas (4,h), regla 05AF0FA5	12
3	Construcción del diagrama de Subconjuntos	18
3.1	Diagrama de Subconjuntos autómatas (2,1), regla 15 . .	19
3.2	Diagrama de Subconjuntos autómatas (4,h), regla 05AF0FA5	22
4	Construcción del diagrama de Parejas	25
4.1	Diagrama de Parejas autómatas (2,1), regla 15	26
4.2	Producto Tensorial	27
4.3	Diagrama de parejas autómatas (4,h), regla 05AF0FA5 .	29
5	Autómatas Celulares Lineales Reversibles	30
5.1	Ejemplos de Autómatas Celulares Lineales no Reversibles	32
5.2	Ejemplos de Autómatas Celulares Lineales Reversibles .	38
6	Modelo de Fredkin (de segundo orden)	44
6.1	Regla de dos generaciones.	44
6.2	Obtención de la regla inversa para un autómatas (2,1) .	45
6.3	Obtención de la regla inversa para un autómatas (4,h) .	46
7	Uso de las Matrices de Conectividad del Diagrama de de Bruijn.	48
8	Construcción de un Autómatas Celular Lineal Reversible (4,h)	51
9	Algoritmo de Hillman	52
9.1	Ejemplos para autómatas (2,1)	55
9.2	Ejemplo para un autómatas (4,h)	57
10	Reversibilidad y Producto Cartesiano	62

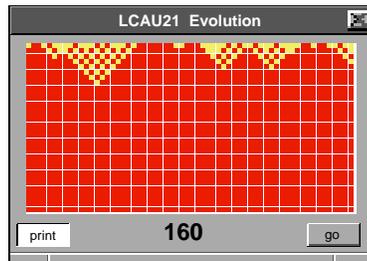
1 Conceptos Básicos

El desarrollo de los Autómatas Celulares (AC), ha tenido tres etapas importantes; la etapa de John von Neumann a principios de los 50's, von Neumann buscaba la auto-reproducción de las máquinas, recuérdese que era la época de la postguerra; como von Neumann carecía de los materiales suficientes siguió la sugerencia de su amigo Stanislaw Ulam para desarrollar un modelo matemático con el cual pudiera demostrar su teoría, con ello se sentaron las bases del estudio de los AC.

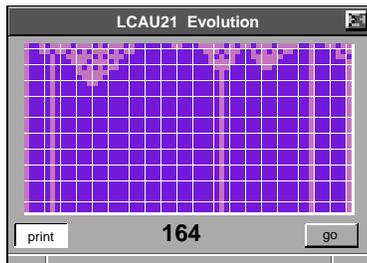
La segunda etapa importante vino en los 70's con el desarrollo del juego de *Life*, por parte de John Conway, el cual podía simular evoluciones interesantes de "seres" a través de un tiempo dado en un espacio bidimensional utilizando conceptos de los AC.

La tercera etapa importante vino a mediados de los 80's con las investigaciones realizadas por Stephen Wolfram [1] sobre los Autómatas Celulares Lineales (ACL), donde plantea una clasificación en cuatro clases dependiendo del comportamiento que muestran:

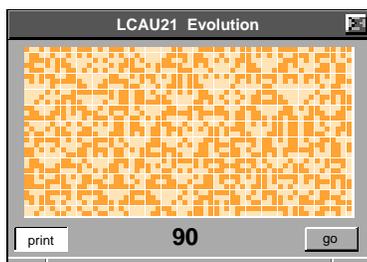
Clase I. AC que evolucionan en un estado uniforme.



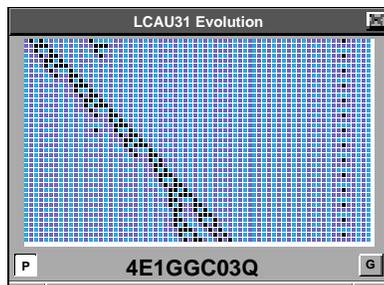
Clase II. AC que evolucionan en estados cíclicos aislados.



Clase III. AC que evolucionan en comportamientos caóticos.



Clase IV. AC que evolucionan en estados complejos aislados.



El ACL consiste en un conjunto de células conectadas linealmente, cada célula puede tener el valor de un número finito de estados y una función de transición (regla de evolución), la cual hace mapeos locales entre estos estados en unidades de tiempo discreto.

Un ACL denota su número de estados por k y el radio de la vecindad por r . La evolución del ACL depende de la función de transición y de la configuración inicial (primer arreglo de células). Para ejemplificar los conceptos dados anteriormente se tomará un autómata $(2,1)$, $k=2$ estados, $r=1$ vecino de cada lado.

Las diferentes vecindades que existen para este autómata están dadas por:

$$vecindades = \begin{cases} 11110000 \\ 11001100 \\ 10101010 \end{cases} \quad (1)$$

La regla de evolución especifica en que estado evoluciona cada vecindad, para este autómata la regla se denomina en valores binarios que van del 0 al 255 (en decimal, ya que hay 256 combinaciones diferentes), números de Wolfram [1], tomemos la regla 15.

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \quad (2)$$

$$2^7 \quad 2^6 \quad 2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0 \quad (3)$$

$$0 \quad 0 \quad 0 \quad 0 \quad 8 \quad 4 \quad 2 \quad 1 \quad \rightarrow 15 \quad (4)$$

Tomando una configuración inicial aleatoria, se observará como evoluciona a través del tiempo con esta regla.

$$\begin{array}{llll} 111001101000 & t & gen\{0\} \\ 100011001011 & t + 1 & gen\{1\} \\ 001110011010 & t + 2 & gen\{2\} \\ 111000110010 & t + 3 & gen\{3\} \\ 100011100110 & t + 4 & gen\{4\} \\ & \cdot & \cdot \\ & \cdot & \cdot \\ & \cdot & \cdot \end{array} \quad (5)$$

Las configuraciones que puede tener un autómata pueden ser de 3 tipos:

i) Configuraciones que no pueden ser generadas por su regla de evolución y que sólo pueden aparecer en su configuración inicial. A estas configuraciones sin pasado se les denomina pertenecientes al *Jardín*

del Edén.

ii) Configuraciones en el tiempo t que pueden ser generadas por diferentes configuraciones en el tiempo $t-1$, es decir, tienen múltiples ancestros. *Ancestro* es una configuración en el tiempo $t-1$ que genera la nueva configuración en el tiempo t .

iii) Configuraciones que son generadas por un ancestro único.

En particular un autómata que genera configuraciones del tipo iii), se les denomina Autómata Celular Lineal Reversible (ACLR), ya que al tener cada configuración un ancestro único, éste puede obtenerse mediante una regla inversa de evolución que revierte el proceso.

2 Construcción del diagrama de de Bruijn

La aplicación de Teoría de Gráficas juegan un papel importante en el estudio de los AC, existen tres gráficas que son fundamentales, estas son el diagrama de *de Bruijn*, el diagrama de *Subconjuntos* y el diagrama de *Parejas*. Una revisión detallada de muchas de las posibles aplicaciones de éstas en el estudio de AC ha sido publicada por McIntosh [2].

El diagrama de *de Bruijn* es una gráfica cuyos nodos son secuencias de símbolos de algún alfabeto, en este caso el conjunto de estados del autómata. Todas estas secuencias tienen la misma longitud, para un autómata éstas serían vecindades parciales, en una dimensión, el estado de una célula liga de un extremo a otro un vecindario dado.

En principio tal diagrama podría ser extendido a autómatas de mayores dimensiones, pero un problema surge con la selección de los vecindarios parciales para los nodos del diagrama, para los cuales su unión formará los vecindarios completos en todas direcciones; pero al menos en una dimensión no hay ninguna dificultad en crear el diagrama de *de Bruijn*.

El número de nodos del diagrama de *de Bruijn* está dado por $k^{\hat{2}r}$

y el número de ligas por $k(2r+1)$. Ya que los nodos representan fracciones de las vecindades que tenga el autómata, habrá una liga del nodo i al nodo j , si y sólo si, la salida del nodo i coincide con la entrada del nodo j (como fichas de dominó), de esta forma, las características de la evolución pueden ser usadas para seleccionar subgráficas del diagrama de *de Bruijn*, por ejemplo; aquella subgráfica compuesta de las vecindades, cuya célula central nunca cambia si tales vecindades existen.

El diagrama de *de Bruijn* transforma el problema del autómata en un problema de trazo de rutas, por ejemplo; ningún loop puede ser más largo que el número total de nodos en la gráfica sin repetir algún segmento. Ya que los símbolos son enteros consecutivos pueden ser tratados como elementos de un anillo o quizás de un campo finito. La discusión de sus propiedades aritméticas o algebraicas es muy valiosa, por ejemplo; la matriz de conectividad de diagrama se convierte simplemente en,

$$M_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{otro caso} \end{cases} = \begin{cases} ki \\ ki + 1 \\ \dots \\ ki + k - 1 \end{cases} \quad (6)$$

**

2.1 Diagrama de de Bruijn autómata (2,1), regla 15

Se mostrará un ejemplo de la construcción de un diagrama de *de Bruijn* tomando un autómata (2,1) bajo la regla de evolución 15.

$$\begin{aligned} k &= 2 = \text{número de estados} \\ r &= 1 = \text{radio de la vecindad} \end{aligned}$$

Vecindades:

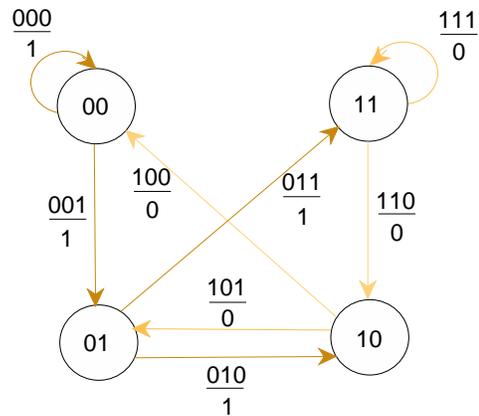
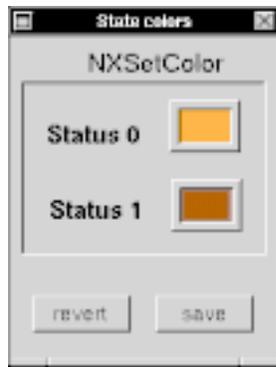
$$\begin{array}{r} 11110000 \\ 11001100 \\ 10101010 \\ \hline 00001111 \end{array} \rightarrow \text{regla15} \quad (7)$$

$$\begin{aligned} k^{2r} &= (2)^{2(1)} = 4, \text{ n\u00famero de nodos en el diagrama} \\ k^{2r+1} &= (2)^{2(1)+1} = 8, \text{ n\u00famero de ligas en el diagrama} \end{aligned} \quad (8)$$

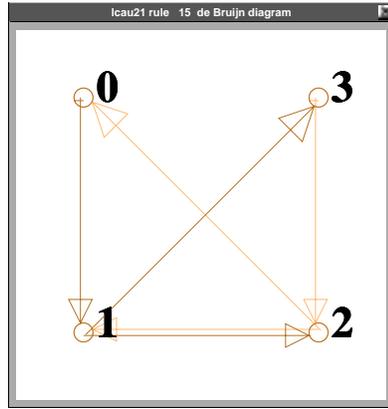
Descripci\u00f3n de los nodos.

$$\text{Fracciones de las vecindades : } \begin{cases} \text{nodo 1 : 00} \\ \text{nodo 2 : 01} \\ \text{nodo 3 : 10} \\ \text{nodo 4 : 11} \end{cases} \quad (9)$$

Diagrama de de Bruijn



Este diagrama puede simplificar su notaci\u00f3n binaria a decimal, como lo denota McIntosh [3].



