

Grados de Reversibilidad
en
Autómatas Celulares Lineales

Genaro Juárez Martínez
e-mail: genaro@sparcomp.cs.cinvestav.mx
Escuela Nacional de Estudios Profesionales Acatlán
Universidad Nacional Autónoma de México
D. F., México

Marzo 30, 1998

Introducción

Un autómata celular, es un sistema dinámico discreto que evoluciona en iteraciones a través de una regla determinística, tal como un sistema dinámico; las variables del sistema cambian como una función de sus valores actuales. Pueden ser vistos como un proceso de cómputo en paralelo, donde los datos son la configuración inicial. Otra aproximación es que un autómata celular es un “universo lógico... con su propia física local”. Tal como un universo de autómata celular; no obstante su construcción matemática, son capaces de soportar comportamientos complejos [13].

El estudio de los Autómatas Celulares se remonta desde los años 50's, época de grandes aportaciones en materia de conjuntos, expresiones regulares, bases de datos, autómatas, neuronas, cibernética, máquinas finitas, computación, entre otros. Los autómatas celulares también fueron canalizados en un principio bajo fines bélicos. Los autómatas celulares pueden ser representados en dos dimensiones y en una dimensión, a los autómatas celulares en dos dimensiones se le encontró un juego ecológico llamado *Life* que hizo muy popular esta teoría gracias a su presentación que resulto muy llamativa. La teoría de autómatas celulares como tal a tenido altibajos a través de la historia, sin embargo se puede decir que ha pasado por tres etapas de alto grado de estudio, la primera de ellas es la etapa de John von Neumann, la segunda es de John Horton Conway y la última es de Stephen Wolfram. Aunque se han dado importantes aportaciones en estos períodos de bajo interés que han contribuido de manera relevante al estudio de dicha teoría, aunque algunas veces de manera indirecta.

Se han encontrado algunas aplicaciones dentro del campo de la biología, la química, la física, las matemáticas, la cristalografía, la termodinámica y cierta relación con la teoría del caos, la teoría fractal, vida artificial, simulación de procesos y ciencias de la computación.

El estudio de los autómatas celulares dentro del campo de la computación es uno de los principales focos de interés para muchos investigadores. Desde los inicios de la teoría de autómata celular se pretendía realizar la construcción de una máquina capaz de soportar comportamientos complejos. Se han hecho estudios sobre si una máquina de Turing [22] puede contener un autómata celular o viceversa, si esto es cierto cuales son las limitaciones y los alcances de cada una de estas máquinas. Por otra parte la representación del cómputo en paralelo es ideal para su representación en autómatas celulares.

El tema de como se manifiesta la *reversibilidad* en los autómatas celulares en los últimos años a sido de notable interés. Ya que hay algunas personas que tratan de buscar aplicaciones bajo estas características, tal como encriptadores de datos, almacenamiento masivo de información, por mencionar algunos. Sin embargo la teoría como tal aún se encuentra en fase de formalización matemática para posteriormente obtener una explicación real y concisa de tales fenómenos.

El presente trabajo titulado “*Grados de Reversibilidad en Autómatas Celulares Lineales*”, se encuentra dividido en tres partes principales: la primera consiste en el

fundamento de la teoría de autómatas celular donde se desglosa un poco de historia y se explica como esta constituida la estructura de los autómatas celulares en una dimensión, la segunda parte lo constituye el problema de explicar como a través de las estructuras particulares se originan los comportamientos globales que a su vez determinan el comportamiento de los mismos autómatas, la tercera y última parte la constituyen los resultados obtenidos a través de sus propiedades globales para autómatas celulares no reversibles pero que contienen grados de reversibilidad para configuraciones de cierto tamaño.

A través de todo el escrito se hizo uso del software NXLCAU¹ desarrollado por el Dr. Harold V. McIntosh del Instituto de Ciencias² de la Universidad Autónoma de Puebla. Estos programas fueron codificados en el lenguaje de programación orientado a objetos **C-Object** en el sistema operativo **NeXTSTEP** y son de dominio público. También existe una versión del software (LCAU) compatible para el sistema operativo Ms-DOS.

El escrito fue realizado en \LaTeX un derivado del formato \TeX sobre el sistema operativo NeXTSTEP v3.3 y compatible con la versión de OPENSTEP, en las instalaciones del CINVESTAV³.

El trabajo fue realizado en común acuerdo de los dos departamentos sobre un proyecto de investigación sobre “*Autómatas Celulares Lineales Reversibles*”, donde se han dado resultados importantes y buenos avances.

¹ Web site <http://www.cs.cinvestav.mx/mcintosh/celular.html>

² Departamento de Aplicación de Microcomputadoras, 49 Poniente 1102, Puebla.

³ Centro de Investigación y Estudios Avanzados del IPN. Departamento de Ingeniería Eléctrica, Sección de Computación.

Contenido

1	Marco Teórico	1
1.1	Autómata celular lineal	4
1.2	Estructura de los autómatas celulares	4
1.3	Clasificación Wolfram	5
2	Teoría de Gráficas en Autómatas Celulares	9
2.1	Topología	9
2.2	Diagrama de de Bruijn	10
2.3	Diagrama de subconjuntos	12
2.4	Diagrama de parejas	14
2.5	Autómata celular binario	15
3	Mapeo Local induce Mapeo Global	19
3.1	Que es un mapeo?	19
3.2	Mapeo local	21
3.3	Mapeo global	24
3.3.1	Diagrama de transiciones	25
3.4	Ancestros	27
3.5	Clasificación de los mapeos	27
3.5.1	Jardín del Edén implica mapeo inyectivo	27
3.5.2	Ancestros múltiples implica mapeo suryectivo	29
3.5.3	Ancestros únicos implica mapeo biyectivo	31
4	Grados de Reversibilidad	33
4.1	Autómatas celulares reversibles	33
4.2	Algunos algoritmos existentes	35
4.2.1	Algoritmo de Fredkin	35
4.2.2	Algoritmo de Hillman	36
4.3	Autómatas celulares con grados de reversibilidad	36
4.3.1	Matriz de de Bruijn simbólica	37
4.3.2	Configuraciones de tamaño par e impar	40
4.3.3	Autómata (2,1) regla 45	40
4.3.4	Autómata (4,h) regla 0056B9EF	45
4.3.5	Autómata (4,h) regla 016ED4BB	52
4.3.6	Algoritmo	55

5 Conclusiones	57
A Árboles Topológicos (4,h)	61
A.1 Regla 0056B9EF	62
A.2 Regla 0056F9AF	63
A.3 Regla 005BADF6	64
A.4 Regla 005BBDE6	65
A.5 Regla 005BEDB6	66
A.6 Regla 00669FF9	67
A.7 Regla 00679EF9	68
A.8 Regla 0552F8AF	69
A.9 Regla 0158AFF6	72
A.10 Regla 0156F8AF	73
A.11 Regla 016ED4BB	75
A.12 Regla 055BACF2	77
A.13 Regla 0196E87F	79
A.14 Regla 05AF14EB	82
A.15 Regla 0156E8BF	84
A.16 Regla 0156B8EF	86
A.17 Regla 0154BAEF	87
A.18 Regla 016AB4DF	90
A.19 Regla 0168D6BF	91
A.20 Regla 05936CFA	94
A.21 Regla 019BEC76	96
A.22 Regla 1BB1E44E	98

Capítulo 1

Marco Teórico

Johannes von Neumann [1] de origen húngaro era un verdadero genio en su tiempo, el encuentro una importante relación entre las computadoras y los organismos naturales. Relaciono ambos sistemas encontrando características muy interesantes que dieron origen a una teoría que pudiera representar dichas propiedades. De esta manera von Neumann llama a su teoría, la *Teoría de Autómata Celular*. La teoría de autómata celular esta formada por un conjunto coherente de conceptos y principios enfocados a la estructura y organización de ambos sistemas, la programación y el control de tales sistemas. Von Neumann trabajó en el diseño de computadoras, programandolas para que reconocieran la lógica matemática y a su vez tuvieran un importante rendimiento en la nueva teoría de autómata celular, logrando de esta manera que la teoría de autómata celular fuera multidisciplinaria. A grandes rasgos la naturaleza general de la teoría de autómata celular es su estructura, sus materiales, algunos de sus problemas, algunas de sus aplicaciones y la forma de su matemática; además von Neumann comenzo un estudio comparativo de autómata artificial y natural. Finalmente formuló un par de preguntas básicas en la teoría de autómata celular: ¿Como puedo construir sistemas confiables desde componentes no confiables?, ¿Qué tipo de organización lógica es suficiente para que un autómata sea capaz de auto-reproducirse?.

La idea de von Neumann, es que la matemática de la teoría de autómata celular debe empezar con la lógica matemática y moverse hacia un análisis, tal como la teoría de probabilidad y la termodinámica y cuando esto sea desarrollado, deberemos entender la gran complejidad del autómata celular, en particular el del sistema nervioso humano. El razonamiento matemático es ejecutado por el sistema nervioso humano y el lenguaje natural en que el razonamiento matemático toma su lugar es análogo para el lenguaje natural en el cómputo de la máquina. Esto es totalmente posible, en que la teoría de autómata celular debe ejecutar lógica y nuestros conceptos fundamentales de matemáticas. Ahora bien, la lógica es la base de las matemáticas; por lo tanto, si von Neumann dice que esto es verdad, la teoría de autómata celular debe moverse en forma circular, es decir, empezar en los fundamentos de las matemáticas y terminar ahí mismo.

Antes del estudio formal acerca de los autómatas celulares, surgieron otras teorías muy importantes que influyeron en los estudios de von Neumann, por ejem-

plo los estudios de Warren S. McCulloch y Walter Pitts sobre neuronas (1943); una interesante abstracción matemática sobre expresiones regulares por S. C. Kleene (1951); los fundamentos generales de la cibernética introducida por Norbert Wiener (1948); por otra parte Henri Poincaré enfatizó en la cantidad de aspectos de la mecánica clásica en términos de estabilidad, propiedades ergódicas y la recurrencia de orbitas, tales temas actualmente constituyen una medida para estudios teóricos, topológicos y de dinámica simbólica, todo esto como resultado de las ideas de Poincaré; la dinámica simbólica elaborada por George Birkhoff y por último Gustav Adolf Hedlund [2] (1969) realizó un amplio trabajo que contiene abundantes resultados aplicables a la teoría de autómatas celulares, aunque este enfoque es totalmente diferente y la relación entre los dos conceptos no son por lo general bien apreciados [7].

Los autómatas celulares son una rama de la teoría de autómatas, la diferencia que existe entre estos es que los autómatas son manejados por señales de entrada que a su vez producen señales de salida. Los autómatas celulares por el contrario disfrutan de toda una simetría; sin embargo los autómatas celulares usan los estados de sus vecinos como señales de entrada y generalmente no producen señales de salida. La teoría de autómatas por si misma es bastante antigua, por un lado podemos ver que los autómatas son mecanismos capaces de ejecutar movimientos complejos y la teoría como tal comienza propiamente con la abstracción neurofisiológica de McCulloch y Pitts. Su fineza dentro de la teoría de expresiones regulares por Kleene, constituye uno de varios puntos de vista, donde término por incluir semigrupos (o monoides) de mapeos a un conjunto dentro de si mismo, que es justo la teoría de la gramática. La teoría de semigrupos es más compleja que la teoría de grupos, cuya clasificación sería una de las realizaciones matemáticas más importantes en los últimos tiempos y a su vez esta clasificación sería de gran utilidad para el estudio de los autómatas celulares. En las décadas de los 50's, 60's y 70's, se dieron grandes aportaciones en el estudio sobre autómatas, lenguajes y temas relacionados, pero a los autómatas celulares en particular no se les daría mucha atención.

El conocimiento público de los autómatas celulares puede ser atribuido enormemente a John Horton Conway. Interesado en buscar una simple configuración como la de von Neumann dio como resultado una serie de configuraciones para un autómata en dos dimensiones, que mostraban de manera muy ilustrativa y atractiva sus comportamientos a través del tiempo dentro de una lattice. Algunos de sus resultados fueron presentados en 1970 como un juego ecológico llamado *Life*, por Martin Gardner [3]. El juego de *Life*, sería el resultado de una regla muy particular, para un autómata celular en dos dimensiones (dos estados por célula), donde las vecindades están formadas por una célula central y ocho más alrededor de esta, como se observa en la Figura 1.1. Conway examinó diferentes reglas para poder encontrar la regla que dio origen a su juego. Algunos de sus resultados más importantes fueron obtenidos en el Laboratorio de Inteligencia Artificial (MIT), con la ayuda de las facilidades gráficas de su computadora PDP-6.

$x_{i-1,j-1}$	$x_{i,j-1}$	$x_{i+1,j-1}$
$x_{i-1,j}$	$x_{i,j}$	$x_{i+1,j}$
$x_{i-1,j+1}$	$x_{i,j+1}$	$x_{i+1,j+1}$

Figura 1.1: Vecindad de Moore.

En los últimos tiempos, la teoría de autómatas celulares recibió un considerable interés a través de las investigaciones de Stephen Wolfram [4], que emprendió un estudio computacional a través de las propiedades de los autómatas celulares en una dimensión; guiado por algunos conceptos de la dinámica no lineal y técnicas estadísticas. La idea del autómata celular en una dimensión es simple y su evolución a través del tiempo es ideal para una presentación en dos dimensiones. Para empezar hay una célula en una cierta región cada una en un punto dado, representados de diferentes formas llamados estados. Por conveniencia, estos estados son usualmente enumerados con pequeños valores, comenzando con cero hacia adelante. Para los propósitos de la teoría de autómatas celulares la naturaleza de los estados no importa ya que pueden ser representados de cualquier manera, lo que sí es muy importante es su relación con los otros estados y el comportamiento de estos cambiando a través del tiempo y su medio ambiente.

Wolfram fue uno de los primeros en comparar las historias de evoluciones de un gran número de diferentes reglas para autómatas binarios, intentando su clasificación a través de su comportamiento en grandes evoluciones, surgiendo de esta manera las clases de Wolfram.

1.1 Autómata celular lineal

Para construir un autómata celular, crearemos un arreglo lineal finito de células donde supondremos la existencia de que todas las células tienen el mismo número de estados similares y a su vez se verán afectadas por una regla de transición; el segundo elemento para definir un autómata celular, es la relación entre las células o más bien los tipos de vecindades que llegan a formar estas. Una vecindad debe estar formada por células o vecinos predefinidos ubicados alrededor de la célula central, que se denota como r vecinos de radio, resultando $2r + 1$ vecinos totales que en conjunto forman una vecindad dada. Como podemos ver, si la cadena es finita seguramente no se tendrán las mismas vecindades a través de todo el arreglo; pero esto puede ser enfocado de manera diferente, el arreglo puede ser tratado como un anillo, es decir, cerrando el arreglo en forma circular y de esta forma conservaremos la uniformidad en todas las vecindades, además con esto se establecen sus condiciones a la frontera. Por lo tanto, es conveniente trabajar con vecindades simétricas cada una centrada en su propia célula y no con vecindades irregulares.

De esta forma surge la notación Wolfram (k, r) para un autómata celular lineal, donde k es el número de estados por cada célula y r el número de vecinos que forman la vecindad. Por último falta definir la regla de transición, en donde todos los estados de las células se ven afectados por la misma en cada generación y asumimos la misma regla para todas las vecindades.

1.2 Estructura de los autómatas celulares

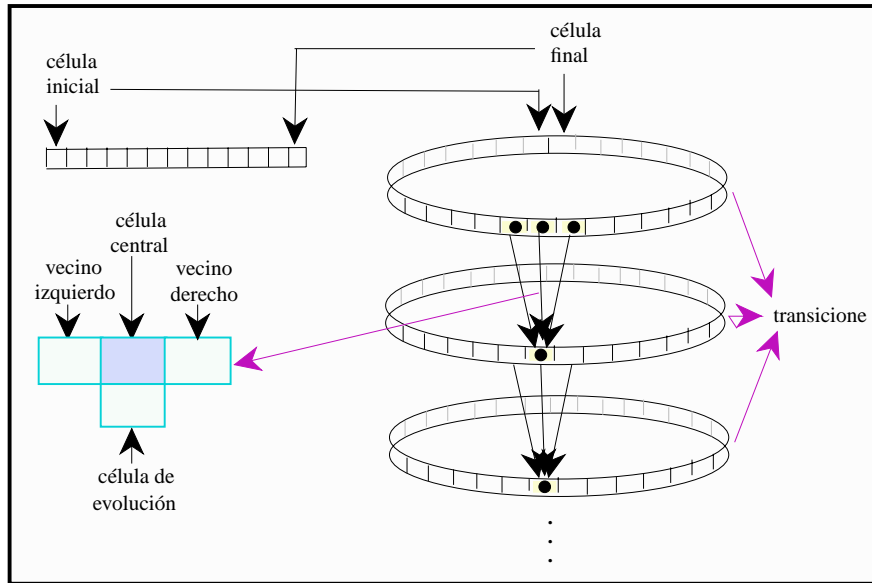


Figura 1.2: Estructura lineal de los autómatas celulares.

Wolfram por su parte describe algunos términos básicos para determinar su clasificación de autómatas de acuerdo a su regla de evolución. Una tabla de transiciones describe la función de mapeos en cada vecindad y esta función se denota como φ . Para un autómata celular su vecindad contiene k^{2r+1} vecindades por regla, que nos da un total de $k^{k^{2r+1}}$ reglas de evolución. Por otra parte uno puede tomar estados globales de la tabla de transiciones, es decir, el patrón de valores a través de los renglones en el arreglo es el estado global del autómata celular en un tiempo dado. Por lo tanto el autómata celular evoluciona a través de una sucesión de estados globales (su trayectoria), además la iteración de su actualización global produce la función de transición.

Dos tipos de autómata celular pueden ser distinguidos, ambos determinísticos: el caso más general puede ser descrito teniendo grados de construcción desordenados, donde el diagrama dado y/o la función en cada célula pueden ser diferentes, como un ejemplo existen las redes de trabajo de funciones booleanas de Walker y las redes de trabajo booleanas de Kauffman. El autómata celular de construcción ordenada es un caso especial, donde el diagrama y la función son el mismo sobre todo el arreglo. En ocasiones, el orden puede ser confinado a una vecindad dentro de una zona ininterrumpida de células, típicamente centradas sobre la célula identificadora.

1.3 Clasificación Wolfram

Para describir la evolución de un autómata celular lineal arbitrario de una configuración inicial aleatoria, observemos el diagrama de evoluciones donde en algunos casos de manera rápida, llega a ser claro que cada regla y tipos de autómatas son diferentes. Por su parte Wolfram [4] presentó una útil clasificación dentro de cuatro clases.

- *Clase I.* Comportamientos uniformes.



Figura 1.3: Autómata $(3, h)$ regla 17793.

Como puede verse en el diagrama de evoluciones, un sólo estado es el que domina a los demás a través del tiempo y este comportamiento es siempre el mismo con cualquier configuración aleatoria.

- *Clase II.* Comportamientos cíclicos aislados.

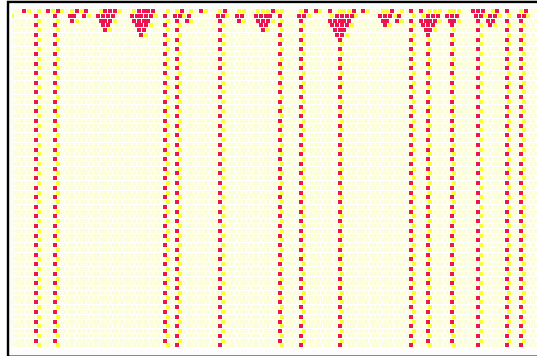


Figura 1.4: Autómata $(3, h)$ regla 14252.

En el diagrama de evoluciones podemos observar como dos estados se comportan de manera repetitiva bajo una vecindad dada. Este comportamiento siempre es el mismo a través del tiempo y cualquier configuración aleatoria.

- *Clase III.* Comportamientos caóticos.

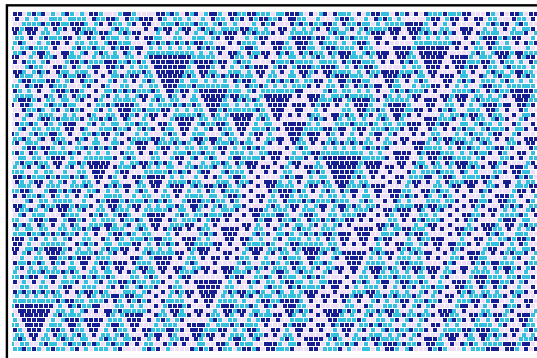


Figura 1.5: Autómata $(3, h)$ regla 11509.

En el diagrama de evoluciones podemos observar como es que el patrón que se muestra a través del tiempo es totalmente inestable, es decir, no podemos determinar a simple vista que tipo de configuración se presentará de generación en generación.

- *Clase IV.* Comportamientos complejos.

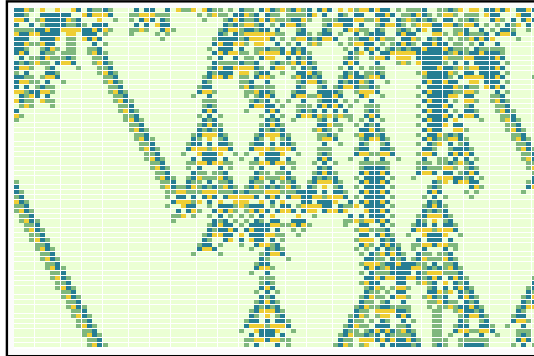


Figura 1.6: Autómata (4, 1) regla C30E39E50E39E59739E5975EE5975E78.

Sin duda la clase más estudiada, esta clase se caracteriza por ser una combinación de la clase dos con la clase tres, es decir, dentro del diagrama de evoluciones podemos observar regiones uniformes originadas por un estado y regiones caóticas originadas por los otros tres estados restantes.

Su interés sería particularmente por los autómatas clase IV. Al parecer estas clases son identificadas por reglas cuyos diagramas de de Bruijn contienen ciertos ciclos, donde estos pueden ser detectados con facilidad en períodos cortos. Esto es seleccionar una regla específica y trabajar su tabla de períodos (tiempo de repetición) y ciclos (espacio de repetición), en donde un renglón dado muestra el número de ciclos de un período dado, en anillos de longitud dada por las columnas.

Para obtener estos ciclos en los autómatas celulares, se tomarán longitudes de arreglos finitos mayores o iguales a dos, estos arreglos son los estados globales. Primero se tomará como semilla un estado global aleatorio, posteriormente se obtendrán sus evoluciones hasta encontrar un estado global que se repita, esto quiere decir que hemos entrado el ciclo (espacio de repetición) de estos mapeos globales. De esta manera cuantificamos todos los estados globales que pueden resultar de la longitud empleada. Finalmente trazamos una gráfica cuyas ligas son determinadas por su evolución. Una buena utilidad puede ser seleccionar un anillo inicial y examinar generaciones sucesivas de su evolución, la nueva generación puede ser totalmente distinta o el mismo punto podemos buscarlo generaciones atrás. Para un anillo finito el numero total de estados globales que se pueden obtener es k^l posibles combinaciones diferentes de estados en un anillo de longitud l .

Otro uso del diagrama de transiciones es obtener ancestros de una cadena dada, una simple aplicación es encontrar los ancestros de cadenas uniformes donde cada liga representa la evolución de la célula central en una vecindad. Todas las ligas que evolucionan en cero determinan las cadenas que deben evolucionar en cero, inversamente los ciclos evolucionan en cero y determinan las reglas para que tal evolución sea posible. De manera menos complicada es la determinación de las cadenas estáticas o mejor conocida como naturaleza muerta, esto significa vecindades que evolucionan sobre un valor constante, es decir las vecindades para la célula

central evolucionan sobre si misma.

Históricamente los diagramas de de Bruijn serían creados para resolver el problema de buscar todas las distintas secuencias de ciertos símbolos. Esta idea puede ser aplicada para un diagrama de períodos, cuestionándose si todas las posibles secuencias de células pueden aparecer como posibles evoluciones. Por lo tanto el diagrama de períodos es una restricción del diagrama de de Bruijn, además puede ser dudoso en el sentido de que ésto no puede ser y a su vez confirma la existencia de los estados conocidos como “Jardín del Edén”. Estas configuraciones son cadenas de células que sólo aparecen como una configuración inicial en las evoluciones de los autómatas por que ellos no tienen ancestros y no pueden surgir durante el curso de la evolución desde otros estados.

Examinar un camino en un diagrama puede ser un proceso tedioso, los diagramas de subconjuntos de Edward F. Moore [5], proveen un camino para sistematizar este tipo de estudio. Este es un nuevo diagrama cuyos nodos son subconjuntos de estados, donde cada subconjunto es ligado para la unión de estados y esta liga que los une se deriva de la evolución de cada uno de los estados que forman el subconjunto con otro subconjunto dado. Usualmente uno empieza desde el conjunto máximo, suponiendo que no importa donde comienza un camino y continuarlo si es posible hasta llegar al subconjunto mínimo, aunque también es verdad que aquí no existen caminos con las mismas características dentro del diagrama original. Un diagrama de subconjuntos más elaborado retiene todos los detalles que pueden ser contruidos desde los caminos originales con todas sus múltiplicidades que pueden ser extraídas.

Capítulo 2

Teoría de Gráficas en Autómatas Celulares

Desde un punto de vista puramente abstracto, una gráfica es una pareja (N, L) que consiste de N nodos y L ligas. Los nodos son justo un conjunto regularmente formado por enteros positivos de 0 hasta $n - 1$ y las ligas son un subconjunto de $N * N$.

Los nodos son proyectados como puntos frecuentemente etiquetados por los miembros del conjunto N , las ligas son representadas por flechas uniendo los nodos, la flecha corre del nodo i hasta el nodo j si $(i, j) \in L$. Nótese que no es necesario que ambos (i, j) y (j, i) correspondan a L , ni que (i, i) corresponda o no corresponda. Este es un problema clásico en teoría de gráficas para saber si una gráfica puede ser dibujada en dos dimensiones además de que sus ligas no se deben mezclar.

2.1 Topología

Las estructuras de células y organismos vivos son organizados ampliamente en un camino que puede ser descrito como topología, caracterizado para especificar sus posiciones como patrones de conectividad [12].

La matriz topológica $M_{i,j}$ esta definida por:

$$M_{i,j} = \begin{cases} 1 & (i, j) \in L \\ 0 & c.o.c \end{cases} .$$

La aplicación más importante de la matriz topológica consiste en su capacidad para describir caminos entre nodos conteniendo múltiples ligas de acuerdo a su evolución, por lo tanto los elementos de tales matrices son enteros y son útiles para contar el número total de caminos, pero no clasifican los caminos en diferentes categorías. Los elementos de la diagonal de la matriz cuentan los ciclos; la traza produce el número total de ciclos y cuentan cada uno de ellos, una vez por cada nodo que ellos continen.

La teoría de gráficas como tal juega un papel muy importante dentro de la teoría de los autómatas celulares. Uno consiste en describir la evolución del autómata y el

otro consiste en relacionar las propiedades locales con las propiedades globales. La evolución es descrita por reglas locales mapeando vecindades en su estado subsiguiente y esto es porque las vecindades sucesivas se solapan, además es importante en el sentido de que deben ser capaces de tomar el solape explicando cuando relacionar el comportamiento de células sucesivas para algunas otras. Un autómata celular se presta para análisis rigurosos, ya que la simetría inherente dentro de su lattice del autómata individual (las células) y suponiendo cambios individuales de estados; dependen exclusivamente de los estados de un número limitado de vecinos.

A primera vista el extenso solape entre vecindades de diferentes células parece confundir el problema, además esto sería buscar ciertas ideas usando teoría de registro de corrimientos y varias técnicas más, además de que la información cuida del solape de una manera muy adecuada.

2.2 Diagrama de de Bruijn

La teoría de registro de corrimientos es una disciplina basada en el tratamiento de secuencias solapando. Los nodos del diagrama de de Bruijn son secuencias de símbolos de algún alfabeto, justo como una expresión regular ya que ellos pueden ser secuencias de nodos de una gráfica específica, las ligas del diagrama describen como tales secuencias pueden solaparse. Por consecuencia diferentes grados de solape conducen a diferentes diagramas.

Cuando los enteros son enteros consecutivos ellos pueden ser tratados como elementos de un anillo o quizás dentro de un campo finito y nos da la facilidad de discutir sus propiedades aritméticas o algebraicas, además de hacer varias elecciones valiosas. La matriz topológica del diagrama de de Bruijn se representa como:

$$M_{i,j} = \begin{cases} 1 & \text{si } j = \begin{cases} ki \\ ki + 1 \\ \vdots \\ ki + k - 1 \end{cases} \pmod{k^2r} \\ 0 & \text{c.o.c} \end{cases}$$

A través del diagrama de de Bruijn las gráficas pueden representar configuraciones o clases de configuraciones en los autómatas celulares, las ligas del diagrama están naturalmente asociadas con las vecindades de un autómata usando los mismos símbolos que asocian las ligas en un paso de evolución. Si hay alguna razón para discriminar entre vecindades, la misma discriminación puede definir un subdiagrama del diagrama original. El diagrama de de Bruijn genérico [8] para un autómata (2, 1), tiene cuatro nodos que corresponden a cuatro vecindades parciales de dos células con ocho ligas representando todas las vecindades de tres células.

La matriz topológica de un diagrama de de Bruijn es totalmente regular aunque es asimétrica y se diferencia de las matrices circulantes, matrices tridiagonales u otra forma especializada que haya sido extensamente analizada dentro de la literatura.

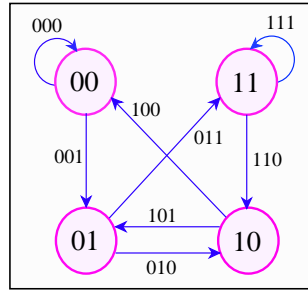


Figura 2.1: Diagrama de de Bruijn genérico para el autómata (2,1).

Las matrices de de Bruijn $D_{k,s}$ ó D_s son caracterizadas por k , el número de símbolos en que las secuencias pueden ser formadas y s el número de estados que es la longitud de la secuencia.

$$D_{k,s} = \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \end{bmatrix}$$

En estos términos deben existir k ligas de entrada en cada nodo y k ligas de salida en cada nodo, en total deben existir k^s nodos con k^{s+1} ligas uniendolos, correspondiendo a las secuencias de longitud $s + 1$.

Un buen ejemplo del uso de subdiagramas es el proceso conocido como *naturaleza muerta*, ya que estas son configuraciones de células que no cambian durante la evolución. El diagrama para la naturaleza muerta puede ser extraído del diagrama genérico removiendo todas las ligas que no conservan su célula central durante la evolución. Los ciclos en el subdiagrama contienen ligas que definen secuencias que son invariantes en la evolución, de esta manera toda la naturaleza muerta para el autómata es determinada automáticamente. En general, cualquier combinación booleana de las células de la vecindad y la célula de evolución pueden determinar el subdiagrama.

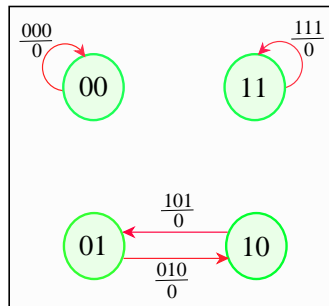


Figura 2.2: Subdiagrama de de Bruijn para un autómata (2,1).

Se hace resaltar de manera muy especial la importancia del uso de los diagramas de de Bruijn, ya que el significado del mismo diagrama no sólo representa el mapeo

de las vecindades por su estado de evolución, si no que a través de estos diagramas y su respectiva matriz de evolución dan pie a la realización de otros diagramas tal como el diagrama de subconjuntos y el diagrama de parejas. Por otra parte su matriz de conectividad es de vital importancia para obtener una estimación del número de ancestros del mismo autómata.

Con esto se hace notar que la importancia de los diagramas de de Bruijn, es que estos contienen en gran parte toda la información relevante que se quiera obtener del autómata en estudio.

2.3 Diagrama de subconjuntos

El diagrama de de Bruijn sirve como guía en la descripción de secuencias de ciertas cadenas de símbolos, para saber si algún otro conjunto de objetos puede producir la misma secuencia. Un ejemplo trivial puede ser tratar de recuperar los símbolos originales desde algún elemento central de cada cadena, cuando el propósito en sí de usar vecindades en los autómatas celulares es obtener la secuencia de evolución de las mismas células. Por el mismo camino, la multiplicación de la matriz de de Bruijn garantiza que los elementos del producto hagan referencia a ciertas secuencias de evolución que corresponden al diagrama de de Bruijn. Esto es de gran utilidad cuando se desea calcular el número promedio de ancestros que pueden tener determinadas secuencias.

El diagrama de subconjuntos tiene 2^{k^2r} nodos, si todas las configuraciones de cierta longitud poseen ancestros entonces todas las configuraciones con extensiones tanto a la izquierda como a la derecha con la misma equivalencia deben tener ancestros. Si este no es el caso, entonces ellos describen las configuraciones conocidas como Jardín del Edén y no es más que la descripción del camino principal que va del conjunto máximo al conjunto mínimo dentro del diagrama de subconjuntos.

Los nodos dentro del diagrama de subconjuntos están formados por la combinación de todos los subconjuntos que se pueden formar a partir del número de estados que conforman el diagrama de de Bruijn, por ejemplo para un autómata $(2, 1)$ tenemos cuatro fracciones de estados en el diagrama de de Bruijn $\{a\}$, $\{b\}$, $\{c\}$ y $\{d\}$, y de ahí podemos formar todos los subconjuntos posibles: $\{a, b, c, d\}$, $\{a, b, c\}$, $\{a, b, d\}$, $\{a, c, d\}$, $\{b, d, c\}$, $\{a, b\}$, $\{a, c\}$, $\{a, d\}$, $\{b, c\}$, $\{b, d\}$, $\{d, c\}$, $\{d\}$, $\{c\}$, $\{b\}$, $\{a\}$ y $\{\}$. Dentro de estos subconjuntos se pueden distinguir de manera clara las cuatro clases unitarias que se forman, la integración del conjunto vacío garantiza que todos los subconjuntos tengan al menos una imagen, aunque esta no exista en el diagrama original. Para determinar el tipo de unión que existen entre los subconjuntos se debe revisar el estado en que evoluciona cada estado y de esta manera saber hacia que estados (subconjunto que lo formen) se puede conectar; de esta manera se construye la siguiente relación para la regla 30.

nodo	liga con 0	liga con 1
a	a	b
b	ϕ	c, d
c	b	a
d	c, d	ϕ

Tabla 2.1: Relación entre estados.

De esta manera podemos construir la matriz de subconjuntos donde la dimensión de la misma esta definida por el número de nodos del diagrama y a su vez puede ser particionada a través de sus clases unitarias. Podemos observar que el elemento fundamental para definir alguna función es que cada argumento tiene justo una imagen. Las ligas del diagrama definen una función cuando existe una liga de una clase hacia un nodo; pero si esta condición no se cumple, se debe a que existe una combinación de múltiples ligas a una sola liga entre subconjuntos que implica la existencia de más de una entrada al nodo a través del mismo estado o diferentes estados.

