Universidad de Buenos Aires

Facultades de Ciencias Económicas, Ciencias Exactas y Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática

Trabajo Final

Título: Autómatas Celulares con Reglas de Evolución Compuestas para Fines Criptográficos

Autor: Ing. Luis Antonio Catanzariti

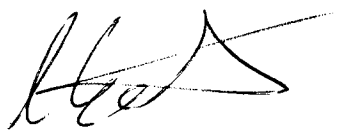
Tutor: Dr. Pedro Hecht

Cohorte: 2014

26 de abril de 2016

Declaración Jurada

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.



Luis A. Catanzariti
DNI 30.724.522

Resumen

Los números aleatorios y pseudo-aleatorios son de gran utilidad en diversos campos. Dos de los algoritmos más sencillos para generar secuencias pseudo aleatorias son los denominados registros de desplazamiento con retroalimentación y los autómatas celulares. Un Autómata Celular es un sistema constituido por un conjunto de celdas en una estructura de una o más dimensiones. Por ejemplo, en una dimensión las celdas se en disponen en forma lineal (ver figura 2.1). En dos dimensiones conforman una matriz en un plano cartesiano.

Cada una de estas celdas posee otras vecinas, conformando un vecindario. Las celdas pueden adoptar, en un momento determinado, un único valor entre un conjunto de valores discretos. Estas celdas actualizan su valor, en instantes de tiempo discretos, siguiendo una regla que especifica de que manera lo hará. Estas cuestiones y otras referentes a la descripción y caracterización de los autómatas celulares serán tratadas en el capítulo 2.

Las reglas que emplean los autómatas celulares para determinar su evolución en el tiempo son diversas. Inclusive, en algunos casos, se pueden emplear dos reglas de evolución y que "algunas" celdas evolucionen de acuerdo la primer regla y "otras" celdas evolucionen en función de la segunda regla.

Asimismo, se pueden establecer reglas compuestas. Una regla de evolución compuesta es aquella que surge de la aplicación sucesiva de 2 o más reglas de evolución simples en cada instante de tiempo, de la misma forma que en el ámbito del análisis funcional se procede a construir una función h a partir de las funciones f y g: $h(x) = g(f(x))$.

En este trabajo se ha elegido un conjunto de reglas compuestas, las cuales serán caracterizadas y estudiadas en el capítulo 3.

Índice general

Agradecimientos

Simplemente agradecer a mis padres por su apoyo incondicional durante toda mi carrera académica y profesional.

A mi hermana Mirtha, quien siempre me alentó en el camino de las ciencias y la investigación.

Al Dr. Pedro Hecht por su tiempo y dedicación en las correcciones y sugerencias que me ha realizado, las cuales me permitieron completar el presente trabajo.

Dado que existe una ley como la de la gravedad, el Universo pudo y se creó de la nada.

Stephen Hawking and Leonard Mlodinow - The Grand Design (2010)

1

Introducción

Los números aleatorios son de gran utilidad en diversos campos, ya sean teóricos o prácticos. Por ejemplo: simulaciones, estadística y criptografía entre otros. La aleatoriedad proviene de sistemas (generalmente físicos) donde los diferentes estados que estos adoptan se suceden en forma impredecible (e irreproducible), es decir, no se conoce con certeza el siguiente estado que tomará el sistema, aún conociéndose todos los estados previos que hubo adoptado.

Desde el punto de vista criptográfico, contar con esta clase de sistemas resulta de suma utilidad. Sin embargo, existen situaciones en las cuales es conveniente contar con secuencias cuyas propiedades estadísticas sean similares a aquellas que presentan los sistemas aleatorios y además que las secuencias de números generadas puedan reproducirse (de forma determinista). Estas secuencias pueden generarse mediante la utilización de procedimientos algorítmicos y se las denomina secuencias pseudo-aleatorias.

Los algoritmos capaces de producir esta última clase de secuencias, son de complejidad diversa. Por razones de computabilidad y rendimiento, se prefieren los algoritmos más simples. Algunos de los más sencillos son los

[1] Texto original: *Because there is a law such as gravity, the universe can and will create itself from nothing.*

denominados registros de desplazamiento con retroalimentación, más conocidos como LFSR (del inglés *Linear Feedback Shift Register*), y los autómatas celulares (**CA**, del inglés *Cellular Automata* o *Cellular Automaton*).

Como se ha mencionado en el inicio de este capítulo, los números aleatorios son útiles en diferentes campos. En particular dentro del campo de la criptografía la utilidad de los métodos, no deterministas o deterministas, para generar números aleatorios o pseudo-aleatorios respectivamente, se puede encontrar en las siguientes áreas[1, p. 169] (entre otras):

- Generación de números de identificación personal (**PIN** del inglés *Personal Identification Number*) y contraseñas (en inglés *passwords*)
- Generación de números primos, que son empleados en algoritmos y protocolos basados en criptografía asimétrica
- Generación de claves empleadas en algoritmos de cifrado de flujo[2] (estas claves se suelen denominar *keystreams*). Ejemplos de este tipo de algoritmos de cifrado son: el cifrador de Vernam o la "libreta de uso único" (*one-time-pad*).
- Generación de claves de sesión[3]
- En protocolos de autenticación mediante el empleo de un desafío-respuesta (*challenge-response protocols*)[4].

El trabajo que se presentará a continuación tiene por finalidad proponer un nuevo enfoque en la utilización de CAs para la generación de secuencias de números pseudo-aleatorios. Es decir, utilizar autómatas celulares como PRNG (del inglés *Pseudo Random Number Generator*). La finalidad del trabajo es proponer nuevas reglas de evolución y mostrar las características intrínsecas de los autómatas celulares que las utilizan; antes que evaluar las características y calidad de las secuencias que estos generan, dejándose esta evaluación para un estudio posterior a realizarse sobre una implementación en particular.

---

[2] *Un **cifrador de flujo** es un algoritmo de cifrado simétrico que opera aplicando distintas transformaciones, en función del tiempo, sobre cada uno de los dígitos individuales del texto plano.* Tomado de [2, pp. 1264-1265].

[3] *Una **clave de sesión** o clave de cifrado de mensaje, es una clave que se utilizará durante una única sesión de comunicaciones.* Tomado de [2, p. 678].

[4] *La **autenticación mediante desafío-respuesta** es un protocolo en el cual una entidad se autentica enviando un valor que depende de (1) un valor secreto y (2) el valor de un desafío variable.* Tomado de [2, p. 199].

Si bien se realizará una introducción al tema de CAs, dirigida al lector novel, el desarrollo exhaustivo se encuentra fuera del alcance de este trabajo. Se sugiere enfáticamente consultar [3], [4] y visitar [5].

El resto de este documento se organizará de la siguiente forma: en el capítulo 2 se presentan los autómatas celulares y algunas de sus características, muchas de las cuales han sido consideradas en el estudio y evaluación del enfoque de CA propuesto en este trabajo. El capitulo 3 se centra en el objeto del trabajo: las reglas de evolución compuestas; se presentarán aquellas que se han estudiado y se dará una descripción de las mismas, empleando términos, técnicas y herramientas propuestas por otros autores en el estudio de autómatas celulares. Muchos de los resultados obtenidos, se presentan en forma gráfica y consecutiva (para una mejor comparación entre ellas) en los anexos A a G. Para cerrar se exponen las conclusiones del trabajo.

Dale un pez a un hombre y comerá un día; enséñale a pescar y comerá siempre.

Proverbio chino

2

Autómatas Celulares: propiedades y características

Un Autómata Celular (**CA**, por sus siglas en inglés: *Cellular Automata*) es un sistema constituido por un conjunto de celdas (o sitios, *sites*), cada una de las cuales puede adoptar en un momento determinado un valor único entre un conjunto de valores posibles. Las celdas que componen el CA actualizan su valor en forma sincrónica (en general) y en intervalos discretos de tiempo, obedeciendo a una regla de evolución determinista o probabilística (es más habitual el primer tipo). Por ello en un instante de tiempo t, el valor de una celda estará definido por:

- la regla de evolución del CA
- el valor de dicha celda en el instante $t-1$
- los valores de las celdas "cercanas" en el instante $t-1$

El conjunto de celdas cercanas a una dada se denomina **vecindad** (*neighborhood*) de la celda. Cuando la vecindad está conformada por las celdas que son directamente adyacentes a una dada, el CA posee **radio** (*range*) igual a uno: $r = 1$. Si la vecindad incluyera, además de las celdas directamente adyacentes, a todas aquellas que se encuentran a 1 celda de distancia entonces se tendrá $r = 2$ y así sucesivamente.

Las disposiciones de celdas más habituales que podemos encontrar en un CA son:

- **Unidimensional (1D):** donde las celdas se disponen una a continuación de otra en una única dimensión.
- **Bidimensional (2D):** el cual las celdas se disponen en forma matricial de dos dimensiones.

  Y en general se pueden utilizar estructuras de n dimensiones [6].

En el caso de un CA unidimensional cuyo radio sea uno ($r = 1$), cada celda que lo conforma tendrá exactamente dos vecinas: una por derecha y la otra por izquierda. Por lo tanto, en este caso cada celda conformará una vecindad de 3 celdas: ella misma más sus dos vecinas. Si $r = 2$ entonces cada celda contará con 2 vecinas por derecha y otras dos por izquierda, configurando una vecindad de cinco celdas. En general, los autómatas celulares 1D con un radio r poseen vecindades de tamaño $2 \times r + 1$.

Estas características determinan cuantas reglas de evolución pueden formarse: para CAs cuyas celdas tomen uno de entre k posibles valores y con un radio r: existen $k^{k^{2\times r+1}}$ reglas distintas. Lo cual refleja que pequeños incrementos en los valores que puede adoptar una regla o la cantidad de celdas en el vecindario, redundará en un incremento exponencial en la cantidad de reglas dificultando la realización de un estudio exhaustivo.

En el caso particular de CAs 1D con $k = 2$ y $r = 1$ (los más simples), estos se denominan **Autómatas celulares elementales** (ECA del inglés *Elementary Cellular Automata*) y existen 256 reglas de evolución distintas, como se sugiere en [4, pp. 7-8]. Este tipo de autómatas celulares evolucionan de acuerdo a reglas que contemplan el estado de la vecindad en el instante inmediato anterior. Si Φ es la regla y $N(c_i)^t$ es el vecindario de la celda c_i en el instante t, entonces: $c_i^{t+1} = \Phi(N(c_i)^t)$ es el estado que la celda tomará en el instante $t + 1$.

Wolfram definió una técnica para etiquetar o nombrar a las diferentes reglas de evolución basándose en el valor al cual evoluciona la celda central de las 8 vecindades ($2^{2\times1+1}$) existentes en este tipo de CA. Estas 8 celdas centrales proporcionan, cuando las mismas toman uno de dos valores posibles, un número de 8 bits (cuyo rango no signado comprende desde 0 hasta 255 en base decimal). La forma de generar el número de regla consiste en disponer

las 8 vecindades en orden lexicográfico descendente y así obtener los valores de la evolución de cada una de ellas como 8 bits ponderados en función de la posición que ocupan (de acuerdo al sistema de numeración binario de notación posicional). Por ejemplo, las siguientes reglas de evolución se clasifican como regla 30 (superior) y regla 90 (inferior):

111	110	101	100	011	010	001	000	
0	0	0	1	1	1	1	0	$= 30_{(10)}$

Listado 2.1: Regla 30

111	110	101	100	011	010	001	000	
0	1	0	1	1	0	1	0	$= 90_{(10)}$

Listado 2.2: Regla 90

Cuando el valor de las celdas en el instante t se encuentra definido en función del estado de la vecindad en el instante inmediato anterior $(t-1)$, el CA se denomina de **primer orden** (*First Order*). En cambio si el valor de las celdas en el instante t depende del estado de la vecindad en los instantes $t-1$ y $t-2$, el autómata celular se denomina de segundo orden [7, p. 45]. Sin embargo, todo CA de orden superior n (orden mayor a 1) y radio r tiene un CA de primer orden equivalente cuya vecindad posee un radio $n \times r$, ver [8, p. 43].