La matriz de conectividad del diagrama de *de Bruijn* queda de la siguiente manera.

$$\begin{array}{l}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \begin{bmatrix}
 1 & 1 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 1 \\
 1 & 1 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 1
 \end{bmatrix}
 \quad (10)$$

Matriz de evolución del diagrama de *de Bruijn*.

$$\begin{array}{l}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \begin{bmatrix}
 1 & 1 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 1 \\
 0 & 0 & \cdot & \cdot \\
 \cdot & \cdot & 0 & 0
 \end{bmatrix}
 \quad (11)$$

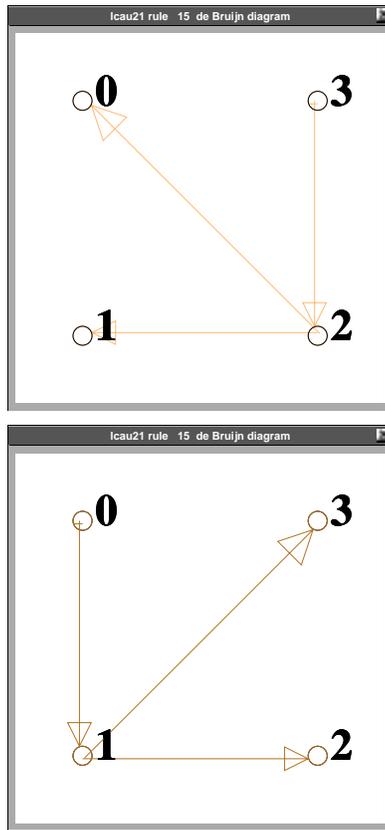
Es claro que se pueden derivar dos matrices, cada una mostrando la evolución a un estado en particular; estas matrices se denominan matriz A para las ligas del estado 0 y matriz B para las ligas del estado 1.

Evolución en el estado 0.

$$A = \begin{array}{l}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \begin{bmatrix}
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot \\
 1 & 1 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 1
 \end{bmatrix}
 \quad (12)$$

Evolución en el estado 1.

$$B = \begin{array}{l}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \begin{bmatrix}
 1 & 1 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 1 \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot
 \end{bmatrix}
 \quad (13)$$



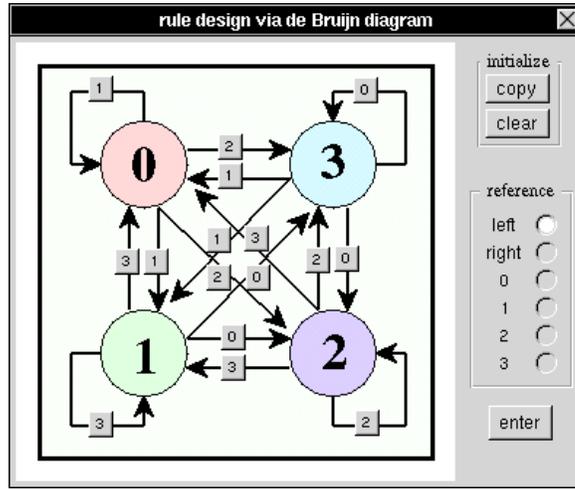
2.2 Diagrama de de Bruijn automática (4,h), regla 05AF0FA5

Cuando el radio de vecindad es fraccionario (0.5,1.5,...), se denotará con letras como en este caso $h=0.5$, media vecindad a cada lado.

$$\begin{aligned} k &= 4 \\ r &= \frac{1}{2} \end{aligned} \tag{14}$$

Vecindades cíclicas:

$$\begin{array}{r} 3333222211110000 \\ 3210321032103210 \\ \hline 0011223300332211 \end{array} \rightarrow \text{regla05AF0FA5} \tag{15}$$



Esta regla se codifica en parejas utilizando base hexadecimal para asignarle un valor a cada par.

$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	(16)
00	11	22	33	00	33	22	11	
0 + 0	4 + 1	8 + 2	12 + 3	0 + 0	12 + 3	8 + 2	4 + 1	
0	5	A	F	0	F	A	5	

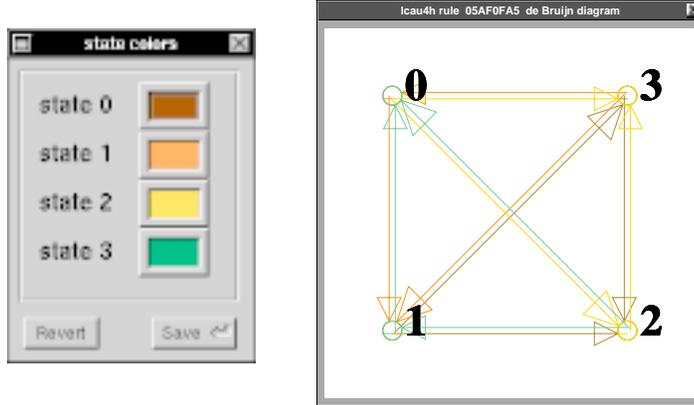
En este caso para mostrar la evolución de este autómata se tomará una configuración inicial aleatoria.

$$\begin{array}{lll}
 0113030122021201023 & t & gen\{0\} \\
 11301211023230313221 & t + 1 & gen\{1\} \\
 3011033322021210023 & t + 2 & gen\{2\} \\
 01133200023230331220 & t + 3 & gen\{3\} \\
 \dots & \cdot & \cdot \\
 \dots & \cdot & \cdot \\
 \dots & \cdot & \cdot
 \end{array} \quad (17)$$

Se construye el diagrama de *de Bruijn*.

$$\begin{aligned}
 k^{2r} &= (4)^{2(\frac{1}{2})} = 4 \text{ nodos en el diagrama} \\
 k^{2r+1} &= (4)^{2(\frac{1}{2})+1} = 16 \text{ ligas en el diagrama}
 \end{aligned} \quad (18)$$

$$\text{Fracciones de las vecindades : } \left\{ \begin{array}{l} \text{nodo } 1 : 0 \\ \text{nodo } 2 : 1 \\ \text{nodo } 3 : 2 \\ \text{nodo } 4 : 3 \end{array} \right. \quad (19)$$



Para los ACL con radio de vecindad $\mathbf{r}=\mathbf{h}$, todos los nodos en el diagrama de *de Bruijn* son conectados aunque no tengan la misma entrada y salida, en los diagramas para un ACL de radio de vecindad \mathbf{h} , su notación se conserva igual.

Matriz de evolución del diagrama de *de Bruijn*.

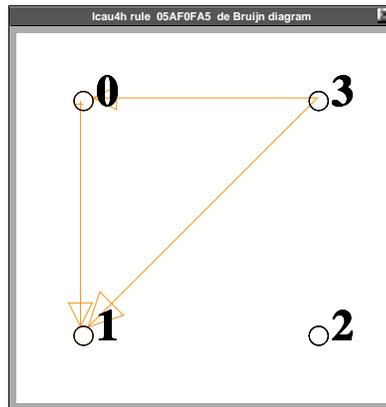
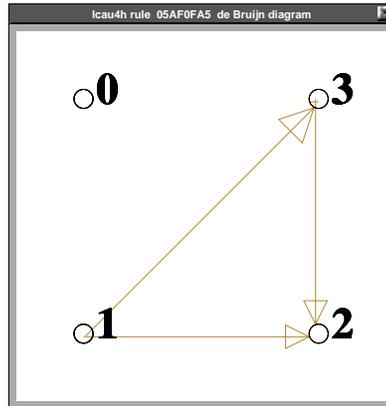
$$\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \left[\begin{array}{cccc} 1 & 1 & 2 & 2 \\ 3 & 3 & 0 & 0 \\ 3 & 3 & 2 & 2 \\ 1 & 1 & 0 & 0 \end{array} \right] \quad (20)$$

Evolución en el estado 0.

$$A = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \left[\begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{array} \right] \quad (21)$$

Evolución en el estado 1.

$$B = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \left[\begin{array}{cccc} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{array} \right] \quad (22)$$

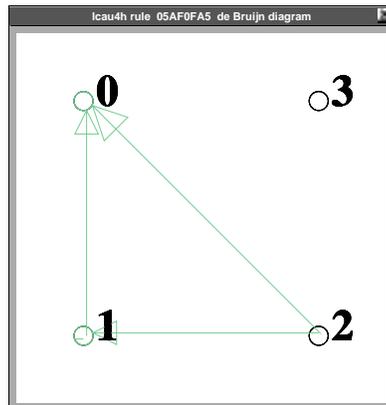
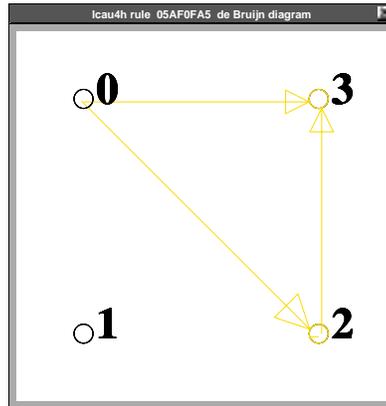


Evolución en el estado 2.

$$C = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} . & . & 1 & 1 \\ . & . & . & . \\ . & . & 1 & 1 \\ . & . & . & . \end{bmatrix} \quad (23)$$

Evolución en el estado 3.

$$D = \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} . & . & . & . \\ 1 & 1 & . & . \\ 1 & 1 & . & . \\ . & . & . & . \end{bmatrix} \quad (24)$$



3 Construcción del diagrama de Subconjuntos

El diagrama de *Subconjuntos* es un derivado del diagrama de de Bruijn, se usa principalmente para obtener todas las configuraciones que puedan evolucionar en una configuración deseada, es decir, encontrar ancestros. Una excelente aplicación de la construcción del diagrama de *Subconjuntos* es la búsqueda del Jardín del Edén, es decir, configuraciones que no tienen ancestros.

McIntosh [2] da dos razones importantes para trabajar con diagramas de *Subconjuntos*.

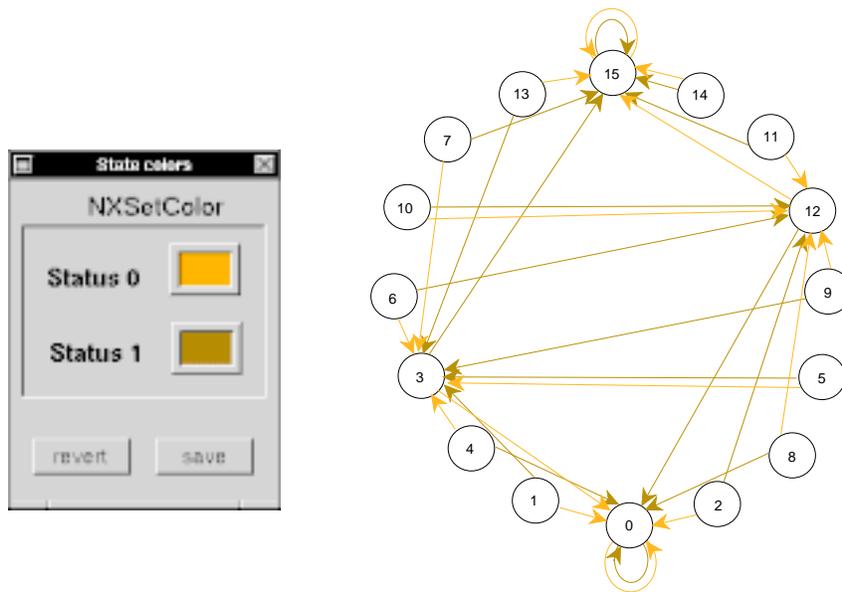
- Etiquetar ligas da la apariencia de funciones, ya que asocia unos objetos con otros, pero si dos ligas con la misma etiqueta surgen de un mismo vértice, ellas difícilmente pueden representar una función; construyendo el diagrama de *Subconjuntos* de todos los destinos, deja una simple liga entre éstos, brindando funcionalidad al diagrama aunque ésta no existiera originalmente, la inclusión del conjunto vacío asegura que cada punto tiene una imagen, permitiendo parcialmente definir funciones.
- Una vez que el diagrama de *Subconjuntos* ha sido formado, se ve si un camino va del conjunto universal al conjunto vacío, esto es evidencia decisiva de que tal ruta no existe en el diagrama original.

Otra aplicación es determinar donde existen rutas partiendo de las clases unitarias, tales rutas pueden ser usadas para forzar un autómata dentro de un estado predeterminado, no importando su condición original.

3.1 Diagrama de Subconjuntos autómeta (2,1), regla 15

El número de nodos del diagrama de de Bruijn es igual a 4, por lo tanto k toma este valor para posteriormente, calcular los nodos del diagrama de *Subconjuntos*.

$$2^k = \text{número de nodos para el diagrama de Subconjuntos} \quad 2^k = 2^4 = 16 \text{ nodos en el diagrama} \quad (25)$$



Tomando las matrices A y B del diagrama de de Bruijn.

$$\begin{bmatrix} 1 & 1 & . & . \\ . & . & 1 & 1 \\ 0 & 0 & . & . \\ . & . & 0 & 0 \end{bmatrix} = \begin{matrix} A \\ \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ 1 & 1 & . & . \\ . & . & 1 & 1 \end{bmatrix} \end{matrix} + \begin{matrix} B \\ \begin{bmatrix} 1 & 1 & . & . \\ . & . & 1 & 1 \\ . & . & . & . \\ . & . & . & . \end{bmatrix} \end{matrix} \quad (26)$$

entonces, obtenemos la siguiente tabla de destinos de nodos individuales.