Aunque las ligas de una gráfica como tal no definen una función las ligas entre subconjuntos son siempre funcionales y esto se debe por la inclusión del conjunto vacío definido para toda la gráfica. Cada clase de ligas definen una función, sea Σ_0 ó Σ_1 . El diagrama de subconjuntos describe la unión de $\Sigma_0 \cup \Sigma_1$, que por si misma no es funcional.

Sean a y b nodos, S un subconjunto y $|S|$ la cardinalidad de S ; entonces el diagrama de subconjuntos esta definido por,

$$\sum_i(S) = \begin{cases} \phi & S = \phi \\ \{b \mid liga_i(a, b)\} & S = \{a\}. \\ \bigcup_{a \in S} \Sigma_i(a) & |S| > 1 \end{cases}$$

De aquí se desprenden tres propiedades importantes:

1. Si existe una cadena principal del subconjunto máximo al conjunto mínimo, entonces debe existir una cadena similar que salga de algún subconjunto menor al conjunto vacío. Por el contrario si las clases unitarias carecen de ligas al conjunto vacío, aquí no existen configuraciones Jardín del Edén.
2. Existe un cierto residuo del diagrama de de Bruijn, en el sentido de que dado un origen y un destino, siempre debe haber un subconjunto conteniendo el destino accesible y otro subconjunto conteniendo el origen, además el destino puede tener nodos adicionales.
3. El diagrama de subconjuntos puede no estar conectado, si este es el caso es interesante conocer el subconjunto más grande accesible desde algún subconjunto dado, así como el subconjunto más pequeño.

Cabe recalcar que el diagrama de subconjuntos proporciona información muy valiosa con respecto a ciertas secuencias dentro del autómata en estudio, el saber si el autómata contiene configuraciones Jardín del Edén, si tiene múltiples ancestros

y en un momento dado determina el comportamiento del autómata dentro del diagrama de evoluciones para casos triviales. Los ancestros pueden ser determinados a través del tipo de mapeo que manifiestan.

2.4 Diagrama de parejas

El diagrama de parejas [8], es un producto carteseano obtenido a través de las matrices de conectividad del diagrama de de Bruijn. El diagrama de parejas es útil para comparar caminos entre dos diagramas diferentes o dos caminos en el mismo diagrama. Las flechas dentro del diagrama de parejas se derivan del hecho de que para algún nodo (formado por una pareja de estados) cualesquiera de la gráfica, evolucione en parejas ordenadas en un mismo estado a otro nodo. Es decir, los índices de la matriz topológica del diagrama de parejas estan formadas por todas las parejas ordenadas que se llegan a formar del número de estados del diagrama de de Bruijn y nos da un total de k^{2r^2} nodos en el diagrama de parejas. Una gran utilidad que nos brinda el diagrama de parejas es obtener los múltiples ancestros dentro de los autómatas celulares, ya que podemos extraer secuencias que formen un ciclo dentro del diagrama y comprobar si estas secuencias por los mismos nodos tienen más de un ancestro para una vecindad dada. Con esto cabe señalar que en el diagrama de parejas se pueden obtener todos los ciclos que pueden existir dentro del autómata en estudio.

Producto carteseano. [17]

Sean A y B dos conjuntos y C el conjunto producto que representamos como $A \otimes B$ cuyos elementos van a ser los distintos pares o parejas que se puedan formar tomando todos y cada uno de los elementos de A , con todos y cada uno de los elementos de B , de esta manera si a es un elemento de A y b un elemento de B entonces,

$$A \otimes B = \{(a, b) | a \in A \wedge b \in B\}. \quad (2.1)$$

Para determinar el orden de la matriz C consideremos las matrices de conectividad por estado del diagrama de de Bruijn de orden n , A_n y $B_n \Rightarrow A_n \otimes B_n = C_{n^2}$. Por lo tanto, el diagrama como tal describe parejas ordenadas y consecuentemente la matriz de conectividad C tiene propiedades simétricas así como reflexivas, por lo que nos permite construir un diagrama desordenado en donde la distinción entre los elementos de las parejas (a, b) y (b, a) es practicamente insignificante.

2.5 Autómata celular binario

En este caso tomaremos un autómata celular con dos estados y un vecino por cada lado, es decir un autómata $(2, 1)$. La regla se representa de dos formas ya sea en notación binaria o en notación decimal. En este caso se tienen ocho vecindades de tamaño tres.

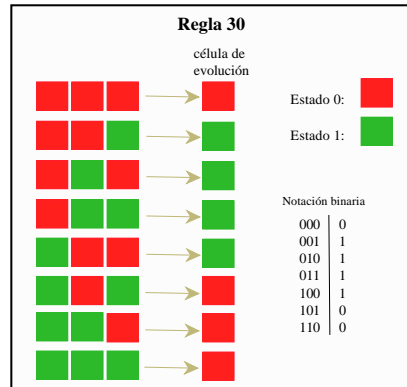


Figura 2.3: Regla de evolución del autómata $(2,1)$ regla 30.

Consecuentemente $k = 2$ y $s = 2$, tenemos $2^2 = 4$ nodos y $2^{2+1} = 8$ ligas, que a su vez se pueden identificar los nodos como el número de estados del autómata y las ligas el número de vecindades del autómata. Los estados están formados por las fracciones de vecindad: 00, 01, 10 y 11, que en notación decimal se representan como los estados: 0, 1, 2 y 3. Como se había mencionado en la sección anterior la naturaleza de los estados no es muy importante si no más bien la regla de evolución que afectan a las vecindades. Para mayor ilustración los estados se representarán a través de colores. La función de transición $\varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \rightarrow x_i^{t+1}$, indica el mapeo de la vecindad en el tiempo t al nuevo estado de evolución en el tiempo $t + 1$.

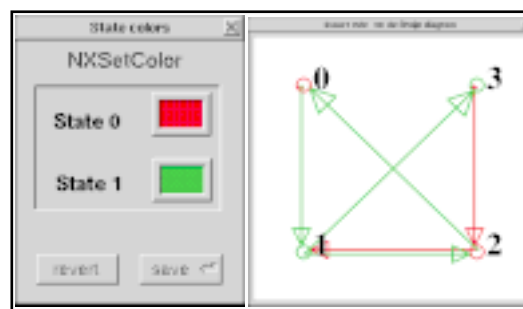


Figura 2.4: Diagrama de de Bruijn para la regla 30.

Para obtener el diagrama de subconjuntos emplearemos la tabla de relaciones de la Sección 2.3, entonces tenemos $2^{2^{2(1)}} = 16$ nodos en el diagrama de subconjuntos, que son exactamente todos los subconjuntos que se formaron en esa misma sección,

ahora bien estos subconjuntos se representarían en notación decimal como se ilustra en el siguiente cuadro:

Subconjunto	valor	liga con 0	liga con 1
0,1,2,3	15	15	15
1,2,3	14	14	13
0,1,2	7	3	15
0,1,3	11	13	14
0,2,3	13	15	3
0,1	3	1	14
0,2	5	3	3
0,3	9	13	2
1,2	6	2	13
1,3	10	12	12
2,3	12	14	1
3	8	12	ϕ
2	4	2	1
1	2	ϕ	12
0	1	2	2
ϕ	0	ϕ	ϕ

Tabla 2.2: Conectando subconjuntos.

en este cuadro podemos observar perfectamente las cuatro clases unitarias que existen dentro del diagrama empezando por el conjunto máximo hasta llegar al conjunto mínimo, las columnas tres y cuatro muestran a que subconjunto se liga el subconjunto de la primera columna y con que estado. De esta manera se obtienen todas las ligas en el diagrama de subconjuntos.

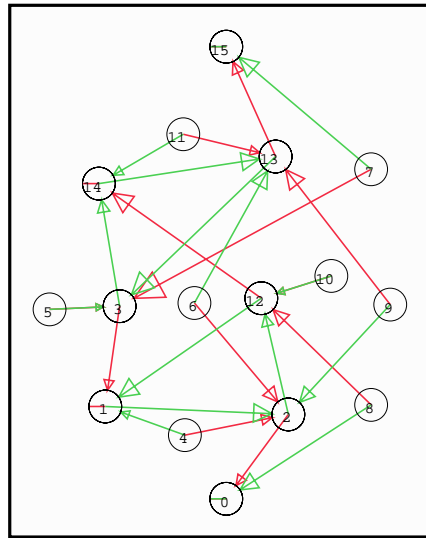


Figura 2.5: Diagrama de subconjuntos para la regla 30.

Para construir el diagrama de parejas efectuaremos el producto cartesiano de las matrices de conectividad del diagrama de de Bruijn. En este caso tendremos $2^{2(1)^2} = 16$ parejas ordenadas que a su vez son el número de nodos del diagrama de parejas. Esta matriz recibe el nombre de la matriz \mathbf{T} y sus subíndices estan formados por todas las parejas posibles (a, b) , donde a y b pertenecen a Σ ,

$$\mathbf{T} = \begin{bmatrix} 1 & . & . & . & | & . & 1 & . & . & | & . & . & . & . & | & . & . & . & . \\ . & . & . & . & | & . & . & 1 & 1 & | & . & . & . & . & | & . & . & . & . \\ . & 1 & . & . & | & 1 & . & . & . & | & . & . & . & . & | & . & . & . & . \\ . & . & 1 & 1 & | & . & . & . & . & | & . & . & . & . & | & . & . & . & . \\ \hline . & . & . & . & | & . & . & . & . & | & . & 1 & . & . & | & . & 1 & . & . \\ . & . & . & . & | & . & . & . & . & | & . & . & 1 & 1 & | & . & . & 1 & 1 \\ . & . & . & . & | & . & . & . & . & | & 1 & 1 & . & . & | & . & . & . & . \\ . & 1 & . & . & | & 1 & . & . & . & | & . & . & . & . & | & . & . & . & . \\ . & . & 1 & 1 & | & . & . & . & . & | & . & . & . & . & | & . & . & . & . \\ 1 & . & . & . & | & . & 1 & . & . & | & . & . & . & . & | & . & . & . & . \\ . & . & . & . & | & . & . & 1 & 1 & | & . & . & . & . & | & . & . & . & . \\ \hline . & . & . & . & | & . & . & . & . & | & 1 & . & . & . & | & 1 & . & . & . \\ . & . & . & . & | & . & . & . & . & | & . & . & . & . & | & . & . & . & . \\ . & . & . & . & | & . & . & . & . & | & . & 1 & . & . & | & . & 1 & . & . \\ . & . & . & . & | & . & . & . & . & | & . & . & 1 & 1 & | & . & . & 1 & 1 \end{bmatrix}$$

es claro notar que los subíndices de la primera submatriz son las parejas ordenadas $(0,0)$, $(0,1)$, $(0,2)$ y $(0,3)$. Como ejemplo, el subíndice $(0,0)$ contiene un ciclo consigo mismo ya que el estado cero evoluciona en el mismo cero y así para toda la matriz. A continuación se mostrara el diagrama de parejas completo, cabe señalar que de este diagrama se pueden producir dos diagramas más, uno de ellos puede ser el diagrama ordenado donde la simetría mostrada en el diagrama original se pierde y el otro diagrama es el que nos muestra exclusivamente los ciclos que se forman a través de ciertas secuencias dadas. Para estos diagramas los ciclos autocontenidos se representan como un radio dentro del nodo.

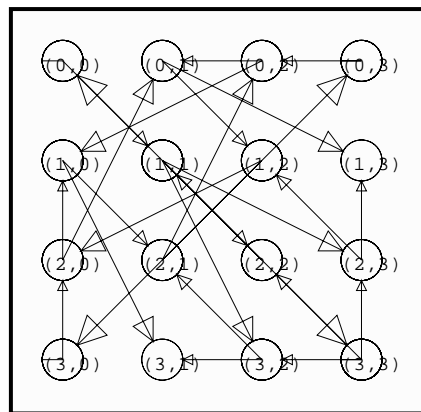


Figura 2.6: Diagrama de parejas para la regla 30.

Capítulo 3

Mapeo Local induce Mapeo Global

3.1 Que es un mapeo?

Sea $a \in A$, $b \in B$ y f es una función. Entonces si $(a, b) \in f$ decimos que f mapea a a sobre b y denotamos esto como $f(a) = b$. Por lo tanto denotamos un mapeo f del conjunto A al conjunto B por $f : A \rightarrow B$. El término mapeo se usa como un sinónimo de función [17].

El mapeo de los estados dentro del diagrama de evoluciones puede ser representado en un diagrama de *Venn*, como conjuntos. Donde el conjunto A representa las configuraciones posibles y el conjunto B los estados de evolución.

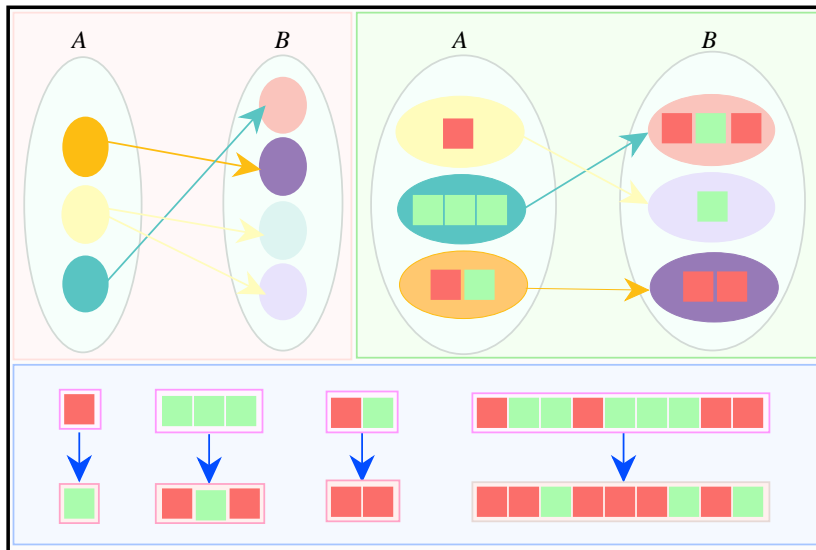


Figura 3.1: Mapeo de estados representado como conjuntos.

En la Figura 3.1 el primer diagrama de Venn ilustra el mapeo entre conjuntos de elementos no definidos, partiendo de este punto de referencia podemos representar estos elementos como configuraciones de tamaño n , mapeando sobre estados que pudiesen ser sus estados de evolución en el siguiente paso. Posteriormente presentamos diferentes tamaños de configuraciones mapeando a sus respectivos estados de evolución en la siguiente generación.

Nótese la poca importancia que se les a dado a k y a r ; esto es por razones de informalidad pues el objetivo de los diagramas es sólo identificar el mapeo de las mismas configuraciones en los diagramas de evoluciones. Posteriormente se especificarán en detalle estos tipos de mapeo de tamaño variado.

El primer detalle que vamos a tratar, es el mapeo no simétrico dentro del diagrama de evoluciones. El caso que más se ha estudiado hasta estos momentos es el autómata $(2,1)$, es decir, dos estados y un vecino por cada lado, pero recordemos que el número de estados puede crecer tal como $(3,1)$, $(4,1)$, $(5,1)$, $(6,1)$, \dots , y el radio de vecindad también $(2,2)$, $(2,3)$, $(2,4)$, \dots . Pero tenemos autómatas tales como $(2, h)$, $(2, t)$, $(2, f)$, \dots , donde $h = \frac{1}{2}$, $t = \frac{3}{2}$, $f = \frac{5}{2}$ y así sucesivamente.

Ilustrando para el caso más trivial el autómata $(2, h)$, la h nos indica medio vecino por cada lado, pero en lugar de tomar medio vecino por cada lado se toma uno completo ya sea a la izquierda o a la derecha y de esta forma conservamos la simetría dentro del diagrama de evoluciones.

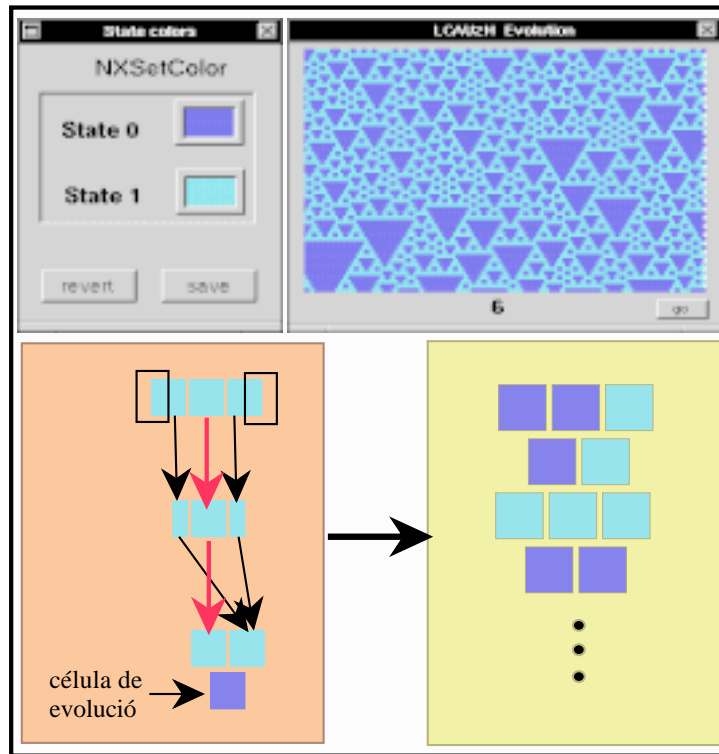


Figura 3.2: Mapeo para el caso $r = h$.

Una vez identificado como se comporta el mapeo en los autómatas celulares se analizarán los diferentes tipos de mapeos que pueden originarse, estos tipos de mapeos son muy importantes para poder analizar el comportamiento del autómata y obtener una buena clasificación de los mismos mapeos.

3.2 Mapeo local

El *mapeo local* [16] determina el comportamiento de los autómatas celulares dentro del diagrama de evoluciones en localidades específicas. Estas propiedades las podemos ver en el mismo diagrama de de Bruijn, donde la función de transición determina el mapeo local para cada una de las vecindades existentes en un estado dado. Ahora formalizaremos estos conceptos para tener una idea más clara del mapeo local.

Sea Σ un conjunto no vacío de símbolos llamado alfabeto o estados. La cardinalidad de Σ se denota como $|\Sigma|$. Una secuencia de longitud finita de símbolos en Σ es llamada una cadena sobre Σ . El conjunto de todas las cadenas sobre Σ es denotado por Σ^* . Si $n \in \mathbf{P} \Rightarrow \Sigma^n$ es el conjunto de todas las cadenas de longitud n sobre Σ . Sea c una configuración sobre Σ . El conjunto de todas las configuraciones sobre Σ es denotado por $C(\Sigma)$. El conjunto de todas las configuraciones finitas sobre Σ es denotado por $C_F(\Sigma)$.

Sea $n \geq 1$, $k \geq 1$ y el mapeo $\sigma : \Sigma^n \rightarrow \Sigma$, entonces σ es llamado un mapeo local de k -símbolos de n -longitud si $|\Sigma| = k$, para $\Sigma = \{0, 1, \dots, k-1\}$. Si esto es verdadero, entonces el conjunto de todos los mapeos locales de k -símbolos de n -longitud es denotado por L_n^k , por lo que $\sigma \in L_n^k$.

Masakazu Nasu desarrolla el mapeo local finito y no finito de un modo general y abstracto. Para comprender esta simbología analizaremos un autómata celular (2,1) con $k = 2$ y $r = 1$. Si Σ es el conjunto del alfabeto o estados, entonces el conjunto de Σ para el autómata (2,1) está representado como $\Sigma = \{0, 1\}$. Además k es el número de estados dentro del autómata y el número de estados es dos por lo que $k = 2$, entonces la cardinalidad de Σ es igual al valor de k , si esta condición se cumple entonces ahora podemos calcular el valor de n .

Como n es la longitud de las cadenas sobre Σ , este es definido como el tamaño de cada una de las vecindades, es decir, para este autómata (2,1) $n = 3$, ya que tenemos una célula central y un vecino a la izquierda y otro a la derecha tal como se había definido en la Sección 1.1, por lo tanto el tamaño de cada una de las vecindades $2r + 1$ es igual a n . Consecuentemente Σ^n denota el conjunto de todas las cadenas de longitud n , este conjunto de cadenas no son más que las mismas vecindades que pueden llegarse a formar $\{(000), (001), (010), (011), (100), (101), (110), (111)\}$. Para el caso del autómata (2,1) tenemos ocho vecindades de tamaño tres, compuestas por los estados que pertenecen a Σ .

Finalmente podemos determinar el mapeo local $\sigma : \Sigma^n \rightarrow \Sigma$ para el autómata (2,1) de k -símbolos y n -longitud y una regla dada. Dado que, si tenemos ocho vecindades diferentes de tamaño tres, entonces a cada una de ellas le corresponde un único estado de evolución o estado al que mapea cada vecindad. Esto es la función de transición para un autómata (2,1) dada por $\varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \rightarrow x_i^{t+1}$ donde las x representan los estados, i la posición donde se encuentra cada estado

dentro de la lattice y t representa el tiempo (generación o iteración) de cada una de las células en cierta configuración c , dentro del diagrama de evoluciones.

Sin embargo observemos la cantidad de mapeos que existen para cadenas de longitud n , este valor puede variar considerablemente de acuerdo al número de estados y al radio de vecindad que contenga el autómata en estudio. Por lo que la función de transición φ depende de k y r .

Si $l \in \mathbf{P} \Rightarrow c_l \in C_F(\Sigma)$, entonces c_l es una configuración finita de longitud l , por lo que se deduce que $c \in C(F)$.

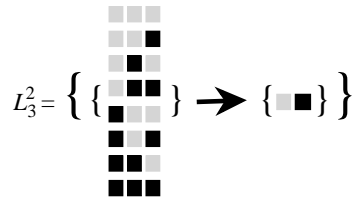
El mapeo local puede efectuarse tanto en configuraciones finitas c_l como en configuraciones no finitas c . Este es un punto importante, una configuración finita es un vector renglón que se encuentra dentro de la lattice de longitud l ; n puede estar ubicado dentro de cualquier i -ésimo sitio de la configuración, sea finita o no finita. Por lo tanto, un mapeo local puede efectuarse sobre configuraciones no finitas.

Autómata celular (2,1)

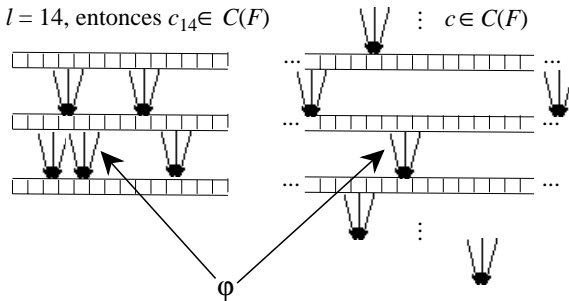
Si $\Sigma = \{0, 1\}$ y $k = 2 \Rightarrow |\Sigma| = k$. Si $r = 1$, entonces $2r + 1 = n = 3$ por lo que la función de transición para este caso es $\varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \rightarrow x_i^{t+1}$. Dada la regla 01111000 (30 en decimal) la función de transición denota la siguiente tabla para mapeos locales sobre configuraciones finitas y no finitas:

	estado 0 = ■	
	estado 1 = ■	
$\varphi(0,0,0) \rightarrow 0$		000 = ■■■
$\varphi(0,0,1) \rightarrow 1$		001 = ■■■
$\varphi(0,1,0) \rightarrow 1$		010 = ■■■
$\varphi(0,1,1) \rightarrow 1$		011 = ■■■
$\varphi(1,0,0) \rightarrow 1$	$\Sigma^n = \Sigma^3 = \left\{ \begin{array}{l} 011 = \text{■■■} \\ 100 = \text{■■■} \\ 101 = \text{■■■} \\ 110 = \text{■■■} \\ 111 = \text{■■■} \end{array} \right\}$	100 = ■■■
$\varphi(1,0,1) \rightarrow 0$		101 = ■■■
$\varphi(1,1,0) \rightarrow 0$		110 = ■■■
$\varphi(1,1,1) \rightarrow 0$		111 = ■■■

$$L_n^k = \{\Sigma^n \rightarrow \Sigma\}$$



si $l = 14$, entonces $c_{14} \in C(F)$



El diagrama de de Bruijn representa adecuadamente los mapeos locales para autómatas celulares de orden (k, r) . Para el caso del autómata celular $(2,1)$ regla 30 los mapeos estan dados por las aristas del diagrama siguiente:

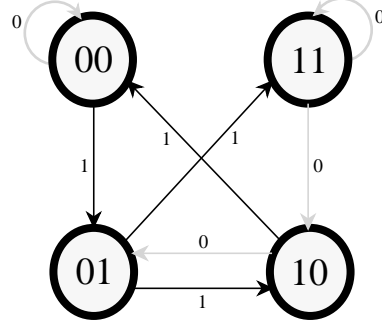


Figura 3.3: Mapeos locales para la regla 30.

Como podemos ver el mapeo local se deriva de una regla específica para un valor dado, aún así Nasu define el conjunto L_n^k como el conjunto de todos los mapeos locales. Esto quiere decir que si tenemos un autómata celular $(2,1)$ con $n = 3$, tendremos $k^{k^{2r+1}}$ mapeos locales.

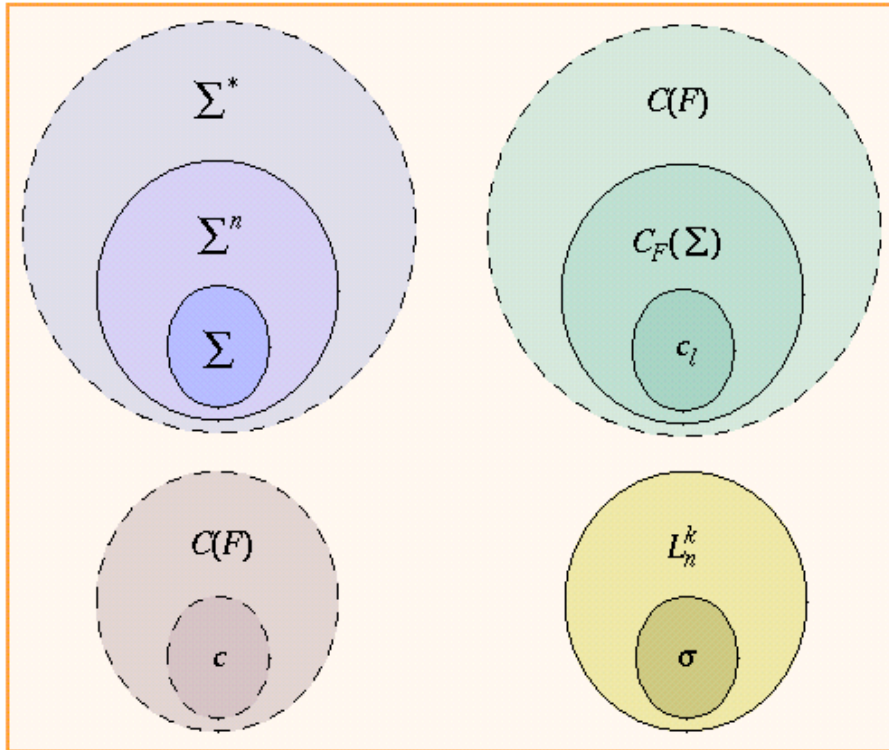


Figura 3.4: Conjuntos de estados, configuraciones y mapeos.

3.3 Mapeo global

Sea $i \in \mathbf{Z}$ y $c \in C(\Sigma)$ entonces $f_{\sigma,m}$ es llamado un *mapeo global* [16] inducido por un mapeo local σ desde un punto de vista m . Un mapeo $f : C(\Sigma) \rightarrow C(\Sigma)$ es llamado global si $f_{\sigma,m} = f$ para alguna $m \in \mathbf{Z}$. Y sea f un mapeo global de $\sigma \in L_n^k$. Si $\hat{f} : C_F(\Sigma) \rightarrow C_F(\Sigma)$ es definido como $\hat{f}(c) = f$ para toda $c \in C_F(\Sigma)$, entonces \hat{f} es llamado un mapeo global finito de σ .

Sea x una cadena entonces $\lg(x)$ denota la longitud de x . Sea Λ una secuencia de longitud cero y además es un cadena. Entonces para alguna $x \in \Sigma^*$, denotamos a \hat{x} una configuración finita de $c \in C_F(\Sigma)$ tal que si $x \neq \Lambda$ entonces $c_0 c_1 \cdots c_{\lg(x)-1} = x$ por lo tanto $c_i = 0$ para todo $i < 0$ y para todo $i \geq \lg(x)$. Pero si $x = \Lambda$ entonces $c_i = 0 \forall i \in \mathbf{Z}$.

Decimos que un mapeo local induce un mapeo global por que el mapeo global conserva todas las propiedades locales a través de todas las configuraciones. Estas propiedades globales representadas por σ son las que determinan el comportamiento del mapeo global a través del tiempo. Desarrollando de una manera más explícita ejemplificaremos las definiciones de los párrafos anteriores sobre un autómata binario.

Como definimos en la Sección 3.2, σ representa los mapeos locales de una regla en particular. Entonces para una $\sigma \in L_n^k$ sobre una configuración $c \in C(\Sigma)$ denotamos un mapeo global desde un punto de vista m y una posición i denotado por,

$$(f_{\sigma,m}(c))_i = \sigma(c_{m+i} c_{m+i+1} \cdots c_{m+i+n-1}). \quad (3.1)$$

Si $k = 2$ y $r = 1$ entonces $n = 3$. Si $i = 0$ y $m = 0$ entonces el mapeo global $(f_{\sigma,0}(c))_0$ se representa como:

$$\begin{array}{c} (f_{\sigma,0}(c))_0 = \sigma(c_0 c_1 c_2) \\ \hline \begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & & f & \\ \cdot & & \downarrow & \\ \cdot & & \sigma & \\ \cdot & & & \cdot \end{array} \\ \hline \end{array}$$

por lo que se deduce que $\sigma(c_0 c_1 c_2) = \hat{x}$, entonces la condición $i < 0$ e $i \geq \lg(x)$ determina los límites de $\lg(x)$ dentro una configuración $c \in C_F(\Sigma)$.

Si $i = 0$ y $m = -1$ entonces el mapeo global $(f_{\sigma,-1}(c))_0$ se representa como:

$$\begin{array}{c} (f_{\sigma,-1}(c))_0 = \sigma(c_{-1} c_0 c_1) \\ \hline \begin{array}{cccc} \cdot & \cdot & \cdot & \cdot \\ \cdot & & f & \\ \cdot & & \downarrow & \\ \cdot & & \sigma & \\ \cdot & & & \cdot \end{array} \\ \hline \end{array}$$

podemos apreciar que el subíndice i indica la posición que ocupan cada una de las configuraciones c_i dentro de la cadena x ; m indica el punto de vista donde se localizan estas cadenas. Las condiciones establecidas por i denotan un mapeo global finito de σ denotado por $\hat{f} : C_F(\Sigma) \rightarrow C_F(\Sigma)$.

Si \hat{f} representa el mapeo de las configuraciones finitas del conjunto $C_F(\Sigma)$ entonces $\lg(x)$ es igual a l , donde $l \in \mathbf{P}$ y denota el tamaño de cualquier configuración $c_l \in C_F(\Sigma)$.

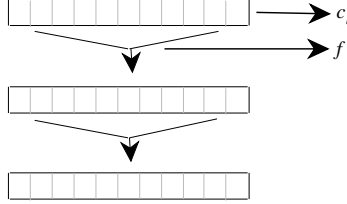


Figura 3.5: Mapeo global del conjunto $C_F(\Sigma)$.

Por lo que concluimos que $\hat{f}(c_l) = f$ para un mapeo global finito. Nótese que se emplea la notación original [16] $c \in C_F(\Sigma)$ con la notación propuesta en este escrito $c_l \in C_F(\Sigma)$, donde la finalidad es la misma pero tratando de especificar con mayor detalle los elementos que pertenecen al conjunto $C_F(\Sigma)$.

Finalmente desarrollamos el mapeo global para configuraciones no finitas. Para un mapeo global de una configuración $c \in C(\Sigma)$ la correspondencia esta dada por las configuraciones mismas como elementos del conjunto $C(\Sigma)$. El mapeo global $f : C(\Sigma) \rightarrow C(\Sigma)$ es el mapeo de las mismas configuraciones de $c \in C(\Sigma)$.

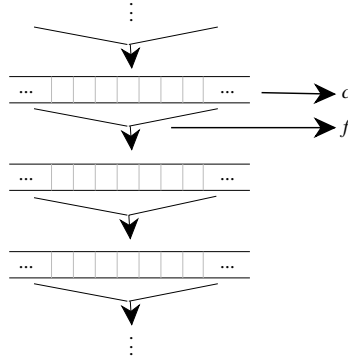


Figura 3.6: Mapeo global del conjunto $C(\Sigma)$.

Por lo tanto $f_{\sigma,m} = f$ denota un mapeo global por que las propiedades locales establecidas por σ y la localización de estas configuraciones denotadas por m , son las mismas para toda configuración $c \in C(\Sigma)$.

3.3.1 Diagrama de transiciones

Los mapeos globales derivados por los mapeos locales restringidos por una σ particular ofrecen un nuevo enfoque de estudio basado en las mismas configuraciones $c_l \in C_F(\Sigma)$. Si $c_l \in C_F(\Sigma)$ es una cadena de longitud $\lg(x) = l$, entonces c_l es un estado global.

Con esta definición podemos contruir un nuevo diagrama, conocido como *diagrama de transiciones* [13]. Estos diagramas de transiciones estan formados por

todos los estados globales del conjunto de $C_F(\Sigma)$. Si l representa el tamaño de una configuración c_l , entonces tenemos k^l estados globales. Estos diagramas de transiciones también son conocidos como árboles topológicos [18] donde cada nodo representa un estado global y las flechas indican la transición en un paso. Se tiene una raíz cíclica formada por uno o más estados globales, de esta raíz se desprenden ramas formadas por las mismas transiciones y finalmente estas ramas contienen hojas donde estas hojas representan la configuración inicial (para algunos casos) del diagrama de evoluciones. La función de transición que determina un mapeo global se representa como:

$$\Phi(x_0^t, x_1^t, \dots, x_{l-1}^t) \rightarrow x_0^{t+1}, x_1^{t+1}, \dots, x_{l-1}^{t+1}. \quad (3.2)$$

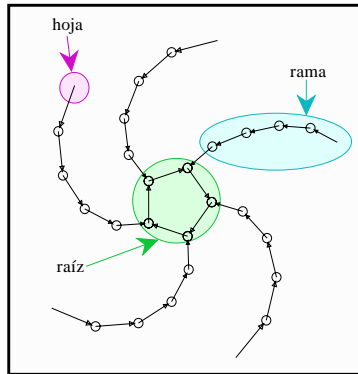


Figura 3.7: Árbol topológico.

Como se había señalado los nodos de tal gráfica son las configuraciones finitas de tamaño l . Si tenemos un autómata con $k = 2$ y $r = 1$ podemos formar configuraciones de tamaño $l \geq 2$ para alguna $\sigma \in L_n^k$.