La definición formal de un CA es la siguiente [9]:

$$CA = \{\tau, S, s, s_0, N, \Phi\}$$

Donde:

1. τ es la estructura del conjunto de celdas c_i, $i \in \mathbb{N}$ perteneciente a un espacio euclídeo n-dimensional, $\mathbb{R}^n$.
2. S es un conjunto finito de k estados que pueden tomar las celdas
3. s indica el valor de la celda c_i en el instante t: $s(c_i, t)$
4. s_0 es el estado inicial de cada celda en el instante $t = 0$: $s(c_i, 0) = s_0(c_i)$
5. N es la función de la vecindad que mapea cada celda c_i con un conjunto de celdas vecinas. De hecho, para un CA 1D: $|N(c_i)| = 2r + 1$
6. Φ es el conjunto de funciones que gobiernan el comportamiento de una celda. Es la función de transición que define el estado local del autómata.

Desde el punto de vista teórico la cantidad de celdas que integran un CA puede ser finita o infinita. En la práctica, los CAs sólo poseen una cantidad

finita de celdas, debido a la limitación de recursos computacionales disponibles para la implementación.

Cuando los CAs poseen una cantidad n finita de celdas, estos poseen una cantidad también finita de estados globales distintos. Si cada celda adopta uno de entre k valores posibles, entonces el CA contará con k^n estados diferentes. Un estado global que el CA tomó en el instante de tiempo t_i, se volverá a presentar en un instante de tiempo menor o igual que t_{i+k^n} (excepto que dicho estado integre el denominado **Jardín del Eden**, como se explicará más adelante).

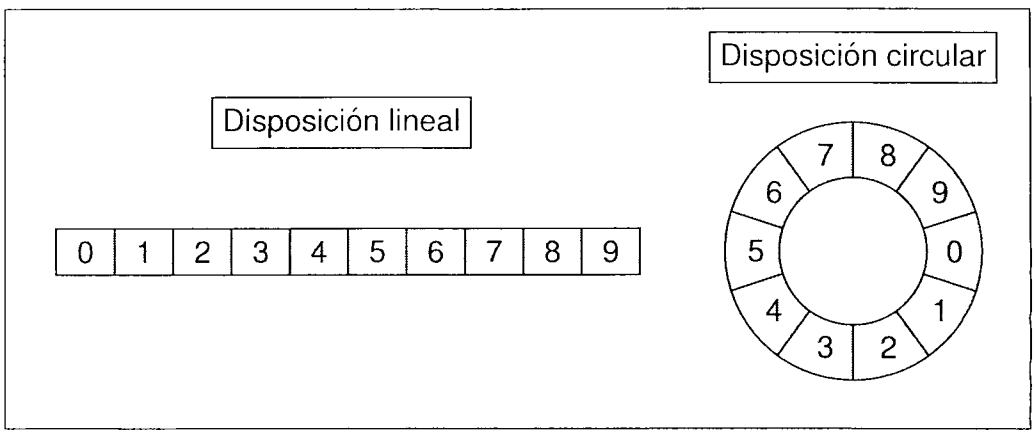


Figura 2.1: Condición de frontera periódica

Debido a que una estructura finita de celdas posee un límite o frontera, existirán celdas cuyo vecindario se encuentre incompleto (en CA 1D con $r = 1$ la celda del extremo derecho y la celda del extremo izquierdo). Para afrontar esta situación de las celdas fronterizas, existen diferentes aproximaciones, las cuales se denominan condiciones de frontera (en inglés *Boundary Conditions*) [7, pp. 47-49], a saber:

- Condición de frontera periódica: si el CA 1D posee n celdas y $r = 1$, numeradas de 1 a n, entonces la celda n tiene como vecina derecha a la celda 1. Y a su vez, la celda 1 tiene como vecina por izquierda a la celda n. En este caso al autómata celular se lo trata como un buffer circular (ver figura 2.1). Bajo este tratamiento de las condiciones de borde, se puede pensar geométricamente la evolución del CA como un cilindro. Esta condición de frontera es la que mejor permite simular un CA infinito, implementado con uno finito.

- Condición de frontera reflectiva: en este caso las celdas que se encuentran en los extremos se reflejarán adecuadamente para completar la vecindad (ver figura 2.2).

- Condición de frontera fija: Se completa el vecindario de las celdas que se encuentran en los extremos con valores fijados arbitrariamente de entre los k posibles.

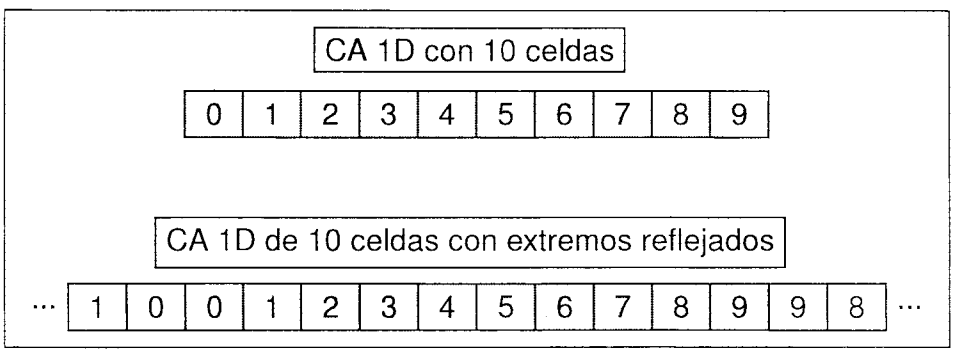


Figura 2.2: Condición de frontera reflectiva

Para poder estudiar y caracterizar a los autómatas celulares, Wolfram propuso una forma de clasificarlos [4, pp.115-157]:

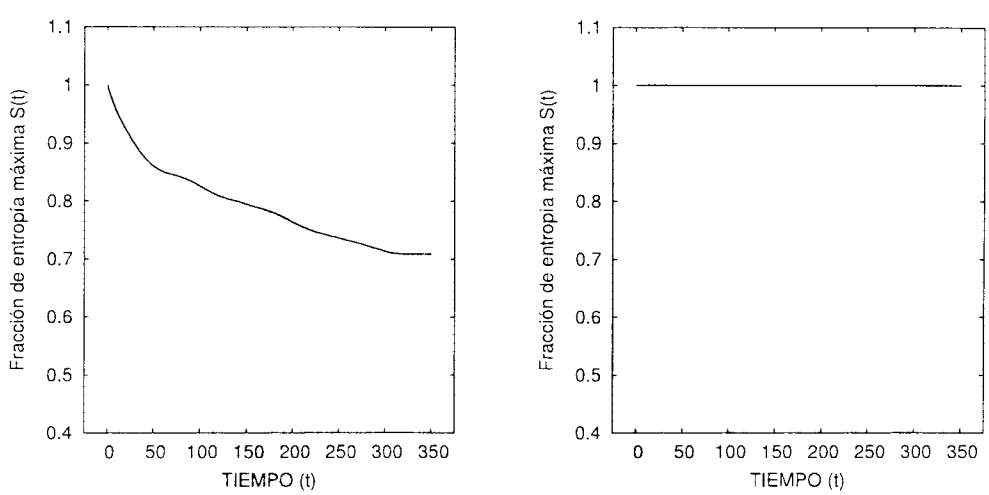
- **Clase I**: Sin importar las condiciones iniciales, la evolución del autómata es constante. Denominado punto de atracción o punto fijo (*fixed point, a sink*) en dinámica no lineal (caos) [10, p. 60].
- **Clase II**: Ciertas celdas o estructuras de las condiciones iniciales se propagan o repiten periódicamente. En dinámica no lineal se lo denomina órbita periódica o ciclo límite (*limit cycle*)[10, p. 60].
- **Clase III**: El comportamiento del autómata celular es caótico (caos determinista). Estos son los CAs de interés para generar secuencias pseudoaleatorias (los ya mencionados PRNG). Este comportamiento corresponde a los llamados **atractores extraños** (*strange attractors*) en dinámica no lineal [10, p. 60].
- **Clase IV**: El autómata rápidamente exhibe estructuras, las cuales interactúan durante la evolución. Los autómatas celulares que pertenecen a esta clase, presentan capacidades de computación universal [4, pp. 115-157]. En 2004, se ha probado que la regla 110, perteneciente a esta clase, posee capacidad de computación universal [11].

Debemos agregar que en la práctica, cuando se emplean CAs finitos, en todos los casos se alcanzará (en algún momento de la evolución) un punto fijo o una órbita periódica.

Otro aspecto a considerar en la evolución de los CA es el comportamiento de los distintos estados globales. Ya se ha expuesto que un autómata

finito de N celdas con $k = 2$, posee 2^N estados conformando el conjunto E de estados posibles del autómata. El estado adoptado en el instante $t = 0$ conformará las **condiciones iniciales** del autómata. Si los 2^N estados son equiprobables ($\frac{1}{2^N}$), entonces el CA posee máxima entropía (fracción de máxima entropía: $S = 1$). Si se consideran los 2^N distintos posibles estados y se les aplica la regla de evolución a cada uno, se obtendrá la evolución en $t = 1$ de cada estado inicial, las cuales también pertenecen necesariamente a E (la evolución es una operación que posee la propiedad de clausura).

Puede suceder que en $t = 1$ exista una única instancia de cada estado global (al igual que en $t = 0$) y por lo tanto, esto significa que todos los estados de E ocurren luego de la evolución. En este caso, luego de 1 paso en la evolución, los estados siguen siendo equiprobables ($\frac{1}{2^N}$) y la regla de evolución mantiene al CA en un estado de máxima entropía. Claramente, para $t \in 1, 2, 3, ...$ se seguirá manteniendo esta situación. En la figura 2.3b se muestra el comportamiento descripto para un ECA de 15 celdas con regla de evolución 45 (otros gráficos de entropía para las reglas de evolución, que serán tratadas en próximo capítulo, se encuentran en el anexo D).



(a) Regla 30 **(b)** Regla 45

Figura 2.3: Entropía informacional por sitio

Pero también, puede ocurrir que dos estados globales iniciales al avanzar a $t = 1$ evolucionen hacia el mismo estado. En este caso, necesariamente uno de los estados de E no ha sido alcanzado (debido a que la sumatoria de ocurrencias de estados se mantiene constante), por lo tanto los estados en $t = 1$ no serán equiprobables ya que el estado que se repite tendrá probabilidad de ocurrencia $\frac{2}{2^N}$ y el estado que no fue alcanzado tiene probabilidad

0. A medida que el CA evolucione, $t \in 1, 2, 3, ...$, puede reducirse aún más la cantidad de estados globales a los cuales pueda evolucionar el autómata. Esto (compresión de estados alcanzados) conduce a una reducción de la entropía del CA, debido a que el autómata evoluciona desde el desorden completo hacia un estado más ordenado [7, p. 62]. En la figura 2.3a se muestra el comportamiento descripto para un ECA de 15 celdas con regla de evolución 30.