<i>nodo</i>	<i>liga con 0</i>	<i>liga con 1</i>
0	\emptyset	0, 1
1	\emptyset	2, 3
2	0, 1	\emptyset
3	2, 3	\emptyset

(27)

Formando los *Subconjuntos*.

<i>Subconjunto</i>	<i>liga con 0</i>	<i>liga con 1</i>
0, 1, 2, 3	0, 1, 2, 3	0, 1, 2, 3
1, 2, 3	0, 1, 2, 3	0, 1, 2, 3
0, 1, 2	0, 1	0, 1, 2, 3
0, 1, 3	2, 3	0, 1, 2, 3
0, 2, 3	0, 1, 2, 3	0, 1
0, 1	\emptyset	0, 1, 2, 3
0, 2	0, 1	0, 1
0, 3	2, 3	0, 1
2, 1	0, 1	2, 3
2, 3	0, 1, 2, 3	\emptyset
1, 3	2, 3	2, 3
3	2, 3	\emptyset
2	0, 1	\emptyset
1	\emptyset	2, 3
0	\emptyset	0, 1
\emptyset	\emptyset	\emptyset

(28)

Para simplificar la notación de los *Subconjuntos*, éstos se renombran de la siguiente manera.

<i>Configuración Binaria</i>	<i>Valor en decimal</i>	<i>Subconjuntos (combinación)</i>
3210		
1111	15	3, 2, 1, 0
1110	14	3, 2, 1
1101	13	3, 2, 0
1100	12	3, 2
1011	11	3, 1, 0
1010	10	3, 1
1001	9	3, 0
1000	8	3
0111	7	2, 1, 0
0110	6	2, 1
0101	5	2, 0
0100	4	2
0011	3	1, 0
0010	2	1
0001	1	0
0000	0	\emptyset

(29)

McIntosh [4] define tres propiedades importantes del diagrama de *Subconjuntos*:

1. Si hay una cadena que va de un subconjunto al conjunto vacío, existirá una cadena la cual irá desde cualquier subconjunto más pequeño incluido al conjunto vacío; por el contrario, si las clases unitarias carecen de ligas al conjunto vacío, ningún otro subconjunto tendrá tales ligas y no existirá el Jardín del Edén, tal sea el caso cuando cada nodo del diagrama de de Bruijn tiene una liga asociada con cada diferente tipo de célula, esto es, cuando es estocástica por renglón o es en sí una función. Similarmente, todo el subconjunto no ligará con otros subconjuntos, no permitiendo la existencia de un Jardín del Edén, si cada nodo tiene una liga entrante para cada tipo de célula tales reglas tienen columnas estocásticas en las matrices de de Bruijn. Las reglas para que el Jardín del Edén no exista por razones más sutiles son mucho más raras y más interesantes, requiriendo un estudio más cuidadoso del di-

agrama de *Subconjuntos* para su descubrimiento.

2. Hay un cierto residuo de la conectividad del diagrama de de Buijn en el sentido que, dada cualquier fuente y cualquier destino, siempre habrá un subconjunto conteniendo el destino accesible desde cualquier subconjunto conteniendo la fuente, pero el destino puede tener nodos adicionales.

3. El diagrama de *Subconjuntos* puede no ser conectado; aún si esto sucede, es interesante conocer el subconjunto más grande accesible desde algún subconjunto dado, así también como el más pequeño.

A continuación se construirá la matriz de conectividad.

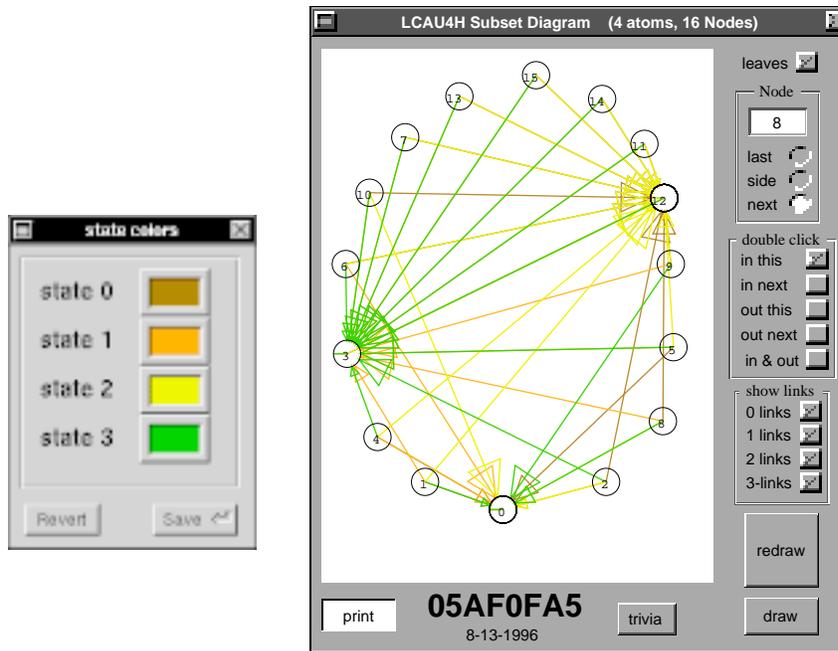
0	2
1	1	.	.	1
2	1	1
4	1	1
8	1	1
3	1	1
5	2
6	1	1
9	1	1
10	2
12	1	1
7	1	1
11	1	1
13	1	1
14	2
15	2

(30)

3.2 Diagrama de Subconjuntos autómeta (4,h), regla 05AF0FA5

El número de nodos del diagrama de de Buijn es igual a 4.

$$2^k = 2^4 = 16 \text{ nodos del diagrama de Subconjuntos} \quad (31)$$



Tomando las matrices de conectividad A, B, C y D del diagrama de de Bruijn.

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{bmatrix}
 \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{bmatrix}
 \begin{bmatrix} \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}
 \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}
 \quad (32)$$

Matriz de evolución.

$$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix}
 \begin{bmatrix} 1 & 1 & 2 & 2 \\ 3 & 3 & 0 & 0 \\ 3 & 3 & 2 & 2 \\ 1 & 1 & 0 & 0 \end{bmatrix}
 \quad (33)$$

Tabla de destinos de nodos individuales.

<i>nodo</i>	<i>liga con 0</i>	<i>liga con 1</i>	<i>liga con 2</i>	<i>liga con 3</i>
0	\emptyset	0, 1	2, 3	\emptyset
1	2, 3	\emptyset	\emptyset	0, 1
2	\emptyset	\emptyset	2, 3	0, 1
3	2, 3	0, 1	\emptyset	\emptyset

(34)

Formando los subconjuntos.

<i>subconjunto</i>	<i>liga con 0</i>	<i>liga con 1</i>	<i>liga con 2</i>	<i>liga con 3</i>
0, 1, 2, 3	2, 3	0, 1	2, 3	0, 1
1, 2, 3	2, 3	0, 1	2, 3	0, 1
0, 1, 2	2, 3	0, 1	2, 3	0, 1
0, 1, 3	2, 3	0, 1	2, 3	0, 1
0, 2, 3	2, 3	0, 1	2, 3	0, 1
0, 1	2, 3	0, 1	2, 3	0, 1
0, 2	\emptyset	0, 1	2, 3	0, 1
0, 3	2, 3	0, 1	2, 3	\emptyset
2, 1	2, 3	\emptyset	2, 3	0, 1
2, 3	2, 3	0, 1	2, 3	0, 1
1, 3	2, 3	0, 1	\emptyset	0, 1
3	2, 3	0, 1	\emptyset	\emptyset
2	\emptyset	\emptyset	2, 3	0, 1
1	2, 3	\emptyset	\emptyset	0, 1
0	\emptyset	0, 1	2, 3	\emptyset
\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

(35)

La simplificación de los nodos es la misma en este caso que la utilizada para el autómata (2,1).

Matriz de Conectividad.

0	4				
1	2	1	.	.	.	1
2	2	1	.	.	.	1
4	2	1	.	.	.	1
8	2	1	.	.	.	1
3	2	.	.	.	2
5	1	2	.	.	.	1
6	1	1	.	.	.	2
9	1	1	.	.	.	2
10	1	2	.	.	.	1
12	2	.	.	.	2
7	2	.	.	.	2
11	2	.	.	.	2
13	2	.	.	.	2
14	2	.	.	.	2
15	2	.	.	.	2

(36)

Aquí se pueden observar las ligas de las clases unitarias, binarias, trinarias y cuatrinarias, así como una uniformidad en las mismas.

4 Construcción del diagrama de Parejas

El producto cartesiano (diagrama de **Parejas**) del diagrama de de Bruijn tiene muchas aplicaciones, ya sea comparar rutas entre dos diferentes diagramas o dos rutas distintas en el mismo diagrama. No siempre es necesario distinguir entre los miembros de una pareja, las flechas en una gráfica requieren que las ligas sean definidas como pares ordenados de los nodos, pero que no requieren necesariamente que una pareja de ligas sea tomada en orden particular, como lo señala McIntosh [4]. Si una configuración tiene múltiples ancestros puede ser visualizado en el diagrama de *Parejas*, los nodos de un diagrama de *Parejas*, son pares de nodos del diagrama de de Bruijn, las ligas unen parejas cuando ambos miembros de la pareja estén unidos por ligas con la misma etiqueta en el diagrama de de Bruijn, las rutas en el diagrama de *Parejas* corresponden a pares de rutas en el diagrama original; que ni siquiera pueden originarse en el mismo nodo.

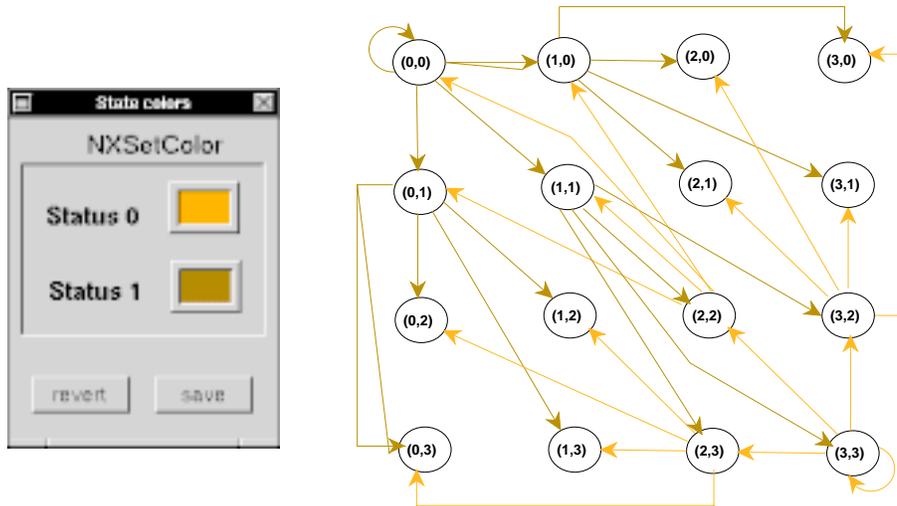
Sin embargo, la misma ruta tomada dos veces puede ser encontrada entre algunos otros, de este modo el diagrama de de Bruijn está siempre contenido en el diagrama de *Parejas*.

4.1 Diagrama de Parejas autómeta (2,1), regla 15

Se construye el diagrama de parejas.

$$[k^{2r}]^2 = \text{número de nodos en el diagrama de Parejas} \quad (37)$$

$$[2^{2(1)}]^2 = 16$$



La matriz asociada al diagrama de *Parejas* se denomina como matriz **T**.

00	1	1	.	.	1	1
01	.	.	1	1	.	.	1	1
02
03
10	1	1	.	.	1	1
11	1	1	.	1	1
12
13
20
21
22	1	1	.	.	1	1
23	.	.	1	1	.	.	1	1
30
31
32	1	1	.	.	1	1
33	1	1	.	1	1

(38)

Se observa que esta matriz se puede obtener a través del producto tensorial de la matriz de conectividad con las matrices A y B, tomando éstas según indique la matriz de evolución y representándose como submatrices de conectividad.