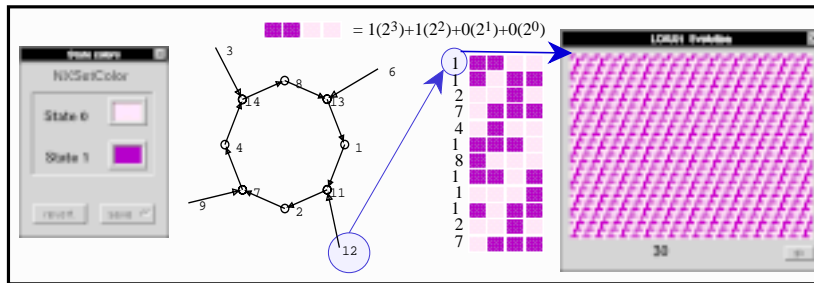


Figura 3.8: Transiciones globales.

Los valores que toman cada uno de los nodos dentro del diagrama de transiciones se obtienen a través de la suma de los valores de cada una de las células multiplicadas por su potencia correspondiente a la posición que ocupan cada una de ellas. Si tenemos un autómata binario con $k = 2$, $r = 1$ y $l = 4$, entonces tenemos k^l posibles configuraciones formadas por elementos que pertenecen a Σ . Para este caso son 16 configuraciones de tamaño cuatro que pertenecen al conjunto $C_F(\Sigma)$.

3.4 Ancestros

Dentro del diagrama de evoluciones podemos observar el comportamiento de los autómatas celulares restringidos por una regla dada, estas son generadas a partir de una configuración inicial aleatoria o predeterminada. La construcción de estas mismas configuraciones a través del tiempo denotan *ancestros* [9] de una configuración c_t^{t-1} para una configuración c_t^t en toda la lattice.

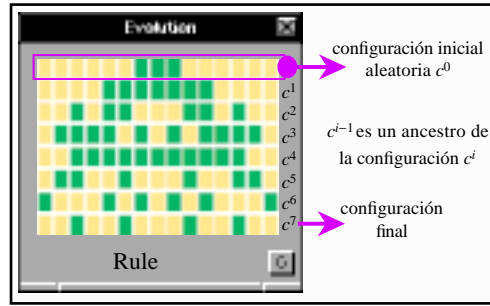


Figura 3.9: Ancestros.

Como podemos apreciar los ancestros implican el hecho de que una configuración tenga al menos un antecesor a través del tiempo. Esto se puede visualizar a través de un mapeo, donde el mapeo determinará los tipos de ancestros que podemos obtener en el diagrama de evoluciones o transiciones.

3.5 Clasificación de los mapeos

3.5.1 Jardín del Edén implica mapeo inyectivo

Un *mapeo inyectivo* [16] es cuando a lo más tenemos un ancestro. Sea A el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo $t - 1$ y B el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo t . Entonces definimos un mapeo inyectivo si existe una correspondencia *uno a uno* [17] del conjunto A al conjunto B .

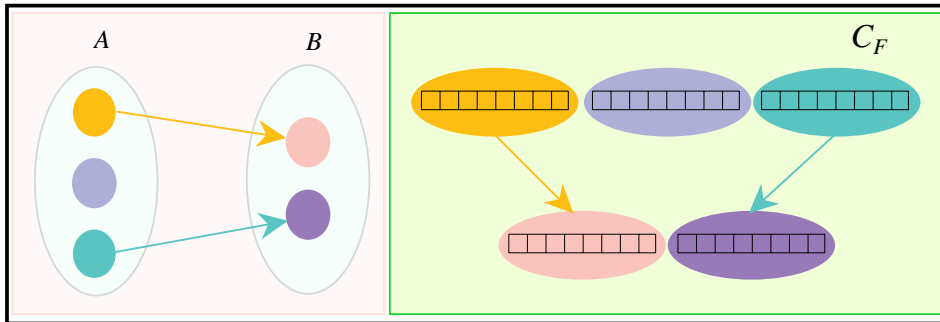


Figura 3.10: Mapeo inyectivo.

El término Jardín del Edén significa que existen cadenas que no pueden ser construidas a partir de otras cadenas. Es decir, si tenemos una configuración $c_l \in C_F(\Sigma)$ esta configuración sólo puede ser la configuración inicial del conjunto de evoluciones y nunca podremos obtenerla a través del tiempo. Entonces si existe un elemento en el conjunto A que no corresponda con ningún elemento del conjunto B , este elemento es un Jardín del Edén.

Dada la naturaleza del surgimiento de este fenómeno las características que describen el origen de estos estados, son aún más complejas y que no desarrollaremos en este escrito. Sin embargo existen artículos tales como los de S. Amoroso y G. Cooper [14], Moore [6], McIntosh [7] y Sven Skyum [15]. Donde se determina de manera rigurosa y formal el surgimiento de tales configuraciones.

Una manera de detectar y obtener las configuraciones pertenecientes al Jardín del Edén, es construir el diagrama de subconjuntos. Examinemos un autómata binario dado por la regla 18. Si existe alguna secuencia que vaya del conjunto máximo al conjunto mínimo esta cadena es una cadena perteneciente al Jardín del Edén.

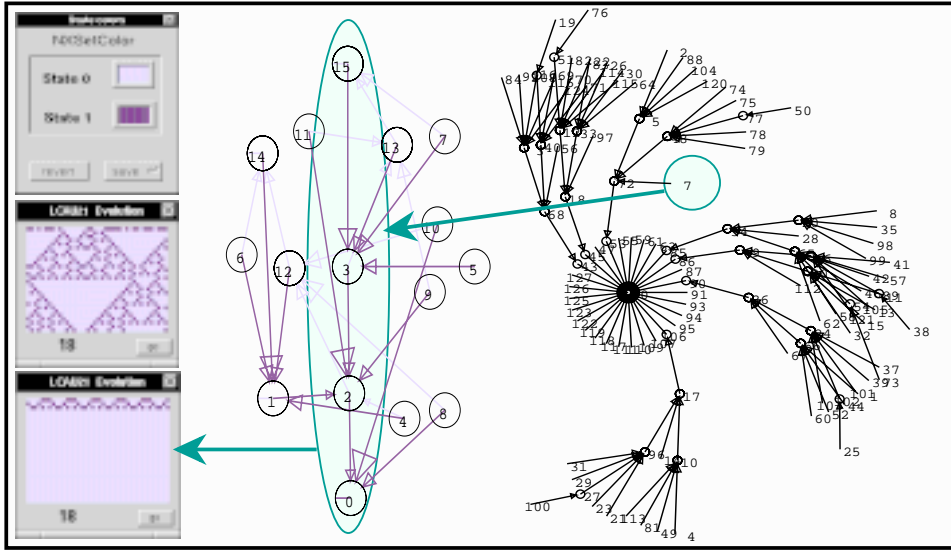


Figura 3.11: Jardín del Edén.

En la Figura 3.11 tenemos el diagrama de subconjuntos de la regla 18, con todas sus clases unitarias y todos los posibles caminos entre ellas. La expresión regular formada por la cadena 1110^* denota una configuración perteneciente al Jardín del Edén, dado que si tenemos cadenas que empiecen con 111 y concatenen con 0^* para cualquier longitud l de cualquier $c_l \in C_F(\Sigma)$. Entonces todas estas configuraciones pertenecen al Jardín del Edén. El diagrama de transiciones confirma nuestra definición, ya que para cualquier anillo de longitud l la configuración 1110^* es una hoja del árbol topológico, por lo tanto se deduce que esta configuración carece de ancestros y presenta un mapeo inyectivo.

3.5.2 Ancestros múltiples implica mapeo suryectivo

Un *mapeo suryectivo* [16] es cuando tenemos al menos un ancestro. Sea A el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo $t - 1$ y B el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo t . Entonces definimos un mapeo suryectivo si existe una correspondencia *sobre* [17] del conjunto A al conjunto B .

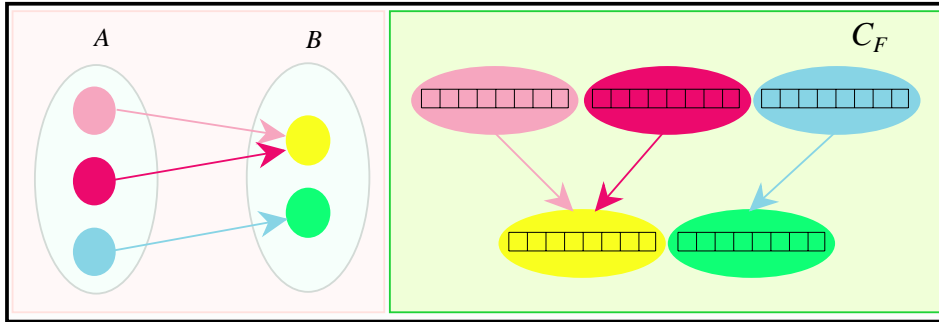


Figura 3.12: Mapeo suryectivo.

Esto quiere decir que siempre existirá al menos una imagen del conjunto A sobre el conjunto B . Dentro del diagrama de evoluciones implica que una configuración $c_l \in C_F(\Sigma)$ tiene uno o más ancestros, esto es podemos construir esa misma cadena a partir de una o más cadenas en el tiempo $t - 1$.

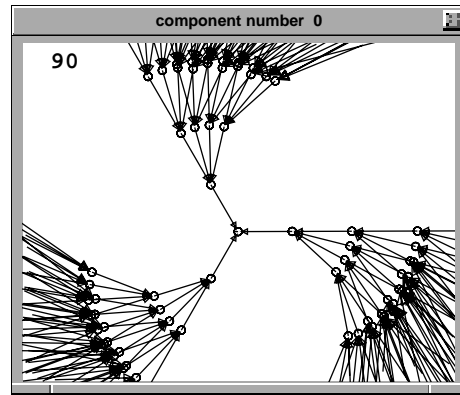


Figura 3.13: Mapeo suryectivo regla 90.

El mapeo suryectivo tiene la característica de que no existen configuraciones pertenecientes al Jardín del Edén, esto se debe por que todas las configuraciones tienen uno o más ancestros. Si examinamos el autómata binario regla 90 en la Figura 3.13 podemos observar dentro el árbol topológico para las configuraciones c_l con $l = 8$ localizados en una i -ésima posición, configuraciones $c_{l,i} \in C_F(\Sigma)$ con $Ancestros \geq 1$. El diagrama de transiciones es útil para poder observar las multiplicidades que existen sobre las mismas transiciones. Sin embargo podemos verificar esta propiedad a partir del mismo diagrama de de Bruijn y el diagrama de

subconjuntos.

De manera informal, el diagrama de de Bruijn nos indica que un mapeo es suryectivo si el número de ligas dado por algún elemento que pertenece a Σ es igual a cada uno de los demás elementos que pertenecen a Σ para el caso cuando $r = h$. Si analizamos las matrices de conectividad y el mapeo es suryectivo entonces:

$$|M_{i,j}(0)| = |M_{i,j}(1)| = \dots = |M_{i,j}(k-1)| \quad \forall k \in \Sigma.$$

El diagrama de subconjuntos debe de carecer de secuencias que vayan del conjunto máximo al conjunto mínimo, esto implica que no existen configuraciones pertenecientes al Jardín del Edén. Para verificar con mayor exactitud todas estas multiplicidades [2] [16] podemos construir el diagrama de parejas y analizar las secuencias que tienen uno o más ancestros. Regresemos al autómata binario regla 90 cuyo mapeo es un mapeo suryectivo.

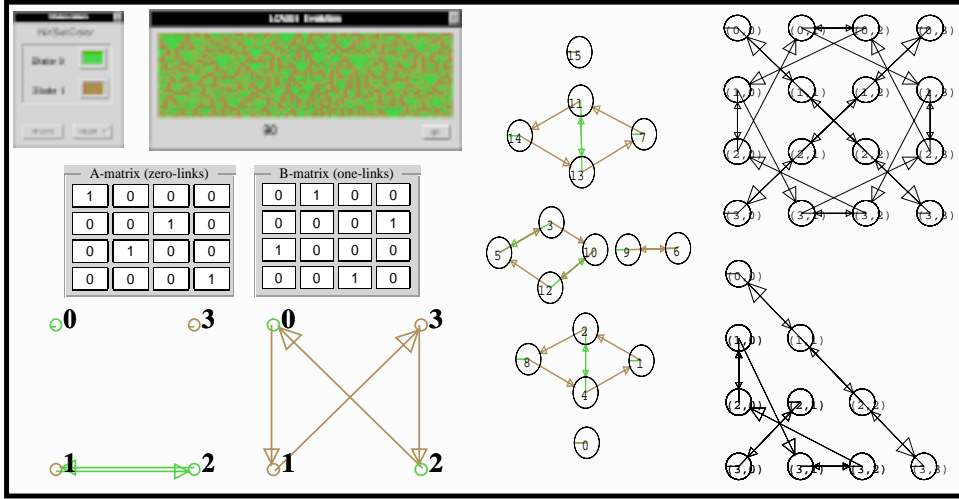


Figura 3.14: Propiedades de un mapeo suryectivo.

En la Figura 3.14 tenemos que las matrices de conectividad por estado para la regla 90 cumple con la condición de que la cardinalidad de la matriz A es igual a la cardinalidad de la matriz B , esto implica que el diagrama de de Bruijn debe tener igual número de aristas del estado cero y del estado uno. El diagrama de subconjuntos nos indica que no existe ruta alguna que vaya del conjunto máximo al conjunto mínimo, además las posibles secuencias que pueden llegarse a formar no exceden de cuatro transiciones a lo más. Esto nos indica que si construimos sus respectivos árboles topológicos ellos tendran una altura máxima de cuatro transiciones y además sobre la misma clase unitaria, ya que las clases unitarias se encuentran autocontenidas en ciclos de longitud mayor igual que uno y menor igual que cuatro sin salir de estas. Finalmente el diagrama de parejas confirma el mapeo suryectivo, pues nos muestra los ciclos que se forman fuera de la diagonal lo que implica que existen secuencias de cadenas que tienen más de un posible ancestro.

3.5.3 Ancestros únicos implica mapeo biyectivo

Un *mapeo biyectivo* [16] es cuando tenemos un único ancestro. Sea A el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo $t - 1$ y B el conjunto de las configuraciones finitas $C_F(\Sigma)$ en el tiempo t . Entonces definimos un mapeo biyectivo si existe una correspondencia *uno a uno* y además *sobre* [17] del conjunto A al conjunto B .

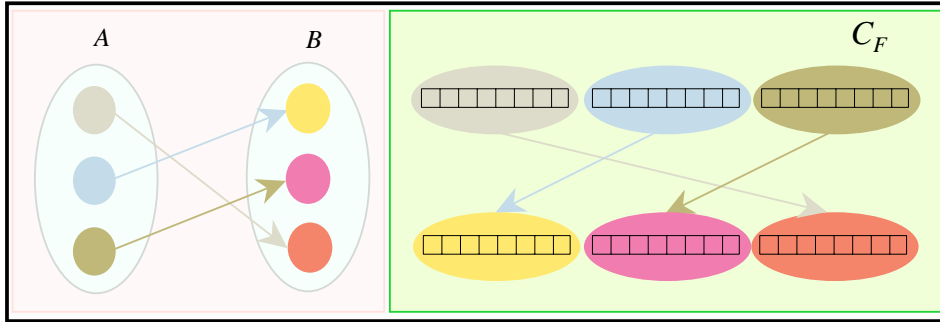
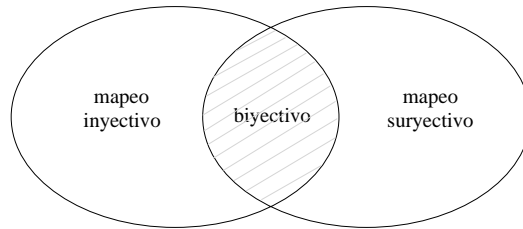


Figura 3.15: Mapeo biyectivo.

Como podemos ver en la Figura 3.15 el mapeo biyectivo implica que por cada elemento que existe en el conjunto A debe de existir un mapeo único para cada uno de los elementos del conjunto B .

El mapeo biyectivo es la intersección del mapeo inyectivo y el mapeo suryectivo [19], por lo que el mapeo biyectivo carece de configuraciones pertenecientes al Jardín del Edén y configuraciones con múltiples ancestros.



Esto quiere decir que para toda configuración $c_t \in C_F(\Sigma)$ en el tiempo t , esta configuración tiene un único ancestro en el tiempo $t - 1$ y un único sucesor en el tiempo $t + 1$. El mapeo biyectivo juega un papel muy importante dentro del estudio de los autómatas celulares *reversibles*, las mismas propiedades globales que manifiestan este tipo de mapeo a sido uno de los principales puntos de investigación [20] en los últimos años. Es importante señalar que el mapeo biyectivo presenta propiedades muy importantes tanto gráficas como matriciales. La determinación y representación adecuada de los autómatas reversibles la trataremos con mayor detalle en el capítulo siguiente, mencionando algunos algoritmos existentes para obtener tales autómatas, así como sus propiedades elementales para una mejor identificación de los mismos.

Por último dentro del diagrama de transiciones el mapeo biyectivo se comporta de manera cíclica como se puede observar en la Figura 3.16; su representación de acuerdo a la definición señalada anteriormente ilustra de manera precisa este comportamiento para cualquier configuración $c_l \in C_F(\Sigma) \forall l \geq 2$, en este caso el autómata celular en estudio es un autómata binario regla 15. Los dos primeros diagramas de transiciones para las configuraciones globales tienen una longitud de $l = 2$, en el diagrama de cuatro nodos $l = 4$, en el diagrama de seis nodos $l = 6$, en el diagrama de ocho nodos $l = 8$ y en el último diagrama de diez nodos $l = 10$.

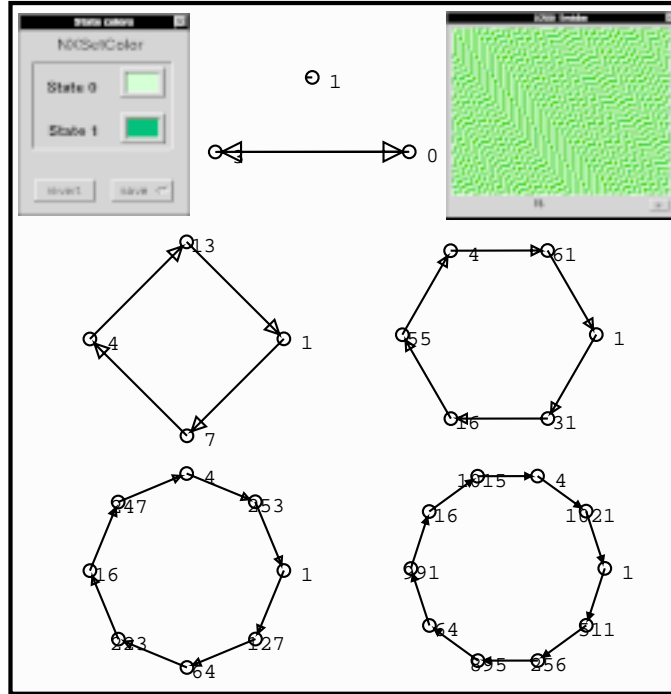


Figura 3.16: Transiciones biyectivas.

Capítulo 4

Grados de Reversibilidad

Se le denomina “Grados de Reversibilidad” a aquellos autómatas celulares que no son reversibles pero que bajo ciertas características estos autómatas tienen comportamientos reversibles. Para comprender mejor este punto de vista, daremos una pequeña introducción sobre los *Autómatas Celulares Reversibles*, mencionaremos algunos algoritmos y posteriormente describiremos con detalle estos tipos de autómatas con grados de reversibilidad.

4.1 Autómatas celulares reversibles

El estudio sobre autómatas celulares reversibles no ha sido resuelto en su totalidad hasta nuestros días para todos los casos posibles, sin embargo se han logrado avances muy importantes que nos han permitido comprender de una manera más clara la explicación de este comportamiento. Mencionaremos de manera general las características más importantes para identificar autómatas celulares reversibles.

Los autómatas celulares reversibles se caracterizan por tener un mapeo inyectivo y a la vez suryectivo, es decir tienen un mapeo biyectivo, carecen de configuraciones pertenecientes al Jardín del Edén y tampoco tienen múltiples ancestros. Esto significa que los árboles topológicos están constituidos sólo de puras raíces cíclicas, no existen ramas ni hojas. Trabajaremos con un autómata $(4, h)$ para tener una idea más clara de estos conceptos.

El autómata celular de la Figura 4.1 ilustra algunas condiciones básicas para detectar si el autómata es reversible o no. Mencionaremos sólo algunas pues este estudio se ha desarrollado detalladamente en [21].

Una primera condición es que los estados deben existir en igual número en el diagrama de de Bruijn, esto es, debe de cumplir el teorema de *Multiplicidad Uniforme* [2] [16] [11] que nos dice; si tenemos un estado con un número de ancestros igual al número de nodos del diagrama de de Bruijn, entonces para toda concatenación que realizemos por la izquierda o por la derecha el número de ancestros debe ser el mismo.

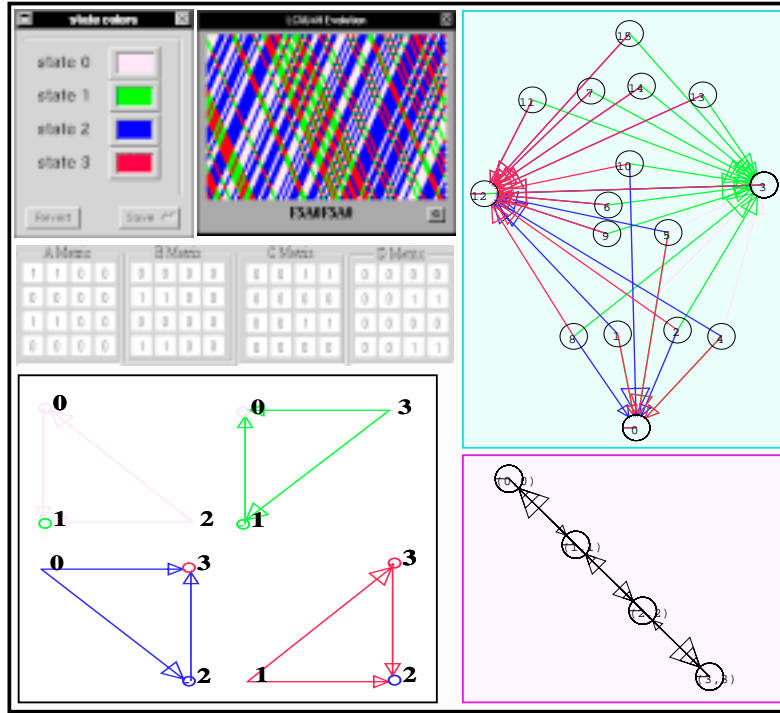


Figura 4.1: Autómata reversible $(4, h)$ regla F5A0F5A0.

La segunda implica que en el diagrama de subconjuntos no deben existir cadenas que vayan del conjunto máximo al conjunto mínimo y además que no existan secuencias de estados que formen ciclos dentro de las mismas clases unitarias, ya que las secuencias deben estacionarse en una clase unitaria del diagrama sin pasar a otra clase unitaria, es decir, estacionarse en un sólo nivel o en su defecto llegar de un sólo paso al conjunto máximo o al conjunto mínimo. Esta propiedad va asociada a la primera condición ya que si el autómata cumple con el teorema de multiplicidad uniforme, también debe cumplir con el teorema de los *Índices de Welch* [2] [16] [11] que nos dice; si todas las secuencias posibles que se encuentran dentro de las clases unitarias que describen los mismos índices de Welch ya sea por la izquierda o por la derecha, forman cadenas de longitud mayor igual que uno y menor igual al número de nodos del diagrama de de Bruijn, esto nos indica que debe existir un único ancestro.

La tercera propiedad es analizar el diagrama de parejas que nos ayuda a verificar la nula existencia de ciclos fuera de la diagonal principal, es decir no existen múltiples ancestros. Los ciclos que existen dentro de la diagonal principal se debe por que los nodos de la diagonal principal no son más que los mismos nodos del diagrama de de Bruijn.

Regresando a la Figura 4.1 podemos identificar estas propiedades mencionadas en el párrafo anterior, las matrices de conectividad derivadas del diagrama de de Bruijn por estado, contienen el mismo número de elementos por cada uno de los

estados posibles, lo que implica que en el diagrama de de Bruijn deben existir igual número de aristas por cada uno de los estados. El diagrama de subconjuntos muestra todas las posibles secuencias originadas desde cualquier nodo, es decir con sus cuatro clases unitarias construidas. Si seguimos cualquier secuencia podemos verificar como de la clase unitaria uno sube a la clase unitaria dos o baja al conjunto vacío, la clase unitaria tres sólo baja a la clase unitaria dos, la clase unitaria cuatro baja a los nodos 12 y 3 de la clase unitaria dos, a estos nodos también se les conoce como nodos atractores. Por lo que deducimos que la clase unitaria nivel dos es la clase que contiene las propiedades de los índices de Welch y cumple con la propiedad de que toda secuencia que exista dentro de esta clase debe de ser mayor igual a uno y menor igual al número de nodos del diagrama de de Bruijn. Finalmente podemos verificar que el autómata presenta un mapeo biyectivo y carece de múltiples ancestros ya que se a gráficado su respectivo diagrama de parejas con la opción *sólo ciclos* y observamos que no existen ciclos fuera de la diagonal principal con lo que se determina que el autómata celular es reversible.

Existen aún más sutilezas por analizar, pero este no es el fin de esta sección. Si se desea un mayor análisis de estas propiedades se puede consultar la siguiente bibliografía [2] [16] [20] [11] [10] [21], donde se desarrolla con gran profundidad y de manera formal la demostración y representación de tales teoremas.

4.2 Algunos algoritmos existentes

Los algoritmos para obtener autómatas celulares reversibles que mencionaremos en esta sección son dos; el primero fue desarrollado por Edward Fredkin (1990) y el segundo fue desarrollado por David Hillman (1991).

4.2.1 Algoritmo de Fredkin

Fredkin [10] descubrió que la paridad o la operación **or exclusivo** produce replicas de patrones de manera interesante dentro de los autómatas celulares, con lo que desarrolla un algoritmo para obtener reglas reversibles de una regla original dada.

La regla de evolución propuesta es para dos generaciones de células. Sea x_i^t la célula que se encuentra en la i -ésima posición dentro de la lattice en el tiempo t , entonces si

$$x_i^{t+1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \quad (4.1)$$

la regla de evolución para el caso de un autómata binario (2,1), la nueva regla esta dada por:

$$x_i^{t+1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \oplus x_i^{t-1} = \Phi(x_i^{t-1}, x_{i-1}^t, x_i^t, x_{i+1}^t) \quad (4.2)$$

además este algoritmo puede ser usado de forma invertida, es decir operar con la célula x_i^{t+1} para obtener la célula x_i^{t-1} como se denota a continuación:

$$x_i^{t-1} = \varphi(x_{i-1}^t, x_i^t, x_{i+1}^t) \oplus x_i^{t+1} = \Phi^{-1}(x_i^{t+1}, x_{i-1}^t, x_i^t, x_{i+1}^t). \quad (4.3)$$

Dadas las propiedades algebraicas del **or exclusivo**, alguna otra función invertible de dos variables puede ser usada en lugar de \oplus , para funciones booleanas puede ser el caso de la operación del **nor exclusivo (equivalencia)**.

4.2.2 Algoritmo de Hillman

Hillman [20] hace notar que los autómatas celulares reversibles no sólo dependen de la regla de evolución, si no que hay otros factores que intervienen en tal proceso y este es el caso del tamaño del anillo de evolución.

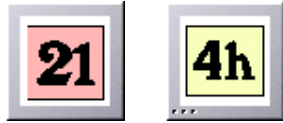
Sea C una cadena de x células en el tiempo t , de este modo C está completamente determinado por la cadena de células P de longitud $x + d - 1$ en el tiempo $t - 1$, donde d es el tamaño de la vecindad del autómata.

1. Si α es una configuración en C , sea P_α el conjunto de configuraciones las cuales de acuerdo a una regla de evolución dada generan α .
2. Sea B_j la j -ésima configuración presente en P_α , entonces formamos una tabla Q_α la cual su j -ésimo elemento será igual a $(b_\alpha(j), e_\alpha(j))$, donde $b_\alpha(j)$ es el conjunto de las primeras $d - 1$ células en B_j y $e_\alpha(j)$ es el conjunto de las últimas $d - 1$ células en B_j .
3. Finalmente formamos R_w como el conjunto de todas las tablas posibles Q_α dado un ancho w especificado. Si $w > 1$ entonces R_w se obtiene de la concatenación de R_1 y R_{w-1} para cada elemento Q_α de R_{w-1} y Q_β de R_1 , uno obtiene un elemento Q_τ de R_w después de comparar cada entrada $Q_\alpha(i)$ con cada entrada $Q_\beta(j)$, si $(e_\alpha(i) = b_\beta(j))$ entonces añadimos a Q_τ una entrada m tal que $b_\tau(m) = b_\alpha(i)$ y $e_\tau(m) = e_\beta(j)$. El proceso recursivo empieza calculando R_1 directamente de la regla de evolución.
4. Los elementos R_w contienen tablas Q_i con las listas de las primeras y las últimas $d - 1$ células de cada ancestro de una cadena de longitud w especificada. 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 sólo si $b_i(j) = e_i(j)$. Por lo tanto, si el autómata es reversible cada tabla Q_i debe tener una y sólo una entrada j tal que $b_i(j) = e_i(j)$.
5. El algoritmo para hasta un ancho w especificado.

4.3 Autómatas celulares con grados de reversibilidad

Como habíamos señalado en el primer párrafo del Capítulo 4 los grados de reversibilidad se manifiestan en autómatas celulares que no son propiamente reversibles. Hillman [20] hace un estudio análogo de estas características ofreciendo un algoritmo para determinar estas propiedades, aunque el fin de este algoritmo es la determinación de los autómatas celulares reversibles, donde el cómputo se hace cada vez más lento conforme va aumentando el tamaño de w .

Dentro de los autómatas celulares tenemos configuraciones finitas $c_I \in C_F(\Sigma)$ que pueden ser de tamaño par o impar, donde su evolución es reversible para ciertas configuraciones dadas. Esto significa que estas configuraciones tienen un único ancestro, carecen de configuraciones pertenecientes al Jardín del Edén y múltiples ancestros. Nuestro estudio se enfoca en las matrices de conectividad por estado del diagrama de de Bruijn que determinan las propiedades fundamentales de estas características y representaremos de manera formal. El estudio se realizará particularmente para el autómata $(4, h)$ e intercalando un autómata $(2, 1)$ para ejemplificar con mayor rapidez estas propiedades, ofreciendo además una perspectiva para todos los demás casos posibles.



4.3.1 Matriz de de Bruijn simbólica

El uso del lenguaje de expresiones regulares puede formar expresiones simbólicas sencillas o complejas, cuyas operaciones están constituidas principalmente por la unión, concatenación e iteración. Otra operación que es de gran utilidad es el *translape* entre dos cadenas cuya representación se puede denotar con el símbolo \diamond . Si las palabras ax y yb consisten de las letras a y b unidas por las palabras x y y , entonces:

$$ax \diamond yb = \begin{cases} axb & x = y \\ \phi & \text{c.o.c} \end{cases}$$

donde ϕ es el conjunto vacío.

Con tal notación una matriz de de Bruijn simbólica [10] puede ser definida como:

$$D_{i,j} = i \diamond j \quad (4.4)$$

la forma simbólica de dicha matriz es construida de la siguiente manera para un autómata $(2, 1)$

$$D = \begin{bmatrix} 000 & 001 & \cdot & \cdot \\ \cdot & \cdot & 010 & 011 \\ 100 & 101 & \cdot & \cdot \\ \cdot & \cdot & 110 & 111 \end{bmatrix}.$$

Donde los puntos representan el conjunto vacío y los elementos los translapes de los mismos índices de la matriz. Además tales matrices deben ser multiplicadas bajo la operación \diamond , para multiplicar los elementos individuales de la matriz. Por lo general es poco usual trabajar con la matriz simbólica totalmente llena, sin embargo sirve para obtener una representación muy explícita de los bloques de evolución que podemos generar.

A través de las matrices simbólicas podemos obtener las matrices de evolución también originada del diagrama de de Bruijn. Sea la n -ésima potencia de una matriz

M , cuya matriz contiene todas las posibles palabras de longitud n distribuida a través de toda la matriz de acuerdo al traslape de los índices que son construidos por las mismas letras de longitud n . Es decir, los elementos de la matriz son indexados por su vecindad parcial inicial y final.

Para obtener la evolución de una configuración dada consideremos una variante de la Eq. 4.4, que describe la evolución de las vecindades individualmente,

$$\mathbf{S}_{i,j} = \varphi(i \diamond j) \quad (4.5)$$

que a su vez puede ser descompuesta en la siguiente suma:

$$S = \sum_{i=0}^{k-1} S(k) \quad (4.6)$$

para definir finalmente

$$S(k)_{i,j} = \begin{cases} k & \varphi(i \diamond j) = k \\ \phi & c.o.c \end{cases} . \quad (4.7)$$

Entonces la célula de evolución es insertada dentro de la matriz simbólica, esto significa que las cadenas con sus respectivas células de evolución aparecen como elementos del producto de las mismas matrices. Por lo tanto denotamos a $(i \diamond j)_{r+1}$ como el estado de la célula central de su vecindad delimitada por su radio r .

De esta manera la Eq. 4.4 esta definida por

$$\mathbf{s}_{i,j} = (i \diamond j)_{r+1} \quad (4.8)$$

que es además descompuesta en una suma,

$$s = \sum_{i=0}^{k-1} s(k) \quad (4.9)$$

definiendo

$$s(k)_{i,j} = \begin{cases} (i \diamond j)_{r+1} & \varphi(i \diamond j) = k \\ \phi & c.o.c \end{cases} . \quad (4.10)$$

Por lo que se deduce que la célula central no es precisamente la que define las matrices $s(i)$, frecuentemente un ancestro esta en un sitio seleccionado por la misma célula de evolución.

Aún podemos obtener una tercera variante sobre las matrice simbólicas de de Bruijn y esta es una matriz numérica, cuyo propósito es meramente contar el número de imágenes, aunque representandolos de otra forma. La variante final de la Eq. 4.4 es representada como

$$D_{i,j} = \begin{cases} 1 & \varphi(i \diamond j) \neq \phi \\ 0 & c.o.c \end{cases} , \quad (4.11)$$

dada esta definición alternativa de la matriz de de Bruijn todavía la podemos descomponer en una suma

$$D = \sum_{i=0}^{k-1} N(k) \quad (4.12)$$

y finalmente definimos

$$N(k)_{i,j} = \begin{cases} 1 & \varphi(i \diamond j) = k \\ 0 & c.o.c \end{cases} . \quad (4.13)$$

El análisis que realizaremos a continuación para algunas reglas en especial es con el fin de detectar el caso más sencillo hasta el caso que puede presentar ciertas características un poco enredosas.

4.3.2 Configuraciones de tamaño par e impar

Cuando nos referimos al tamaño (ancho) de la configuración par o impar, es por el valor que toma l para una configuración $c_l \in C_F(\Sigma)$.