Esta compresión de estados, determinada por el decrecimiento de estados alcanzables en la evolución del autómata celular, junto a la disminución de entropía; son propiedades habitualmente exhibidas por CAs irreversibles [7, p. 62].

Cuando el autómata celular se comporta de esta forma, los estados globales que se pueden alcanzar se reducen en función del avance del tiempo. Si bien en los sistemas dinámicos la irreversibilidad implica un aumento *de la entropía* (con el avance del tiempo), ¡en los autómatas celulares ocurre lo opuesto! La irreversibilidad en un CA implica que algunos estados no son alcanzables y que en algún punto dos (o más) estados globales evolucionan y convergen hacia un mismo estado haciendo imposible revertir dicha evolución, debido a que los predecesores se vuelven indistinguibles. Pero, como ya se ha dicho, ello implica que algunos estados globales no sean alcanzables, probabilidad 0, lo que implica una reducción de la entropía o en otras palabras: el CA exhibe mayor orden interno.

Todos los estados que solo existen en $t = 0$, es decir, aquellos estados que solo existieron como condiciones iniciales y que no aparecerán cuando $t \geq 1$ conforman un conjunto denominado **Jardín del Edén** (*Garden of Eden*). El conjunto se ha denominado en esta forma debido a que todos los elementos que componen el conjunto carecen de estados predecesores, no tienen ancestros. Este comportamiento se puede observar fácilmente en los **diagramas de transición de estados.** Estos diagramas muestran los estados como nodos y las transiciones como aristas orientadas, en los cuales se aprecia que todos los estados poseen sucesor (todos los nodos son origen de una arista orientada), pero en algunos casos, no todos tienen antecesor (no a todos los nodos arriba una arista). Por ejemplo ver la figura 2.4.

Como se mencionó, la evolución de un autómata celular puede reflejarse en un diagrama denominado: Diagrama de Transiciones de Estados (**STD**,

por sus iniciales en inglés *State Transition Diagram*). Este tipo de gráficos se construye con nodos representando los estados globales del CA y aristas dirigidas que conectan el nodo origen con el nodo destino, si el nodo origen evoluciona al estado destino. Esto refleja el hecho de que los STD son grafos dirigidos y puede realizarse un análisis de los mismos utilizando teoría de grafos.

En los gráficos se puede observar que todos los nodos (estados) poseen uno y solo un estado sucesor (una única arista de salida), lo cual refleja el carácter determinista del CA. Asimismo, un nodo puede ser alcanzado, por 0, 1 o más aristas. Este hecho es interesante de observar, debido a que si un autómata celular posee N estados, entonces el gráfico estará compuesto por N nodos y también de N aristas (sólo una arista abandona cada nodo). Si existen N aristas en el grafo y algunos nodos pueden ser alcanzados por más de una arista, claramente existirán nodos que no serán alcanzados por ninguna debido a que la cantidad de nodos y aristas es constante e igual: N. Como ya se ha mencionado, esos serán los nodos (estados), a los que no arriba ninguna arista, conforman el Jardín del Edén del CA.

Por ejemplo, en la figura 2.4 se observa que los estados 011, 110 y 101 constituyen el Jardín del Edén de este autómata (R30 y 3 celdas), ya que no poseen estados predecesores y es imposible que al autómata evolucione hacia alguno de estos estados, los mismos sólo pueden existir como condiciones iniciales del sistema. También se observa que el autómata tiene un único atractor: el estado 000, el cual constituye, como se indicó anteriormente, un punto fijo. En teoría de sistemas dinámicos (y los autómatas celulares se los puede considerar como representaciones discretas de sistemas dinámicos [4, p. 33]) un punto fijo es aquel en el cual convergen todas las trayectorias de un determinado centro de atracción (*basin of atraction*) y corresponde al tipo de atractor más simple [8, p. 171].

Además de la irreversibilidad de los autómatas celulares, se debe mencionar la irreductibilidad de los mismos. Un CA es computacionalmente irreducible cuando, dada una condición inicial ($t = 0$) conocida, la manera optima para conocer su estado en el instante $t = n$ es calcular uno por uno los estados en los n instantes discretos. En otras palabras, no existe una forma computacionalmente más "económica" de calcular el estado del autómata en el instante n [4,].

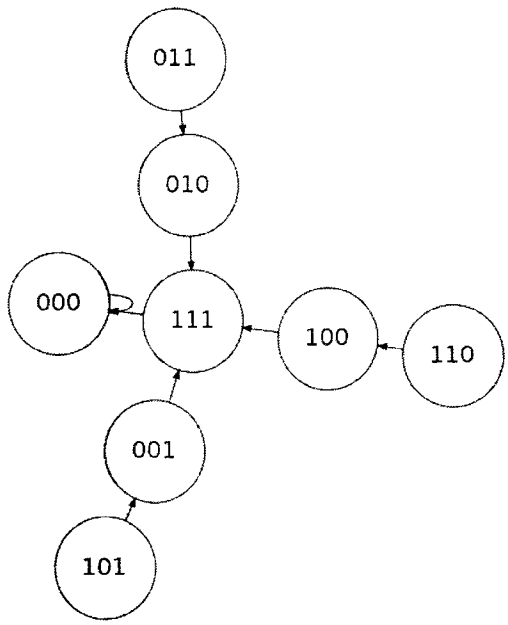


Figura 2.4: STD para un CA de 3 celdas y Regla 30

Otro aspecto a tener en cuenta en la caracterización de los CAs, es el parámetro de Langton: λ. Este indica la proporción de estados (locales) que permanecen activos, es decir cuya celda central del vecindario sigue activa. En los CAs con $k = 2$, una celda se encuentra activa si adopta el valor 1, en caso contrario la celda se encuentra en un estado inactivo (*quiescent state*). Si el autómata celular está formado por un vecindario de p celdas y cada una de ellas puede adoptar uno de k valores posibles se tienen, como ya se ha mencionado, k^p vecindarios. Si n_q son aquellos vecindarios que evolucionan a una celda inactiva, entonces el parámetro de Langton está dado por [7, pp. 74-81]:

$$\lambda = \frac{k^p - n_q}{k^p} \tag{2.1}$$

Cuando los k estados posibles son equiprobables se tiene: $\lambda = 1 - \frac{1}{k}$ y en el caso particular de $k = 2$: $\lambda = \frac{1}{2}$. Por lo que en general es de interés el intervalo: $0 \leq \lambda \leq 1 - \frac{1}{k}$ mostrando la relación con las clases de Wolfram que se observa a continuación [7, p.79]:

Así como se observa en la figura 2.4 el carácter determinista del autómata celular, no todos los CAs se comportan de esta forma. Se pueden construir CAs que contengan cierta cantidad de aleatoriedad en el cálculo del estado al cual evolucionará cada celda. Esta clase de autómatas se denomina: Autómatas Celulares Estocásticos o Autómatas Celulares Probabilísticos (**PCA** del inglés *Probabilistic Cellular Automaton*). En estos casos las celdas pueden

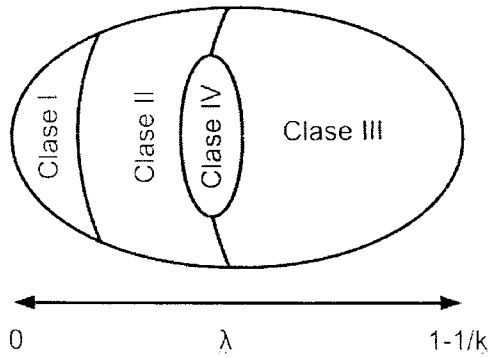


Figura 2.5: Relación entre el parámetro de Langton y las 4 clases de CAs de Wolfram. Tomado de [7, p.79]

evolucionar en forma completamente aleatoria (su valor en el instante $t+1$ se elige en forma aleatoria) o puede evolucionar a uno de entre varios estados diferentes, cada uno de ellos con cierta probabilidad de ocurrencia asociada. Según el artículo [12]: "*los PCA poseen, al menos en teoría, un mayor grado de generalidad*", de la misma forma que una máquina de Turing probabilística es más general que una determinista.

Se ha mencionado al inicio de este capítulo que las celdas de un CA actualizan su valor en forma sincrónica, es decir: el valor de las N celdas de un CA evolucionan simultáneamente en cada intervalo de tiempo. Pero también existen autómatas celulares en los cuales las celdas evolucionan en forma asincrónica. En este último tipo de autómata no todas las celdas actualizan su valor en forma simultanea. Existen dos formas de comportamiento asincrónico [7]:

- Por paso: se actualiza una celda por intervalo de tiempo en algún orden establecido (o en forma aleatoria).
- Por tiempo: cada celda posee un "reloj" interno que le indica el momento de actualizarse.

También se ha mencionado que todas las celdas que conforman el autómata evolucionan de acuerdo a una (única) regla, pero se pueden crear CAs en los cuales cada celda evolucione empleando diferentes reglas. En el caso más sencillo, se pueden emplear dos reglas de evolución y que "algunas" celdas evolucionen siguiendo la primer regla y "otras" celdas evolucionen en función de la segunda regla. Esto puede abrir todo un universo de posibilidades, ya que las reglas no necesariamente se deben aplicar en forma fija, sino que se puede aplicar una de ellas u otra en función de algún criterio (mayoría de celdas "activas" en el vecindario, mayoría de celdas "activas" en la mitad

derecha o izquierda, etc.) [13].

Para finalizar este capítulo, otro tipo de autómatas celulares, son los denominados Autómatas Celulares Cuánticos (**QCA**, del inglés *Quantum Celullar Automata*). Este tipo de CA en lugar de emplear reglas de evolución deterministas, emplean *amplitudes* [14] aplicadas los vectores que constituyen una base del espacio vectorial (de los estados/valores de las celdas). En el ámbito de mecánica cuántica el estado de un sistema se encuentra representado por un vector (en inglés *state vector*) ψ, el cual resulta de la combinación lineal de los vectores de la base del espacio vectorial:

$$|\psi\rangle = C_0 |X_0\rangle + C_1 |X_1\rangle + \cdots + C_i |X_i\rangle \,^1 \tag{2.2}$$

donde los coeficientes complejos C_i son las amplitudes y $|C_i|^2$ es la probabilidad de que el sistema asuma (colapse en) el estado $|X_i\rangle$. Por lo tanto, $\sum_i |C_i|^2 = 1$ (por condición de normalización) ver [15, Capítulo 4].

[1] Ejemplo tomado de la página 106 de [15]. Para mayor detalle de la notación de Dirac ver [16, Capítulo 1]

La vida es compleja: tiene una parte real y una imaginaria.[1]

Anónimo

3

Reglas de evolución compuestas

Como se presentó en el capítulo anterior, pequeños incrementos en el rango, orden o valores posibles (k) en los CA, implican que la cantidad de diferentes CAs susceptibles de ser estudiados, se incremente en forma exponencial haciendo impracticable (incluso con la potencia computacional actual) un análisis exhaustivo de todos ellos.

Los ECA, ampliamente estudiados por Wolfram, son los autómatas celulares más sencillos que existen: CA 1D, orden 1, $r = 1$ y $k = 2$ y aún así existen 256 reglas de evolución diferentes para este tipo de autómatas. Si se incrementara en la mínima unidad alguno de los parámetros mencionados, es decir construir un CA levemente más complejo, la variedad de autómatas se tornaría tan grande que volvería impracticable un análisis detallado de cada uno de los mismos. Por ejemplo con $r = 2$, la cantidad de reglas de evolución distintas crecerá de 256 a: $k^{k^{2\times r+1}} = 2^{2^{2\times 2+1}}$ =4.294.967.296!.