4.2 Producto Tensorial

En la matriz de evolución se ve que el elemento $(\mathbf{0},\mathbf{0})$ evoluciona en uno, por lo tanto, el producto tensorial multiplicará el elemento $(\mathbf{0},\mathbf{0})$ de la matriz de conectividad con la matriz B (evolución en el estado 1) y de esta forma se obtiene su submatriz de conectividad:

$$\begin{array}{l}
 00 \\
 01 \\
 02 \\
 03
 \end{array}
 \begin{bmatrix}
 1 & 1 & . & . \\
 . & . & 1 & 1 \\
 . & . & . & . \\
 . & . & . & .
 \end{bmatrix}
 \tag{39}$$

la cual pertenece a la matriz asociada del diagrama de *Parejas*.

Esta submatriz es estocástica por columna, la matriz de parejas es una herramienta eficaz para observar el número de ancestros de una

configuración, también se puede observar que existe una propiedad de simetría como se ve a continuación:

$$si \quad (2, 3) \quad \boxed{\begin{matrix} (0, 2) \\ 1 \end{matrix}} \Rightarrow (3, 2) \quad \boxed{\begin{matrix} (2, 0) \\ 1 \end{matrix}} \quad (40)$$

$$si \quad (1, 1) \quad \boxed{\begin{matrix} (2, 3) \\ 1 \end{matrix}} \Rightarrow (1, 1) \quad \boxed{\begin{matrix} (3, 2) \\ 1 \end{matrix}} \quad (41)$$

$$si \quad (2, 0) \quad \boxed{\begin{matrix} (i, j) \\ \cdot \end{matrix}} \Rightarrow (0, 2) \quad \boxed{\begin{matrix} (j, i) \\ \cdot \end{matrix}} \quad \forall \quad i, j \quad (42)$$

por lo tanto, se puede generalizar,

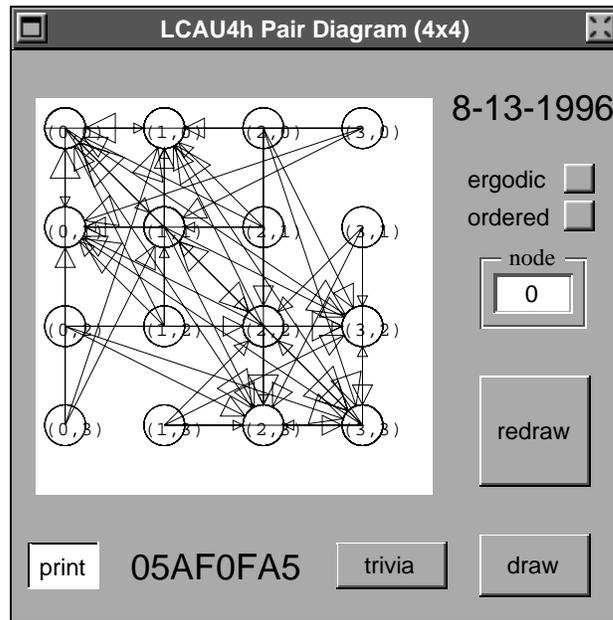
si (x, y) conecta a (x', y') entonces (y, x) conecta a (y', x')

El diagrama de *Parejas* describe las relaciones de parejas ordenadas; pero su propiedad de simetría nos permite obtener un diagrama desordenado, en el cual la distinción entre (x, y) y (y, x) se considera insignificante. De esta manera se produce un diagrama más pequeño, éste puede observarse uniendo los nodos correspondientes del diagrama de *Parejas* original.

4.3 Diagrama de parejas automática (4,h), regla 05AF0FA5

Se construye el diagrama.

$$[k^{2r}]^2 = [(4)^{(2)(\frac{1}{2})}]^2 = 16, \text{ número de nodos en el diagrama} \quad (43)$$



Como se observa la propiedad de simetría aparece de manera muy clara.

Matriz \mathbf{T} del diagrama de *Parejas*.

00	1	1	.	.	1	1	1	1	.	.	1	1
01
02	1	1	.	.	1	1	
03	1	1	.	.	1	1
10
11	1	1	.	.	1	1	.	.	.	1	1	.	.	1	1	
12	1	1	.	.	1	1
13	1	1	.	.	1	1	
20	1	1	.	.	1	1	
21	1	1	.	.	1	1
22	1	1	.	.	1	1	.	.	.	1	1	.	.	1	1	
23
30	1	1	.	.	1	1
31	1	1	.	.	1	1	
32
33	1	1	.	.	1	1	.	.	.	1	1	.	.	1	1	

(44)

En esta matriz también se observa la simetría señalada anteriormente y muestra un cierto comportamiento uniforme en sus submatrices.

5 Autómatas Celulares Lineales Reversibles

El trabajo de Moore [5] (1962), sobre la existencia del Jardín del Edén probó que si existen configuraciones que no tuvieran ningún ancestro, entonces debían existir configuraciones que tuvieran más de uno; durante este tiempo los matemáticos profesionales no se interesaban en el desarrollo de AC, por lo que el estudio de la reversibilidad permaneció estancado; catorce años después, los resultados del trabajo de Toffoli [6] probaron la existencia de ACLR que podían ser relevantes para el modelado de fenómenos físicos. Independientemente Fredkin [7] había estado estudiando la invertibilidad de modelos con comportamiento dinámico y desarrolló técnicas para sintetizar un comportamiento secuencial arbitrario reversible con funciones booleanas.

Uno de los objetivos de la ciencia de la computación es obtener modelos abstractos de computadoras concretas, es decir, cualquier aparato

computador que puede ser construido físicamente. En estos modelos uno busca expresividad (que el modelo capture todos los aspectos relevantes de la computadora), y exactitud (que el modelo realice computaciones correctas). Los Autómatas Celulares Lineales Reversibles (ACLR), son un importante desarrollo en esa dirección de significancia comparable a la máquina de Turing [8]. Los AC son más expresivos que las máquinas de Turing, ya que nos dan una manera explícita de modelar computación en paralelo, sin embargo, ambas clases de modelos son indiferentes a un aspecto fundamental de la física, la reversibilidad y así visualizar la posibilidad de una computación perpetua. En general, los ACLR proveen un modelado más exacto y productivo para estos fines, hace unos pocos años lo que se conocía acerca de ACLR era escaso y poco interesante, sin embargo, actualmente ha alcanzado un mayor desarrollo. Los campos que trabajan actualmente con ACLR incluyen: computación concurrente en redes que tienen estructura uniforme, preservación de información en procesos computacionales, encontrar conexiones fundamentales entre la física y la computación, computación cuántica, fundamentos de la relatividad, modelado de sistemas físicos granulares, mayor calidad en la simulación de AC y encriptación de datos.

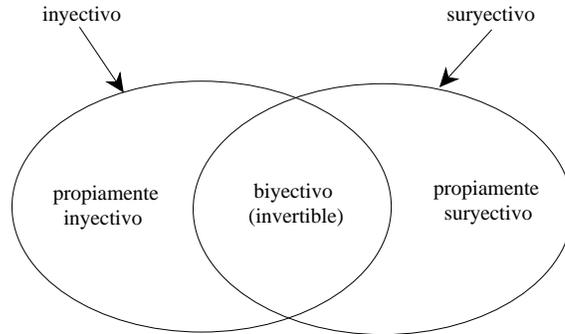
Un AC es un sistema determinístico, es decir, para cada posible configuración su regla de evolución específica un sólo sucesor, existen ACL en los que podemos encontrar una nueva regla que obligue al autómata a invertir su evolución en el tiempo. Por supuesto, esto es sólo posible si la regla de evolución es determinística hacia atrás, como lo señalan Toffoli y Margolus [6], es decir, si para cada posible configuración existe uno y sólo un ancestro, a este tipo de ACL se les denomina reversibles, en estos ACLR se pueden observar dos importantes propiedades:

a) *Suryectividad*, se define un ACL como suryectivo si cada configuración tiene al menos un ancestro.

b) *Inyectividad*, un ACL es inyectivo si cada una de sus configuraciones tiene a lo más un ancestro.

Si ambas propiedades se cumplen para un ACL se dice que es *biyectivo* o *reversible*. El siguiente diagrama de Venn definido por Toffoli y

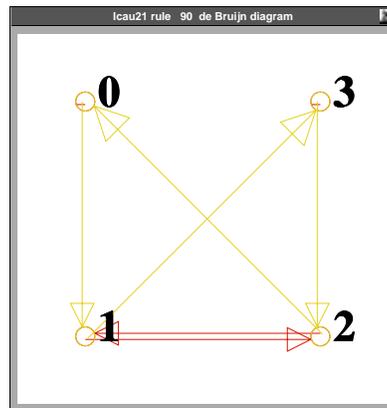
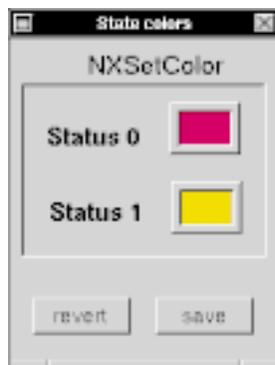
Margolus (Gutowitz [7]), ilustra estas propiedades.



La regla que hace que el sistema regrese en el tiempo, es decir, la reversible a la regla de evolución original se denomina regla inversa, en la mayoría de los casos la regla inversa es diferente de la original; finalmente, excepto para el caso de un ACL no se conoce un procedimiento sistemático para decidir cuando o no un AC es reversible.

5.1 Ejemplos de Autómatas Celulares Lineales no Reversibles

Primero. Autómata (2,1), regla 90.



$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 \hline
 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0
 \end{array} \rightarrow \text{regla90}$$
(45)

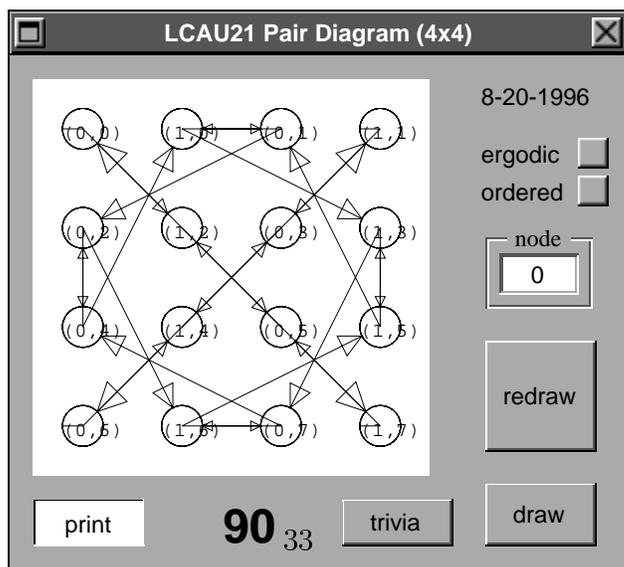
Matrices de conectividad.

$$\begin{array}{cc}
 A & B \\
 \left[\begin{array}{cccc} 1 & . & . & . \\ . & . & 1 & . \\ . & 1 & . & . \\ . & . & . & 1 \end{array} \right] & \left[\begin{array}{cccc} . & 1 & . & . \\ . & . & . & 1 \\ 1 & . & . & . \\ . & . & 1 & . \end{array} \right]
 \end{array}$$
(46)

Se observa que las matrices de conectividad son estocásticas por renglón y por columna, que representan a la matriz de evolución.

$$\begin{array}{c}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \left[\begin{array}{cccc}
 0 & 1 & . & . \\
 . & . & 0 & 1 \\
 1 & 0 & . & . \\
 . & . & 1 & 0
 \end{array} \right]$$
(47)

Veamos el diagrama de Parejas.



Como se puede observar en el diagrama de parejas, existen ciclos fuera de la diagonal, lo que indica que algunas configuraciones tienen múltiples ancestros.