Sea $n \in \mathbf{P}$ entonces definimos una configuración de tamaño par si $l = 2n$ y definimos una configuración de tamaño impar si $l = 2n + 1$.

- Si $n = 1 \Rightarrow l = 2(1)$ por lo que la configuración c_l es de tamaño par.

$$c_l = \square\square$$

- Si $n = 1 \Rightarrow l = 2(1) + 1$ por lo que la configuración c_l es de tamaño impar.

$$c_l = \square\square\square$$

- Si $n = 2 \Rightarrow l = 2(2)$ por lo que la configuración c_l es de tamaño par.

$$c_l = \square\square\square\square$$

- Si $n = 2 \Rightarrow l = 2(2) + 1$ por lo que la configuración c_l es de tamaño impar.

$$c_l = \square\square\square\square\square$$

Y así sucesivamente hasta un n -ésimo número que determinará el valor de l para una configuración $c_l \in C_F(\Sigma)$.

4.3.3 Autómata (2,1) regla 45

El autómata celular (2,1) regla 45 presenta grados de reversibilidad para configuraciones de tamaño impar, tomamos este autómata como un ejemplo práctico ya que para mostrar de manera detallada estas propiedades para el autómata (4, h) el trabajo es demasiado laborioso y poco práctico, siendo que la finalidad es la misma para todos los casos.

La matriz de de Bruijn simbólica para el autómata (2,1) presentada en la Sección 4.3.1, es general para todo autómata con estos valores, ya que sus elementos son las mismas vecindades que forman la regla de evolución.

$$\text{regla} = \begin{array}{r} 11110000 \\ 11001100 \\ 10101010 \\ \hline 10110100 \end{array} \implies D = \begin{bmatrix} 000 & 001 & \cdot & \cdot \\ \cdot & \cdot & 010 & 011 \\ 100 & 101 & \cdot & \cdot \\ \cdot & \cdot & 110 & 111 \end{bmatrix}$$

Ahora bien sabemos de antemano que este autómata posee un mapeo suryectivo, es decir carece de configuraciones pertenecientes al Jardín del Edén y tiene múltiples ancestros. Sin embargo lo curioso de esta regla es que tiene comportamientos reversibles para configuraciones de tamaño impar y para que esta propiedad se cumpla es necesario que el autómata posea un mapeo biyectivo, o sea que tenga un único ancestro por cada configuración.

Si construimos una tabla de vecindades para obtener sus respectivas células de evolución, estas vecindades están denotadas por k^{2r+1} posibles combinaciones, para obtener configuraciones de longitud $l = \overline{1, 4}$.

$l = 1$	$l = 2$	$l = 3$	$l = 4$
000 → 1	0000 → 11	00000 → 111	000000 → 1111
001 → 0	0001 → 10	00001 → 110	000001 → 1110
010 → 1	0010 → 01	00010 → 101	000010 → 1101
011 → 1	0011 → 01	00011 → 101	000011 → 1101
100 → 0	0100 → 10	00100 → 010	000100 → 1010
101 → 1	0101 → 11	00101 → 011	000101 → 1011
110 → 0	0110 → 10	00110 → 010	000110 → 1010
111 → 0	0111 → 10	00111 → 010	000111 → 1010
	1000 → 01	01000 → 101	001000 → 0101
	1001 → 00	01001 → 100	001001 → 0100
	1010 → 11	01010 → 111	001010 → 0111
	1011 → 11	01011 → 111	001011 → 0111
	1100 → 00	01100 → 100	001100 → 0100
	1101 → 01	01101 → 101	001101 → 0101
	1110 → 00	01110 → 100	001110 → 0100
	1111 → 00	01111 → 100	001111 → 0100
		10000 → 011	010000 → 1011
		10001 → 010	010001 → 1010
		10010 → 001	010010 → 1001
		10011 → 001	010011 → 1001
		10100 → 110	010100 → 1110
		10101 → 111	010101 → 1111
		10110 → 110	010110 → 1110
		10111 → 110	010111 → 1110
		11000 → 001	011000 → 1001
		11001 → 000	011001 → 1000
		11010 → 011	011010 → 1011
		11011 → 011	011011 → 1011
		11100 → 000	011100 → 1000
		11101 → 001	011101 → 1001
		11110 → 000	011110 → 1000
		11111 → 000	011111 → 1000
			100000 → 0111
			100001 → 0110
			100010 → 0101
			100011 → 0101
			100100 → 0010
			100101 → 0011
			100110 → 0010
			100111 → 0010
			101000 → 1101
			101001 → 1100
			101010 → 1111
			101011 → 1111
			101100 → 1100
			101101 → 1101
			101110 → 1100
			101111 → 1100
			110000 → 0011
			110001 → 0010
			110010 → 0001
			110011 → 0001
			110100 → 0110
			110101 → 0111
			110110 → 0110
			110111 → 0110
			111000 → 0001
			111001 → 0000
			111010 → 0011
			111011 → 0011
			111100 → 0000
			111101 → 0001
			111110 → 0000
			111111 → 0000

Tabla 4.1: Tabla de evoluciones con $r = 1, 1.5, 2$ y 2.5 .

Si $r = 1$ entonces tenemos $2^{2(1)+1} = 8$ posibles bloques de evoluciones, si $r = 1.5$ entonces tenemos 16 posibles bloques y así sucesivamente. El crecimiento es por demás exponencial, si hubieramos trabajado con el autómata $(4, h)$ los valores por columna serían 64, 256, 1024 y 4096 respectivamente. Por lo que el estudio hubiera sido muy laborioso por realizar.

Para el caso cuando $r = 1$ tenemos que sus bloques de evoluciones coinciden con las mismas vecindades del autómata y sus respectivas células de evolución; se puede observar que en este caso existen igual número de ceros y de unos en las células de evolución de estos bloques. A esta propiedad se le conoce como multiplicidad uniforme, esto es cada uno de los estados tiene el mismo número de imágenes que los demás estados que conforman el autómata celular.

En este caso el estado cero tiene cuatro imágenes y el estado uno también,

111	101
110	011
100	010
<u>001</u>	<u>000</u>
0	1

Tabla 4.2: Configuraciones con $l = 1$.

esto quiere decir que con las cadenas 001, 100, 110 y 111 podemos construir un 0 en la siguiente generación. Ahora bien si las dos células de la izquierda coinciden con las dos células de la derecha podemos translapar estas células con la finalidad de determinar que configuración es un posible ancestro de las células de evolución en los estados globales.

Este concepto es más visible para configuraciones de tamaño mayor igual que dos, ya que podemos formar bloques de evoluciones de tamaño cuatro y obtener configuraciones de tamaño dos con sus respectivas transiciones.

1111	1011
1110	1010
1100	0101
<u>1001</u>	<u>0000</u>
00	11

Tabla 4.3: Configuraciones con $l = 2$.

Como podemos observar en la Tabla 4.3 para las células de evolución 00 el bloque 1111 cumple con la condición de tener la misma pareja de estados en la izquierda como en la derecha, por lo que deducimos que la configuración 00 tiene un ancestro formado por la cadena 11.

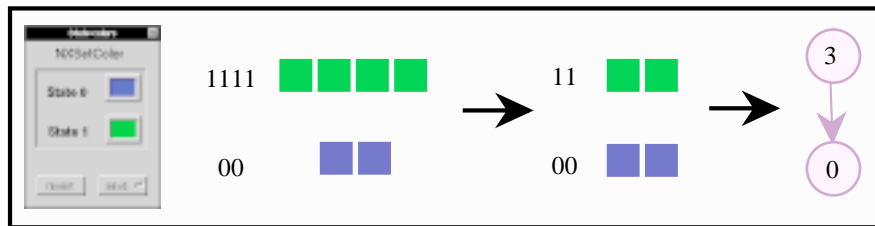


Figura 4.2: Formando anillos.

La Figura 4.2 ilustra paso a paso como formamos las configuraciones globales a partir de los bloques de evoluciones; la cadena 00 tiene cuatro imágenes pero la

imagen 1111 tienen los estados 11 a la izquierda del bloque y 11 a la derecha del mismo bloque, por lo que definimos su traslape si

$$xa \diamond by = \begin{cases} ab & x = y \\ \phi & c.o.c \end{cases} .$$

Si analizamos las células de evolución para la cadena 11 de la Tabla 4.3 podemos notar que tal configuración tiene tres bloques de evolución que cumplen con dicha condición, estos bloques son 1010, 0101 y 0000. Lo que implicaría que los estados globales 2, 1 y 0 mapean al estado global 3. De esta manera podemos obtener todos los ancestros de cada uno de los bloques de evolución que podemos formar con r .

Para comprobar nuestros resultados construyamos su respectivo árbol topológico y revisemos cada uno de sus mapeos posibles.

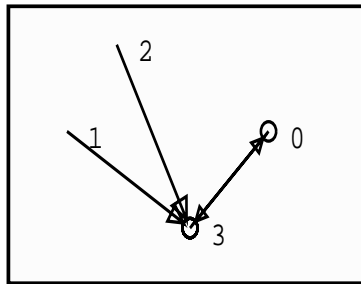


Figura 4.3: Árbol topológico con $l = 2$.

Finalmente podemos comprobar que cuando $l = 2$ las configuraciones c_l tienen más de un ancestro. El estado 0 mapea al estado 3 y el estado 3 mapea al estado 0 formando la raíz cíclica de dicho árbol topológico. El estado 2 mapeo al estado 3 y el estado 1 también mapea al estado 3, por lo que los estados 1 y 2 forman hojas dentro del árbol topológico y no tienen ancestros cuando $l = 2$.

Ahora analicemos el caso cuando $l = 3$ revisando sus respectivos bloques de evolución calculados en la Tabla 4.1,

11111	11101	10001	11011	01111	01101	10111	10101
11110	11000	00111	11010	01110	01000	10110	01011
11100	10011	00110	10000	01100	00011	10100	01010
<u>11001</u>	<u>10010</u>	<u>00100</u>	<u>00101</u>	<u>01001</u>	<u>00010</u>	<u>00001</u>	<u>00000</u>
000	001	010	011	100	101	110	111

Tabla 4.4: Configuraciones con $l = 3$.

nuevamente todas las células de evolución tienen el mismo número de bloques que producen dichas cadenas. Pero en este caso todas las configuraciones tienen una sola imagen, estas son: $111 \rightarrow 000$, $001 \rightarrow 001$, $010 \rightarrow 010$, $101 \rightarrow 011$, $100 \rightarrow 100$, $110 \rightarrow 101$, $011 \rightarrow 110$ y $000 \rightarrow 111$ respectivamente.

Esto significa que tenemos ciclos de tamaño uno, dos y tres, dentro de los árboles topológicos cuando $l = 3$. Los ciclos de longitud igual a uno están formados por los estados globales $1 \leftrightarrow 1$, $2 \leftrightarrow 2$ y $4 \leftrightarrow 4$. El ciclo de longitud igual a dos está formado

por los estados globales $0 \leftrightarrow 7$. El ciclo de longitud igual a tres esta formado por los estados globales $5 \rightarrow 3, 3 \rightarrow 6$ y $6 \rightarrow 5$.

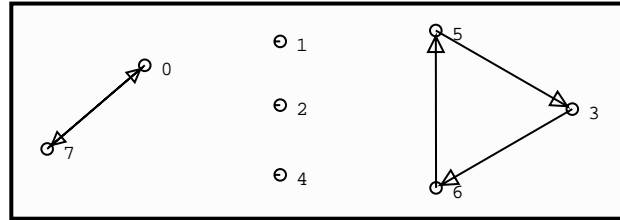


Figura 4.4: Árboles topológicos con $l = 3$.

La Figura 4.4 muestra todos los estados globales posibles cuando $l = 3$, el diagrama de transiciones confirma nuestros resultados sobre los ciclos calculados con los bloques de evoluciones, donde habíamos obtenido cinco ciclos en total. Como señalamos en la Sección 4.1 los autómatas celulares reversibles deben de tener un mapeo biyectivo que implica que no tengan múltiples ancestros y tampoco configuraciones pertenecientes al Jardín del Edén. Esta condición se cumple para este autómata cuando la longitud de c_l es de tamaño impar.

Para confirmar nuestra afirmación realizaremos el cálculo de los bloques de evolución cuando l es igual a cinco, siete y nueve. Comprobando que tienen grados de reversibilidad para todos sus árboles topológicos de este tamaño.

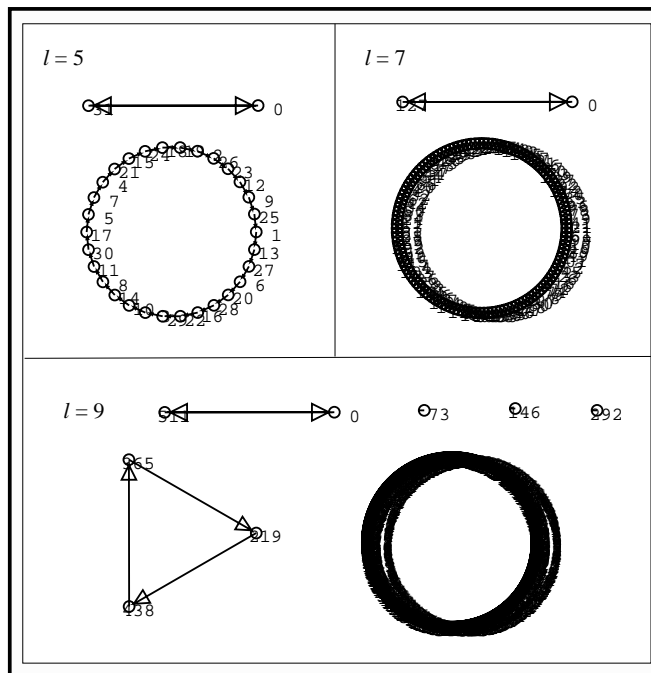


Figura 4.5: Árboles topológicos con $l = 5, 7$ y 9 .

Podemos ver que si $l = 5$ el cálculo se hace un poco laborioso, si $l = 7$ el cálculo se hace más laborioso y si $l = 9$ el cálculo es sumamente complicado. Sin embargo todas estas configuraciones son reversibles en todos sus árboles topológicos posibles.

4.3.4 Autómata (4,h) regla 0056B9EF

El autómata celular (4, h) regla 0056B9EF presenta grados de reversibilidad para configuraciones de ancho impar. La notación Wolfram para el caso (4, h) dentro de la regla de evolución se representa de la siguiente manera:

3 3	3 3	2 2	2 2	1 1	1 1	0 0	0 0
3 2	1 0	3 2	1 0	3 2	1 0	3 2	1 0
0 0	0 0	1 1	1 2	2 3	2 1	3 2	3 3
$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$	$4^1 4^0$
0	0	5	6	B	9	E	F

Tabla 4.5: Regla de evolución.

las vecindades se agrupan de dos en dos y se multiplican por sus respectivas potencias de acuerdo a la posición que ocupen sobre la misma regla, finalmente se suman estas parejas y se obtiene el número Wolfram.

Para construir los bloques de evoluciones para el autómata (2,1) cuando $l = \overline{1, 4}$ estos se realizaron de forma manual obteniendo sus células de evolución de acuerdo a una regla dada. En el caso del autómata (4, h) es aún más laborioso construir cada uno de estos bloques conforme r se va incrementando. Para resolver este problema haremos uso de las matrices de conectividad, con lo que obtendremos los mismos resultados para una l específica.

La matriz simbólica de de Bruijn para el autómata (4, h) se denota de la siguiente manera:

$$D = \begin{bmatrix} 00 & 01 & 02 & 03 \\ 10 & 11 & 12 & 13 \\ 20 & 21 & 22 & 23 \\ 30 & 31 & 32 & 33 \end{bmatrix}$$

y dada su regla de evolución obtenemos su matriz de evoluciones correspondiente

$$S = \begin{bmatrix} 3 & 3 & 2 & 3 \\ 1 & 2 & 3 & 2 \\ 2 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Si efectuamos el producto tensorial de la matriz S obtendremos las siguientes evoluciones

$$S^2 = \begin{bmatrix} 33 & 33 & 32 & 33 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 31 & 32 & 33 & 32 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 22 & 21 & 21 & 21 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 33 & 33 & 32 & 33 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 30 & 30 & 30 & 30 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 21 & 22 & 23 & 22 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 32 & 31 & 31 & 31 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 20 & 20 & 20 & 20 & \cdot \\ 23 & 23 & 22 & 23 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 11 & 12 & 13 & 12 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 12 & 11 & 11 & 11 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 10 & 10 & 10 & 10 & \cdot \\ 03 & 03 & 02 & 03 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 01 & 02 & 03 & 02 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 02 & 01 & 01 & 01 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 00 & 00 & 00 & 00 & \cdot \end{bmatrix}$$

que calcula todos los bloques de evoluciones cuando $r = 1$. Ahora podemos calcular los anillos correspondientes para sus respectivas células de evolución.

Trabajemos con los estado cero y dos, calculando sus bloques de evolución.

333	202
332	113
331	111
330	020
00	22

Tabla 4.6: Configuraciones con $l = 2$.

En la Tabla 4.6 podemos ver que la configuración 00 tiene un único ancestro formado por la configuración 33, sin embargo la configuración 22 tiene tres ancestros formados por las configuraciones 20, 11 y 02.

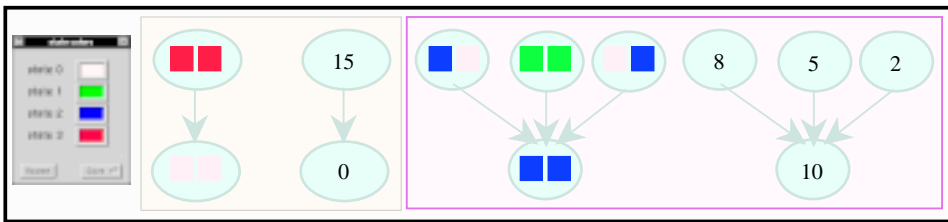


Figura 4.6: Formando anillos.

La Figura 4.6 muestra las configuraciones por estados globales y su valor decimal para cada una de ellas. El primer diagrama nos indica que la configuración 33 mapea a la configuración 00 en un paso, si revisamos la matriz S^2 podemos ver que el índice (00,00) tiene por elemento la cadena 33 lo que confirma su mapeo global, sin embargo notemos que el índice (33,33) tiene por elemento la cadena 00 por lo que deducimos que existe un ciclo de longitud dos entre las cadenas 00 y 33. El segundo diagrama nos indica que la configuración 20, 11 y 02 mapean a

la configuración 22 en un paso, esto quiere decir que los índices de la matriz S^2 (02,20) tienen por elemento la cadena 22, los índices (20,02) tienen por elemento la cadena 22 y el índice (11,11) tiene por elemento la cadena 22.

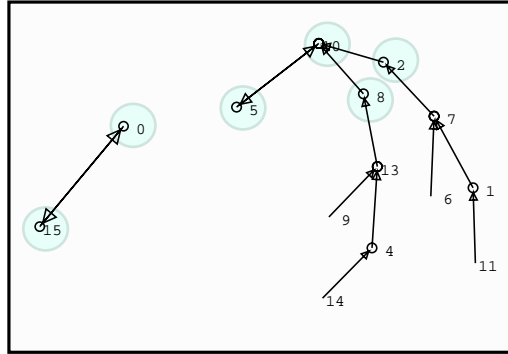


Figura 4.7: Árboles topológicos con $l = 2$.

Los árboles topológicos de la Figura 4.7 confirma nuestros resultados encerrando en un círculo aquellas configuraciones que calculamos con la matriz S^2 .

Si deseamos saber específicamente los ancestros de una configuración en particular sin calcular toda la matriz S^2 , podemos obtenerlos a través de un producto matricial dada por la matriz $s(k)$.

$$\begin{aligned}
 s(2)^2 &= \begin{bmatrix} \cdot & \cdot & 2 & \cdot \\ \cdot & 2 & \cdot & 2 \\ 2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \cdot & \cdot & 2 & \cdot \\ \cdot & 2 & \cdot & 2 \\ 2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \\
 &= \begin{bmatrix} 22 & \cdot & \cdot & \cdot \\ \cdot & 22 & \cdot & 22 \\ \cdot & \cdot & 22 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}
 \end{aligned}$$

Analizemos la configuración 22 calculada anteriormente, donde la matriz $s(2)^2$ determinará todas las posibles imágenes que tiene dicha configuración, como podemos observar la matriz $s(2)$ tiene cuatro imágenes y esto es por que las configuraciones de tamaño uno son los mismos mapeos de cada una de las vecindades posibles para la regla en estudio.

Si calculamos la matriz $s(2)^2$ obtendremos todas las imágenes posibles de la cadena 22 en un sólo paso, ya que los índices de las matrices que representan dicho producto, también determinaran las cadenas de evolución que existen para la cadena 22. Como podemos ver el número de imágenes es el mismo.

Entonces tenemos cuatro imágenes de tamaño tres y los bloques de evoluciones van a estar representados por el valor del renglón y la columna de la primera matriz; con el valor del renglón que determine el producto de dicho elemento con la segunda matriz. Por lo que finalmente podemos decir que para la cadena 22 los

bloques de evoluciones obtenidos en la matriz $s(2)^2$ estan representados por: 020, 111, 113 y 202 respectivamente, como habíamos determinado en la Tabla 4.6.

El diagrama de subconjuntos de este autómata celular muestra que cadenas tienen múltiples ancestros; estas cadenas se identifican por que forman ciclos de longitud mayor igual que uno en las clases unitarias intermedias.

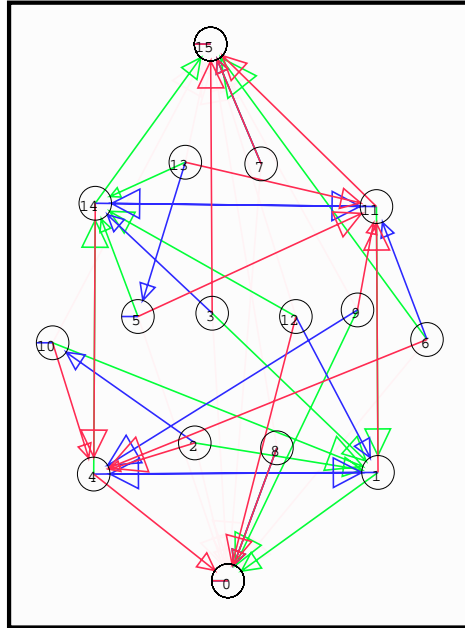


Figura 4.8: Diagrama de subconjuntos de la regla 0056B9EF.

Notemos en la Figura 4.8 que las cadenas formadas por el estado 2 en los nodos $11 \leftrightarrow 14$ y $1 \leftrightarrow 4$ forman ciclos de longitud dos, el estado dos es el estado que origina múltiples ancestros de forma inmediata como ilustramos en la Figura 4.6, ya que la cadena 22 por si misma tiene tres ancestros esto quiere decir que para toda expresión regular formada por las cadenas 22^* , esta cadena tiene más de un ancestro.

Ahora veamos la cadena de los nodos $4 \leftrightarrow 14$ formada por los estados 13 cuya expresión forma un ciclo que puede ser mayor igual que dos, esto nos indica que esta cadena tiene más de un ancestro.

212	123
103	122
101	121
<u>100</u>	<u>011</u>
13	31

Tabla 4.7: Configuraciones con $l = 2$.

La Tabla 4.7 muestra los bloques de evolución para las cadenas 13 y 31, la

cadena 13 tiene dos ancestros formados por las cadenas de 01 y 12, lo que equivale que la configuración 7 tiene por ancestros las configuraciones 1 y 6 como ilustramos en la Figura 4.7. La cadena 31 también forma un ciclo equivalente al de la cadena 13 en los nodos $1 \leftrightarrow 11$, pues es una permutación de dicha cadena que conserva las mismas propiedades. Finalmente podemos verificar todas estas múltiplicidades en el diagrama de parejas

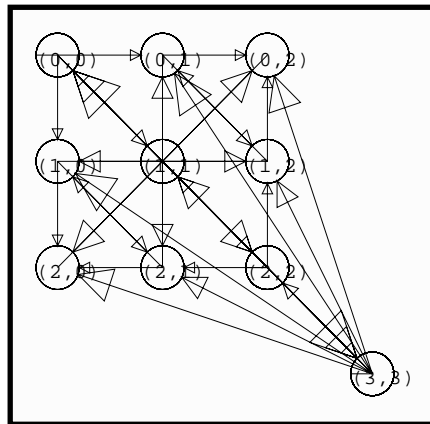


Figura 4.9: Diagrama de parejas de la regla 0056B9EF.

Para obtener las configuraciones que forman los ciclos dentro del diagrama de parejas, analizemos el diagrama de parejas de la Figura 4.9 y apartemos algunos casos.

La secuencia formada por los nodos $(0,2)$ y $(2,0)$ debe producir un bloque de evolución que tenga más de un ancestro, construyamos una secuencia de longitud tres dada de la siguiente manera:

$$(0,2) \rightarrow (2,0) \rightarrow (0,2)$$

si tomamos los elementos de la derecha de cada nodo formamos un bloque de evolución dado por

$$\begin{array}{c} \underbrace{020} \\ 22 \end{array}$$

que a su vez produce las células de evolución 22. Ahora bien si tomamos los elementos de la derecha obtendremos otro ancestro de la cadena 22 denotado de la siguiente manera

$$\begin{array}{c} \underbrace{202} \\ 22 \end{array}$$

por lo que se deduce que la cadena 22 tiene dos ancestros de evolución diferentes, los bloques 020 y 202 construidos manualmente en la Tabla 4.3 y representado en la Figura 4.6. Lo mismo sucede para la secuencia de nodos dada por los ciclos

$(0, 1) \leftrightarrow (1, 2)$ y $(1, 0) \leftrightarrow (2, 1)$. Esto significa que podemos obtener todos los ancestros de las secuencias del diagrama de parejas y a su vez estas secuencias las podemos incrementar de acuerdo al número de células que queramos analizar en una configuración de longitud l .

Las herramientas gráficas tales como el diagrama de subconjuntos y el diagrama de parejas son de mucha utilidad ya que nos ayudan a verificar nuestros resultados de una manera más rápido y seguro. El diagrama de transiciones reafirma estas propiedades revisando que el mapeo global coincida con los mapeos globales determinados anteriormente.

Finalmente proponemos un método para saber si un autómata celular tiene grados de reversibilidad o no. Nuestro método se basa en el producto matricial de las matrices de conectividad por estado del autómata en estudio. Las matrices de conectividad $N(k)$ para este autómata se construyen de la siguiente manera:

$$N(0) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad N(1) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & 1 \end{bmatrix}$$

$$N(2) = \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(3) = \begin{bmatrix} 1 & 1 & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Si elevamos al cuadrado la matriz $N(0)$ obtendremos el siguiente resultado

$$N(0)^2 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Lo que significa que para toda cadena formada por el estado 0 siempre obtendremos un ancestro único. Si la cadena esta formada por 00^* entonces su único ancestro va a estar formado por la cadena 33^* . Notemos que este comportamiento es el mismo para los estados 1 y 3.

$$N(1)^2 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(3)^2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Donde el único ancestro para la cadena 11^* esta formado por la cadena 22^* , para la cadena 33^* su único ancestro esta formada por la cadena 00^* . Esto implica

que la traza de la matriz debe ser igual a uno: por lo que la $Tr(N(0)) = 1$, la $Tr(N(1)) = 1$ y la $Tr(N(3)) = 1$. Entonces estas configuraciones tendrán comportamientos reversibles.

Ahora analicemos la matriz $N(k)$ cuando $k = 2$,

$$N(2)^2 = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

una vez más podemos verificar que el estado 2 por sí mismo tiene tres ancestros en configuraciones de tamaño dos formados por los estados 0, 1 y 2, como lo habíamos señalado en la Tabla 4.6, Figura 4.6 y Figura 4.9. Por lo que deducimos que el estado 2 es el estado que determina los grados de reversibilidad en esta regla.

Para verificar que esto es verdad sólo elevemos la matriz $N(2)$ sucesivamente.

$$N(2)^3 = \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(2)^4 = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

$$N(2)^5 = \begin{bmatrix} \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(2)^6 = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

4.3.5 Autómata (4,h) regla 016ED4BB

El autómata celular (4, h) regla 016ED4BB presenta grados de reversibilidad para configuraciones que no sean múltiplos de tres. En primer instancia obtengamos los productos matriciales por estado para determinar si alguno de los cuatro estados determina la reversibilidad para este autómata.

Primeramente construyamos sus matrices de conectividad.

$$N(0) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & 1 \end{bmatrix} \quad N(1) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \\ 1 & \cdot & \cdot & \cdot \end{bmatrix}$$

$$N(2) = \begin{bmatrix} \cdot & 1 & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(3) = \begin{bmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Todas las matrices cumplen con la condición de que su traza es igual a uno, lo que significa que las configuraciones de tamaño uno tienen un único ancestro. Ahora analizaremos cada uno de los estados y sus respectivas potencias.

$$N(0)^2 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad N(0)^3 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Las matriz $N(0)$ elevada a una n -ésima potencia es idempotente para toda $n \geq 2$, por lo que para toda cadena formada por el estado 00^* , esta cadena va a tener un único ancestro.

$$N(1)^2 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & 1 & 1 \\ 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(1)^3 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

La matriz de conectividad $N(1)$ alcanza la idempotencia cuando $n \geq 3$, y conserva la $Tr(N(1)) = 1$ para cualquier potencia de $N(1)^n$.

$$N(2)^2 = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

Para el estado 2 su matriz de conectividad alcanza rápidamente la idempotencia cuando $n = 2$. Mostrando un único ancestro para configuraciones 22^* .

$$N(3)^2 = \begin{bmatrix} 1 & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \quad N(3)^3 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

La matriz $N(3)$ alcanza la idempotencia cuando $n \geq 3$ igual que el estado 1 y la $Tr(N(3)) = 1$ lo que confirma un único ancestro para toda configuración formada por la cadena 33^* .

Todo parece indicar que el autómata celular es reversible para todas las configuraciones de tamaño par e impar. Sin embargo si analizamos su diagrama de parejas veremos que no es reversible.

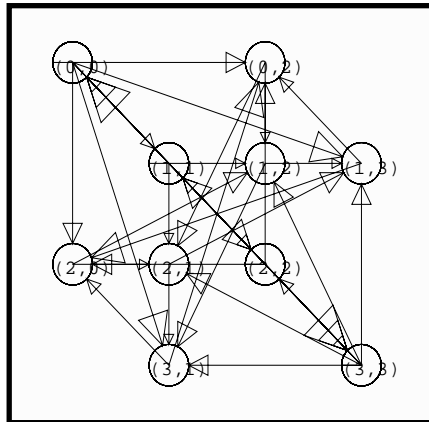


Figura 4.10: Diagrama de parejas de la regla 016ED4BB.

La Figura 4.10 ilustra la existencia de ciclos fuera de la diagonal, lo que implica que este autómata celular no es reversible. La explicación de este fenómeno es que los ciclos se forman por cadenas que están formadas por combinaciones de los mismos estados. Si efectuamos todas las combinaciones posibles cuando $l = 2$, entonces tenemos dieciseis casos posibles donde la $Tr(N(k)_{i,j}) = 1 \forall k \in \Sigma$.

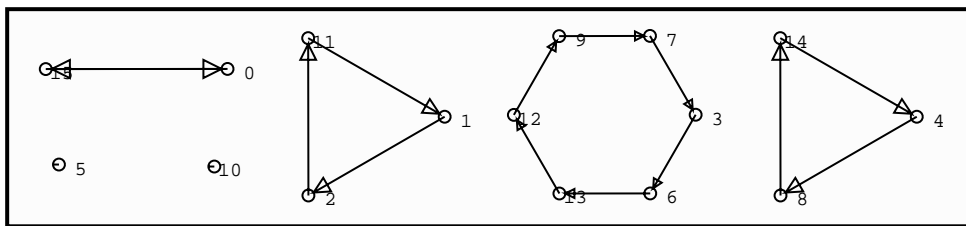


Figura 4.11: Árboles topológicos con $l = 2$.

La Figura 4.11 ilustra todas las configuraciones c_2 donde el mapeo global es biyectivo y lógicamente también es reversible para este valor. Por que sólo existe una configuración que construye a otra configuración para todos los estados globales que podemos formar con $l = 2$. Los ciclos 5 y 10 son de longitud uno, esto quiere decir que la evolución para este caso va a ser siempre 5 ó 10 en todo el diagrama de evoluciones.

Ahora analizemos la secuencia formada por los estados 123 donde sus productos matriciales los representamos de la siguiente manera:

$$N(1)N(2)N(3) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

podemos ver que la $Tr(N(123)) = 2$ lo que implica que la configuración tiene dos ciclos que a su vez indican dos ancestros para dicha cadena. Estas cadenas están formadas por los bloques de evolución 1221 y 3013, sus ancestros de la cadena 123 son los estados 221 y 013 respectivamente.

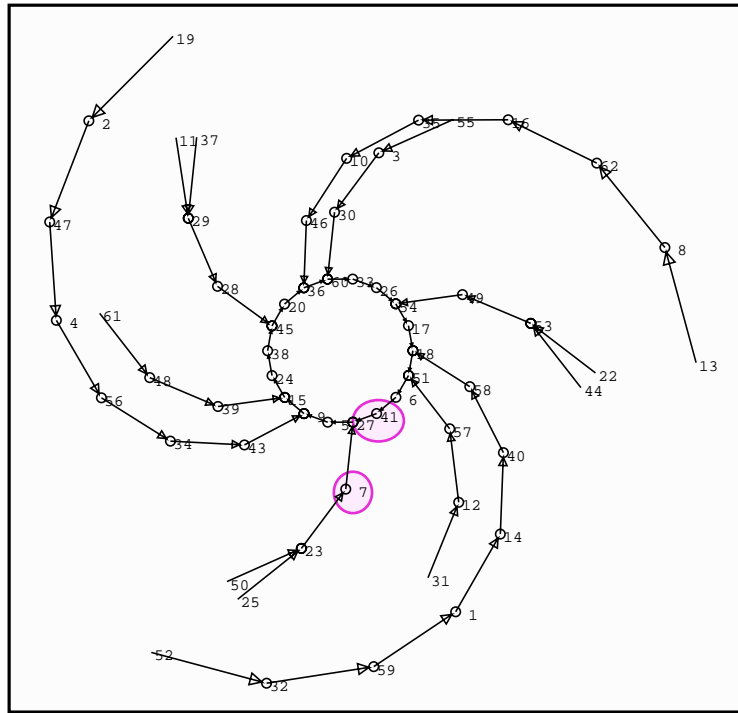


Figura 4.12: Diagrama de transiciones con $l = 3$.

La Figura 4.12 ilustra uno de los árboles topológicos cuando $l = 3$ y podemos verificar que la cadena 123 representada por el estado global 27 tiene dos ancestros, los estados globales 41 y 7 representadas por las cadenas 221 y 013 respectivamente.

Por lo que deducimos que para determinar los grados de reversibilidad de un ancho específico tenemos que efectuar todas las combinaciones posibles determinadas por k . Y el orden de n determinará el tamaño de l para toda configuración $c_l \in C_F(\Sigma)$.