Otra aproximación, que permite construir autómatas celulares más complejos, pero que a la vez permite mantener dentro de límites manejables la *cantidad* de reglas a analizar, implica emplear las 256 reglas de los ECA para construir reglas de evolución compuestas.

[1] Texto original en [17]: ... *Life is complex – it has both real and imaginary parts.*

Una regla de evolución compuesta es aquella que surge de la aplicación sucesiva de 2 o más reglas de evolución simples en cada instante de tiempo (*time step*), de la misma forma que en el ámbito del análisis funcional se procede a construir una función h a partir de las funciones f y g: $h(x) = g(f(x))$ (o en forma más compacta $h = g \circ f$); se procede a construir reglas de evolución compuestas a partir de otras reglas de evolución simples. Por ejemplo, se puede construir una regla de evolución compuesta aplicando la regla 30 ($R30$) y luego la regla 45 ($R45$) y se obtendría la regla $R45 \circ R30$ (ver cuadro 3.4 y figura 3.2).

Si bien, el conjunto de reglas que se obtienen de esta forma es cerrado para la operación composición, el tamaño del vecindario se incrementará (además la composición suele ser una operación no conmutativa)[8, p. 43]. Como se aprecia en el detalle de evolución de las reglas que se tratarán en este capítulo (para la primera regla ver cuadro 3.4), la composición de dos reglas de evolución para autómatas celulares con vecindario de radio igual a uno, es equivalente a la regla de evolución de un autómata celular con vecindario de radio igual a dos.

Para el caso de la composición de sólo dos reglas de ECA la cantidad de alternativas distintas resultantes que se obtendrá será el producto cartesiano de todas las reglas elementales: 256×256 =65.536, que si bien es una cantidad importante de reglas, es mucho más manejable que las 4.294.967.296 existentes para los CA unidimensionales de radio 2 y $k = 2$.

En este trabajo se ha elegido un subconjunto de reglas compuestas, de las 65.536 posibles, comparando la longitud de ciclo máxima de un autómata celular constituido por 17 celdas, con la longitud de ciclo máxima de un CA R30 con una estructura también de 17 celdas. La referencia del CA R30 se ha definido considerando el trabajo de Wolfram en el que se propone a la regla 30 como la mejor candidata para generar secuencias pseudo-aleatorias, como se encuentra descripto en [4, pp. 267-367]. Además la selección de una estructura de 17 celdas se basa en: la aparición de singularidades en el comportamiento de los ciclos para autómatas celulares 1D clase III, cuando poseen estructuras de celdas con cantidades entre $2^k - 1$ y $2^k + 1$, para $k \in \mathbb{Z}$, como se expone en [8, pp. 77]. Debido a las restricciones de la capacidad computacional disponible, se han elegido estructuras de estudio en el entorno de $2^4 \pm 1$ y particularmente la de 17 celdas.

El mayor ciclo para la R30 con 17 celdas transita 10846 estados. La cantidad de reglas compuestas (del tipo $R_2 \circ R_1$) con estructuras de 17 celdas que superan esa longitud son 128, pero si simplificamos esta información utilizando la Tabla de forma y equivalencia de reglas que se presenta en [4, pp. 516-521] para eliminar composiciones con reglas equivalentes, y además eliminamos la composición de reglas sobre sí mismas (o sus equivalentes), la cantidad se reduce a 18:

Regla	Ciclo Máximo	Observaciones
$\boldsymbol{R154 \circ R15}$	**78812**	
$\boldsymbol{R105 \circ R45}$	**80835**	
$R150 \circ R45$	64702	
$\boldsymbol{R166 \circ R45}$	**100878**	
$R170 \circ R45$	78812	(C.R.) de $R154 \circ R15$
$R204 \circ R45$	78812	Idem R45
$R154 \circ R51$	78812	Idem R89
$R45 \circ R105$	80835	
$\boldsymbol{R154 \circ R105}$	**64702**	
$R45 \circ R150$	64702	(C.R.) de $R154 \circ R105$
$\boldsymbol{R154 \circ R150}$	**80835**	
$R15 \circ R154$	78812	(C.) de $R154 \circ R15$
$R45 \circ R154$	100878	
$R51 \circ R154$	78812	Idem R101
$R105 \circ R154$	64702	(C.) de $R154 \circ R105$
$R150 \circ R154$	80835	
$R45 \circ R170$	78812	(C.R.) de $R154 \circ R15$
$R45 \circ R204$	78812	Idem R45

Cuadro 3.1: Reglas compuesta cuyos ciclos máximos para una estructura de 17 celdas superan al ciclo máximo de la regla 30 (ver texto). (C.) refiere a la conjugación: 0 y 1 intercambian sus roles en la regla. (R.) refiere a reflexión. (C.R.) refiere a las operaciones de conjugación y reflexión combinadas (siguiendo el criterio de clasificación de reglas equivalentes usada por Wolfram en [4, pp. 521-522])

De las reglas listadas en la tabla precedente se eligieron las siguientes: $R154 \circ R15$, $R105 \circ R45$, $R166 \circ R45$, $R154 \circ R105$ y $R154 \circ R150$ (se presentaron marcadas en negritas en la tabla anterior).

Además se han considerado las reglas:

Regla	Ciclo Máximo
$R45 \circ R30$	**5100**
$R105 \circ R30$	**1938**
$R106 \circ R85$	**10846**
$R166 \circ R106$	**8483**

Cuadro 3.2: Reglas compuesta cuyos ciclos máximos para una estructura de 17 celdas no superan al ciclo máximo de la regla 30 (ver texto).

Los autómatas celulares son considerados el equivalente discreto de los sistemas dinámicos [4, p.116], los cuales se comportan en forma estable o inestable. Una herramienta para para medir las características topológicas y la impredecibilidad de un sistema [18] es el exponente de Lyapunov. Cuando el exponente de Lyapunov es menor que 0, entonces el sistema converge hacia un punto fijo o una órbita periódica (el sistema está en equilibrio metaestable). Pero si el exponente de Lyapunov es mayor que cero, el sistema se comporta de forma caótica [7, p.36].

Para el estudio de los CAs 1D mediante esta herramienta, Wolfram propuso un exponente de Lyapunov izquierdo y otro derecho, que permiten calcular la tasa de propagación de la información. Además Wolfram indica que los autómatas celulares clase III poseen exponentes de Lyapunov positivos [4, p. 279].

Otra aproximación es la introducida por Bagnoli et. al.: el exponente de Lyapunov máximo (**MLE**, por sus sigla en inglés *Maximal Lyapunov Exponent*) el cual es independiente de la dimensión del autómata y permite calcular un único valor del mismo para un CA dado[19].

La siguiente tabla exhiben los exponentes de Lyapunov calculados para cada una de las reglas elegidas:

Regla	λ_L	λ_R	λ_{max}
30	0.244 ± 0.003	1	0.6594 ± 0.0005
45	0.172 ± 0.004	1	0.7181 ± 0.0002
45 ∘ 30	0.615 ± 0.004	2	1.0290 ± 0.0006
105 ∘ 30	1.142 ± 0.003	2	1.10444 ± 0.00007
105 ∘ 45	1.002 ± 0.004	2	1.1060 ± 0.0005
166 ∘ 45	1.205 ± 0.004	1.023 ± 0.004	0.9180 ± 0.0006
106 ∘ 85	2	-0.757 ± 0.003	0.6593 ± 0.0005
154 ∘ 105	2	1.011 ± 0.004	1.0913 ± 0.0003
166 ∘ 106	2	0.074 ± 0.006	0.9398 ± 0.0006
154 ∘ 150	2	1.002 ± 0.004	1.1053 ± 0.0005
154 ∘ 15	0	1.173 ± 0.004	0.7181± 0.0002

Cuadro 3.3: Detalle del exponente de Lyapunov para cada regla. λ_L y λ_R son los exponentes de Lyapunov izquierdo y derecho de CA 1D, de acuerdo a [4, pp. 278-280]. En cambio, λ_{max} se corresponde con el exponente de Lyapunov máximo e independiente de la dimensión del CA, de acuerdo a [19], [20] y [9].

En el resto del capítulo se exponen las reglas elegidas, así como también resultados de los estudios realizados sobre ellas. En los anexos A a G también se encontrará información adicional relativa a cada una de las reglas que serán tratadas en las secciones 3.1 a 3.9.

3.1. Regla $45 \circ 30$

Esta regla se compone aplicando las reglas de ECA 30 y 45 (en ese orden) por lo cual se denominará $45 \circ 30$ (regla 45 compuesta con 30). También es posible identificarla expresando su valor numérico (bits de la tabla de verdad en orden lexicográfico descendente, ver capítulo 2) en base32[2] (como se propone en [4, p. 303]). Para esta regla, su valor numérico decimal es 3.824.294.925, y su equivalente en base32hex es **SFP1O38**.

A continuación se presenta su tabla de verdad:

[2]La representación en base 32 permite representar valores numéricos grandes utilizando menor cantidad de dígitos, es decir, en forma más compacta. A diferencia de la representación base64, base32 no es sensible a mayúsculas. Existen diferentes formas de representar números en base32, dos de ellas se encuentran detalladas en RFC4648 [21]. De ellas se utilizará en este trabajo, como ha hecho Wolfram en [4, p. 303], la codificación base32 con alfabeto hexadecimal extendido (base32hex). Esta forma de codificación emplea para los primeros 16 dígitos los símbolos $0 \sim F$ (de forma similar a la codificación base 16 o hexadecimal, de ahí su nombre) y luego continúa con el alfabeto (inglés) utilizando para los dígitos 17° a 31° las letras $G \sim V$.