Tomemos la siguiente secuencia de nodos:

$$\begin{array}{ccccccc}
 & (3,1) & \rightarrow & (2,3) & \rightarrow & (0,2) & \rightarrow & (1,0) \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 a) & 3 & & 2 & & 0 & & 1 & (48) \\
 & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 b) & 11 & & 10 & & 00 & & 01 \\
 \hline
 \end{array}$$

$$\begin{array}{ccc}
 c) & & 11001 \\
 d) & & 000 & (49)
 \end{array}$$

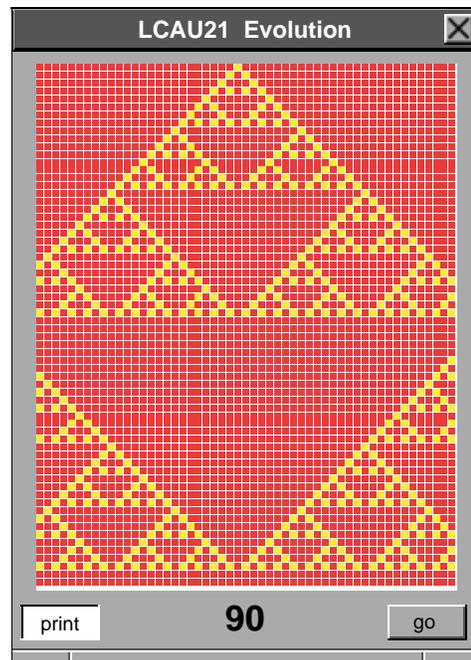
- a) Elementos **i** (izquierdos) de cada nodo.
- b) Nodos del diagrama de de Bruijn.
- c) Configuración en el tiempo **t**.
- d) Evolución en el tiempo **t+1**.

La configuración 000 evoluciona en el pasado en el estado 0, tomemos ahora los elementos **j** (derechos):

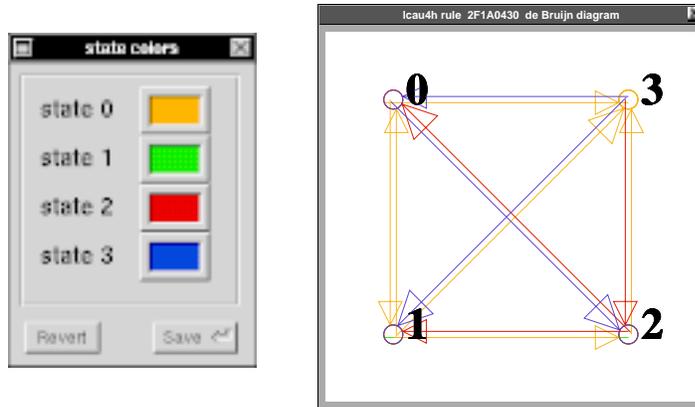
$$\begin{array}{ccccccc}
 (3,1) & \rightarrow & (2,3) & \rightarrow & (0,2) & \rightarrow & (1,0) \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 1 & & 3 & & 1 & & 0 & 01100 \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow & 000 & (50) \\
 01 & & 11 & & 10 & & 00 \\
 \hline
 \end{array}$$

La configuración 000 evoluciona en el pasado en el estado 1, por lo tanto, este autómata presenta múltiples ancestros, así como ciclos

fuera de la diagonal en el diagrama de parejas y consecuentemente no es reversible.



Segundo. Autómata (4,h), regla 2F1A0430.



$$\begin{array}{r} 3333222211110000 \\ 3210321032103210 \\ \hline 0233012200100300 \end{array} \rightarrow \text{regla2F1A0430} \quad (51)$$

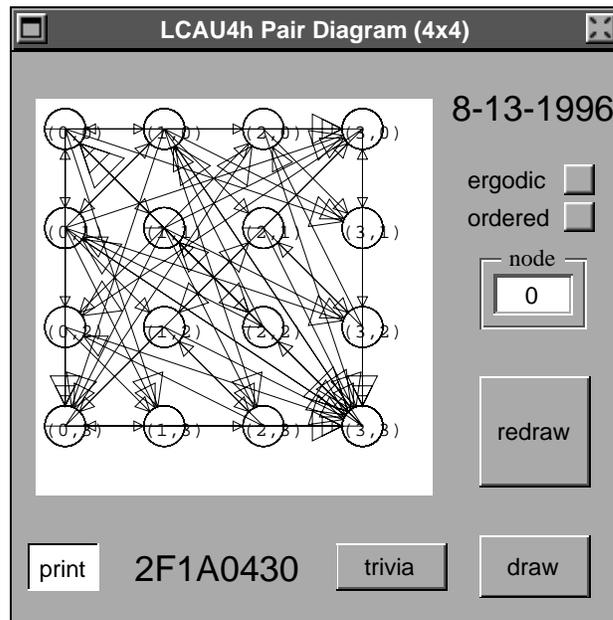
Matrices de conectividad A, B, C y D del diagrama de de Bruijn.

$$\begin{array}{c} A \\ \begin{bmatrix} 1 & 1 & . & 1 \\ 1 & . & 1 & 1 \\ . & . & . & 1 \\ . & . & . & 1 \end{bmatrix} \end{array} \quad \begin{array}{c} B \\ \begin{bmatrix} . & . & . & . \\ . & 1 & . & . \\ . & . & 1 & . \\ . & . & . & . \end{bmatrix} \end{array} \quad \begin{array}{c} C \\ \begin{bmatrix} . & . & . & . \\ . & . & . & . \\ 1 & 1 & . & . \\ . & . & 1 & . \end{bmatrix} \end{array} \quad \begin{array}{c} D \\ \begin{bmatrix} . & . & 1 & . \\ . & . & . & . \\ . & . & . & . \\ 1 & 1 & . & . \end{bmatrix} \end{array} \quad (52)$$

Matriz de evolución.

$$\begin{array}{c}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \begin{bmatrix}
 0 & 0 & 3 & 0 \\
 0 & 1 & 0 & 0 \\
 2 & 2 & 1 & 0 \\
 3 & 3 & 2 & 0
 \end{bmatrix}
 \quad (53)$$

Veamos su diagrama de Parejas.

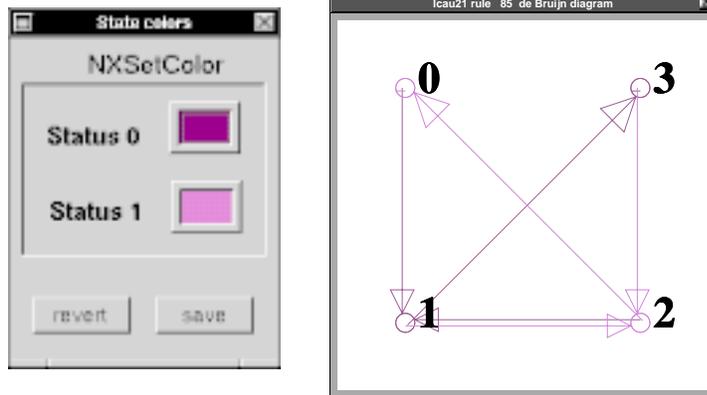


En este diagrama también existen ciclos fuera de la diagonal, indicando que existen configuraciones que tienen múltiples ancestros.

Tomemos la siguiente secuencia de nodos:

$$\begin{array}{cccc}
 (0, 2) & \rightarrow & (1, 3) & \rightarrow & (2, 3) & \rightarrow & (0, 2) \\
 \downarrow & & \downarrow & & \downarrow & & \downarrow \\
 0 & & 1 & & 2 & & 0
 \end{array}
 \quad (54)$$

$$\begin{array}{c}
 0120 \\
 002
 \end{array}
 \quad (55)$$



$$\begin{array}{r}
 11110000 \\
 11001100 \\
 10101010 \\
 \hline
 01010101 \rightarrow \text{regla85}
 \end{array} \tag{57}$$

Matrices de conectividad A y B del diagrama de de Bruijn.

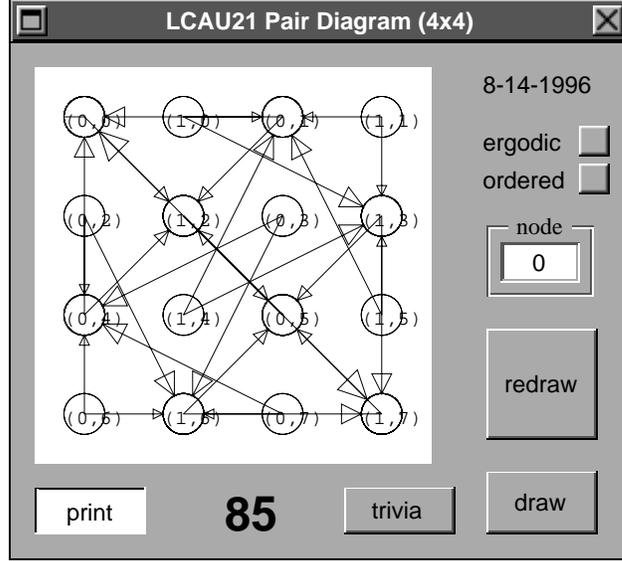
$$\begin{array}{c}
 A \qquad B \\
 \left[\begin{array}{cccc}
 \cdot & 1 & \cdot & \cdot \\
 \cdot & \cdot & \cdot & 1 \\
 \cdot & 1 & \cdot & \cdot \\
 \cdot & \cdot & \cdot & 1
 \end{array} \right] \quad \left[\begin{array}{cccc}
 1 & \cdot & \cdot & \cdot \\
 \cdot & \cdot & 1 & \cdot \\
 1 & \cdot & \cdot & \cdot \\
 \cdot & \cdot & 1 & \cdot
 \end{array} \right]
 \end{array} \tag{58}$$

Matriz de evolución.

$$\begin{array}{c}
 0 \\
 1 \\
 2 \\
 3
 \end{array}
 \left[\begin{array}{cccc}
 1 & 0 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 0 \\
 1 & 0 & \cdot & \cdot \\
 \cdot & \cdot & 1 & 0
 \end{array} \right] \tag{59}$$

Estas matrices de conectividad son estocásticas por renglón, veamos su diagrama de Parejas.

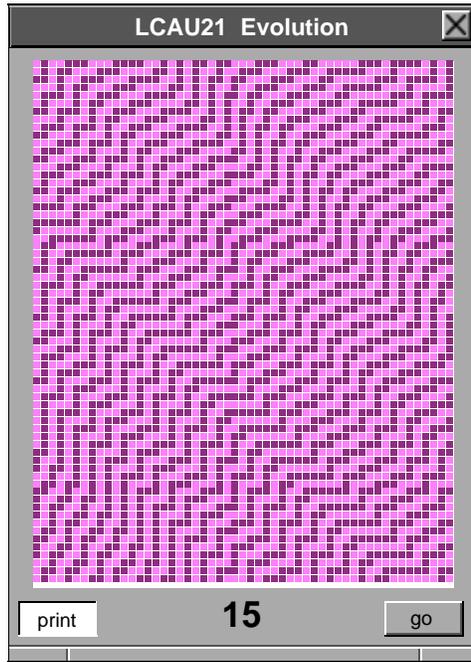
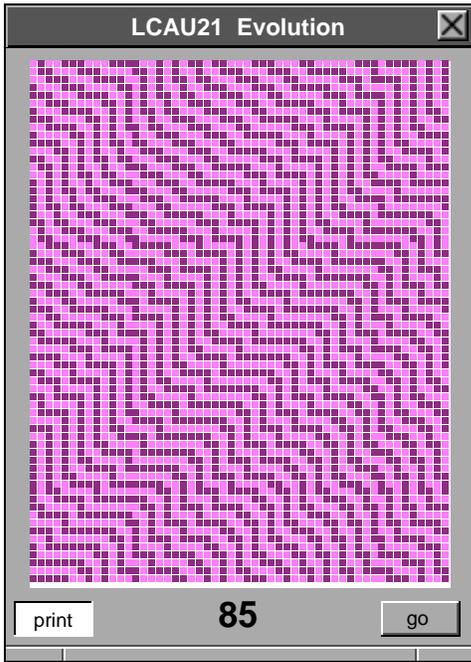
Se observa que no existen ciclos fuera de la diagonal, lo cual es una condición necesaria más no suficiente para que este autómata sea



reversible, dada una configuración aleatoria se verá como evoluciona para obtener la evolución hacia atrás en el tiempo de cada posible vecindad.

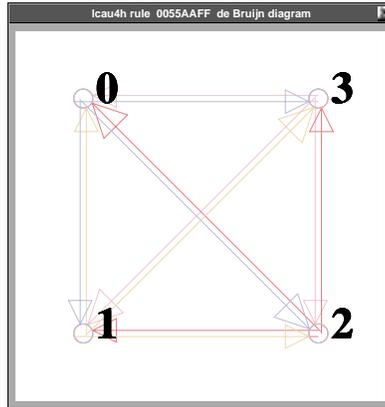
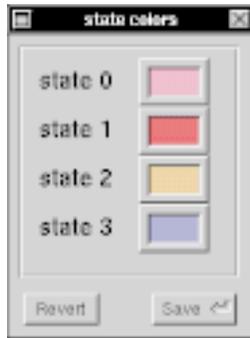
$$\begin{array}{lll}
 00111010010 & t & gen\{0\} \\
 10001011011 & t + 1 & gen\{1\} \\
 111010010000 & t + 2 & gen\{2\} \\
 \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot
 \end{array} \tag{60}$$

Como se nota en este fragmento de evolución, las vecindades evolucionan hacia atrás, en un único elemento.