Comprobando los grados de reversibilidad para este autómata revisemos la secuencia 102303 que debe de tener más de un ancestro, ya que l es un múltiplo de tres en este caso.

$$N(102303) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

La matriz de conectividad para la cadena representada por los estados 102303 presenta una $Tr(N(102303)) = 2$ lo que implica que dicha configuración posee dos ancestros y no es reversible cuando $l = 6$.

4.3.6 Algoritmo

El algoritmo esta propuesto para cualquier autómata celular lineal de orden (k, r) .

1. Obtener las matrices de conectividad $N(k)_{i,j}$ para cualquier autómata celular de orden (k, r) . Donde la función de transición φ determina el traslape y el número de elementos para dicha matriz.
2. Efectuar el producto matricial de estas matrices de conectividad, calculando todas las combinaciones posibles dadas por k^l . Donde el valor de l determina el número de combinaciones posibles restringido por el orden de k .
3. Calcular la traza de cada uno de estos productos matriciales y comprobar que la $Tr(N(k)_{i,j}^l) = 1 \forall k \in \Sigma$, para determinar la reversibilidad de una l dada.
4. El algoritmo para hasta una l específica.

Evidentemente conforme l se vaya incrementando el cálculo es cada vez más laborioso. Y si el orden de k también se incrementa, el cómputo es muchísimo más lento.

Capítulo 5

Conclusiones

Los autómatas celulares que presentan grados de reversibilidad para ciertas configuraciones $c_l \in C_F(\Sigma)$, deben tener algunas de las siguientes características:

- El autómata celular debe tener un mapeo suryectivo. Esto implica que todas las configuraciones del autómata celular deben tener forzosamente al menos un ancestro.
- El diagrama de subconjuntos debe carecer de rutas principales que vayan del conjunto máximo al conjunto mínimo. Esto implica que el autómata celular debe carecer de configuraciones pertenecientes al Jardín del Edén.
- El diagrama de parejas debe presentar ciclos fuera de la diagonal principal. Lo que implica que el autómata celular tiene múltiples ancestros.
- La matriz de de Bruijn debe tener el mismo número de elementos por estado en toda la matriz. Lo que implica que todos los estados tienen al menos un ancestro.
- Las matrices de conectividad deben tener una traza igual a uno por cada uno de los estados posibles que tenga el autómata celular. Esto implica que el autómata celular debe cumplir que la traza de $Tr(N(k)_{i,j}) = 1 \forall k \geq 2$, efectuando todas las combinaciones posibles determinadas por k dentro del producto matricial de sus respectivas matrices.
- Los grados de reversibilidad deben ser originados por un estado o por combinaciones de todos los estados que conforman el autómata celular en estudio. Esto implica que no forzosamente un sólo estado origine estos comportamientos, ya que pueden ser originados bajo ciertas combinaciones de los mismos estados o viceversa.
- Los nodos del diagrama de transiciones siempre tendrán al menos un ancestro. Esto implica que si los árboles topológicos cuando l tiene un valor determinado presentan hojas, estas hojas tendrán un ancestro o más cuando l sea mayor a dicho valor. Aquellas configuraciones que cumplan con la condición cinco,

implica que dichos árboles topológicos son sólo ciclos de longitud mayor igual a uno.

El estudio que se llevó a cabo se enfocó para el caso $(4, h)$, sin embargo para el caso $(2, 1)$ tenemos que la regla 45 muestra grados de reversibilidad para configuraciones de tamaño impar, mientras que la regla 105 muestra grados de reversibilidad para configuraciones que no sean múltiplos de tres.

Para el caso $(3, h)$ la única regla que presenta grados de reversibilidad es la regla 1313, para configuraciones de tamaño impar. Como habíamos señalado en la Sección 4.3.3 el cálculo se va complicando conforme k, r y l , se van incrementando.

Finalmente presentamos una lista de reglas para el caso $(4, h)$ que muestran grados de reversibilidad para ciertos valores de l . Esta lista fue revisada manualmente y ordenada de acuerdo a su cluster mínimo correspondiente por cada una de las reglas. Para comprobar esta lista se anexó un apéndice que ilustra todos los árboles topológicos que muestran sus respectivos grados de reversibilidad para todas las reglas que cumplen con estas condiciones.

Regla de evolución	valor de l
0056B9EF	3 5
0056F9AF	3 5
005BADF6	3 5
005BBDE6	3 5
005BEDB6	3 5
00669FF9	3 5
00679EF9	3 5
0552F8AF	3 5
0158AFF6	3 5
0156F8AF	3 5
016ED4BB	2 4 5
055BACF2	3 5
0196E87F	3 5
05AF14EB	3 5
0156E8BF	3 5
0156B8EF	3 5
0154BAEF	3 5
016AB4DF	2 4 5
0168D6BF	3 5
019BEC76	3 5
05936CFA	3 5
1BB1E44E	2 4 5

Tabla 5.1: Reglas con grados de reversibilidad.

En total existen 685 reglas suryectivas de las cuales 22 reglas tienen grados de reversibilidad para valores de l igual a dos, tres, cuatro y cinco. El cálculo se efectuó hasta un valor de $l = 6$ por eso el valor de l no va más allá de cinco.

Evidentemente estos cálculos sugieren un método computacional para todos los demás casos $(2, t)$, $(2, 2)$, $(2, f)$, $(2, 3)$, $(2, n)$, $(2, 4)$, $(2, s)$, $(2, 5)$, $(3, 1)$, $(3, t)$, $(3, 2)$, $(4, 1)$, $(4, t)$, $(4, 2)$, $(5, h)$, $(5, t)$, $(6, h)$, $(6, 1)$, $(7, h)$, $(7, 1)$, $(8, h)$, $(8, 1)$, $(9, h)$ y $(9, 1)$.

El estudio sobre los autómatas celulares lineales reversibles ha sido muy importante en los últimos tiempos dado por las contribuciones de Hillman, Nasu, Moore, Hedlund, Tommaso Toffoli, Norman Margolus, McIntosh, Fredkin, Jarkko Kari, S. Amoroso, Y. N. Patt, Erica Jen, por mencionar algunos. Todos estos estudios han ido formando una mejor idea para explicar el surgimiento y comportamiento de tal fenómeno. Buscando una formalización matemática por demás general, aunque esta todavía esta por verse.

Los grados de reversibilidad son un derivado de los autómatas celulares reversibles, su estudio es importante por el hecho de que podemos clasificar un autómata reversible aunque este no lo sea. Es importante determinar con detalle estas relaciones y determinar con claridad sus respectivas características.

A los autómatas celulares reversibles se les han encontrado aplicaciones tales como la simulación de reacciones químicas, procesos biológicos, simulación de epidemias, simulación de incendios forestales, simulación de procesos matemáticos, simulación de cómputo en paralelo, simulación del control de tráfico y encriptación de datos.

Siendo un poco aventurado la aplicación de encriptación de datos puede ser enfocada a los autómatas que presentan grados de reversibilidad, dificultando aún más la descripción de tal información ya que ahora intervendría un factor más y este factor es l . Esto significa que podríamos trabajar con varias configuraciones de varias l 's dificultando el proceso para descifrar tal configuración inicial.

No cabe duda que la teoría de autómata celular aún tiene varios problemas por resolver, sin embargo es una teoría que se presta al uso de otras herramientas tales como la estadística, la probabilidad, el cálculo diferencial e integral, ecuaciones diferenciales, teoría de gráficas, métodos numéricos, estructuras algebraicas, geometría proyectiva, algebra booleana, algebra lineal, teoría de conjuntos, expresiones regulares, procesos estocásticos, simulación y ciencias de la computación. Como habíamos señalado desde la Capítulo 1 es una teoría multidisciplinaria.

Finalmente señalamos que la teoría de autómata celular tiene un amplio campo de estudio y consecuentemente un amplio campo de futuras aplicaciones. Por lo que se siguen obteniendo nuevos e importantes avances así como nuevas aplicaciones.

Apéndice A

Árboles Topológicos (4,h)

A continuación ilustramos todos los árboles topológicos que presentan grados de reversibilidad para el autómata celular (4,h). Clasificados por su cluster mínimo, la longitud de las configuraciones calculadas es de tamaños dos, tres, cuatro, cinco y seis. Estos diagramas se calcularon con el programa NXLC4U4h.

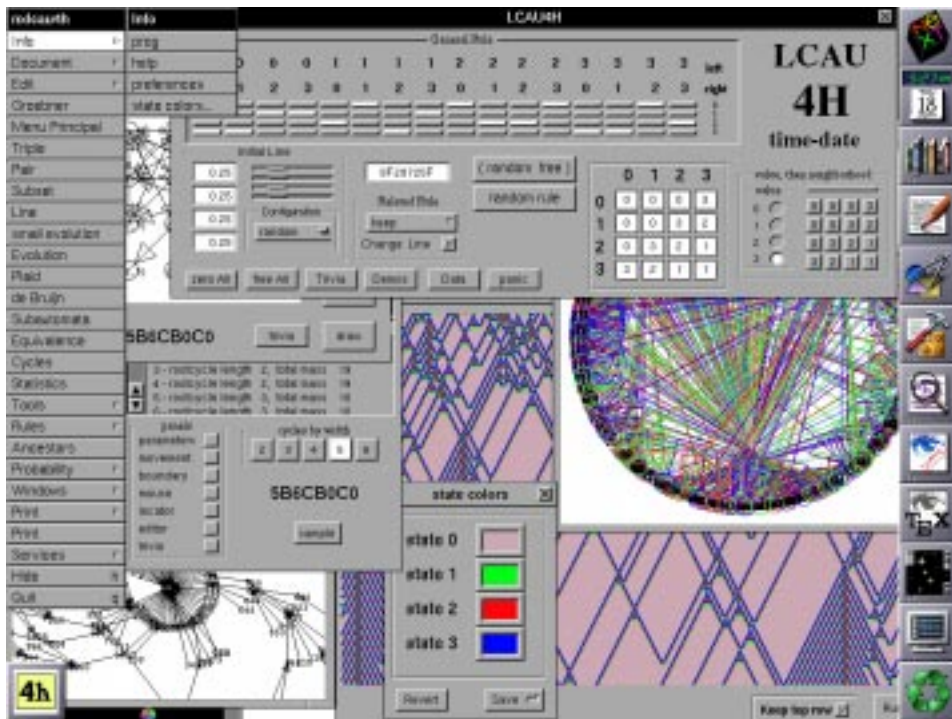
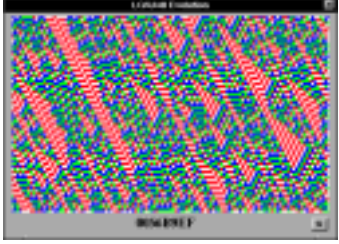
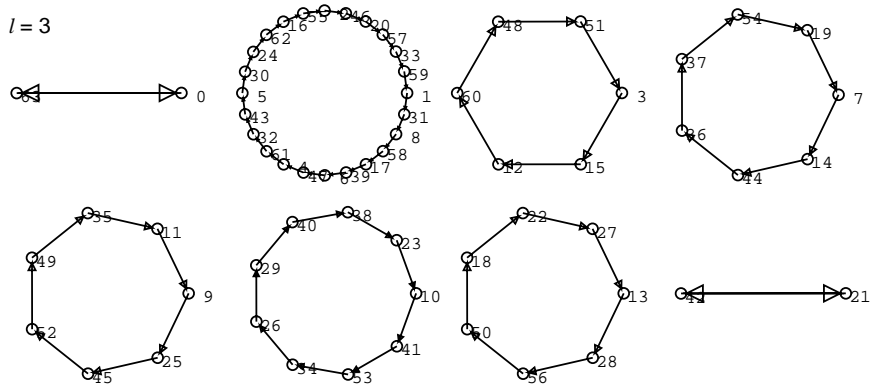
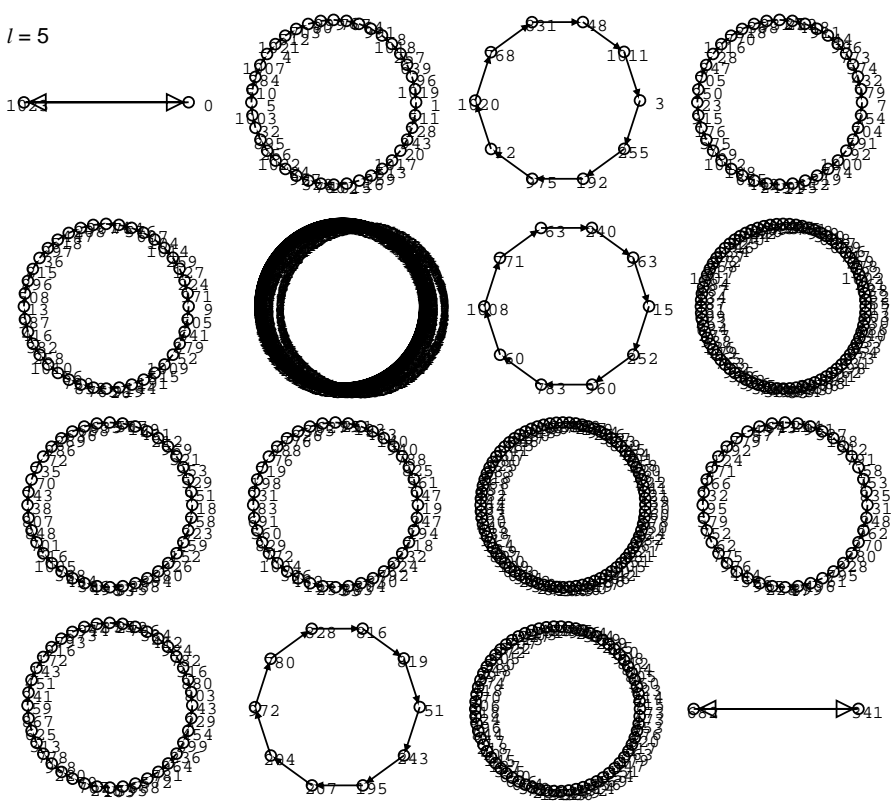


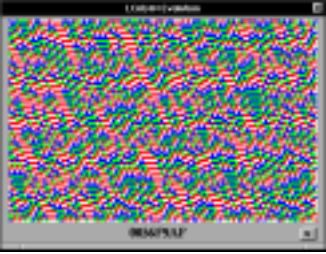
Figura A.1: NXLC4U4h con algunas opciones.

A.1 Regla 0056B9EF

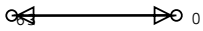
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> </tr> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">2</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> </tr> <tr> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">1</td> </tr> <tr> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">0</td> </tr> </table>	0	1	2	3	0	3	3	2	1	1	2	3	2	2	1	1	3	0	0	0	
0	1	2	3																		
0	3	3	2																		
1	1	2	3																		
2	2	1	1																		
3	0	0	0																		
<p>$l = 3$</p> 																					
<p>$l = 5$</p> 																					


A.2 Regla 0056F9AF

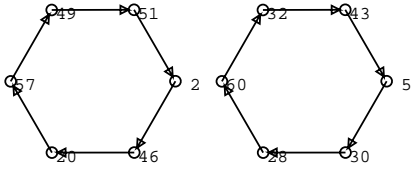
	0	1	2	3
0	3	3	2	2
1	1	2	3	3
2	2	1	1	1
3	0	0	0	0

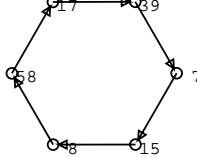



$l = 3$



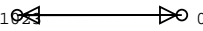


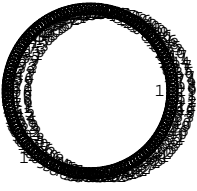


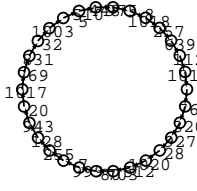


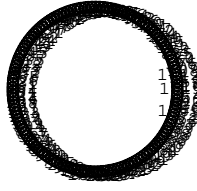


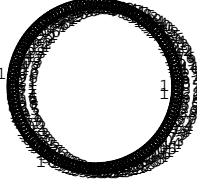
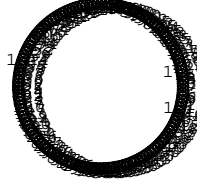
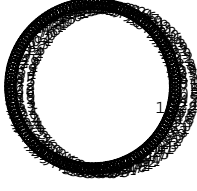
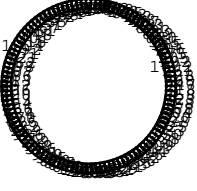
$l = 5$

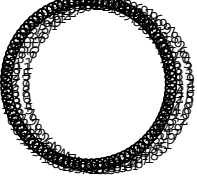


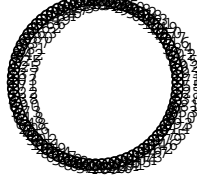


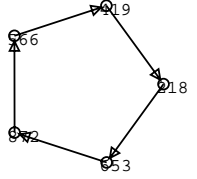





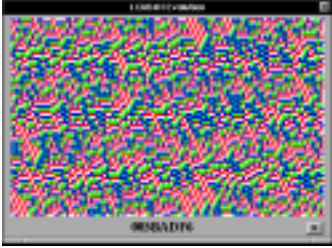
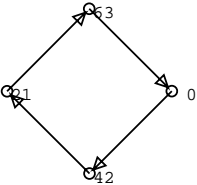

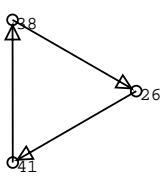
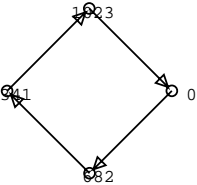
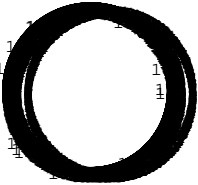
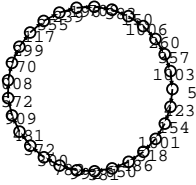
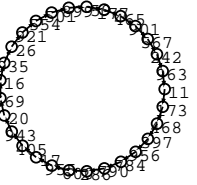
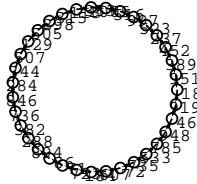
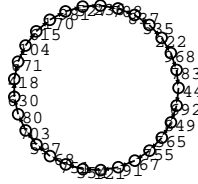
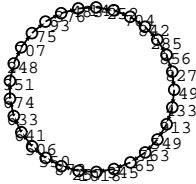
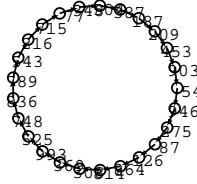
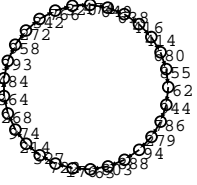




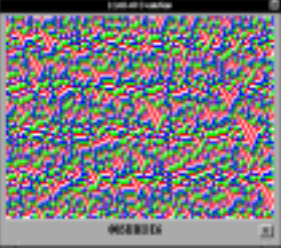
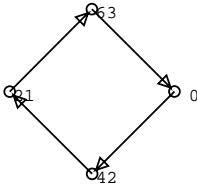
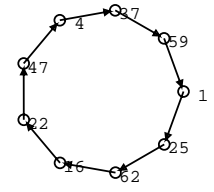
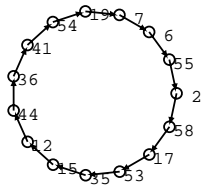
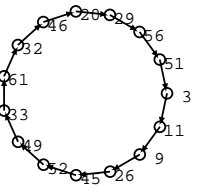
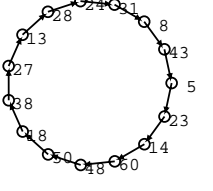
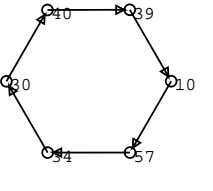
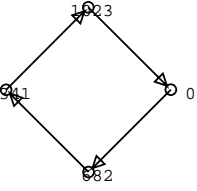
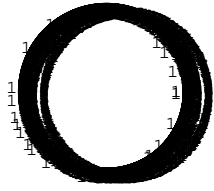
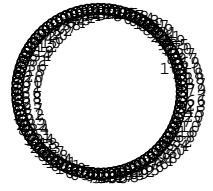
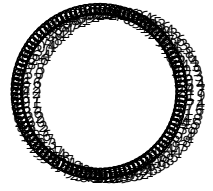
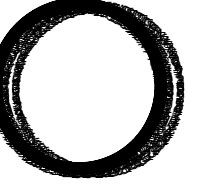
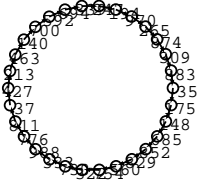




A.3 Regla 005BADF6

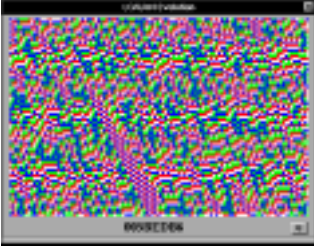
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">3</td> </tr> <tr> <td style="padding: 5px;">0</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">3</td> </tr> <tr> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">2</td> </tr> <tr> <td style="padding: 5px;">2</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="padding: 5px;">3</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> </tr> </table>	0	1	2	3	0	2	1	3	1	1	3	2	2	3	2	1	3	0	0	0	
0	1	2	3																		
0	2	1	3																		
1	1	3	2																		
2	3	2	1																		
3	0	0	0																		
<p>$l = 3$</p> <div style="display: flex; justify-content: space-around; align-items: center;">    </div>																					
<p>$l = 5$</p> <div style="display: flex; justify-content: space-around; align-items: center;">     </div> <div style="display: flex; justify-content: space-around; align-items: center; margin-top: 20px;">     </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 20px;">  </div>																					

A.4 Regla 005BBDE6

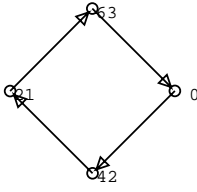
<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 5px;">0</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">3</td> </tr> <tr> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">2</td> </tr> <tr> <td style="padding: 5px;">2</td> <td style="padding: 5px;">3</td> <td style="padding: 5px;">2</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="padding: 5px;">3</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> </tr> </table>	0	2	1	2	3	1	1	3	3	2	2	3	2	1	1	3	0	0	0	0	
0	2	1	2	3																	
1	1	3	3	2																	
2	3	2	1	1																	
3	0	0	0	0																	
<p>$l = 3$</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>																					
<p>$l = 5$</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>																					

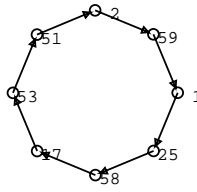
A.5 Regla 005BEDB6

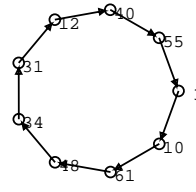
	0	1	2	3
0	2	1	3	2
1	1	3	2	3
2	3	2	1	1
3	0	0	0	0

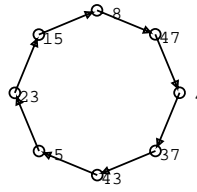


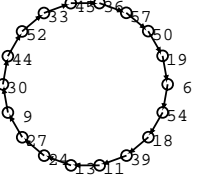
l = 3

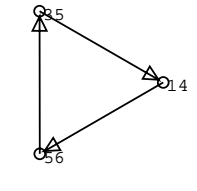


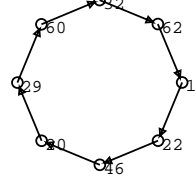


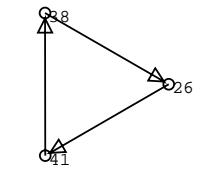






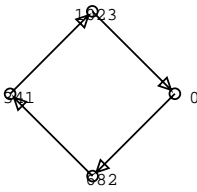


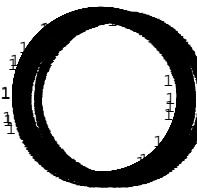


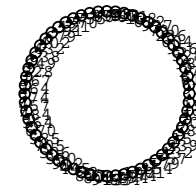


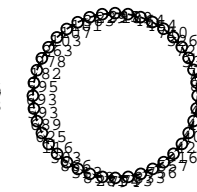
⊙₂₈
⊙₄₉
⊙₇

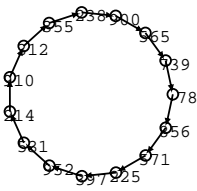
l = 5



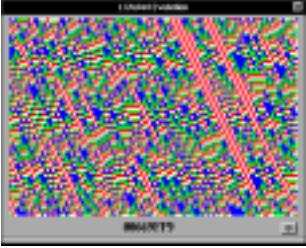
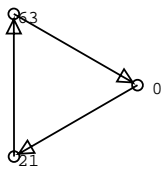
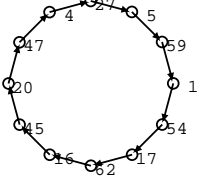
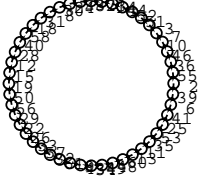
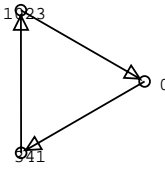
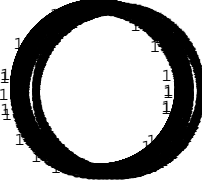
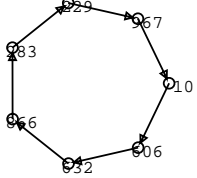
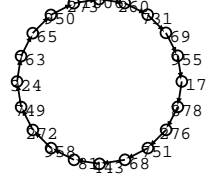
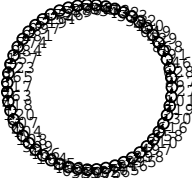
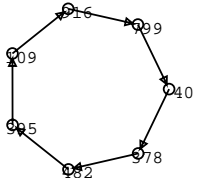
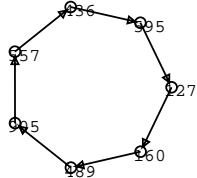
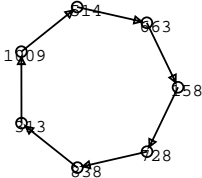
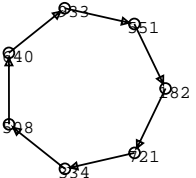






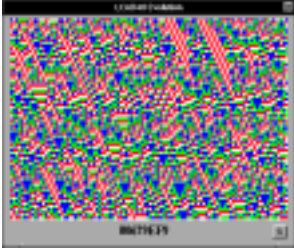


A.6 Regla 00669FF9

<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> </tr> <tr> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">1</td> </tr> <tr> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> </tr> <tr> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">0</td> </tr> </table>	0	1	2	3	0	1	2	3	1	3	3	1	2	2	1	2	3	0	0	0	
0	1	2	3																		
0	1	2	3																		
1	3	3	1																		
2	2	1	2																		
3	0	0	0																		
<p>$l = 3$</p> <div style="display: flex; justify-content: space-around; align-items: center;">    \odot_{12} </div>																					
<p>$l = 5$</p> <div style="display: flex; flex-wrap: wrap; justify-content: space-around;"> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin: 5px;">  </div> <div style="margin-left: 20px;">\odot_{12}</div> </div>																					

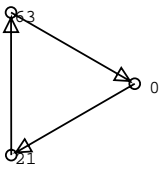
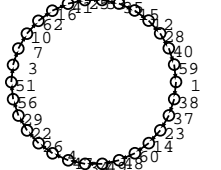
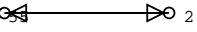
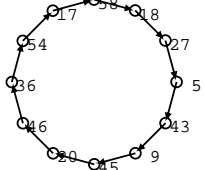
A.7 Regla 00679EF9

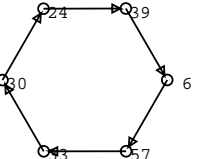
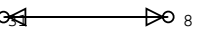
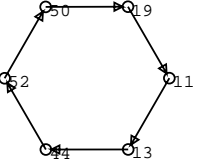
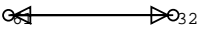
	0	1	2	3
0	1	2	3	3
1	2	3	1	2
2	3	1	2	1
3	0	0	0	0



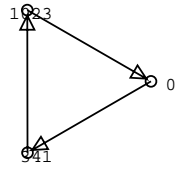
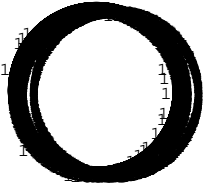
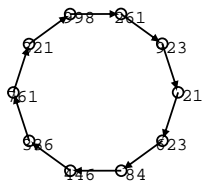
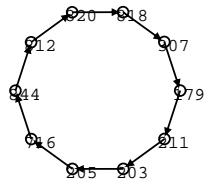
$l = 3$

\mathbb{O}_{42}

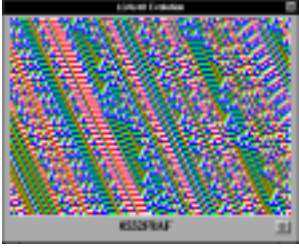
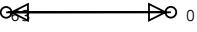
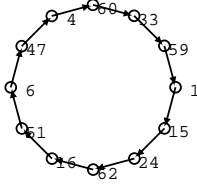
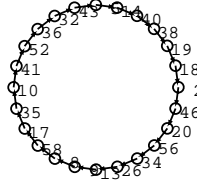
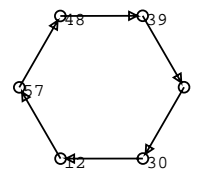
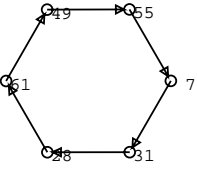
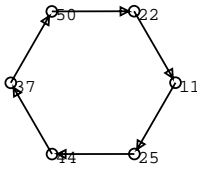
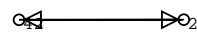
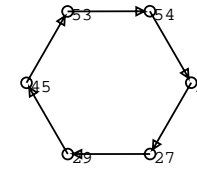
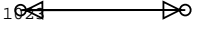
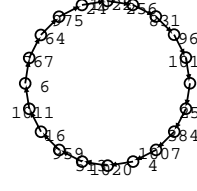
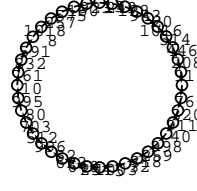
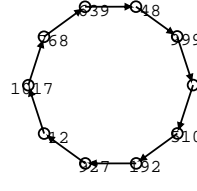
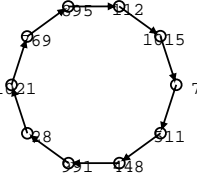
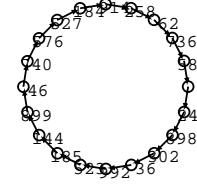
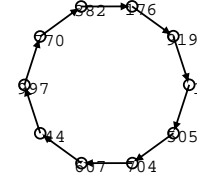
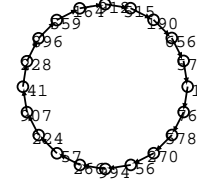
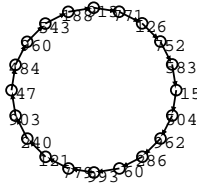
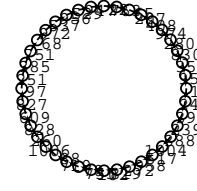
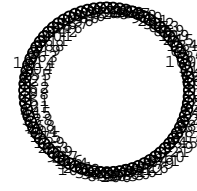
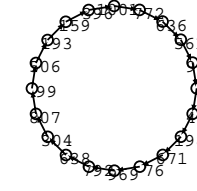