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Cuadro 3.4: Tabla de verdad para CA $R45 \circ R30$

A continuación se puede apreciar la evolución de todos los vecindarios posibles:

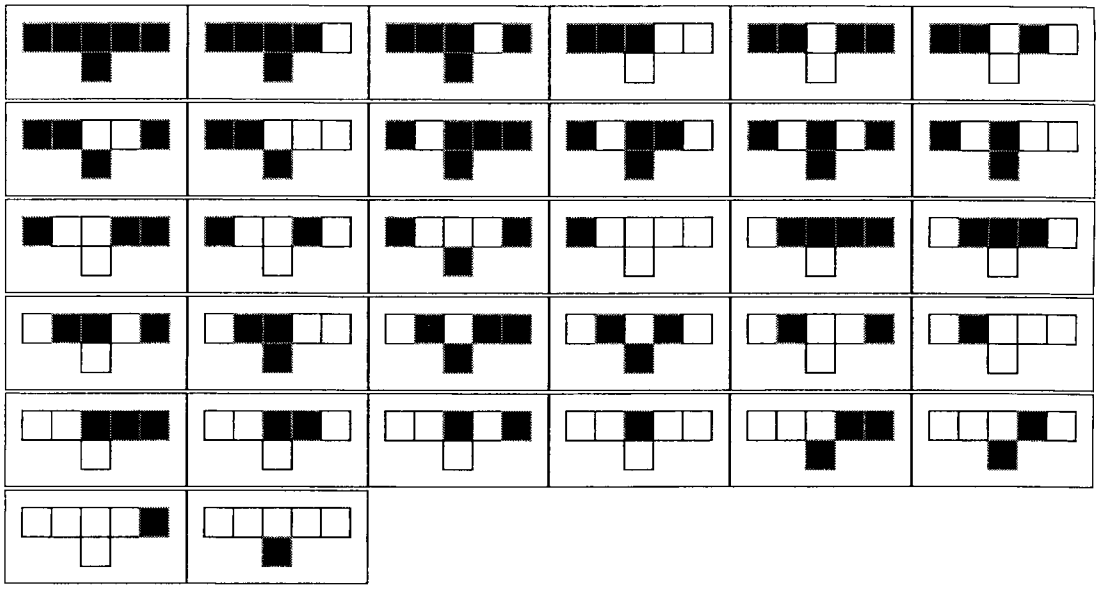


Figura 3.1: Regla $R45 \circ R30$

Observando la figura 3.1, el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.1}$$

El siguiente gráfico muestra el detalle de la evolución de esta regla en un instante de tiempo:

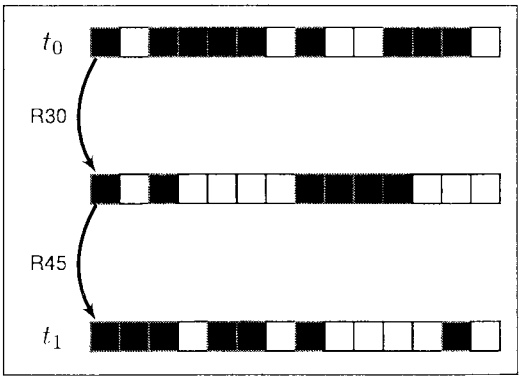


Figura 3.2: Evolución CA $R45 \circ R30$

Esta regla de evolución es la composición de las reglas 30 y 45 (en ese orden), cuyas expresiones booleanas son:

$$\boldsymbol{R30} = c_{i-1} \veebar (c_i \vee c_{i+1}) \tag{3.2}$$

$$\boldsymbol{R45} = c_{i-1} \veebar (c_i \vee \overline{c_{i+1}}) \tag{3.3}$$

En ambos casos al tratarse de autómatas con $r = 1$, cada celda posee un vecindario de 3 celdas. Al aplicar dos reglas, con estas características, en forma simultanea la regla resultante equivalente posee $r = 2$, por lo tanto, cada celda tendrá un vecindario de 5 celdas. Dicho esto, una celda genérica (c_i) posee el siguiente vecindario: c_{i-2}, c_{i-1}, c_i, c_{i+1} y c_{i+2} y al aplicar ambas reglas de evolución se obtendrá la siguiente expresión booleana:

$$[c_{i-2} \veebar (c_{i-1} \vee c_i)] \veebar \{[c_{i-1} \veebar (c_i \vee c_{i+1})] \vee \overline{[c_i \veebar (c_{i+1} \vee c_{i+2})]}\} \tag{3.4}$$

Si bien en la expresión booleana precedente los literales se relacionan empleando los conectivos lógicos **XOR** (operación lógica "o exclusivo", representada mediante $\veebar$), **OR** (operación lógica "o", representada mediante $\vee$) y **NOT** (negación lógica, representada por medio de $\neg$), estos conectivos (o cualquier otro) pueden representarse empleando solamente los conectivos **AND** (operación lógica "y", representada utilizando $\wedge$), OR y NOT. Inclusive, cualquier conectivo puede ser representado empleando únicamente AND y NOT ó OR y NOT (conjuntos suficientes). Sin embargo, por cuestiones de practicidad, en las expresiones se emplean los 3 conectivos (AND, OR y NOT), tal como se detalla en [22, p. 56].

Se utiliza la tabla de verdad (cuadro 3.4) que muestra todos los valores posibles que puede adoptar la expresión (3.4), para construir una expresión equivalente empleando la denominada suma de productos (**SOP** del inglés

Sum Of Products). Esta suma corresponde a la operación OR de sus términos. Y cada término en dicha suma es un *término producto.* Los términos producto corresponden a la operación AND entre cada uno de los literales (un literal es una variable) que en él se encuentran ([22, p.118]). Para proceder a la construcción de la SOP, primero se deberán considerar sólo aquellas filas en la tabla de verdad, cuyo resultado es 1. Para cada una de las filas consideradas, se construye un término producto teniendo en cuenta el valor de cada uno de los literales en esa fila. Si el literal es 1 se lo emplea sin cambios. Por el contrario, si el literal es 0, se lo utiliza negado.

Para el caso de la regla $R45 \circ R30$, los términos producto que se obtienen se muestran en la siguiente tabla:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}	Término
0	0	0	0	0	1	$\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}$
0	0	0	1	0	1	$\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}}$
0	0	0	1	1	1	$\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}\ c_{i+2}$
0	1	0	1	0	1	$\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}}$
0	1	0	1	1	1	$\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ c_{i+1}\ c_{i+2}$
0	1	1	0	0	1	$\overline{c_{i-2}}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}}$
1	0	0	0	1	1	$c_{i-2}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}$
1	0	1	0	0	1	$c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}}$
1	0	1	0	1	1	$c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ c_{i+2}$
1	0	1	1	0	1	$c_{i-2}\ \overline{c_{i-1}}\ c_i\ c_{i+1}\ \overline{c_{i+2}}$
1	0	1	1	1	1	$c_{i-2}\ \overline{c_{i-1}}\ c_i\ c_{i+1}\ c_{i+2}$
1	1	0	0	0	1	$c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}$
1	1	0	0	1	1	$c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}$
1	1	1	0	1	1	$c_{i-2}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2}$
1	1	1	1	0	1	$c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}\ \overline{c_{i+2}}$
1	1	1	1	1	1	$c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}\ c_{i+2}$

Cuadro 3.5: Construcción de los términos producto de la SOP para CA $R45 \circ R30$ (por simplicidad se omite el símbolo $\wedge$ que representa la operación AND, además la barra sobre los literales corresponde a la forma negada (NOT) de los mismos)

De esta forma, la expresión de SOP equivalente a la expresión (3.4),

es:

$$\begin{aligned}
&(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}\ c_{i+2}) + \\
&(\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ c_{i+1}\ c_{i+2}) + (\overline{c_{i-2}}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + \\
&(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + \\
&(c_{i-2}\ \overline{c_{i-1}}\ c_i\ c_{i+1}\ \overline{c_{i+2}}) + (c_{i-2}\ \overline{c_{i-1}}\ c_i\ c_{i+1}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + \\
&(c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}\ \overline{c_{i+2}}) + \\
&(c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}\ c_{i+2})
\end{aligned} \tag{3.5}$$

Utilizando la tabla de verdad se puede representar cualquier función booleana como SOP, es decir, en forma AND-OR (ó OR-AND) de dos niveles (ver [22, pp. 118-124]). Pero se debe tener en cuenta que el procedimiento antes mencionado no, necesariamente, permite obtener un expresión mínima de la función booleana. Una expresión mínima, es aquella que representa una función booleana utilizando la menor cantidad de elementos. Esto facilita la implementación y ahorra costos, además de mejorar la eficiencia del circuito (tanto en términos de velocidad como de disipación de energía).

Para hallar la expresión mínima de una función booleana existen diversos métodos (por ejemplo empleando diagramas de Karnaugh o el método de Quine-McKlusky[3]).

En el caso de la expresión (3.5), luego de aplicar los procedimientos de minimización quedará representada empleando lógica de 2 niveles (AND-OR), tal como se expone en la expresión (3.6) y su correspondiente diagrama del circuito lógico se puede apreciar en la figura 3.3a.

$$\begin{aligned}
&(\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1}) + (c_{i-2}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ \overline{c_{i-1}}\ c_i) + (c_{i-2}\ c_i\ c_{i+1}) + \\
&(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+2}}) + (c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i-1}}) + (\overline{c_{i-2}}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}})
\end{aligned} \tag{3.6}$$

No obstante todo lo mencionado hasta aquí sobre las expresiones mínimas en SOP, se hace notar al lector que en la implementación del circuito la utilización de un sólo tipo de elemento (compuerta lógica) contribuye en gran medida a la eficiencia de producción. Por ello en la producción de circuitos se

[3]La descripción y el detalle de los métodos de simplificación o minimización quedan fuera del alcance de este trabajo. Para aquel lector que desee profundizar en el tema, se recomienda la lectura de los capítulos 6 y 7 de [22]. Asimismo, se encuentran herramientas de software que automatizan los procedimientos de minimización, una de ellas puede encontrarse en `http://www.32x8.com`.

emplea un único tipo de elemento lógico. Este puede ser la compuerta lógica **NAND** (NOT AND) ó la **NOR** (NOT OR), tal como se explica en [22, p. 111].

Asimismo, todo circuito AND-OR de dos niveles posee un circuito equivalente NAND-NAND de dos niveles[22, p. 110] y el procedimiento para la conversión es sencillo, como se lo explica a continuación.

Teniendo en cuenta la característica de idempotencia en la negación lógica ($A = \overline{\overline{A}}$) y las leyes de DeMorgan ($\overline{A+B} = \overline{A} \cdot \overline{B}$ y $\overline{A \cdot B} = \overline{A} + \overline{B}$) (ver [22, cap. 4]). el proceso de conversión, partiendo del circuito lógico AND-OR (figura 3.3a), consiste en aplicar un inversor (NOT) en la salida de cada una de las compuertas AND y un inversor en cada una de las entradas de la compuerta OR (como los inversores se cancelan mutuamente, el circuito sigue siendo equivalente), tal como se muestra en la figura 3.3b. Luego sobre la compuerta OR con sus entradas negadas se aplica DeMorgan obteniéndose la compuerta NAND buscada (ver la figura 3.3c).

Al finalizar el procedimiento de conversión desde AND-OR hacia NAND-NAND, la expresión booleana mínima, basada en compuertas universales NAND, quedará:

$$\overline{\overline{(\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1})}\ \overline{(c_{i-2}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ \overline{c_{i-1}}\ c_i)}\ \overline{(c_{i-2}\ c_i\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}})}\ \overline{(\overline{c_{i-2}}\ c_{i-1}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}})}} \tag{3.7}$$

Sin invalidar todo lo hasta aquí expuesto sobre circuitos lógicos de dos niveles, no se debe dejar de mencionar que existen casos especiales en los cuales dicho tipo de circuitos no mejoran la eficiencia de los circuitos de múltiples niveles, tal como se expone en [22, pp. 222-226].

Para finalizar esta digresión, se hace notar al lector que en las reglas de evolución subsiguientes se podría realizar el mismo análisis lógico que el expuesto en esta sección. Pero para evitar redundar en el tema sólo se expondrán las expresiones mínimas de dos niveles (tanto AND-OR como NAND-NAND) ya calculadas.