Autómata (2,1) regla 85 y su regla inversa 15.

Segundo. Autómata (4,h), regla 0055AAFF.



$$\begin{array}{r}
 3333222211110000 \\
 3210321032103210 \\
 \hline
 0000111122223333 \rightarrow \text{regla0055AAFF}
 \end{array} \quad (61)$$

Matrices de conectividad A, B, C y D del diagrama de de Bruijn.

LCAU4h Pair Diagram (4x4)

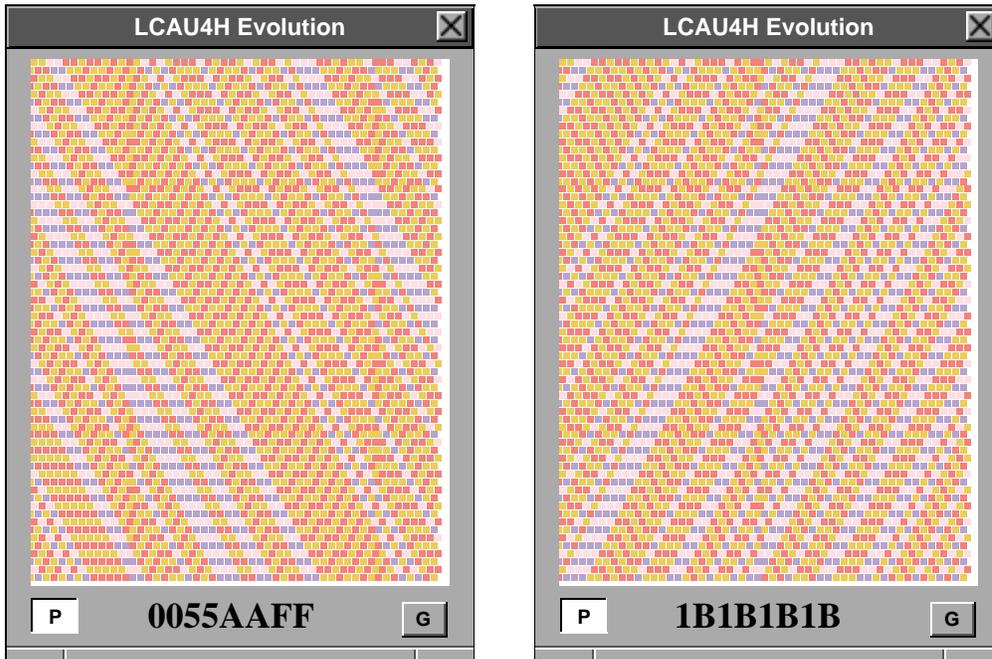
8-13-1996

(0,0)	(1,0)	(2,0)	(3,0)
(0,1)	(1,1)	(2,1)	(3,1)
(0,2)	(1,2)	(2,2)	(3,2)
(0,3)	(1,3)	(2,3)	(3,3)

print 0055AAFF draw

0302101021131323021	t	$gen\{0\}$	(64)
33031232312202010313	$t + 1$	$gen\{1\}$	
0030210102113132302	$t + 2$	$gen\{2\}$	
13303123231220201031	$t + 3$	$gen\{3\}$	
2003021010211313230	$t + 4$	$gen\{4\}$	
...	.	.	
...	.	.	
...	.	.	

En estas evoluciones también se puede observar que las vecindades evolucionan hacia atrás en un único elemento.



Autómata (4,h) regla 0055AAFF y su regla inversa 1B1B1B1B.

6 Modelo de Fredkin (de segundo orden)

Edward Fredkin [7] observó que la paridad o la operación or-exclusivo puede conducir a una réplica de patrones en los AC y de esta manera desarrolló un modelo para obtener la regla inversa de un ACLR.

Tabla del operador lógico or-exclusivo (XOR)

<i>Operando1</i>	<i>Operando2</i>	<i>Operando1 XOR Operando2</i>
<i>Verdadero</i>	<i>Verdadero</i>	<i>Falso</i>
<i>Verdadero</i>	<i>Falso</i>	<i>Verdadero</i>
<i>Falso</i>	<i>Verdadero</i>	<i>Verdadero</i>
<i>Falso</i>	<i>Falso</i>	<i>Falso</i>

(65)

6.1 Regla de dos generaciones.

Se propone una regla de evolución que toma dos generaciones de células. Sea

$$x^{(t)}_i \quad (66)$$

que representa el estado de la i -ésima célula en la generación t , entonces si:

$$x_i^{t+1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \quad (67)$$

sería la regla de evolución de algún autómata ordinario (2,1), la nueva regla se denota como:

$$x_i^{t+1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t), \dots \quad (68)$$

$$x_i^{t-1} = \Phi(x_i^{t-1}, x_{i-1}^t, x_i^t, x_{i+1}^t)$$

y toma dos generaciones, pero se puede invertir,

$$x_i^{t-1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t), \dots \quad (69)$$

$$x_i^{t+1} = \Phi^{-1}(x_i^{t+1}, x_{i-1}^t, x_i^t, x_{i+1}^t)$$

utilizando las propiedades algebraicas del or-exclusivo.

De esta forma puede obtenerse una regla de evolución inversa a la regla original de un ACLR, cabe señalar que cualquier otra función invertible de dos variables puede ser usada en lugar del or-exclusivo.

6.2 Obtención de la regla inversa para un autómata (2,1)

De un autómata (2,1), regla 85; se obtendrá su regla inversa utilizando la tabla del operador or-exclusivo $[a+b-2(ab)]$, base 2.

$$\begin{array}{r} 1100 \\ 1010 \\ \hline 0110 \end{array} \quad (70)$$

Regla de Evolución.

$$\begin{array}{r} 11110000 \\ 11001100 \\ 10101010 \\ \hline 01010101 \end{array} \rightarrow \text{regla85} \quad (71)$$

Evolución de la regla 85 tomando una configuración inicial aleatoria.

$$\begin{array}{lll} 00111010010 & t & \text{gen}\{0\} \\ 10001011011 & t + 1 & \text{gen}\{1\} \\ 11101001000 & t + 2 & \text{gen}\{2\} \end{array} \quad (72)$$

empleando el or-exclusivo se construirá la regla auxiliar,

$$\begin{array}{ll} 0 & \\ \otimes & \text{elemento } 0 \text{ en el tiempo } t, \text{ bajo el or-exclusivo} \\ 100 & \text{con la vecindad } 100 \text{ en el tiempo } t + 1, \text{ evoluciona en} \\ \downarrow & 1 \text{ en el tiempo } t + 2, \text{ por lo tanto, la vecindad } 100 \\ 1 & \text{evoluciona en } 1. \end{array} \quad (73)$$

Regla auxiliar.

$$\begin{array}{r}
11110000 \\
11001100 \\
10101010 \\
\hline
01011010 \rightarrow \text{regla90}
\end{array} \quad (74)$$

La regla 90 evoluciona hacia atrás y hacia adelante con el or-exclusivo para este autómata celular, la regla inversa se obtendrá con la regla 90 y el or-exclusivo.

$$\begin{array}{r}
100 \quad (\text{vecindad } x_{i-1}, x_i, x_{i+1} \text{ en el tiempo } t + 1) \\
\downarrow \\
1 \quad (\text{evolución bajo la regla 90}) \\
\otimes \\
1 \quad (\text{elemento } x_i \text{ en el tiempo } t + 2) \\
\downarrow \\
0 \quad (\text{evolución } x_i \text{ en el tiempo } t).
\end{array} \quad (75)$$

Regla Inversa.

$$\begin{array}{r}
11110000 \\
11001100 \\
10101010 \\
\hline
00001111 \rightarrow \text{regla15}
\end{array} \quad (76)$$

Ilustrando la evolución.

$$\text{regla 85} \left\{ \begin{array}{l} t+2 \quad 11101001000 \quad t \\ t+1 \quad 10001011011 \quad t+1 \\ t \quad 00111010010 \quad t+2 \end{array} \right\} \text{regla15} \quad (77)$$

De esta manera se obtuvo de la regla 85, su regla inversa, es decir, la regla 15.

6.3 Obtención de la regla inversa para un autómata (4,h)

De un autómata (4,h), regla 05AF0FA5; se obtendrá su regla inversa utilizando la tabla del operador or-exclusivo $[a+b-2(ab)]$, base 4.

$$\begin{array}{r}
3221110000 \\
3323213210 \\
\hline
0102303210
\end{array} \tag{78}$$

La tabla se construye como:

$$1.01 \Rightarrow 0 + 1 - 2(01) = 1 - 0 = 1$$

$$2.12 \Rightarrow 1 + 2 - 2(12) = 3 - 0 = 3(\text{ya que el 4 decimales 0 en base 4})$$

$$3.23 \Rightarrow 2 + 3 - 2(23) = 1 - 0 = 1(\text{5 en decimales 1 y 6 es 2 en base 4}) \tag{79}$$

Regla de evolución.

$$\begin{array}{r}
3333222211110000 \\
3210321032103210 \\
\hline
0011223300332211 \rightarrow \text{regla } 05AF0FA5
\end{array} \tag{80}$$

Evolución de la regla 05AF0FA5 tomando una configuración inicial aleatoria.

$$\begin{array}{lll}
001020311213223033121101 & t & \text{gen}\{0\} \\
3113232130300221201033313 & t + 1 & \text{gen}\{1\} \\
130020301211223031320010 & t + 2 & \text{gen}\{2\}
\end{array} \tag{81}$$

Regla auxiliar.

$$\begin{array}{r}
3333222211110000 \\
3210321032103210 \\
\hline
1010000010320022 \rightarrow \text{regla } 44004E0A
\end{array} \tag{82}$$

La regla 44004E0A evoluciona hacia atrás y hacia adelante con el or-exclusivo para este autómata celular.

Regla inversa.

$$\begin{array}{r}
3333222211110000 \\
3210321032103210 \\
\hline
1001223310012233 \rightarrow \text{regla } 41AF41AF
\end{array} \tag{83}$$

Ilustrando la evolución.

$$\left\{ \begin{array}{lll} t+2 & 130020301211223031320010 & t \\ t+1 & 3113232130300221201033313 & t+1 \\ t & 001020311213223033121101 & t+2 \end{array} \right\} \quad (84)$$

Con lo cual se obtuvo de la regla 05AF0FA5 su regla inversa, es decir, la regla 41AF41AF.

7 Uso de las Matrices de Conectividad del Diagrama de de Bruijn.

El uso de estas matrices es una alternativa para obtener la regla inversa de un autómata (4,h). Dentro de las matrices de conectividad del diagrama de de Bruijn para autómatas con $\mathbf{r=h}$, se observa una propiedad que indica si es reversible. Esta es que el número de nodos del diagrama de de Bruijn, es igual a la máxima suma por renglones multiplicada por la máxima suma por columnas. Tomemos las matrices de conectividad del diagrama de de Bruijn de un autómata (4,h), regla 05AF0FA5.

$$\begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (85)$$

Por ejemplo, sea

$$a_{ij} \quad (86)$$

cualquier elemento de la matriz A, entonces tenemos que:

$$\sum_{i=0}^3 a_{i2} \cdot \sum_{j=0}^3 a_{1j} = 4 \quad (87)$$

para el caso de un ACLR(4,h) donde la suma máxima por renglón y por columna se denominan índices de Welch. Utilizando las matrices de conectividad anteriormente dadas, se obtendrá su regla inversa para este autómata.

Algoritmo para obtener la regla inversa

a) Tomar una vecindad para la cual se desee conocer el elemento con el que evoluciona en el pasado.

b) Del primer elemento de la vecindad, tomar su matriz de conectividad y de ahí obtener los elementos cuya suma por columna sea diferente de cero.

c) Del segundo elemento de la vecindad, tomar su matriz de conectividad y de ahí obtener los elementos cuya suma por renglón sean diferente de cero.

d) De los elementos obtenidos en b) y c), tomar el elemento que se intersecta.

e) De la vecindad tomada en a), especificar su evolución en el pasado como el elemento en común obtenido en d).

f) Repetir los pasos anteriores para cada una de las vecindades.

Esta característica sólo se da para los ACLR con radio de vecindad h, tomando nuevamente el autómata (4,h), regla 05AF0FA5.