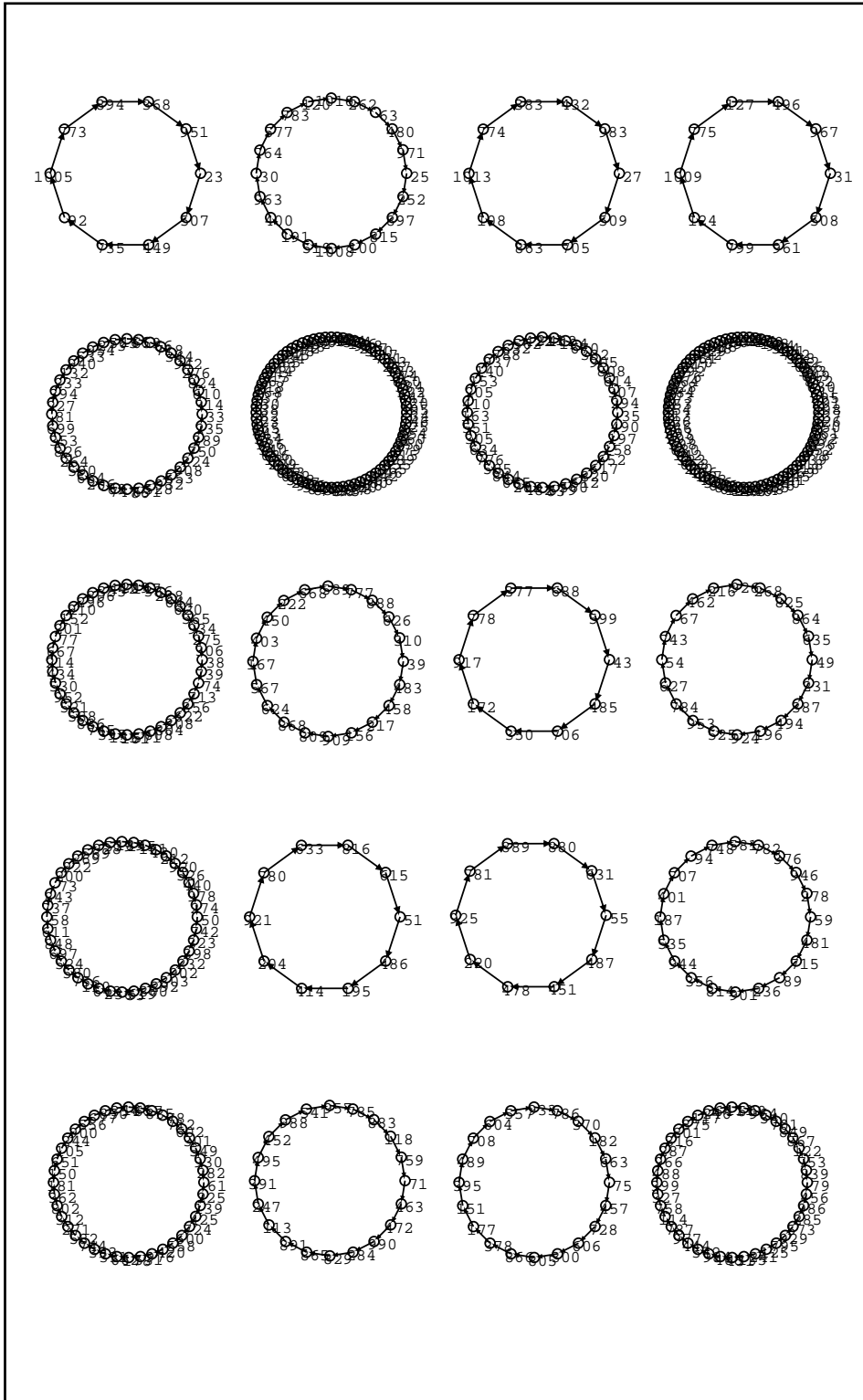
$l = 5$

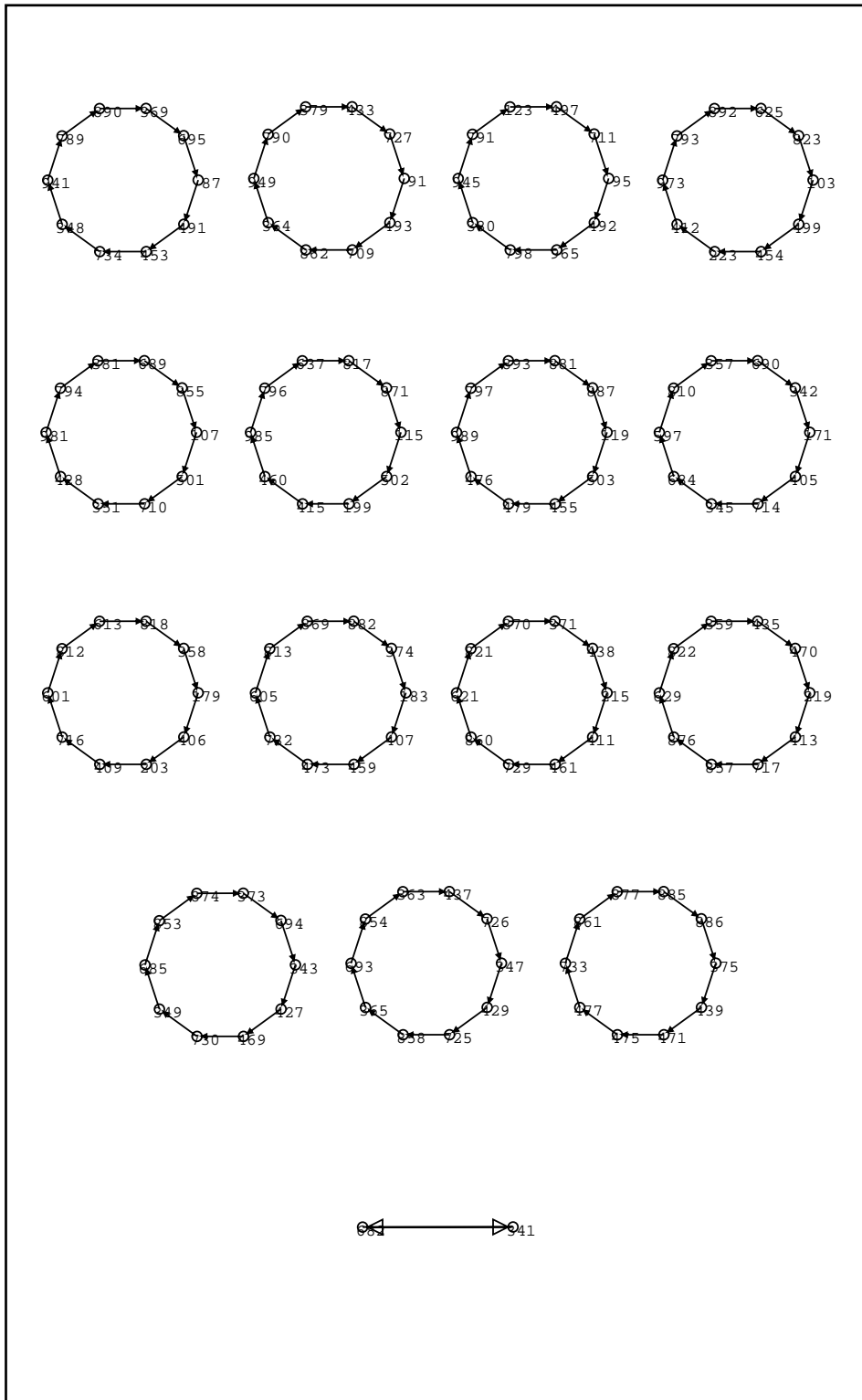





\mathbb{O}_{82}

A.8 Regla 0552F8AF

<table border="1" style="margin: auto;"> <tr> <td></td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> </table>		0	1	2	3	0	3	3	2	2	1	0	2	3	3	2	2	0	1	1	3	1	1	0	0	
	0	1	2	3																						
0	3	3	2	2																						
1	0	2	3	3																						
2	2	0	1	1																						
3	1	1	0	0																						
<p>$l = 3$</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start; margin-top: 20px;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>																										
<p>$l = 5$</p> <div style="display: flex; justify-content: space-around; align-items: flex-start;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start; margin-top: 20px;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div> <div style="display: flex; justify-content: space-around; align-items: flex-start; margin-top: 20px;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>																										





A.9 Regla 0158AFF6

	0	1	2	3
0	2	1	3	3
1	3	3	2	2
2	0	2	1	1
3	1	0	0	0

l = 3

⊙ 6

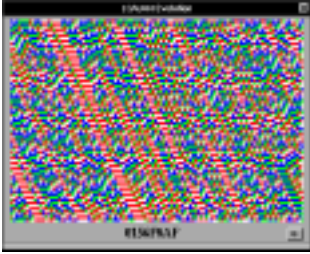
⊙ 24

⊙ 33

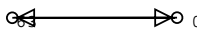
l = 5

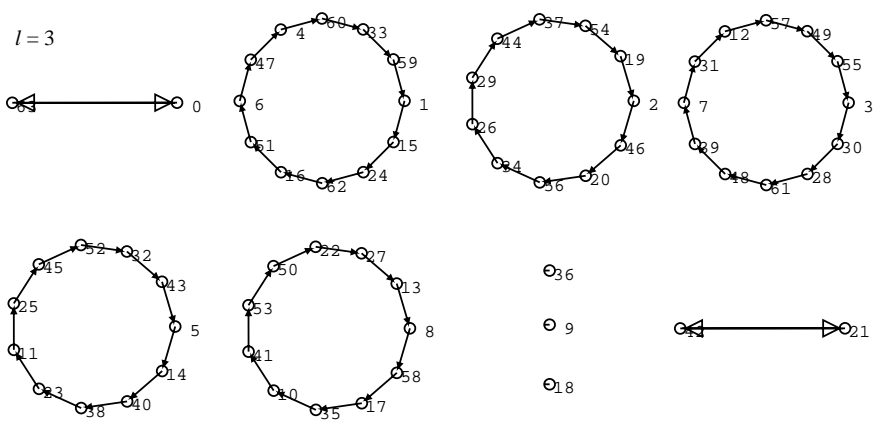
A.10 Regla 0156F8AF

	0	1	2	3
0	3	3	2	2
1	0	2	3	3
2	2	1	1	1
3	1	0	0	0

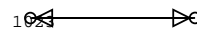


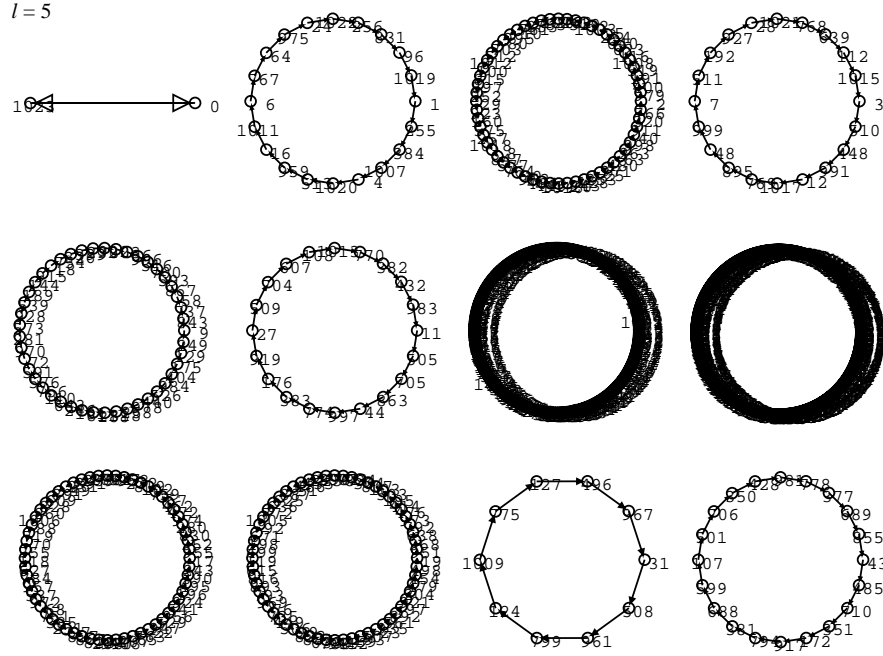
$l = 3$

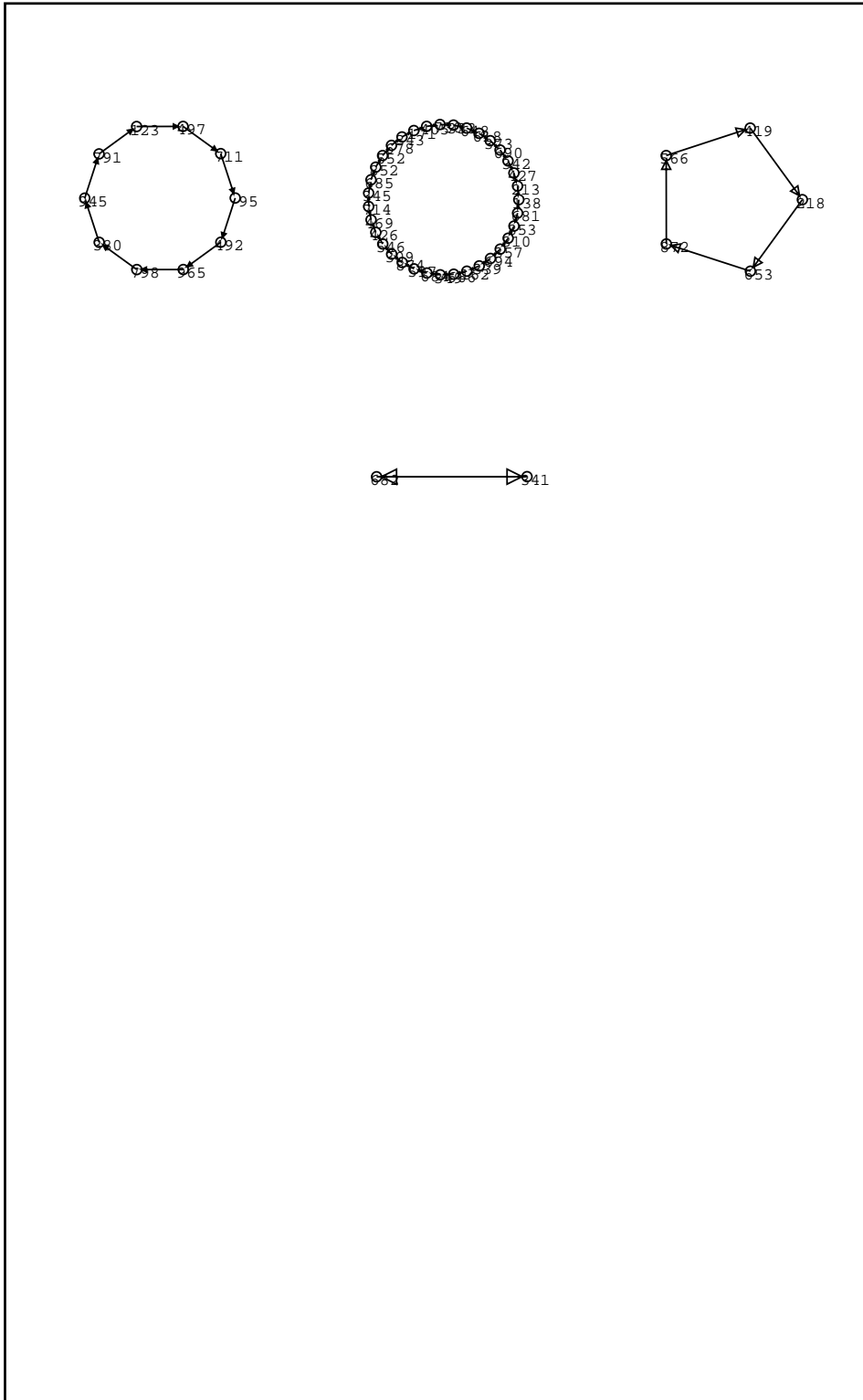




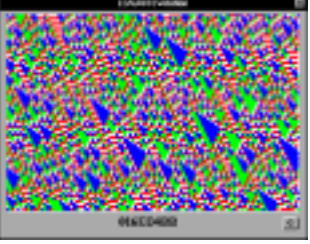
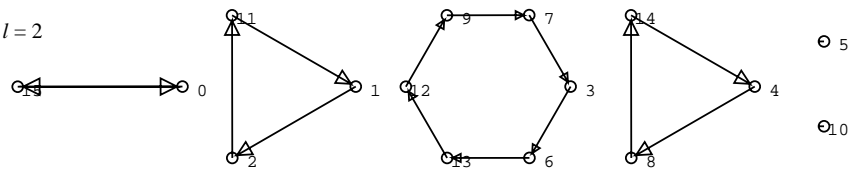
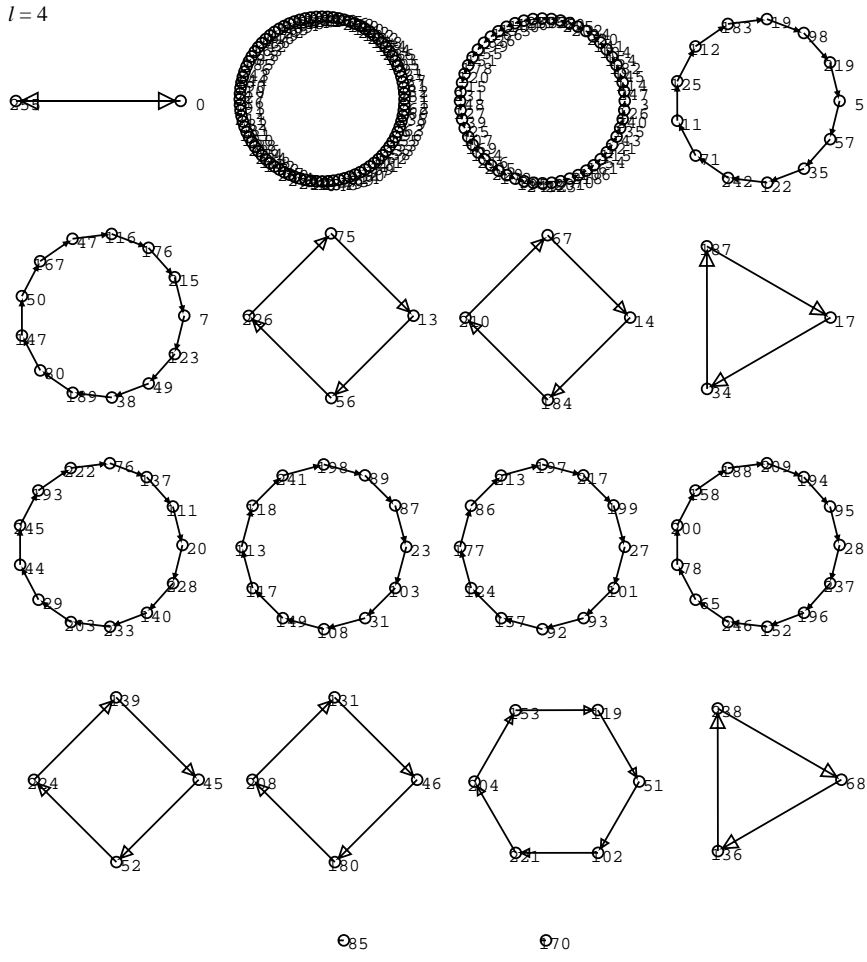
$l = 5$

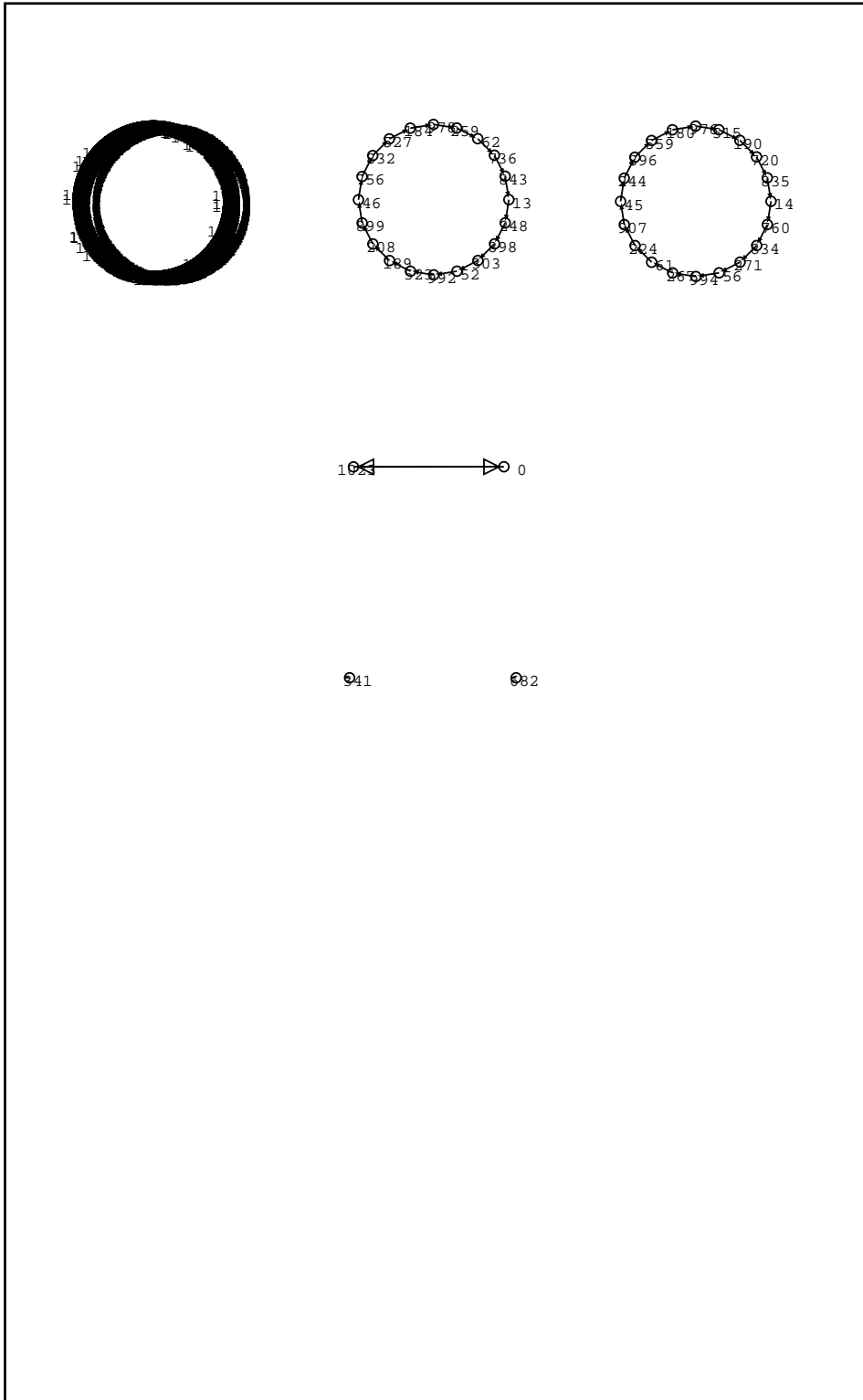




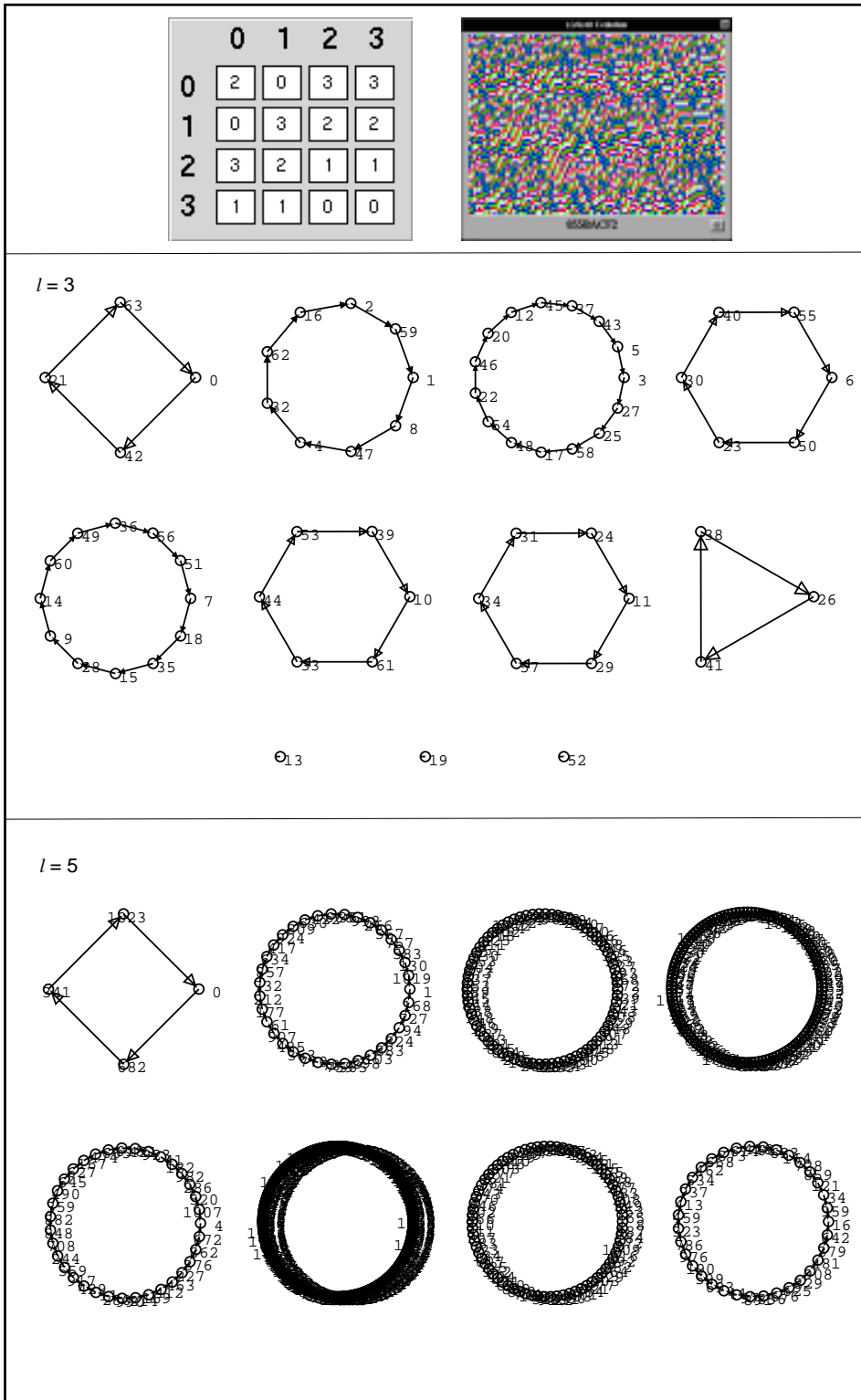


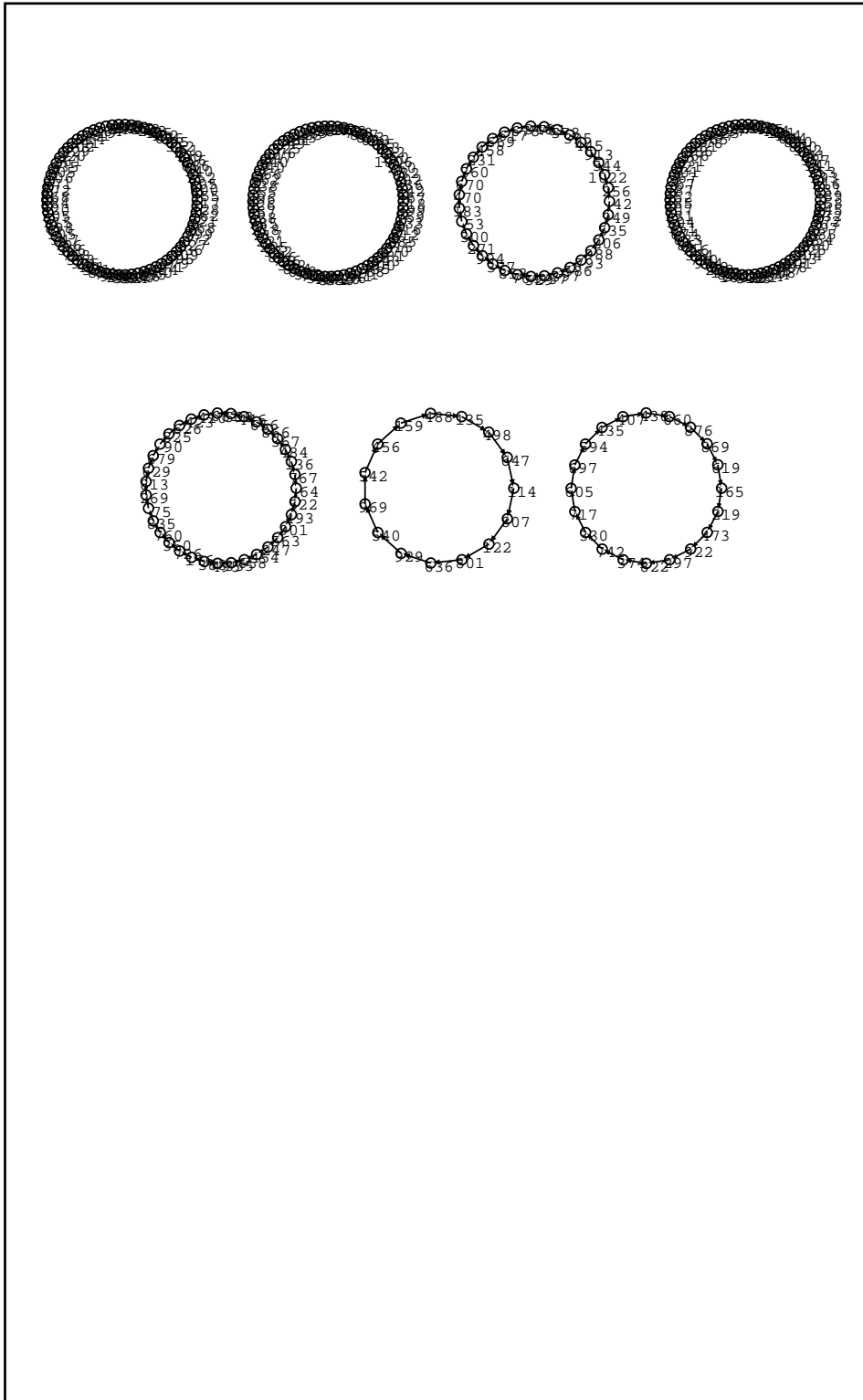
A.11 Regla 016ED4BB

<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">3</td> </tr> <tr> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">3</td> </tr> <tr> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">3</td> </tr> <tr> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">2</td> </tr> <tr> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> <td style="padding: 2px 5px;">0</td> </tr> </table>	0	1	2	3	0	3	2	3	1	0	1	3	2	2	3	2	3	1	0	0	
0	1	2	3																		
0	3	2	3																		
1	0	1	3																		
2	2	3	2																		
3	1	0	0																		
<p>$l = 2$</p> 																					
<p>$l = 4$</p> 																					