También se pueden tener en cuenta, las expresiones algebraicas de las

reglas 30 y 45:

$$\boldsymbol{R30} = (c_{i-1} + c_i + c_{i+1} + c_i c_{i+1}) MOD2 \tag{3.8}$$

$$\boldsymbol{R45} = (1 + c_{i-1} + c_{i+1} + c_i c_{i+1}) MOD2 \tag{3.9}$$

Cuando se efectúa la composición, la expresión algebraica de regla $R45 \circ R30$ queda:

$$\begin{aligned} \{1 + (c_{i-2} + c_{i-1} + c_i + c_{i-1}c_i) MOD2 + (c_i + c_{i+1} + c_{i+2} + \\ c_{i+1}c_{i+2}) MOD2 + [(c_{i-1} + c_i + c_{i+1} + c_i c_{i+1}) MOD2 \times \\ (c_i + c_{i+1} + c_{i+2} + c_{i+1}c_{i+2}) MOD2]\} MOD2 \end{aligned} \tag{3.10}$$

Luego de aplicar propiedades de aritmética modular, la expresión algebraica mínima será:

$$\begin{aligned} (1 + c_{i-2} + c_{i-1} + c_i + c_{i+2} + c_{i-1}c_{i+1} + c_{i-1}c_{i+2} + c_i c_{i+2} + c_{i+1}c_{i+2} + \\ c_{i-1}c_{i+1}c_{i+2} + c_i c_{i+1}c_{i+2}) MOD2 \end{aligned} \tag{3.11}$$

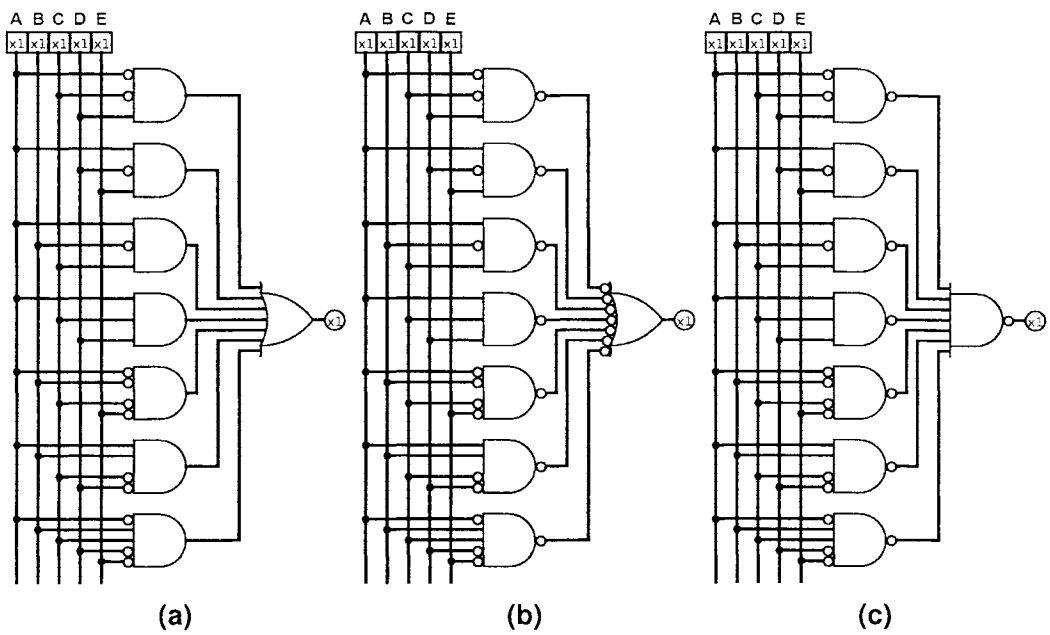
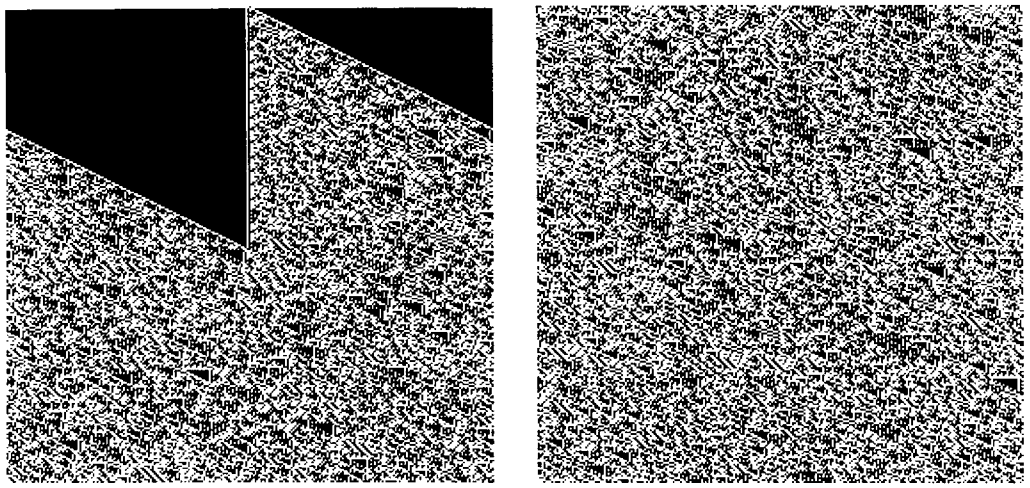


Figura 3.3: Circuitos lógicos equivalentes de la regla $R45 \circ R30$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para

una condición inicial aleatoria:



(a) (b)

Figura 3.4: Diagramas espacio-temporales para la Regla $45 \circ 30$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El estudio de entropía realizado para esta regla (y con una estructura de 15 celdas) se muestra en la figura D.3 del anexo D, donde se puede apreciar que el autómata evoluciona hacia un estado de mayor organización. Comparativamente con la regla 30, se observa que el grado de organización es menor.

Finalmente, la evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito, para diferentes cantidades de celdas, se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.3. El cálculo estadístico se ha realizado para 1000 secuencias de 100.000 bits (esta aclaración vale para las tablas que se presentarán para las demás reglas compuestas de este trabajo):

# Celdas	μ	σ
6	3,826	1,95641
7	83,326	8,29087
8	18,213	8,78856
9	47,818	7,82177
10	16,743	10,9938
11	235,74	46,281
12	36,3	23,7738
13	636,551	190,879
14	119,673	69,6456
15	2773,3	668,711

# Celdas	μ	σ
16	3270,57	2512,59
17	3528,81	1870,2
18	1291,26	1095,13
19	11773,5	1382,37
20	9604,64	1433,03
21	4247,01	2638,47
22	25101,2	17386,8
23	46612,8	9654,61
24	23322,5	13193,8
25	35947,8	12752,9

Cuadro 3.6: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R45 \circ R30$ con diferentes cantidades de celdas

3.2. Regla $105 \circ 30$

Esta regla se compone aplicando las reglas de ECA 30 y 105 (en ese orden) por lo cual se denominará $105 \circ 30$. Su valor numérico decimal es 3.792.838.125, y su equivalente en base32hex es **S891RR8**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Cuadro 3.7: Tabla de verdad para CA $R105 \circ R30$

Se muestra a continuación la evolución para cada uno de los posibles vecindarios:

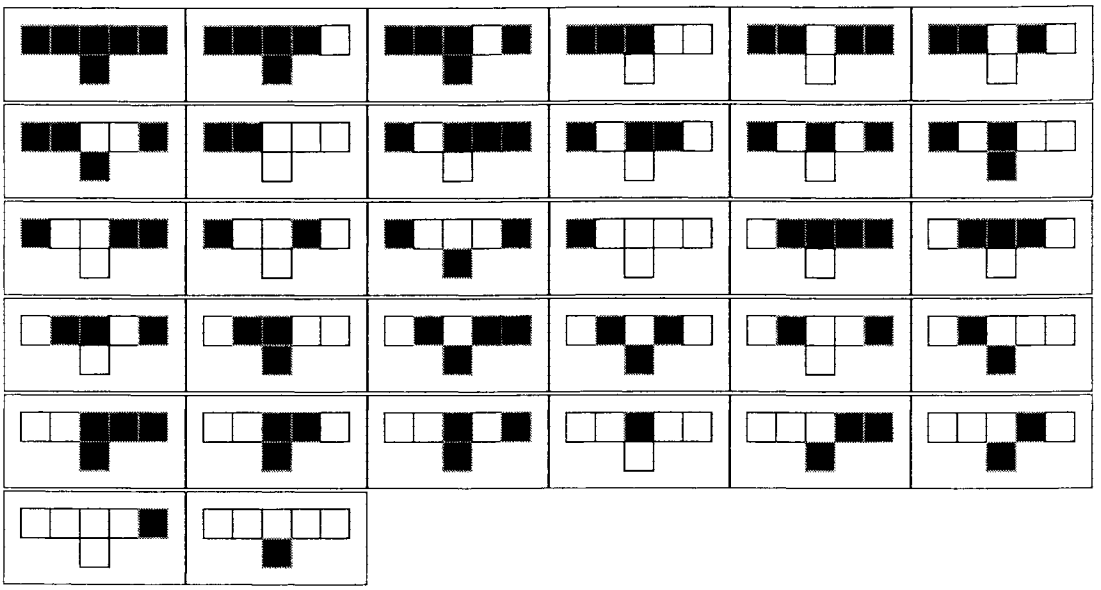


Figura 3.5: Regla $R105 \circ R30$

Como se puede apreciar en la figura 3.5, el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0.5 \tag{3.12}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

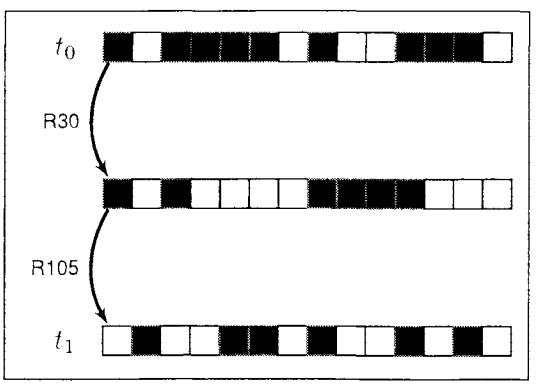


Figura 3.6: Evolución CA $R105 \circ R30$

Esta regla de evolución es la composición de las reglas 30 y 105 (en ese orden), cuyas expresiones booleanas son:

$$\boldsymbol{R30} = c_{i-1} \veebar (c_i \vee c_{i+1}) \tag{3.13}$$

$$\boldsymbol{R105} = c_{i-1} \veebar c_i \veebar \overline{c_{i+1}} \tag{3.14}$$

Como se ha indicado en la sección anterior, la composición de este

tipo de reglas es equivalente a una regla cuyo radio es $r = 2$ y la expresión booleana resultante de la composición es:

$$[c_{i-2} \veebar (c_{i-1} \vee c_i)] \veebar [c_{i-1} \veebar (c_i \vee c_{i+1})] \veebar \overline{[c_i \veebar (c_i \vee c_{i+2})]} \tag{3.15}$$

A partir de la tabla de verdad que se presenta en el cuadro 3.7 y utilizando los diagramas de Karnaugh, como se ha visto en la sección 3.1, se construye la expresión de dos niveles AND-OR mínima:

$$\begin{gathered}(\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}) + (\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_i\ c_{i+2}) + \\ (\overline{c_{i-2}}\ c_{i-1}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (c_{i-2}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ c_i\ c_{i+1}) + \\ (c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}})\end{gathered} \tag{3.16}$$

Y esa expresión booleana (3.16) tiene una representación mínima equivalente utilizando compuertas universales NAND, la cual se puede apreciar a continuación. El detalle de la conversión, en diagramas de circuitos lógicos, se muestra en la figura 3.7.

$$\begin{gathered}\overline{\overline{(\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_i\ c_{i+2})}} \\ \overline{\overline{(\overline{c_{i-2}}\ c_{i-1}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ c_{i-1}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ c_{i-1}\ c_i\ c_{i+1})}} \\ \overline{\overline{(c_{i-2}\ \overline{c_{i-1}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}})}}\end{gathered} \tag{3.17}$$