Regla de evolución :

$$\begin{array}{r} 3333222211110000 \\ \hline 3210321032103210 \\ 0011223300332211 \end{array} \rightarrow \text{regla05AF0FA5} \quad (88)$$

Algoritmo:

a) Se toma la vecindad 01.

b) Matriz del elemento 0.

$$A = \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{bmatrix} \quad (89)$$

Columnas con suma de elementos diferentes de cero: 2 y 3.

c) Matriz del elemento 1.

$$B = \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{bmatrix} \quad (90)$$

Renglones con suma de elementos diferentes de cero: 0 y 3.

$$d) (2, 3) \cap (0, 3) = 3, \text{ elemento en común.} \quad (91)$$

$$e) \text{Entonces 01 evoluciona en el pasado con el estado 3.} \quad (92)$$

De este modo las matrices de conectividad de la regla inversa quedarán de la siguiente manera:

$$\begin{matrix} A & B & C & D \\ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \end{matrix} \quad (93)$$

de estas matrices de conectividad se obtiene la regla inversa.

Regla inversa.

$$\begin{array}{r} 3333222211110000 \\ 3210321032103210 \\ \hline 1001223310012233 \end{array} \rightarrow \text{regla41AF41AF} \quad (94)$$

Justamente la misma regla que se había obtenido por el modelo de Fredkin para este autómata.

8 Construcción de un Autómata Celular Lineal Reversible (4,h)

Como hemos visto, las matrices de conectividad de un ACLR con radio de vecindad \mathbf{h} cumplen que su máxima suma por renglón multiplicada por su máxima suma por columna, índices de Welch, tiene que ser igual al número de nodos del diagrama de de Bruijn, de esta forma uno puede proponer matrices de conectividad que cumplen con esta propiedad.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (95)$$

La máxima suma por renglones es 4 y la máxima suma por columna es 1, entonces el producto de éstos es igual a 4, que es exactamente igual al número de nodos del diagrama de de Bruijn.

Regla del autómata.

$$\begin{array}{r} 3333222211110000 \\ 3210321032103210 \\ \hline 1111222233330000 \end{array} \rightarrow \text{regla55AAF00} \quad (96)$$

De este modo se construirá por medio de las matrices originales, las matrices de conectividad de la regla inversa.

$$\begin{array}{cccc}
A & B & C & D \\
\begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & 1 & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \end{bmatrix}
\end{array} \quad (97)$$

Con estas matrices de conectividad se obtiene su regla inversa del autómata.

$$\begin{array}{r}
3333222211110000 \\
3210321032103210 \\
\hline
1230123012301230 \rightarrow \text{regla6C6C6C6C}
\end{array} \quad (98)$$

9 Algoritmo de Hillman

En 1991, David Hillman [9] propone un algoritmo para encontrar ACLR con \mathbf{k} estados y un radio de vecindad \mathbf{r} .

Hillman hace notar que la reversibilidad no sólo depende de la regla de evolución, sino también del ancho del anillo del autómata, es decir, el número de células en cada configuración; así, un autómata puede ser reversible si su ancho es par o non o múltiplo de algún número.

Algoritmo de Hillman

Sea \mathbf{C} una cadena de \mathbf{x} células en el tiempo \mathbf{t} , de este modo, \mathbf{C} está completamente determinado por la cadena de células \mathbf{P} de longitud $\mathbf{x}+\mathbf{d}-\mathbf{1}$, en el tiempo $\mathbf{t}-\mathbf{1}$, donde \mathbf{d} es el tamaño de la vecindad del autómata.

a) Si

$$\alpha \quad (99)$$

es la configuración en \mathbf{C} , sea

$$P_\alpha \quad (100)$$

el conjunto de configuraciones las cuales de acuerdo a la regla de evolución, generan

$$\alpha. \tag{101}$$

b) Sea

$$B_j \tag{102}$$

la j -ésima configuración presente en

$$P_\alpha, \tag{103}$$

tomando ésto, formar una tabla

$$Q_\alpha \tag{104}$$

la cual, su j -ésimo elemento será igual a

$$(b_\alpha(j), e_\alpha(j)), \tag{105}$$

donde

$$b_\alpha(j) \tag{106}$$

es el conjunto de las primeras d-1 células en

$$B_j \text{ y } e_\alpha(j) \tag{107}$$

es el conjunto de las últimas d-1 células en

$$B_j. \tag{108}$$

c) Por último, formar

$$R_w \tag{109}$$

como el conjunto de todas las tablas

$$Q_\alpha \tag{110}$$

posibles dado un ancho w. Si w es mayor que 1 ,

$$R_w \tag{111}$$

se obtiene de la concatenación de

$$R_1 \text{ y } R_{w-1}; \tag{112}$$

para cada elemento

$$Q_\alpha \text{ de } R_{w-1} \text{ y } Q_\beta \text{ de } R_1, \quad (113)$$

uno obtiene un elemento

$$Q_\tau \text{ de } R_w \quad (114)$$

después de comparar cada entrada

$$Q_\alpha(i) \text{ concadaentrada } Q_\beta(j); \quad (115)$$

$$\text{si}(e_\alpha(i) = b_\beta(j), \quad (116)$$

añadir a

$$Q_\tau \quad (117)$$

una entrada m tal que

$$b_\tau(m) = b_\alpha(i) \text{ y } e_\tau(m) = e_\beta(j). \quad (118)$$

El proceso recursivo empieza calculando

$$R_1 \quad (119)$$

directamente de la regla de evolución.

d) Los elementos

$$R_w \text{ contiene tablas } Q_i \quad (120)$$

con las listas de las primeras y las últimas $d-1$ células de cada ancestro de una cadena de longitud w dada. Las primeras $d-1$ células deben ser iguales a las últimas $d-1$ células si existe un ancestro, ya que su ancho será de $w+d-1$. Así, el j -ésimo ancestro existirá si y solo si

$$b_i(j) = e_i(j); \quad (121)$$

por lo tanto, si el autómata es reversible, cada tabla

$$Q_i \quad (122)$$

debe tener una y solo una entrada

$$j \text{ talque } b_i(j) = e_i(j). \quad (123)$$

e) El algoritmo para hasta un ancho w especificado. Como w puede llegar a ser muy grande, el cómputo de

$$R_w \tag{124}$$

puede no ser práctico, sin embargo, existen dos propiedades importantes que agilizan grandemente el cálculo.

1.- Si las tablas en

$$R_i \tag{125}$$

no tienen todas el mismo número de elementos, el autómata no es reversible si su ancho es mayor o igual que $i+d-1$.

2.- Un elemento

$$R_z \text{ será igual a } R_x \tag{126}$$

con x menor que z ; por lo que el cálculo de las

$$R_w \tag{127}$$

se minimiza.

9.1 Ejemplos para autómatas (2,1)

Primero. Autómata (2,1), regla 161.

Regla de evolución.

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 \hline
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \rightarrow \text{regla161}
 \end{array} \tag{128}$$

Construyendo la tabla.

	Q_0	Q_1	
R_1	00	01	
	01	10	00 00
	01	11	10 01
	10	00	11 11
	11	10	

(129)

este ACL no es reversible ya que el estado 1 tiene varios ancestros y las tablas \mathbf{Q}_0 y \mathbf{Q}_1 son desiguales.

Segundo. Autómata (2,1), regla 85; la cual sabemos que es reversible.

Regla de evolución.

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 \hline
 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
 \end{array} \rightarrow \text{regla85} \tag{130}$$

Construyendo la tabla.

R_1	Q_0		Q_1		(131)
	00	01	00	00	
	01	11	01	10	
	10	01	10	00	
	11	11	11	10	

Concatenando \mathbf{R}_1 con \mathbf{R}_1 .

R_2	Q_{00}		Q_{01}		Q_{10}		Q_{11}		(132)
	00	11	00	10	00	01	00	00	
	01	11	01	10	01	01	01	00	
	10	11	10	10	10	01	10	00	
	11	11	11	10	11	01	11	00	

Concatenando \mathbf{R}_2 con \mathbf{R}_2 , las tablas son iguales que en \mathbf{R}_2 .

R_3	Q_{000}		Q_{001}		Q_{010}		Q_{011}		(133)
	00	11	00	10	00	01	00	00	
	01	11	01	10	01	01	01	00	
	10	11	10	10	10	01	10	00	
	11	11	11	10	11	01	11	00	
	Q_{100}		Q_{101}		Q_{110}		Q_{111}		
	<i>igualque</i>		<i>igualque</i>		<i>igualque</i>		<i>igualque</i>		
	Q_{000}		Q_{001}		Q_{010}		Q_{011}		

$$R_4 : \text{igualque } R_3 \text{ y a s\u00eden adelante.} \quad (134)$$

De esta forma vemos que la regla 85 para un ACL(2,1) es reversible para cualquier ancho.

9.2 Ejemplo para un aut\u00f3mata (4,h)

Estudiamos ahora un ACL(4,h), con una regla de evoluci\u00f3n 05AF05EB.

Regla de evoluci\u00f3n.

$$\begin{array}{r} 3333222211110000 \\ 3210321032103210 \\ \hline 0011223300113223 \end{array} \rightarrow \text{regla } 05AF05EB \quad (135)$$

Construyendo las tablas.

	Q_0	Q_1	Q_2	Q_3	
R_1	3 3	3 1	2 3	2 1	(136)
	3 2	3 0	2 2	2 0	
	1 3	1 1	0 2	0 3	
	1 2	1 0	0 1	0 0	

Concatenando $\mathbf{R}_-(1)$ con $\mathbf{R}_-(1)$ tenemos:

R_2	Q_{00}	Q_{01}	Q_{02}	Q_{03}	(137)
	3 3 3 2 1 3 1 2	3 1 3 0 1 1 1 0	<i>igualque</i> Q_{00}	<i>igualque</i> Q_{01}	
	Q_{10}	Q_{11}	Q_{12}	Q_{13}	
	<i>igualque</i> Q_{00}	<i>igualque</i> Q_{01}	3 2 3 1 1 2 1 1	3 3 3 0 1 3 1 0	
	Q_{20}	Q_{21}	Q_{22}	Q_{23}	
	2 3 2 2 0 3 0 2	2 1 2 0 0 1 0 0	<i>igualque</i> Q_{20}	<i>igualque</i> Q_{21}	
	Q_{30}	Q_{31}	Q_{32}	Q_{33}	
	<i>igualque</i> Q_{20}	<i>igualque</i> Q_{21}	2 2 2 1 0 2 0 1	2 3 2 0 0 3 0 0	

$R_-(3)$ se construye de manera análoga.

Según el algoritmo, el ACL parece ser reversible, sin embargo, tomemos las siguientes vecindades y sus evoluciones.

$$\begin{array}{cccc}
 0 & 1 & 0 & 2 & 3 & 1 \\
 & 2 & 1 & & 2 & 1
 \end{array} \quad (138)$$

Se observa que la vecindad 21 evoluciona en el pasado con 1 y con 3, lo que es imposible si el autómata es reversible. Lo que sucede aquí no es que el algoritmo esté equivocado, sino que es algo muy sutil, el autómata si es reversible, pero la regla de evolución inversa no tiene un radio de vecindad \mathbf{h} , más bien maneja un radio de vecindad $\mathbf{1}$, es decir, 1 vecino a cada lado.

Por consiguiente, el tamaño de vecindad de la regla inversa es 3; tomemos las matrices de conectividad del autómata (4,h) con la regla 05AF05EB.

$$\begin{array}{cccc}
 A & B & C & D \\
 \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{bmatrix} & \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} & \begin{bmatrix} 1 & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \\
 & & & (139)
 \end{array}$$

Con ellas encontraremos la evolución hacia atrás para cada una de las posibles vecindades de tamaño 3 de la regla inversa, utilizando una extensión del algoritmo de matrices de conectividad.

a) Se toma la vecindad 221.

b) Matriz del elemento 2.

$$C = \begin{bmatrix} \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (140)$$

Columnas con suma de elementos diferentes de cero: 1, 2 y 3.

c) Matriz del elemento 2.

$$C = \begin{bmatrix} \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (141)$$

Renglones con suma de elementos diferentes de cero: 0 y 2.

$$d)(\mathbf{1}, \mathbf{2}, \mathbf{3}) \cap (\mathbf{0}, \mathbf{2}) = 2, \text{ elemento en común.} \quad (142)$$

e) Matriz del elemento 2.

$$C = \begin{bmatrix} \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad (143)$$

Columnas con suma de elementos diferentes de cero: 1, 2 y 3.

f) Matriz del elemento 1.

$$B = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \end{bmatrix} \quad (144)$$

Renglones con suma de elementos diferentes de cero: 1 y 3.

g) Sabemos que la vecindad 22 evoluciona en el pasado con 2, y que la vecindad 21 evoluciona en el pasado con 1 ó 3.

$$\begin{array}{cccc} & 2 & 1\text{ó}3 & \\ 2 & 2 & & 1 \end{array} \quad (145)$$

Sin embargo, las vecindades 21 ó 23 deben evolucionar en 2; revisando las matrices de conectividad, tenemos que la vecindad que cumple este mapeo es 23, por lo tanto, 3 es el elemento seleccionado.