A.12 Regla 055BACF2





A.13 Regla 0196E87F

	0	1	2	3
0	3	3	3	1
1	0	2	2	3
2	2	1	1	2
3	1	0	0	0

l = 3

0

1

2

3

4

5

6

7

10

11

21

22

l = 5

0

1

2

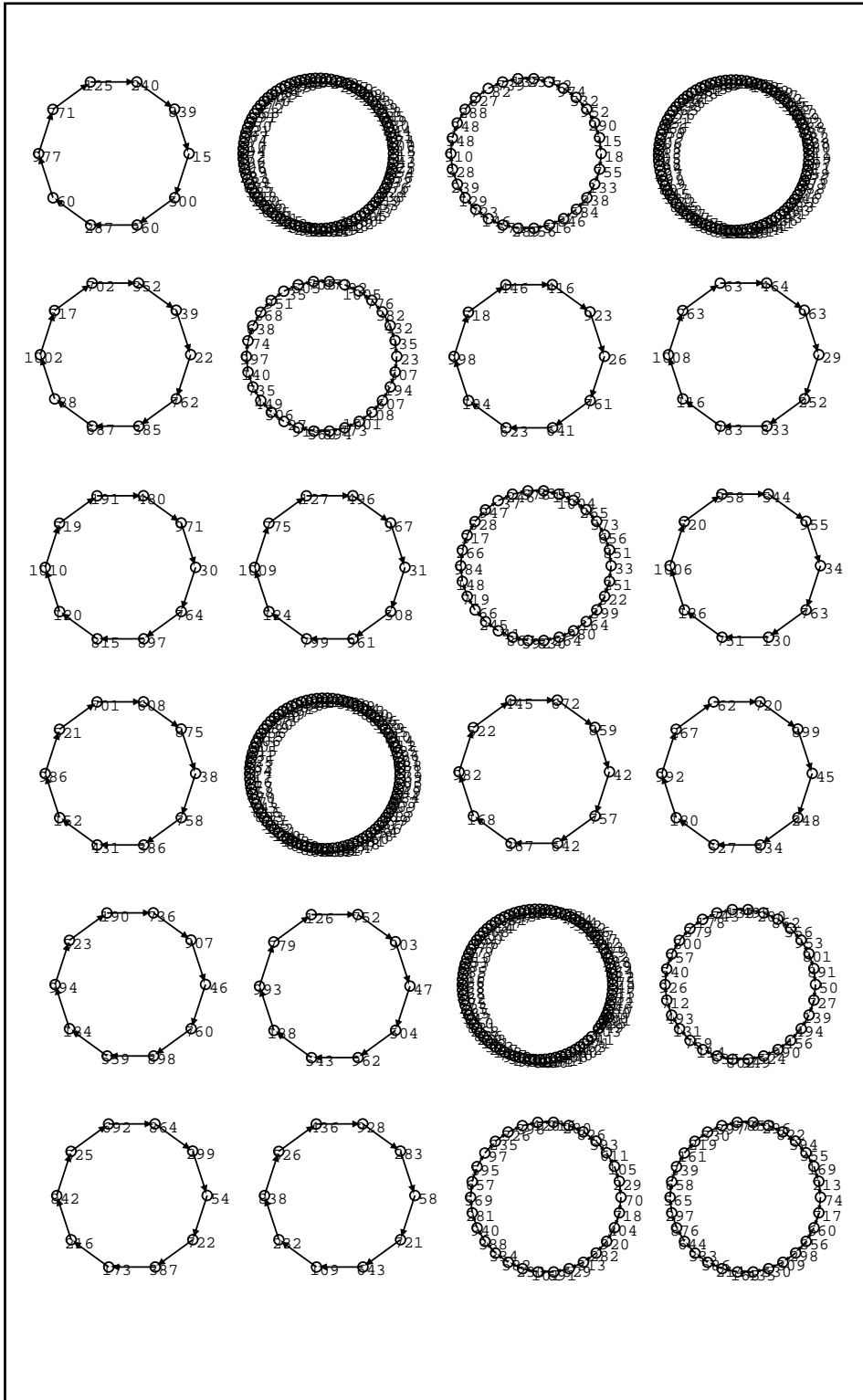
3

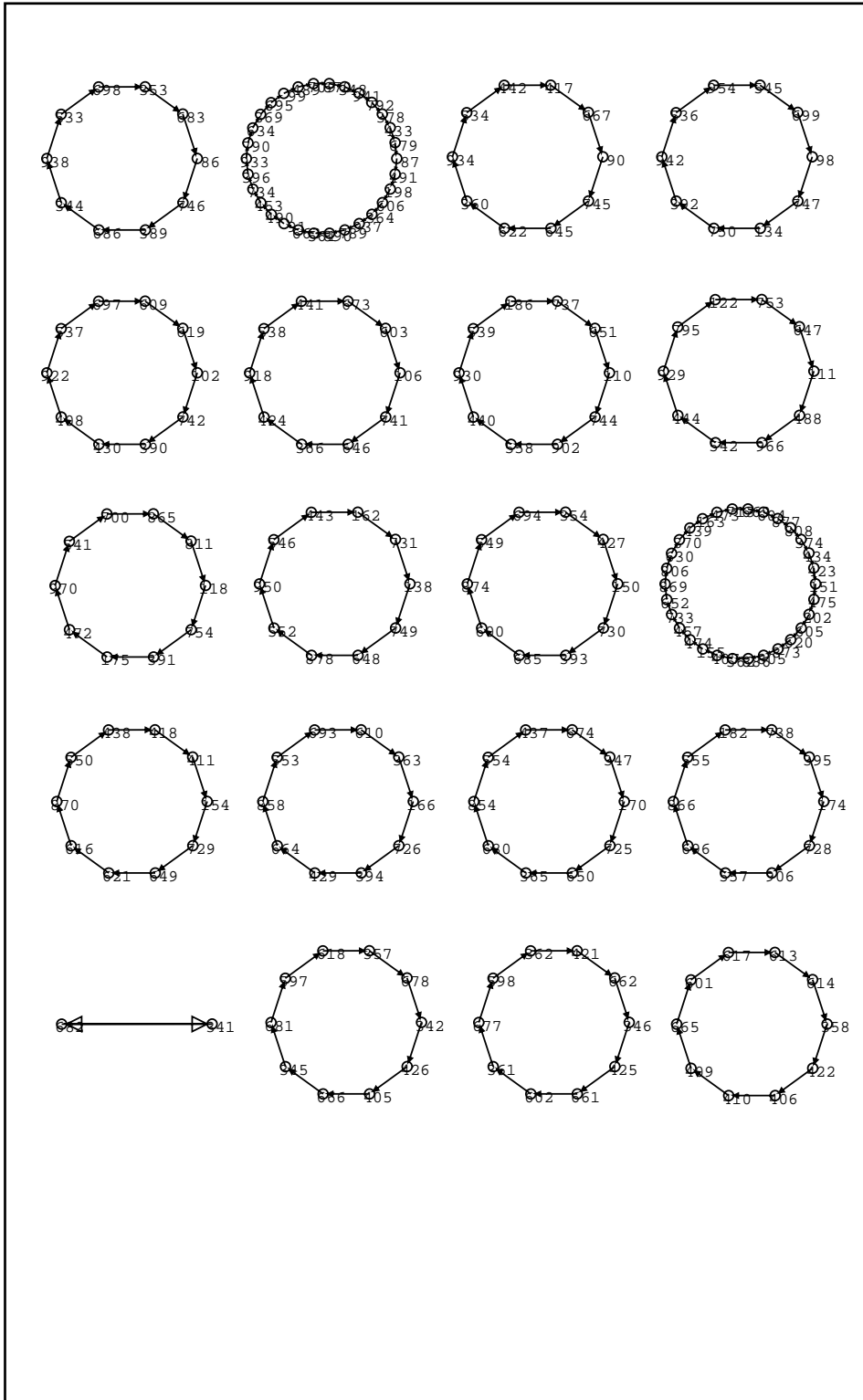
4

5

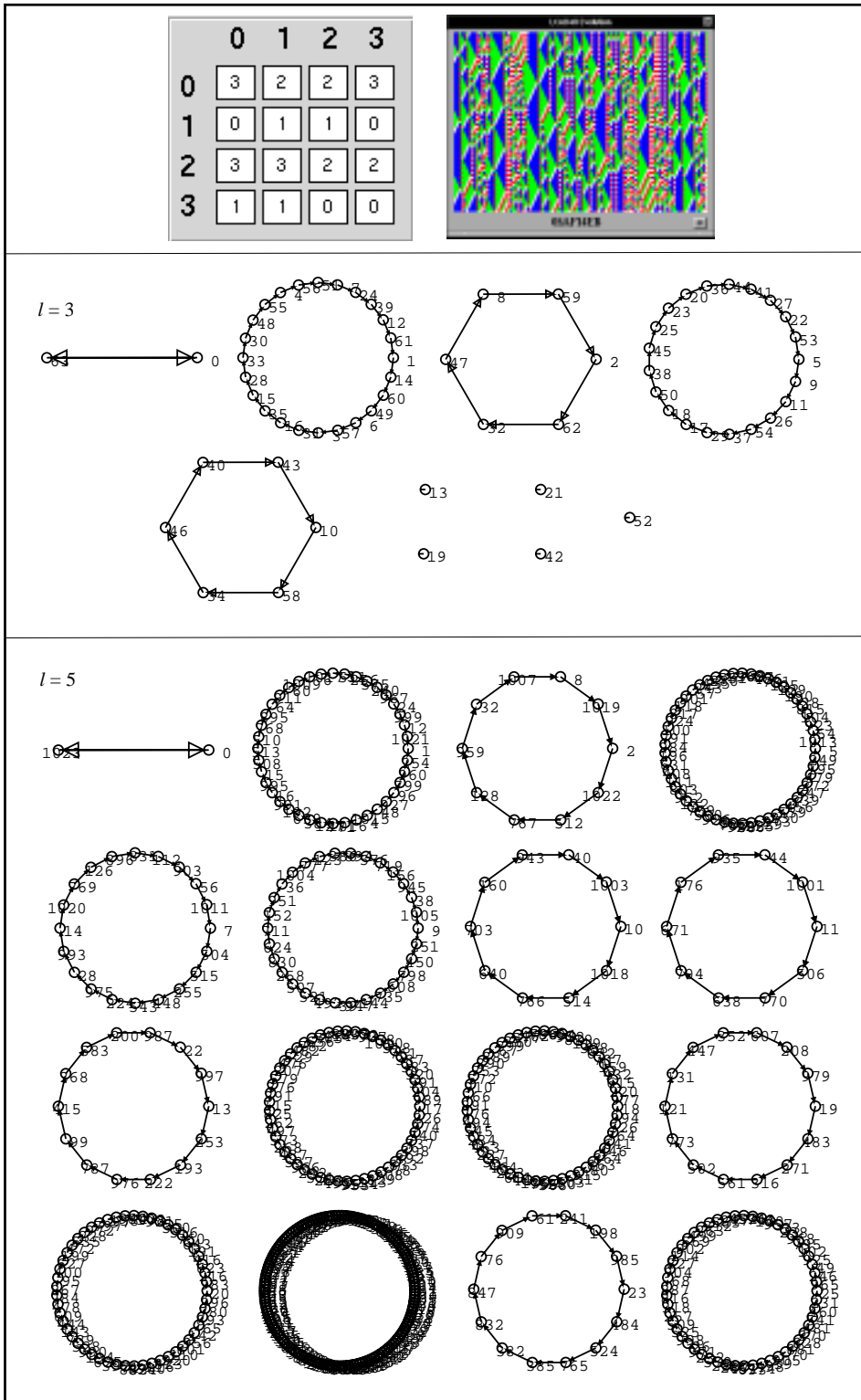
6

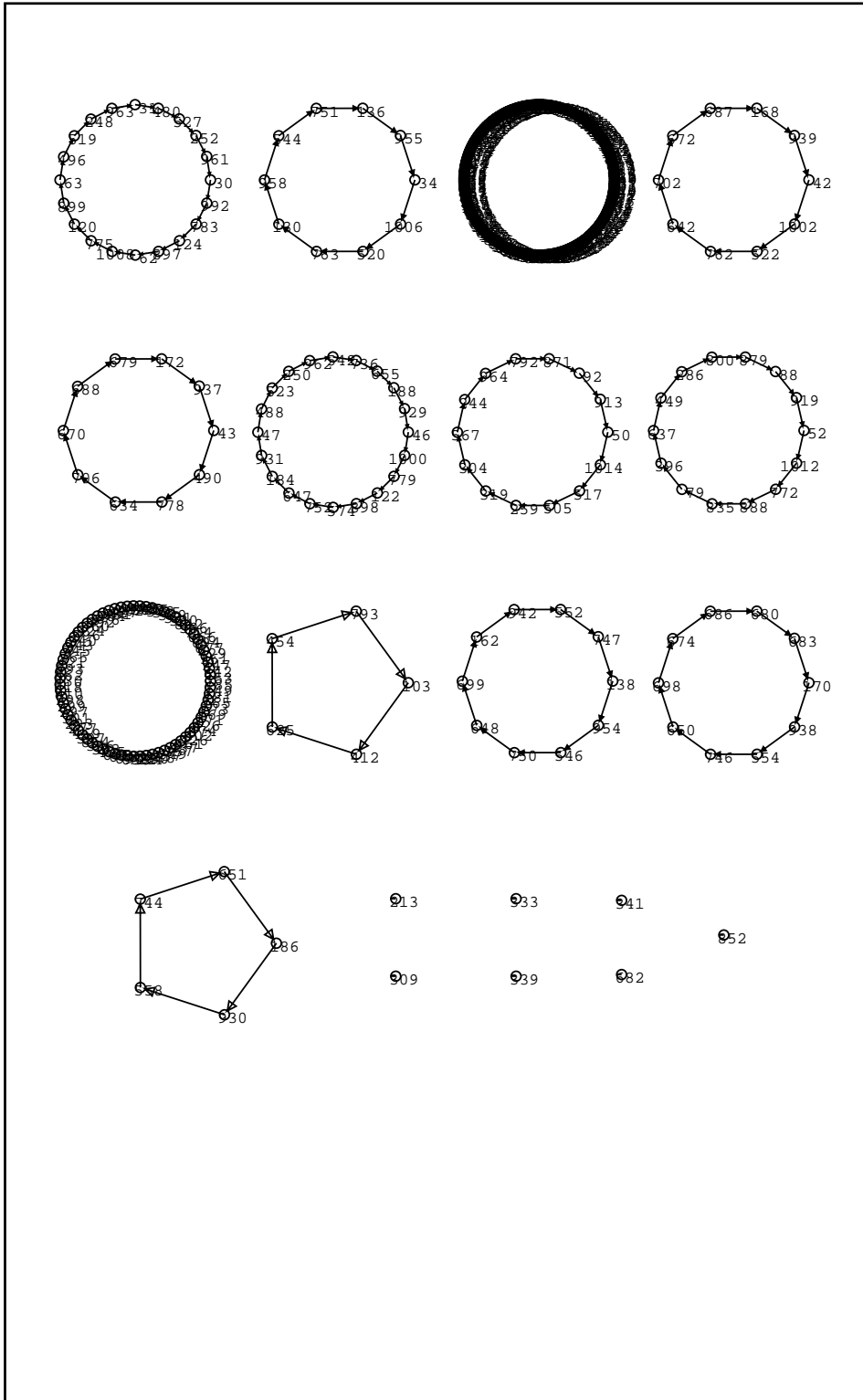
7



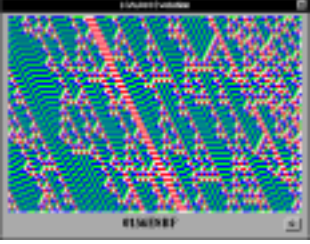


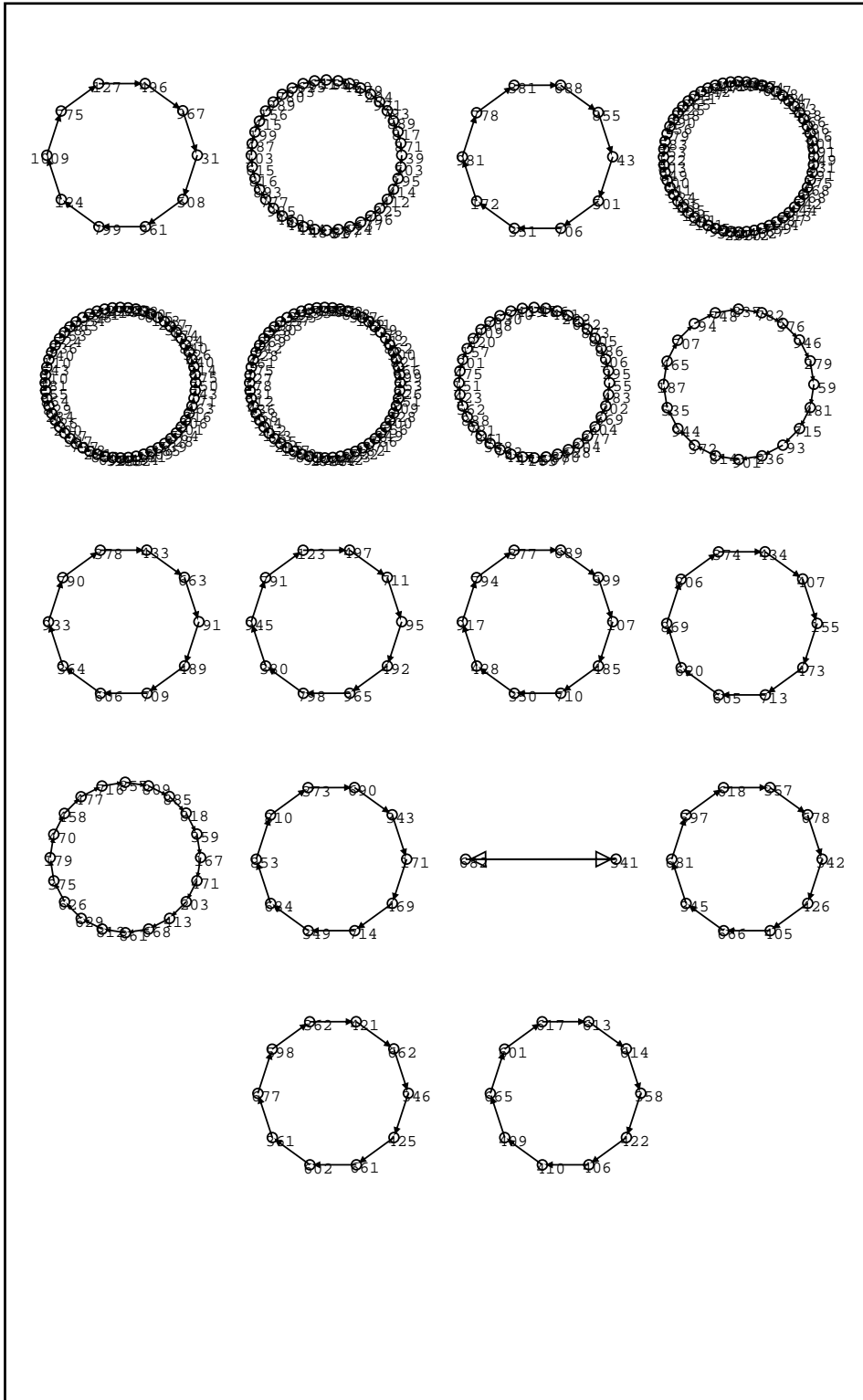
A.14 Regla 05AF14EB



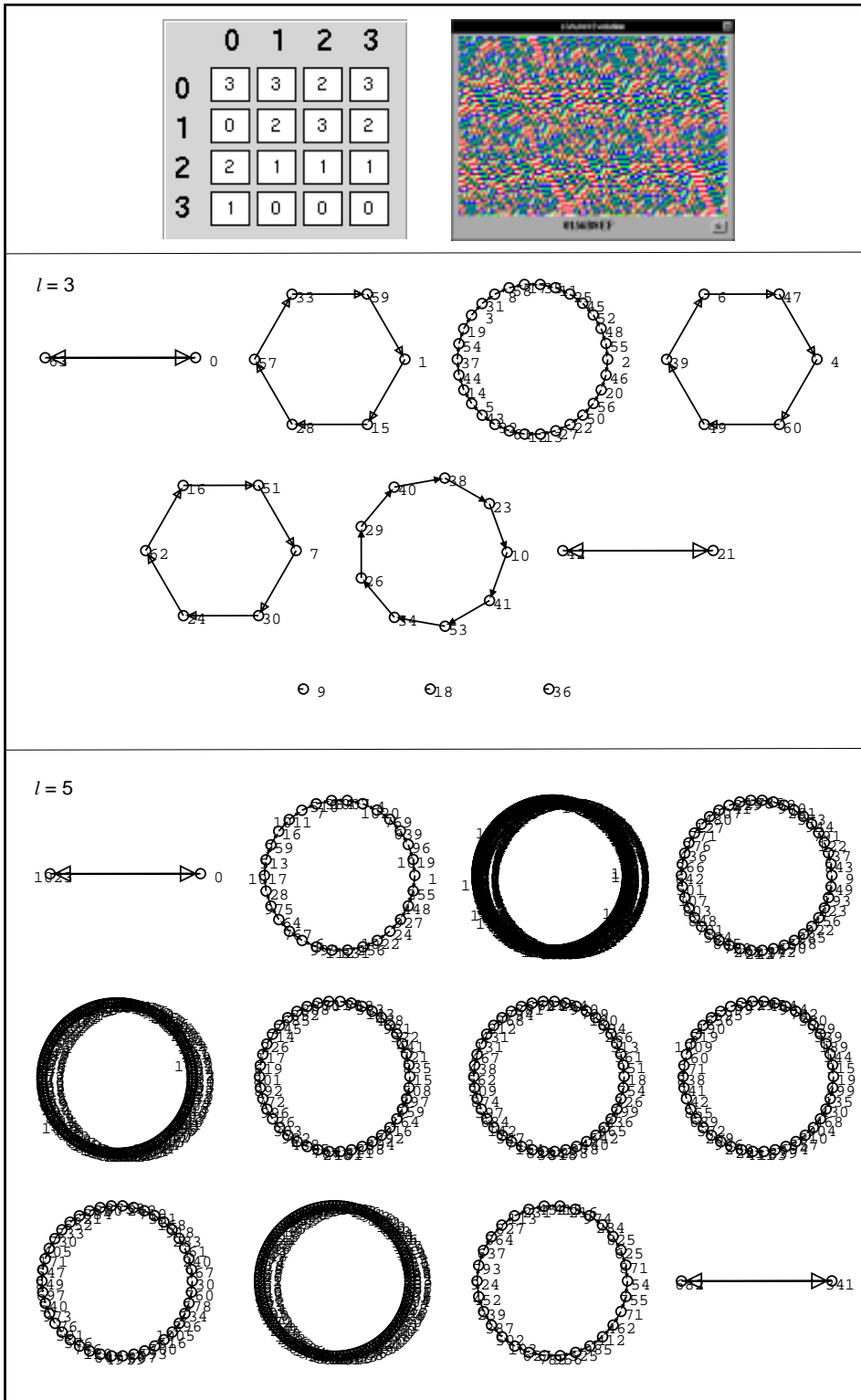


A.15 Regla 0156E8BF

<table border="1" style="margin: auto;"> <tr> <td></td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> </table>		0	1	2	3	0	3	3	3	2	1	0	2	2	3	2	2	1	1	1	3	1	0	0	0	
	0	1	2	3																						
0	3	3	3	2																						
1	0	2	2	3																						
2	2	1	1	1																						
3	1	0	0	0																						
<p>$l = 3$</p>																										
<p>$l = 5$</p>																										

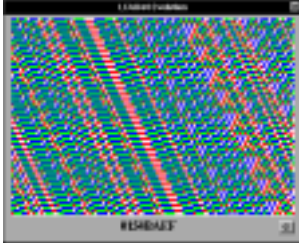


A.16 Regla 0156B8EF

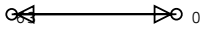


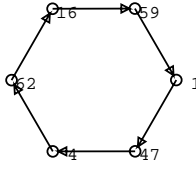
A.17 Regla 0154BAEF

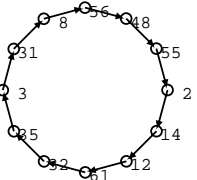
	0	1	2	3
0	3	3	2	3
1	2	2	3	2
2	0	1	1	1
3	1	0	0	0

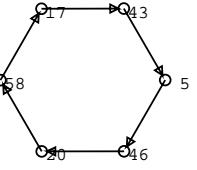


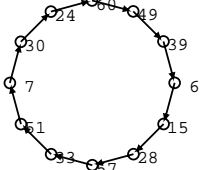
l = 3

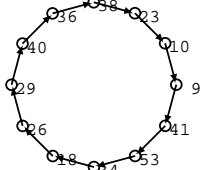


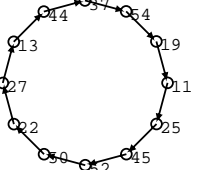


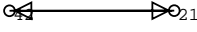




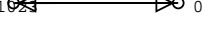


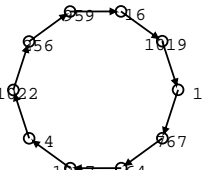


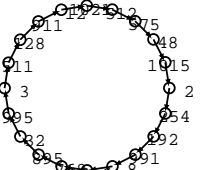


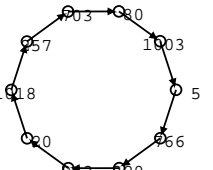


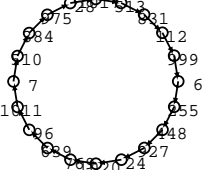
l = 5

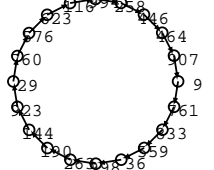


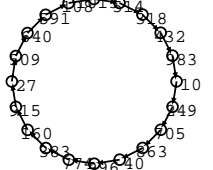


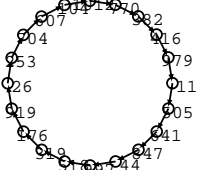


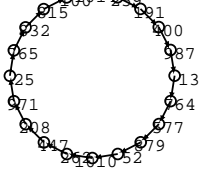


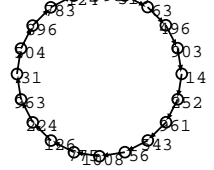


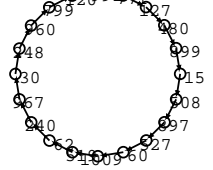


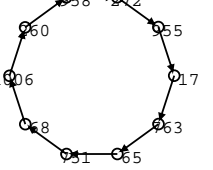


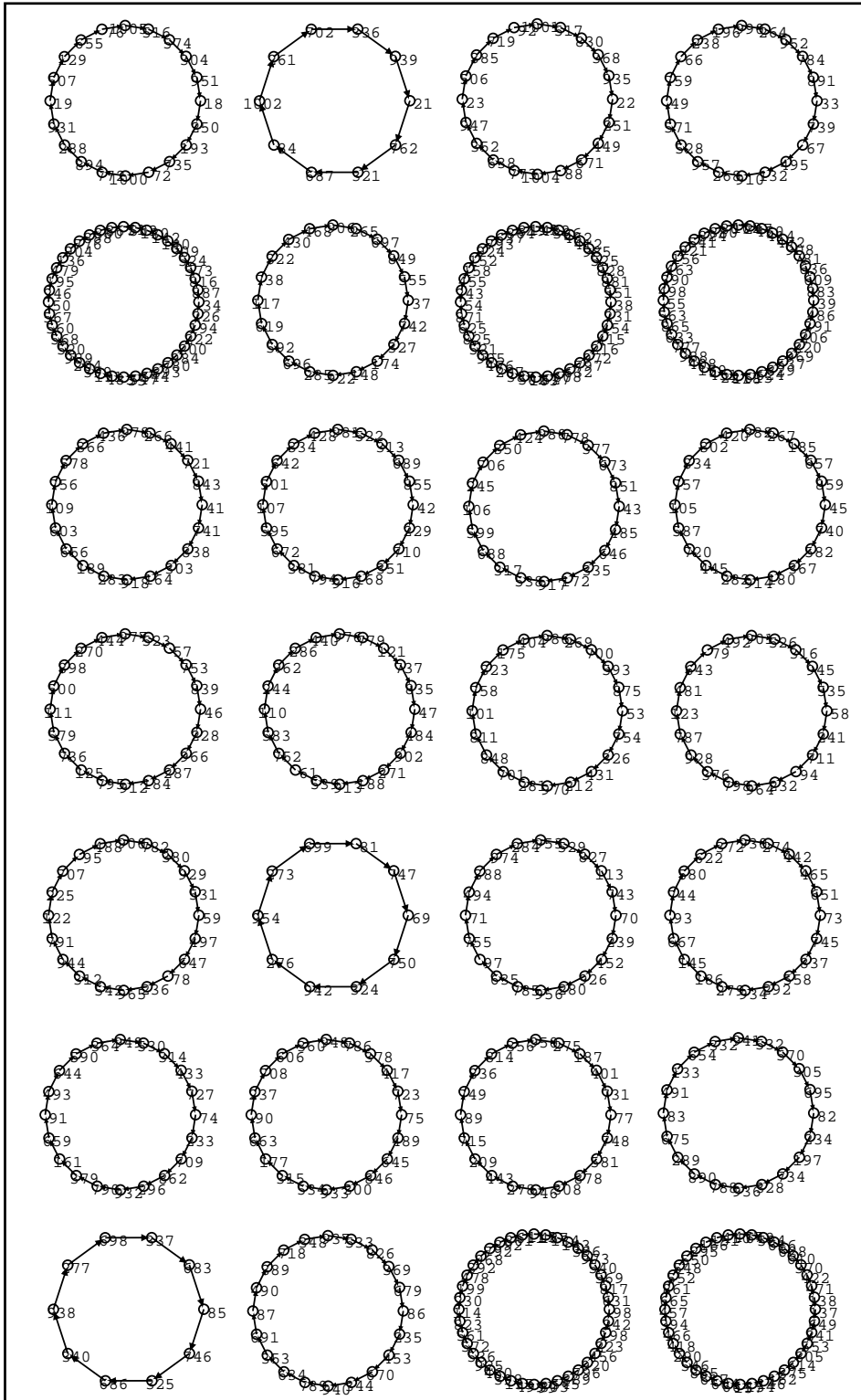


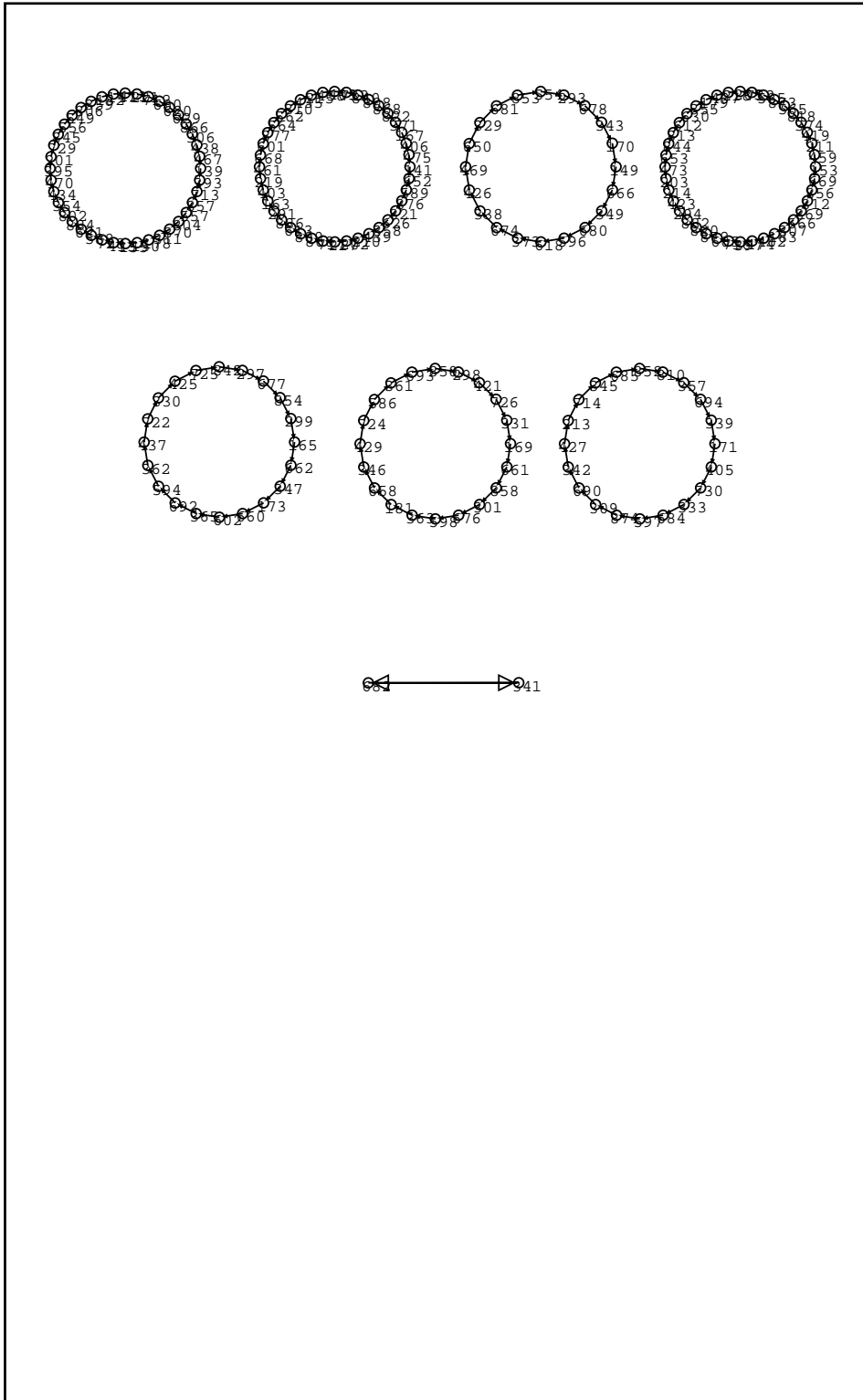




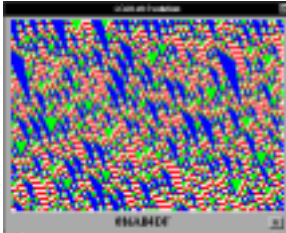
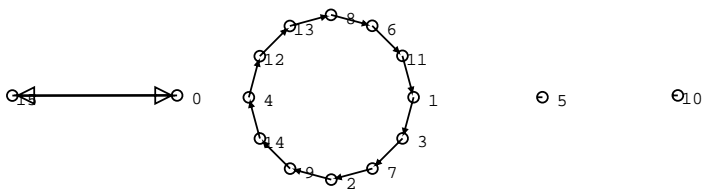
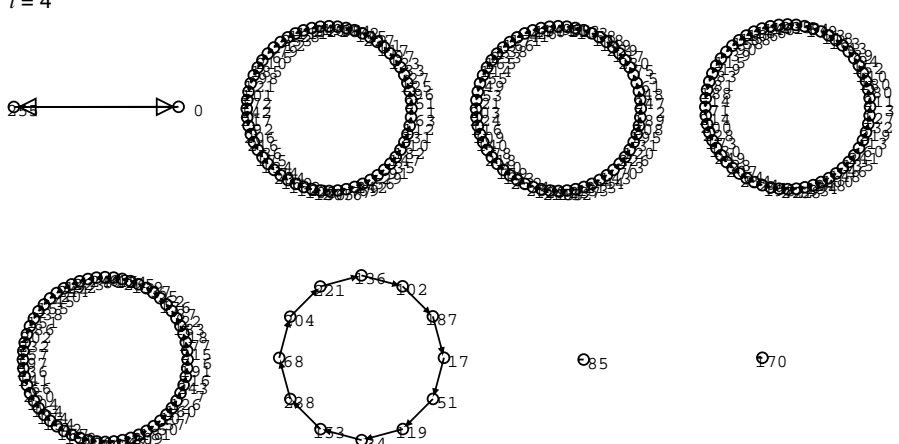
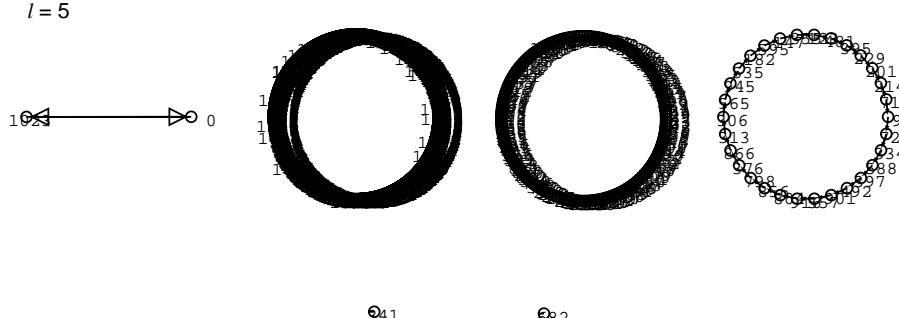






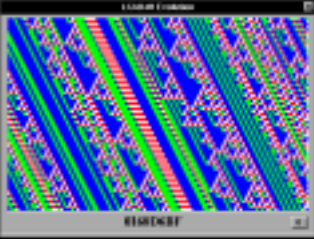


A.18 Regla 016AB4DF

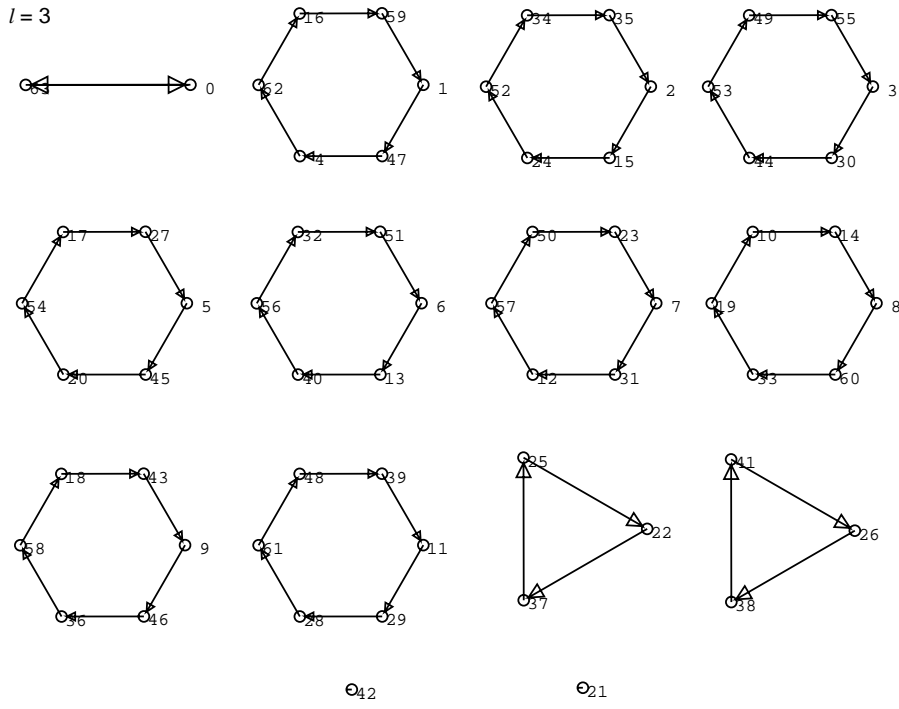
<table border="1" style="margin: auto;"> <tr> <td></td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">3</td> <td style="text-align: center;">3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">3</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">1</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> <td style="text-align: center;">0</td> </tr> </table>		0	1	2	3	0	3	3	1	3	1	0	1	3	2	2	2	2	2	1	3	1	0	0	0	
	0	1	2	3																						
0	3	3	1	3																						
1	0	1	3	2																						
2	2	2	2	1																						
3	1	0	0	0																						
<p>$l = 2$</p> 																										
<p>$l = 4$</p> 																										
<p>$l = 5$</p> 																										