Las expresiones algebraicas de las reglas 30 y 105 son:

$$\boldsymbol{R30} = (c_{i-1} + c_i + c_{i+1} + c_i c_{i+1}) MOD2 \tag{3.18}$$

$$\boldsymbol{R105} = (1 + c_{i-1} + c_i + c_{i+1}) MOD2 \tag{3.19}$$

Y la expresión algebraica resultante de la composición es:

$$\begin{gathered}[1 + (c_{i-2} + c_{i-1} + c_i + c_{i+1} c_i) MOD2 + (c_{i-1} + c_i + c_{i+1} + \\ c_i c_{i+1}) MOD2 + (c_i + c_{i+1} + c_{i+2} + c_{i+1} c_{i+2}) MOD2] MOD2\end{gathered} \tag{3.20}$$

la cual puede reducirse al aplicar las propiedades de la aritmética modular:

$$(1 + c_{i-2} + c_i + c_{i+2} + c_{i-1} c_i + c_i c_{i+1} + c_{i+1} c_{i+2}) MOD2 \tag{3.21}$$

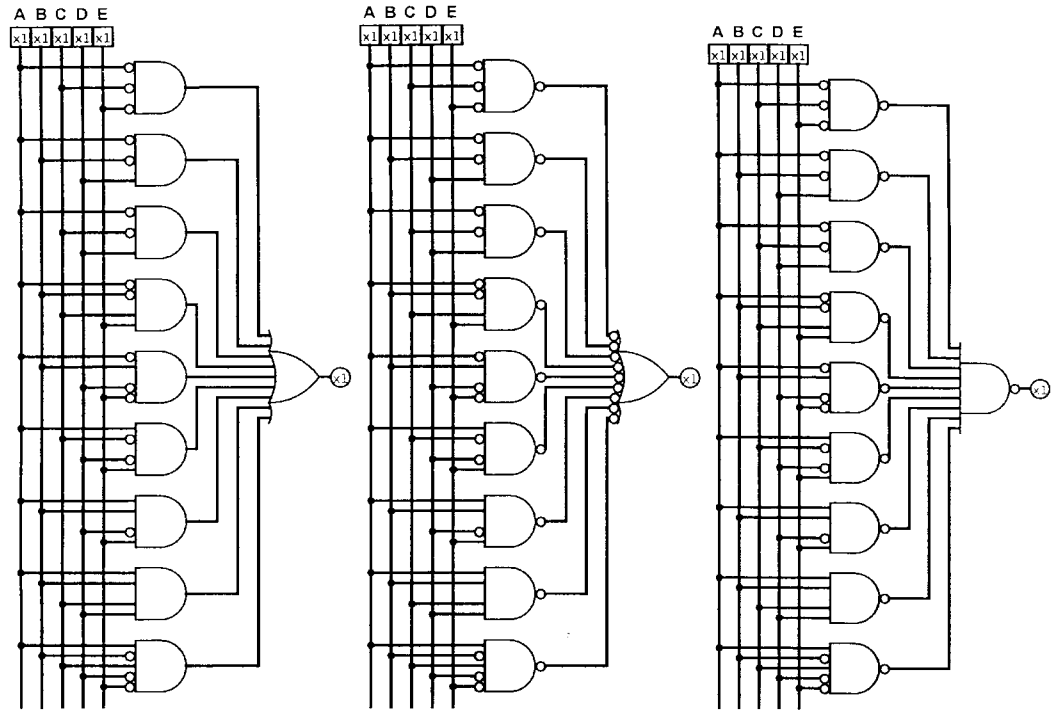


Figura 3.7: Circuitos lógicos equivalentes de la regla $R105 \circ R30$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i-2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para una condición inicial aleatoria:

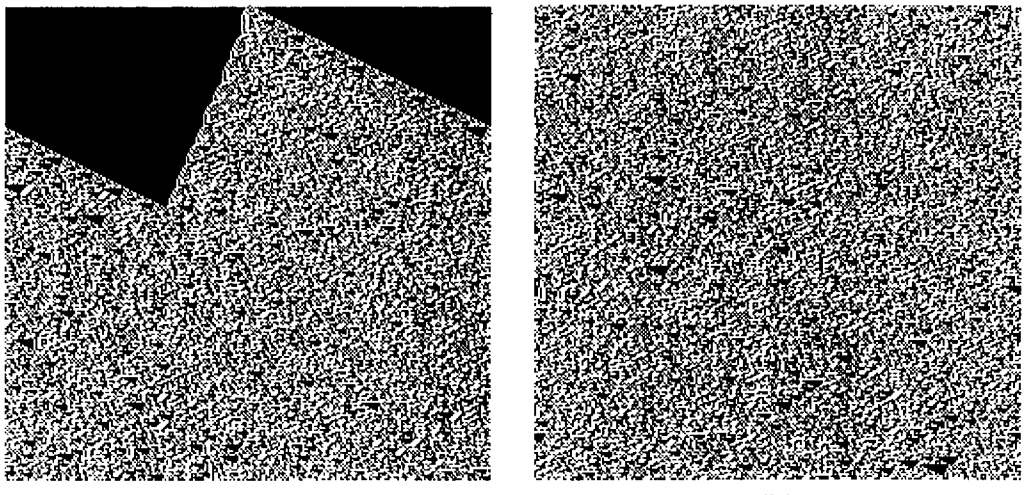


Figura 3.8: Diagramas espacio-temporales para la Regla $105 \circ 30$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El estudio de entropía realizado para esta regla (y con una estructura de 15 celdas) se muestra en la figura D.4 del anexo D, donde se puede apreciar que el autómata rápidamente (en menos de 50 *time steps*) adquiere mayor organización interna. Comparativamente con la regla 30, se observa mayor grado de organización.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.4:

# Celdas	μ	σ	# Celdas	μ	σ
6	2,28	0,685617	16	6876,76	1522,19
7	28,712	9,2767	17	650,825	333,225
8	7,623	3,88613	18	86,268	60,1039
9	7,576	2,59518	19	8702,46	2853,07
10	29,365	22,2777	20	2149,75	1078,33
11	245,33	20,3296	21	162,851	54,9123
12	9,593	6,0734	22	4545,52	1837,81
13	1292,93	151,165	23	49768,7	3006,96
14	216,814	71,784	24	490,454	268,899
15	58,302	25,7375	25	43926	7913,45

Cuadro 3.8: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R105 \circ R30$ con diferentes cantidades de celdas

3.3. Regla $105 \circ 45$

Esta regla se compone aplicando las reglas de ECA 45 y 105 (en ese orden) por lo cual se denominará $105 \circ 45$. Su valor numérico decimal es 3.738.247.470, y su equivalente en base32hex es **RR8I2BG**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	1

Cuadro 3.9: Tabla de verdad para CA $R105 \circ R45$

Para cada uno de los posibles vecindarios, a continuación se puede apreciar su evolución:

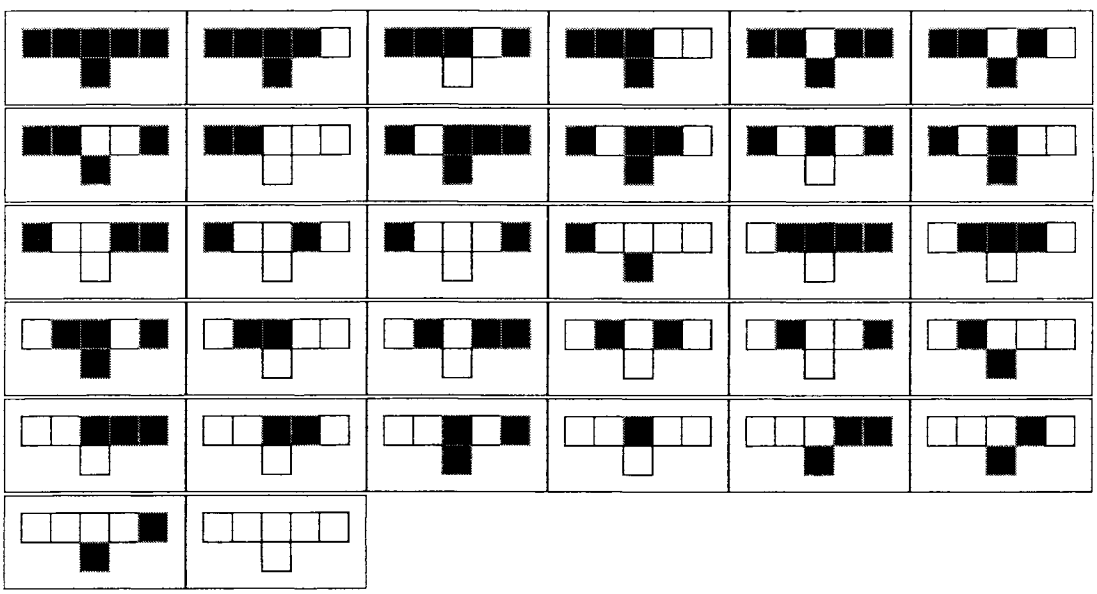


Figura 3.9: Regla $R105 \circ R45$

En la figura precedente, se pueden apreciar las 16 celdas inactivas y el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.22}$$

El gráfico que muestra la evolución de la regla en un sólo instante de

tiempo es el siguiente:

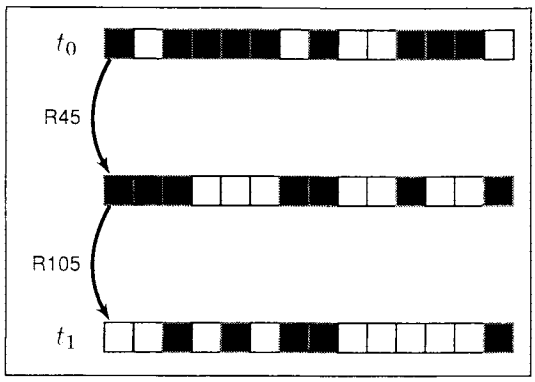


Figura 3.10: Evolución CA $R105 \circ R45$

Esta regla de evolución es la composición de las reglas 45 y 105 (en ese orden), cuyas expresiones booleanas son:

$$\boldsymbol{R45} = c_{i-1} \veebar (c_i \vee \neg c_{i+1}) \tag{3.23}$$

$$\boldsymbol{R105} = c_{i-1} \veebar c_i \veebar \neg c_{i+1} \tag{3.24}$$

La composición de ambas reglas ($R105 \circ R45$) produce la expresión booleana siguiente:

$$[c_{i-2} \veebar (c_{i-1} \vee \overline{c_i})] \veebar [c_{i-1} \veebar (c_i \vee \overline{c_{i+1}})] \veebar \overline{[c_i \veebar (c_i \vee \overline{c_{i+2}})]} \tag{3.25}$$

Como ya se ha explicado, se puede minimizar (3.25) en una expresión de 2 niveles AND-OR:

$$\begin{aligned}(c_{i-2}\ c_i\ \overline{c_{i+2}}) + (c_{i-2}\ c_i\ c_{i+1}) + (c_{i-2}\ c_{i-1}\ c_{i+1}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2}) + \\ (\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}) + (\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + \\ (c_{i-2}\ c_{i-1}\ \overline{c_i}\ c_{i+2}) + (\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}})\end{aligned} \tag{3.26}$$

Empleando el procedimiento descripto en la sección 3.1 y mostrado en la figura 3.11 se obtiene la representación mínima equivalente con compuertas universales NAND:

$$\begin{aligned}&\overline{\overline{(c_{i-2}\ c_i\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ c_i\ c_{i+1})}\ \overline{(c_{i-2}\ c_{i-1}\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2})}} \\ &\overline{\overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_i}\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ c_{i-1}\ \overline{c_i}\ c_{i+2})}} \\ &\overline{\overline{(\overline{c_{i-2}}\ c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}}\end{aligned} \tag{3.27}$$