De esta manera, obtenemos que:

$$\begin{array}{ccc} 2 & 3 & \\ 2 & 2 & 1 \end{array} \quad \begin{array}{l} \text{tiempo } t-1 \\ \text{tiempo } t \end{array} \quad (146)$$

por lo tanto, se concluye que la vecindad 221 evoluciona en el pasado con 3.

Finalmente construimos su regla de evolución inversa ((4,1), cuatro estados un vecino a cada lado) para este autómata, representando sus vecindades en forma matricial con los nodos de su diagrama de de Bruijn.

00	3	3	2	2
01	1	1	0	0
02	3	3	2	2
03	1	1	0	0	.	.	.
10	3	3	2	2
11	1	1	0	0
12	1	1	2	2
13	3	3	0	0	.	.	.
20	3	3	2	2
21	1	1	0	0
22	3	3	2	2
23	1	1	0	0	.	.	.
30	3	3	2	2
31	1	1	0	0
32	1	1	2	2
33	3	3	0	0	.	.	.

(147)

Para obtener la evolución inversa se debe recorrer ésta medio vecino a la derecha para conservar la paridad con la regla original, ilustrando la evolución de la regla 4h, cuya inversa es la regla 4,1.

$$\left\{ \begin{array}{l} t + 2 \\ t + 1 \\ t \end{array} \quad \begin{array}{l} 03201130310231221311 \\ 120131032112210230311 \\ 01303120110231221211 \end{array} \quad \begin{array}{l} t \\ t + 1 \\ t + 2 \end{array} \right\} \quad (148)$$

Cabe señalar que recurriendo al mismo procedimiento se obtuvo otra regla ((4,t), cuatro estados, 1.5 vecinos a cada lado) que también es inversa a la original, sus vecindades, representadas por los nodos del diagrama de de Bruijn evolucionan hacia atrás como muestra su regla

(en forma matricial).

00	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
01	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
02	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
03	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
10	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
11	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
13	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0
20	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
21	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
22	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
23	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
30	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
31	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
32	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
33	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0

(149)

10 Reversibilidad y Producto Cartesiano

El producto cartesiano se puede utilizar, para especificar reglas de evolución de un ACL, de radio de vecindad \mathbf{h} , por ejemplo la regla de evolución de un autómata (4,h), se pueden formar a través de un producto cartesiano de dos autómatas (2,h) con sus respectivas reglas. Este producto cartesiano se puede extender para obtener autómatas (6,h) (producto de un (2,h) por (3,h) y viceversa), (8,h), (10,h) y hasta donde se puedan calcular (si tiene tiempo) más autómatas con vecindad \mathbf{h} .

Si tomamos un autómata (2,h), regla 10; y observemos como evoluciona con una configuración inicial aleatoria.

$$\begin{array}{r}
 1100 \\
 1010 \\
 \hline
 1010 \rightarrow \text{regla10}
 \end{array}
 \tag{150}$$

$$\begin{array}{llll}
111001101000 & t & gen\{0\} \\
1110011010001 & t + 1 & gen\{1\} \\
110011010001 & t + 2 & gen\{2\} \\
1100110100011 & t + 3 & gen\{3\} \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot
\end{array} \tag{151}$$

Como podemos ver la evolución de este autómata muestra un corrimiento a la izquierda que es muy característico en un ACLR, por medio de sus matrices de conectividad encontramos que su regla inversa es la regla 12, el cual muestra también un corrimiento a la derecha en su evolución (el contrario de la regla 10).

$$\begin{array}{r}
1100 \\
1010 \\
\hline
1100
\end{array} \rightarrow regla12 \tag{152}$$

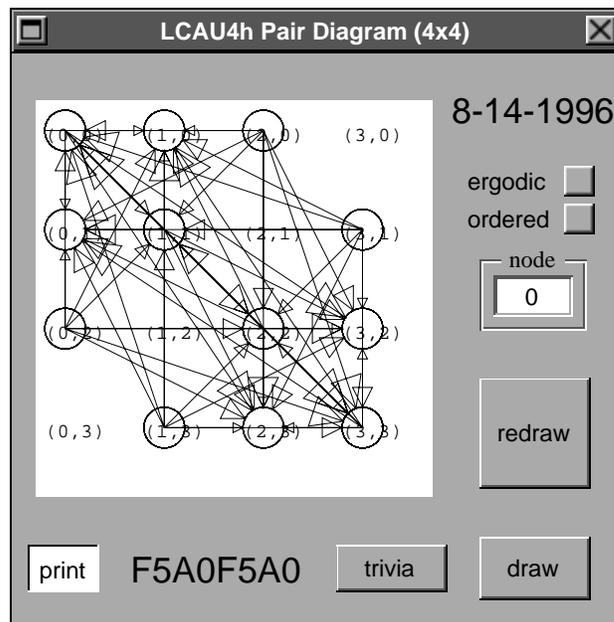
$$\begin{array}{llll}
1100110100011 & t & gen\{0\} \\
110011010001 & t + 1 & gen\{1\} \\
1110011010001 & t + 2 & gen\{2\} \\
111001101000 & t + 3 & gen\{3\} \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot
\end{array} \tag{153}$$

Obtengamos el producto cartesiano de estos dos autómatas.

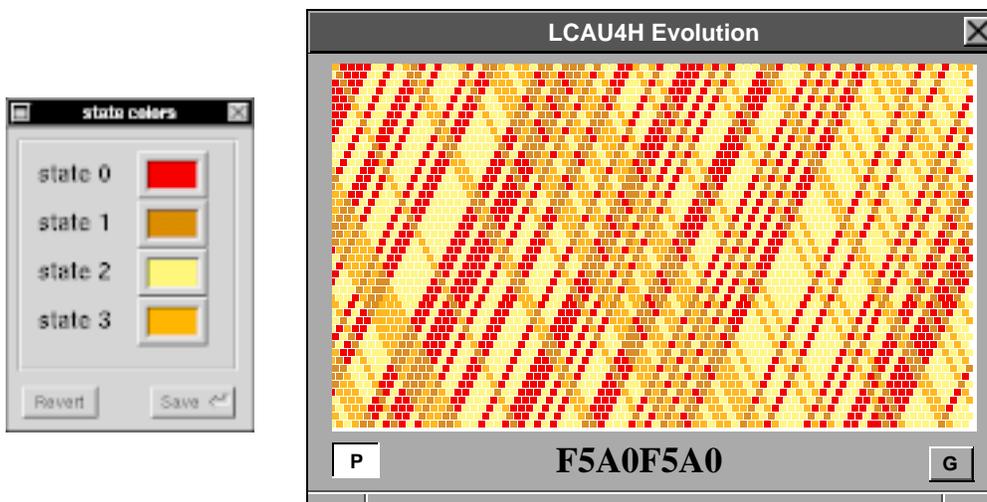
$$\begin{array}{c}
10 \\
\left[\begin{array}{cc} 0 & 1 \\ 0 & 1 \end{array} \right]
\end{array} \cdot \begin{array}{c}
12 \\
\left[\begin{array}{cc} 0 & 0 \\ 1 & 1 \end{array} \right]
\end{array} = \begin{array}{c}
F5A0F5A0 \\
\left[\begin{array}{cccc} 0 & 0 & 2 & 2 \\ 1 & 1 & 3 & 3 \\ 0 & 0 & 2 & 2 \\ 1 & 1 & 3 & 3 \end{array} \right]
\end{array} \tag{154}$$

El producto cartesiano se tomó multiplicando cada elemento de la primera matriz por todos los elementos de la segunda matriz, donde

<i>producto</i>		<i>resultado</i>	
00	→	0	
01	→	1	(155)
10	→	2	
11	→	3	



Observemos el diagrama de Parejas del autómata (4,h), regla F5A0F5A0; no existen ciclos fuera de la diagonal y por lo tanto, puede ser reversible, analicemos una pequeña evolución de este autómata.

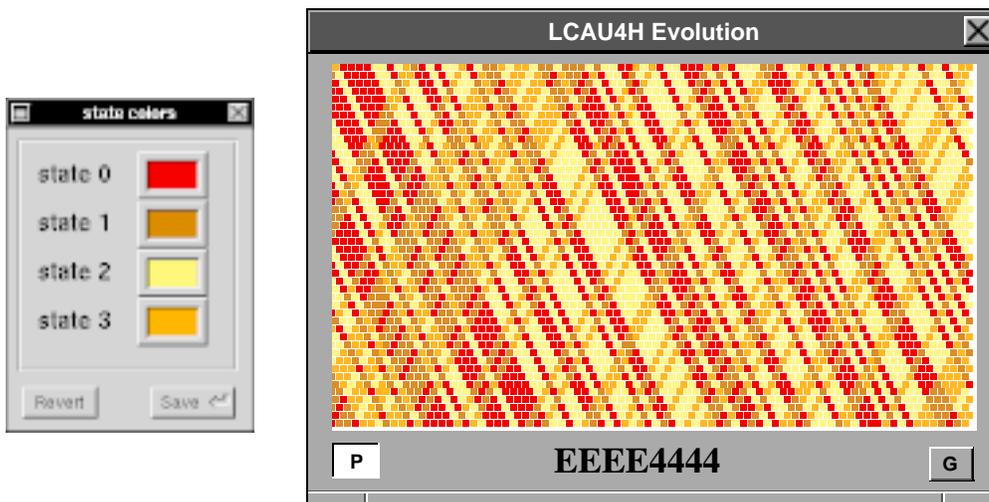


Tenemos que esta evolución es simplemente una mezcla de los corrimientos de los autómatas (2,h), regla 10 y regla 12. Calculando por medio de las matrices de conectividad del diagrama de de Bruijn del autómata (4,h), regla F5A0F5A0; se tiene que la regla inversa es

EEEE4444, ahora bien cambiando el orden de multiplicación del producto cartesiano anterior el resultado es el siguiente:

$$\begin{matrix} & 12 & & 10 & & EEEE4444 \\ \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} & \bullet & \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} & = & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 2 & 3 & 2 & 3 \\ 2 & 3 & 2 & 3 \end{bmatrix} & (156) \end{matrix}$$

el resultado del producto cartesiano es la regla inversa del autómata (4,h), regla EEEE4444, analizemos una pequeña evolución de esta regla inversa.



Vemos que el corrimiento de esta evolución es la contraria de la regla F5A0F5A0 y también la combinación de corrimientos de los autómatas (2,h), regla 12 y regla 10.

Se pueden concluir dos cosas:

- El producto cartesiano de dos autómatas reversibles (2,h), genera un autómata reversible (4,h).

- Si el producto cartesiano anterior es de un reversible por su regla inversa, entonces, se puede obtener el reversible $(4,h)$ y su regla inversa cambiando solamente el orden del producto cartesiano.

Bibliografía

- [1] Stephen Wolfram *Ed. Theory and Applications of Cellular Automata* World Scientific Press Singapore (1986).
- [2] Harold V. McIntosh *Linear Cellular Automata via de Bruijn Diagrams* Próximo a publicarse.
- [3] Harold V. McIntosh *Linear Cellular Automata* Universidad Autónoma de Puebla (1990).
- [4] Harold V. McIntosh *Reversible Cellular Automata* Physica D999 100-130 (1990).
- [5] E. Moore *Machine Models of Self-Reproduction* Proc. Symp. Appl. Math. 14 17-33 (1962).
- [6] Tommaso Toffoli and Norman Margolus *Cellular Automata Machines* The MIT Press Cambridge Massachussetts (1987).
- [7] Howard Gutowitz *Cellular Automata: Theory and Experiment* North-Holland (1990).
- [8] Taylor L. Booth *Sequential Machines and Automata Theory*, John Wiley and Sons, Inc. (1967).
- [9] David Hillman *The Structure of Reversible One-dimensional Cellular Automata* Physica D52 277-292 (1991).