A.19 Regla 0168D6BF

	0	1	2	3
0	3	3	3	2
1	2	1	1	3
2	0	2	2	1
3	1	0	0	0

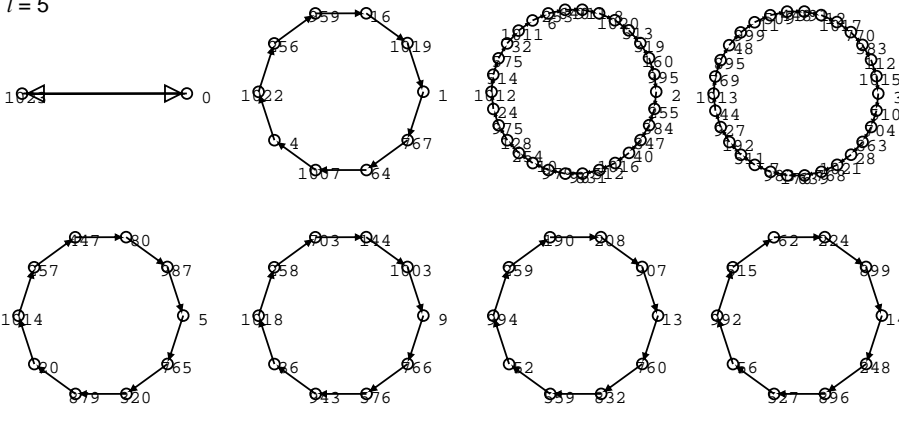


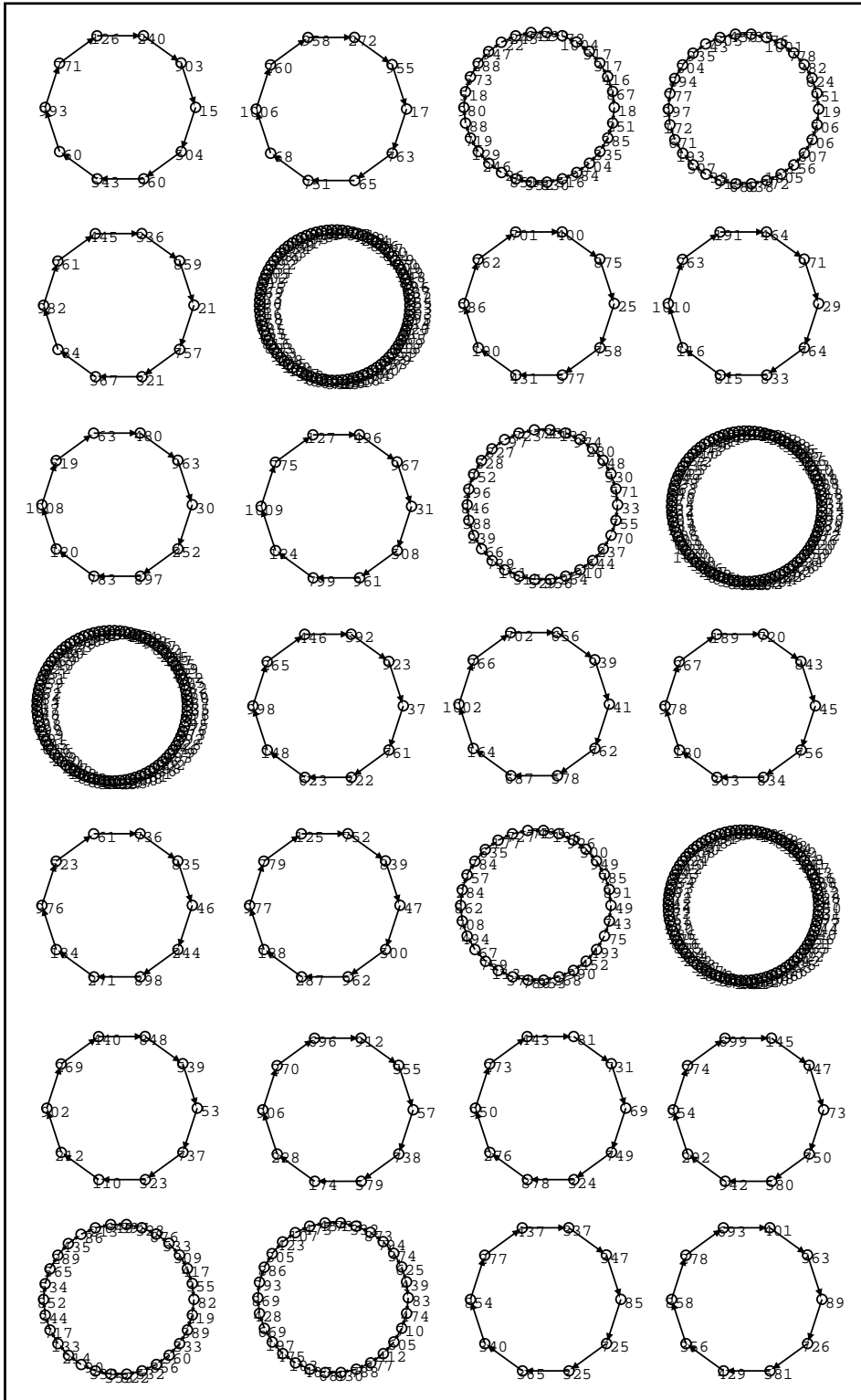
$l = 3$

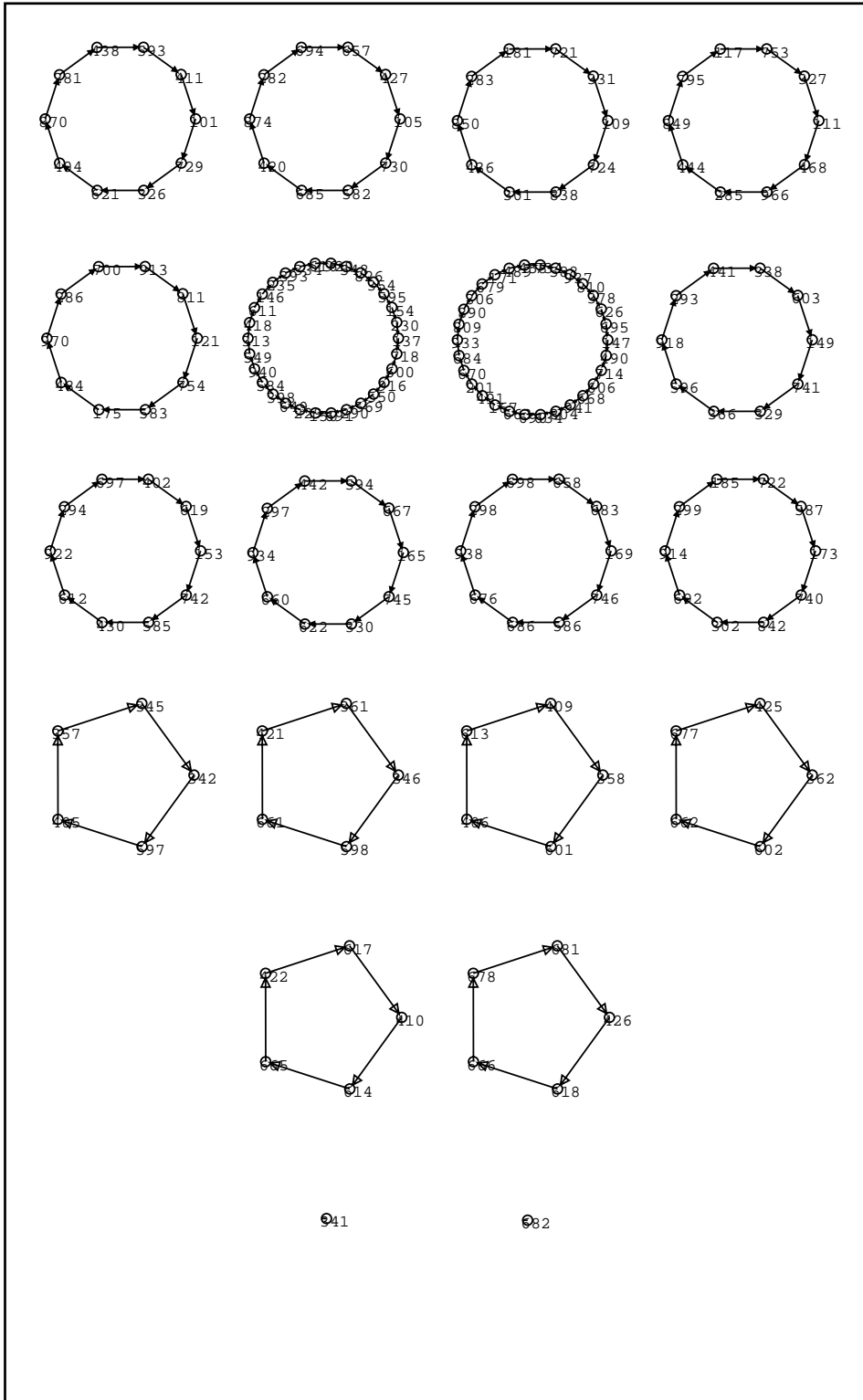


Q_{42} Q_{21}

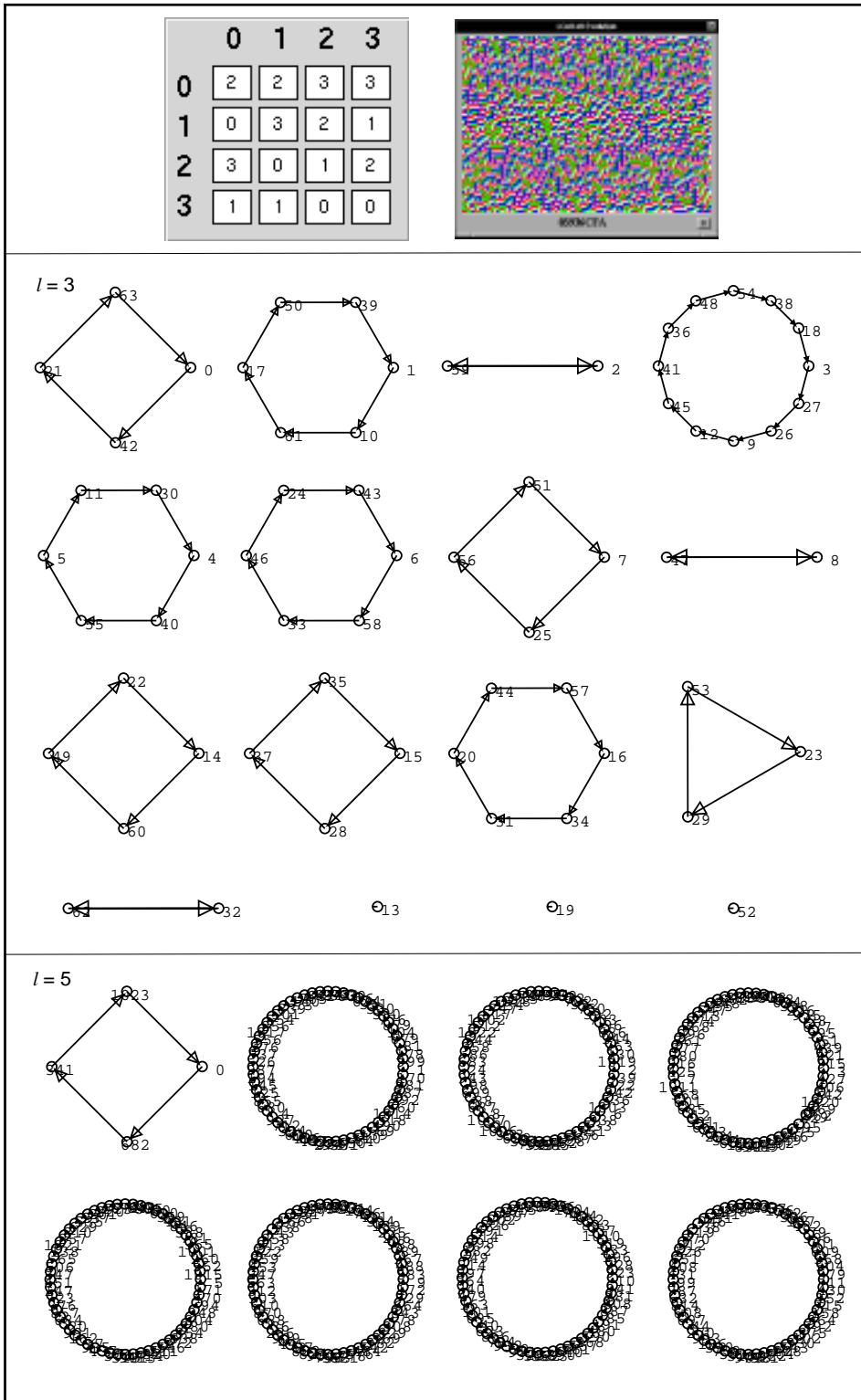
$l = 5$

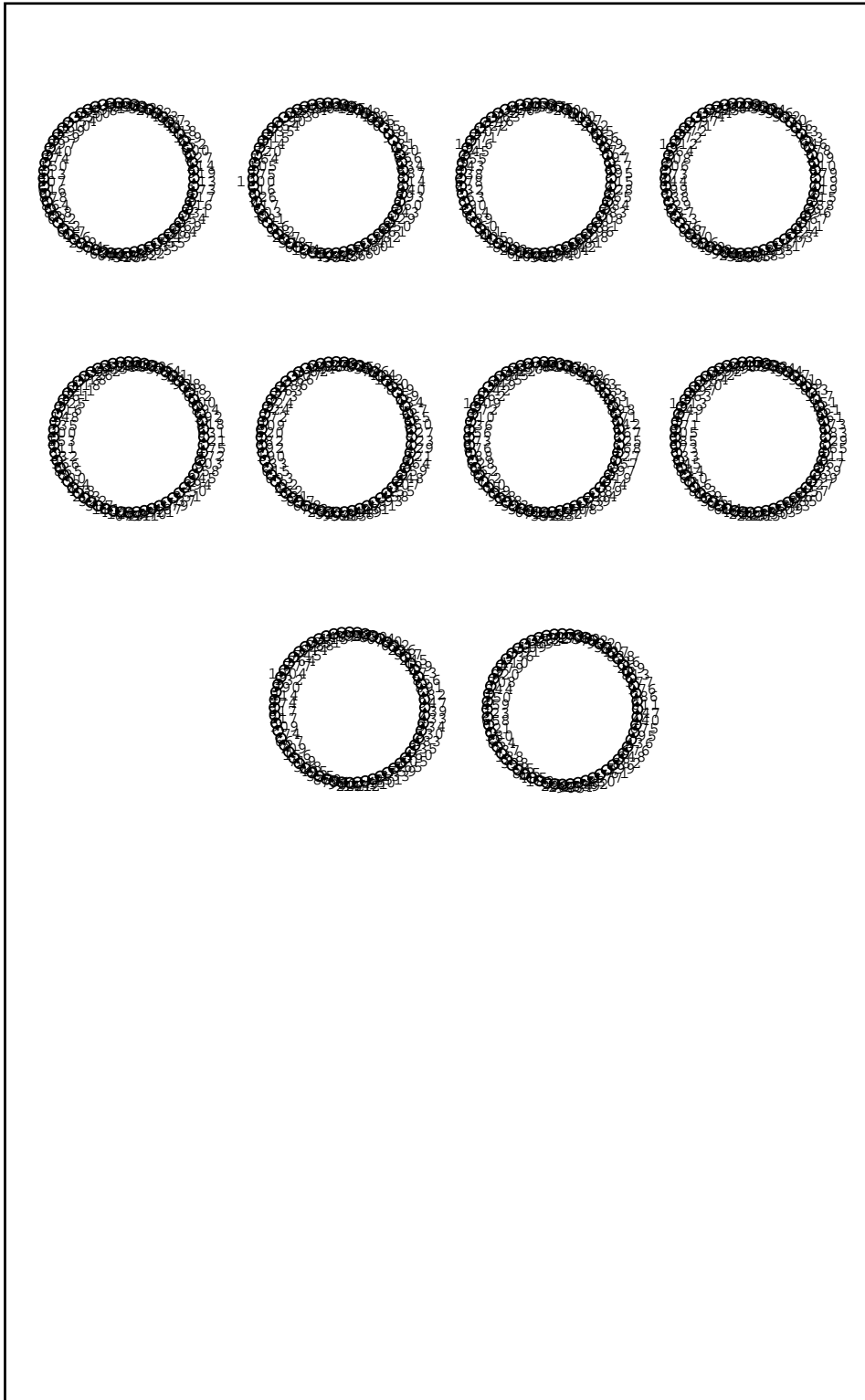






A.20 Regla 05936CFA





A.21 Regla 019BEC76

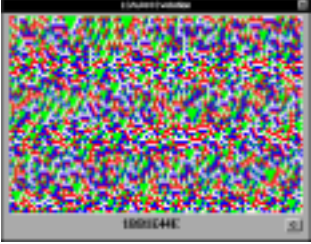
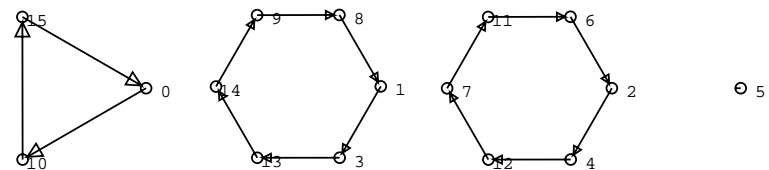
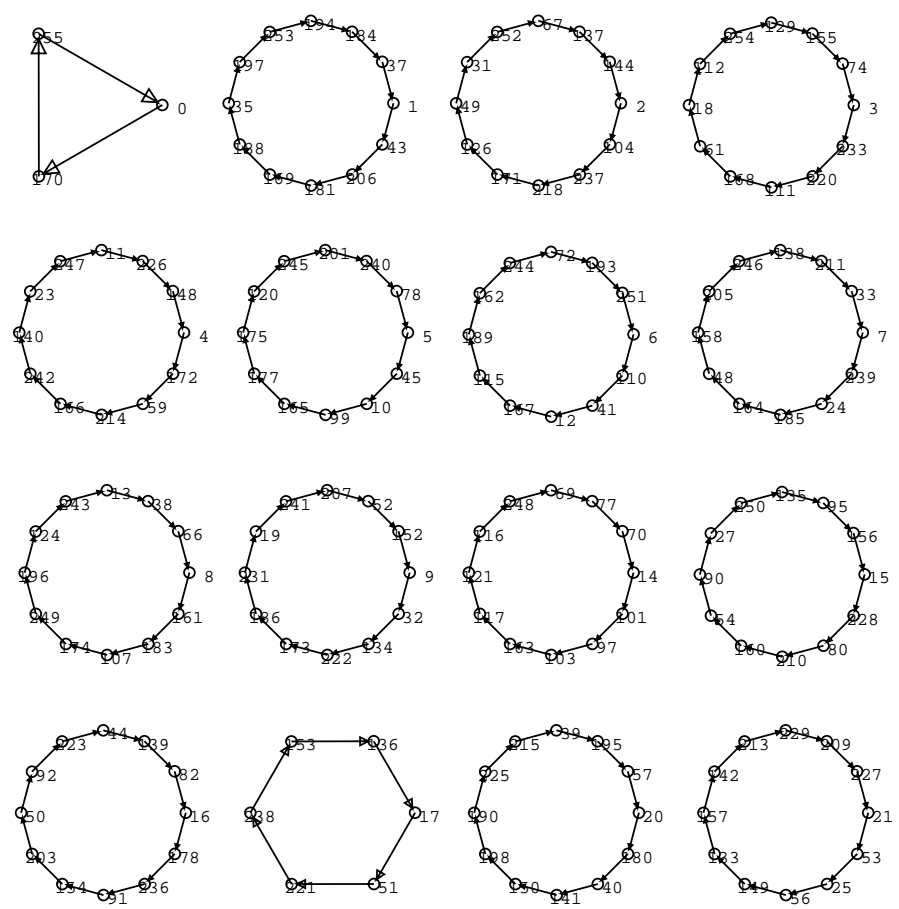
	0	1	2	3
0	2	1	3	1
1	0	3	2	3
2	3	2	1	2
3	1	0	0	0

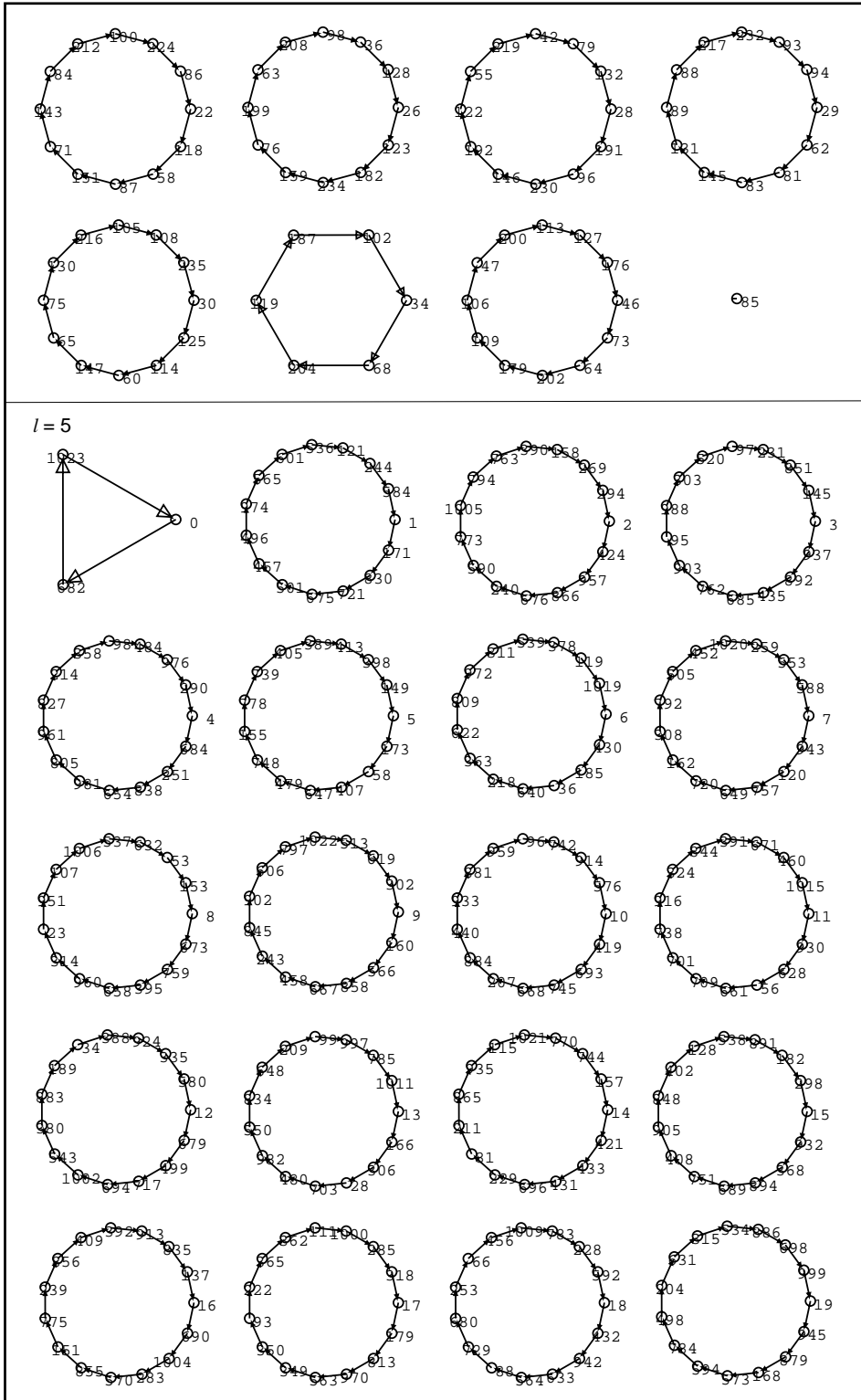
l = 3

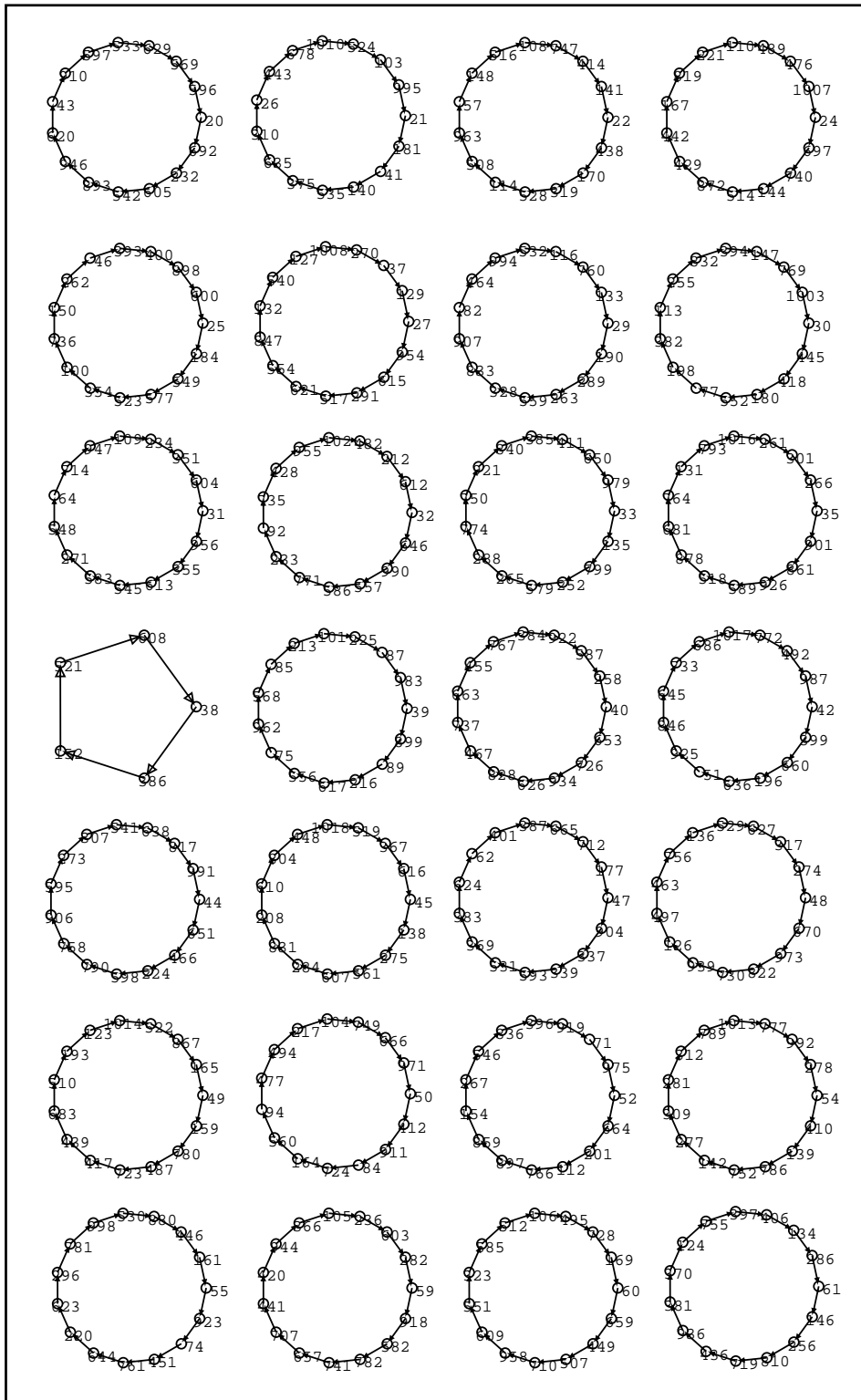
l = 5

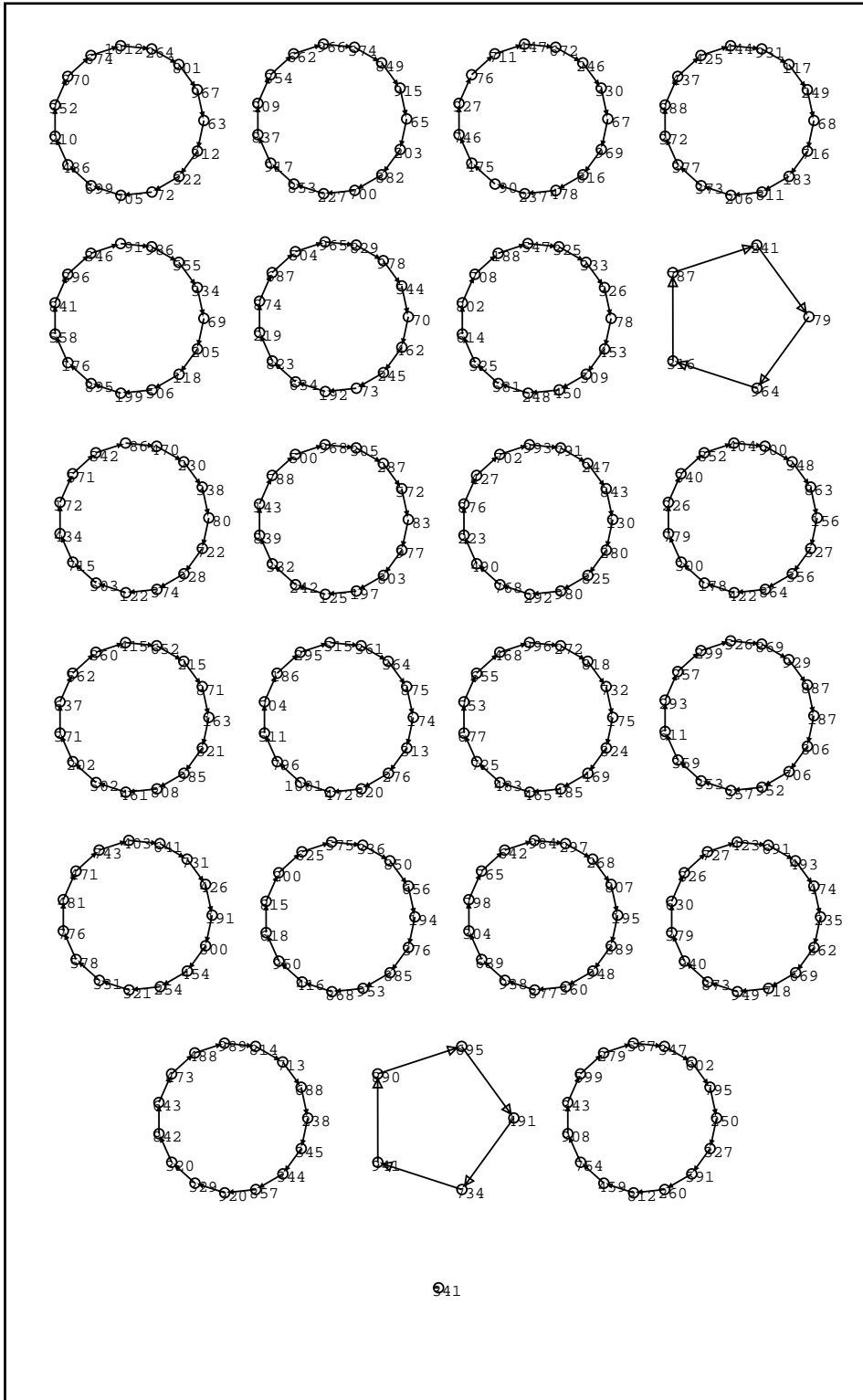


A.22 Regla 1BB1E44E

	<table border="1"><tr><th>0</th><th>1</th><th>2</th><th>3</th></tr><tr><th>0</th><td>2</td><td>3</td><td>0</td><td>1</td></tr><tr><th>1</th><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><th>2</th><td>1</td><td>0</td><td>3</td><td>2</td></tr><tr><th>3</th><td>3</td><td>2</td><td>1</td><td>0</td></tr></table>	0	1	2	3	0	2	3	0	1	1	0	1	2	3	2	1	0	3	2	3	3	2	1	0	
0	1	2	3																							
0	2	3	0	1																						
1	0	1	2	3																						
2	1	0	3	2																						
3	3	2	1	0																						
$l = 2$																										
$l = 4$																										







Lista de Tablas

2.1	Relación entre estados.	13
2.2	Conectando subconjuntos.	16
4.1	Tabla de evoluciones con $r = 1, 1.5, 2$ y 2.5	41
4.2	Configuraciones con $l = 1$	42
4.3	Configuraciones con $l = 2$	42
4.4	Configuraciones con $l = 3$	43
4.5	Regla de evolución.	45
4.6	Configuraciones con $l = 2$	46
4.7	Configuraciones con $l = 2$	48
5.1	Reglas con grados de reversibilidad.	58

Lista de Figuras

1.1	Vecindad de Moore.	3
1.2	Estructura lineal de los autómatas celulares.	4
1.3	Autómata $(3, h)$ regla 17793.	5
1.4	Autómata $(3, h)$ regla 14252.	6
1.5	Autómata $(3, h)$ regla 11509.	6
1.6	Autómata $(4, 1)$ regla C30E39E50E39E59739E5975EE5975E78.	7
2.1	Diagrama de de Bruijn genérico para el autómata $(2,1)$	11
2.2	Subdiagrama de de Bruijn para un autómata $(2,1)$	11
2.3	Regla de evolución del autómata $(2,1)$ regla 30.	15
2.4	Diagrama de de Bruijn para la regla 30.	15
2.5	Diagrama de subconjuntos para la regla 30.	16
2.6	Diagrama de parejas para la regla 30.	17
3.1	Mapeo de estados representado como conjuntos.	19
3.2	Mapeo para el caso $r = h$	20
3.3	Mapeos locales para la regla 30.	23
3.4	Conjuntos de estados, configuraciones y mapeos.	23
3.5	Mapeo global del conjunto $C_F(\Sigma)$	25
3.6	Mapeo global del conjunto $C(\Sigma)$	25
3.7	Árbol topológico.	26
3.8	Transiciones globales.	26
3.9	Ancestros.	27
3.10	Mapeo inyectivo.	27
3.11	Jardín del Edén.	28
3.12	Mapeo suryectivo.	29
3.13	Mapeo suryectivo regla 90.	29
3.14	Propiedades de un mapeo suryectivo.	30
3.15	Mapeo biyectivo.	31
3.16	Transiciones biyectivas.	32
4.1	Autómata reversible $(4, h)$ regla F5A0F5A0.	34
4.2	Formando anillos.	42
4.3	Árbol topológico con $l = 2$	43
4.4	Árboles topológicos con $l = 3$	44
4.5	Árboles topológicos con $l = 5, 7$ y 9	44

4.6	Formando anillos.	46
4.7	Árboles topológicos con $l = 2$	47
4.8	Diagrama de subconjuntos de la regla 0056B9EF.	48
4.9	Diagrama de parejas de la regla 0056B9EF.	49
4.10	Diagrama de parejas de la regla 016ED4BB.	53
4.11	Árboles topológicos con $l = 2$	53
4.12	Diagrama de transiciones con $l = 3$	54
A.1	NXLCAU4h con algunas opciones.	61

Bibliografía

- [1] John von Neumann, *Theory of Self-reproducing Automata* (edited and completed by A. W. Burks), University of Illinois Press, 1966.
- [2] Gustav A. Hedlund, *Endomorphisms and Automorphisms of the Shift Dynamical System*, Mathematical Systems Theory **3** 320-375, 1969.
- [3] Martin Gardner, *Mathematical Games - The fantastic combinations of John H. Conway's new solitaire game Life*, Scientific American, October 1969.
- [4] Stephen Wolfram, *Theory and Applications of Cellular Automata*, World Scientific Press, Singapore, 1986.
- [5] Edward F. Moore, *Gedanken Experiments on Secuencial Machines*, Princeton University Press, Princeton, New Jersey, 1956.
- [6] Edward F. Moore, *Mathematics in the Biological Sciences*, Scientific American, September 1964.
- [7] Harold V. McIntosh, *Linear Cellular Automata*, Universidad Autónoma de Puebla, Puebla, 1990.
- [8] Harold V. McIntosh, *Linear Cellular Automata via de Bruijn Diagrams*, Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, 1991.
- [9] Harold V. McIntosh, *Commentaries on: The Global Dynamics of Cellular Automata* (by A. Wuensche and M. Lesser), Instituto de Ciencias, Universidad Autónoma de Puebla, Puebla, 1993.
- [10] Harold V. McIntosh, *Reversible Cellular Automata*, Physica D **999** 100-130, 1990.
- [11] Harold V. McIntosh, *Welch Indices and Welch Diagrams*, Próximo a publicarse.
- [12] Christopher G. Langton, J. Doyne Farmer and Steen Rasmussen, *Artificial Life II*, Proceedings Volume X, Santa Fe Institute in the Sciences of Complexity, 1990.
- [13] Andrew Wuensche and Mike Lesser, *The Global Dynamics of Cellular Automata*, Santa Fe Institute in the Sciences of Complexity, 1992.

- [14] S. Amoroso and G. Cooper, *The Garden-of-Eden theorem for finite configurations*, Proceedings of the American Mathematical Society, 1970.
- [15] Sven Skyum, *Confusion in the Garden of Eden*, Proceedings of the American Mathematical Society, **50** July 1975.
- [16] Masakazu Nasu, *Local Maps Inducing Surjective Global Maps of One-Dimensional Tessellation Automata*, Mathematical Systems Theory **11** 327-351, 1970.
- [17] Harold S. Stone, *Discrete Mathematical Structures and their Applications*, Computer Science Series, 1973.
- [18] Howard Gutowitz y C. Domian, *The Topological Skeleton of Cellular Automaton Dynamics*, Physica D (submitted), 1995.
- [19] Howard Gutowitz, *Cellular Automata: Theory and Experiment*, MIT Press, Bradford Books, 1991.
- [20] David Hillman, *The Structure of Reversible One-dimensional Cellular Automata*, Physica D **52** 277-292, 1991.
- [21] J. Carlos Seck Tuoh Mora, *Autómatas Celulares Reversibles*, Tesis, Universidad Nacional Autónoma de México, 1998.
- [22] Taylor L. Booth, *Sequential Machines and Automata Theory*, John Wiley and Sons Inc., 1967.