Las expresiones algebraicas de las reglas 45 y 105 son:

$$\boldsymbol{R45} = (1 + c_{i-1} + c_{i+1} + c_i c_{i+1}) MOD2 \tag{3.28}$$

$$\boldsymbol{R105} = (1 + c_{i-1} + c_i + c_{i+1}) MOD2 \tag{3.29}$$

Y la expresión algebraica resultante de la composición es:

$$\begin{aligned}[1 + (c_{i-2} + c_i + c_{i-1}c_i) MOD2 + (1 + c_{i-1} + c_{i+1} + c_i c_{i-1}) MOD2 + \\ (1 + c_i + c_{i+2} + c_{i+1}c_{i+2}) MOD2] MOD2\end{aligned} \tag{3.30}$$

la cual puede reducirse al aplicar las propiedades de la aritmética modular:

$$(c_{i-2} + c_{i-1} + c_{i+1} + c_{i+2} + c_{i-1}c_i + c_i c_{i+1} + c_{i+1}c_{i+2}) MOD2 \tag{3.31}$$

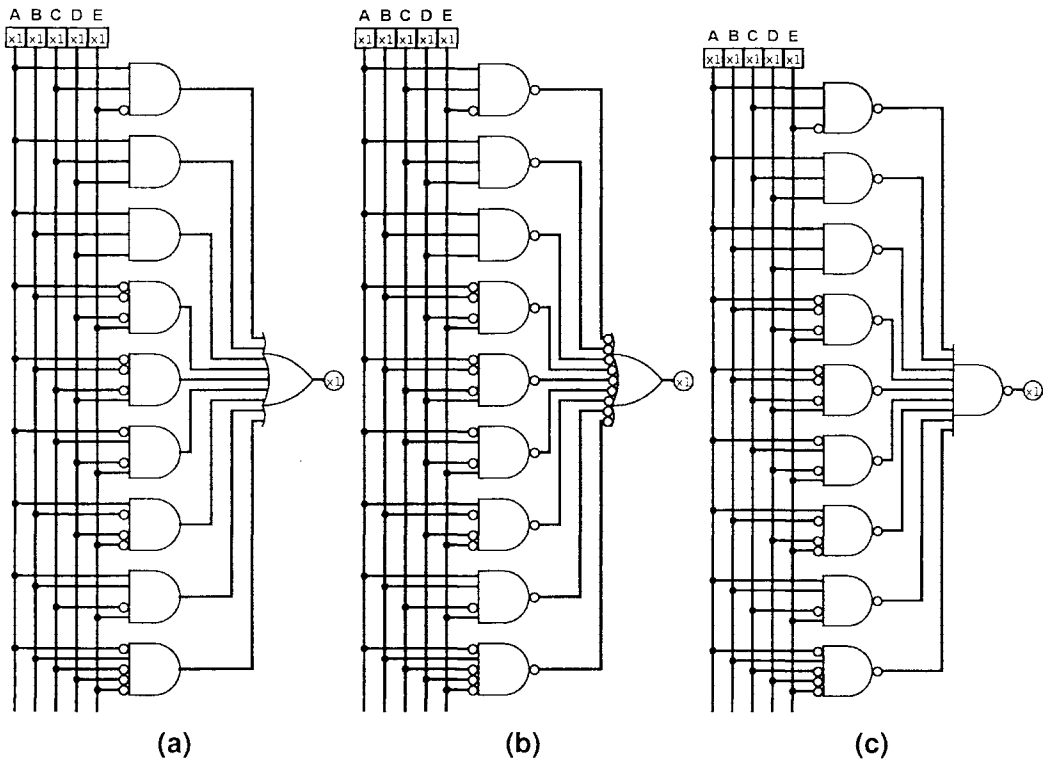
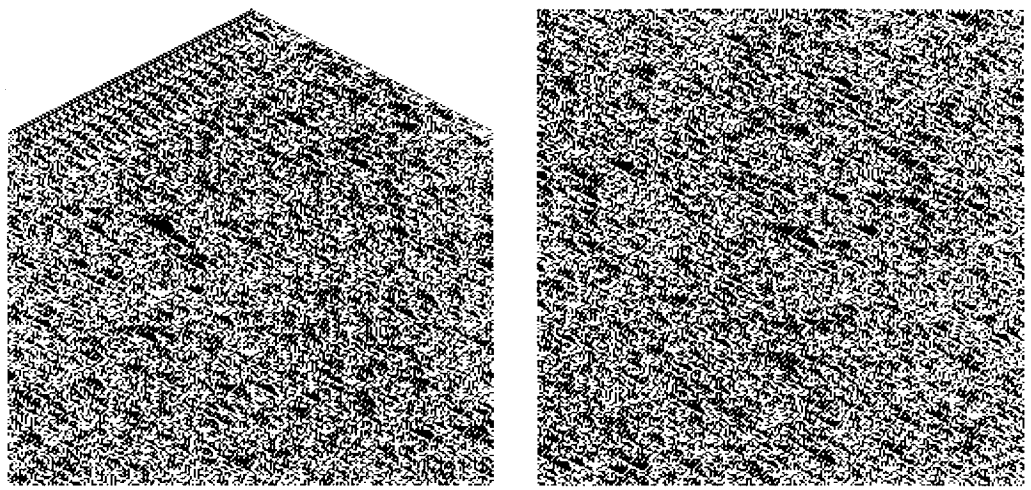


Figura 3.11: Circuitos lógicos equivalentes de la regla $R105 \circ R45$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para

una condición inicial aleatoria:



(a) (b)

Figura 3.12: Diagramas espacio-temporales para la Regla $105 \circ 45$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El estudio de entropía realizado para esta regla (y con una estructura de 15 celdas) se muestra en la figura D.5 del anexo D, donde se puede apreciar que, al igual que para $R30 \circ R105$, el autómata rápidamente adquiere mayor organización interna, aunque mantiene una entropía ligeramente superior a $R30 \circ R105$. Comparativamente con la regla 30, se observa que el grado de organización es mayor.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.5:

# Celdas	μ	σ
6	0,862	0,713768
7	84,093	29,0405
8	34,005	36,3588
9	24,218	15,0329
10	86,325	21,8389
11	1460,68	582,081
12	9,036	4,76712
13	6600,95	2094,3
14	222,201	71,7492
15	135,824	76,5113

# Celdas	μ	σ
16	657,103	397,905
17	36227,9	18053,3
18	711,228	271,114
19	46019	10178,5
20	17515,3	11141,9
21	1377,19	200,97
22	43846,6	14517,3
23	49741,7	2391,18
24	490,423	333,295
25	50000,3	1,10138

Cuadro 3.10: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R105 \circ R45$ con diferentes cantidades de celdas

3.4. Regla $166 \circ 45$

Esta regla se compone aplicando las reglas de ECA 45 y 166 (en ese orden) por lo cual se denominará $166 \circ 45$. Su valor numérico decimal es 556.674.525, y su equivalente en base32hex es **44N2RN8**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	0

Cuadro 3.11: Tabla de verdad para CA $R166 \circ R45$

Los siguientes gráficos muestran todos los vecindarios existentes junto a su correspondiente evolución:

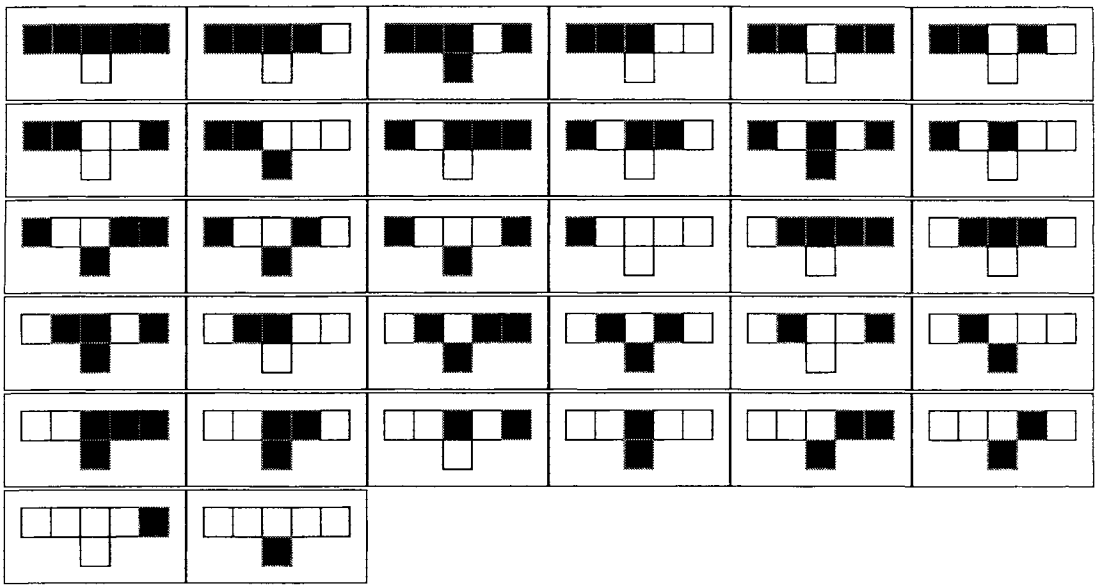


Figura 3.13: Regla $R166 \circ R45$

La figura 3.13 permite calcular el parámetro de Langton :

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.32}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

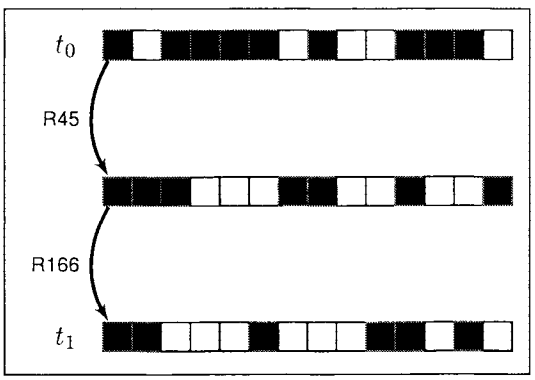


Figura 3.14: Evolución CA $R166 \circ R45$

Esta regla de evolución es la composición de las reglas 45 y 166 (en ese orden), cuyas expresiones booleanas son:

$$\boldsymbol{R45} = c_{i-1} \veebar (c_i \vee \overline{c_{i+1}}) \tag{3.33}$$

$$\boldsymbol{R166} = (c_{i-1} \wedge c_i) \veebar c_i \veebar c_{i+1} \tag{3.34}$$

La composición de ambas reglas ($R105 \circ R45$) es equivalente produce

la expresión booleana siguiente:

$$[c_{i-2} \veebar (c_{i-1} \vee \neg c_i)] \wedge [c_{i-1} \veebar (c_i \vee \overline{c_{i+1}})] \veebar [c_{i-1} \veebar (c_i \vee \overline{c_{i+1}})] \veebar [c_i \veebar (c_{i+1} \vee \overline{c_{i+2}})] \tag{3.35}$$

Utilizando diagramas de Karnaugh se puede obtener la SOP mínima:

$$(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}) + (\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1}) + (\overline{c_{i-1}}\ \overline{c_i}\ c_{i+1}) + (c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2}) \tag{3.36}$$

Y empleando compuertas NAND, se puede expresar el circuito equivalente (tal como se muestra en la figura 3.15):

$$\overline{\overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1})}\ \overline{(\overline{c_{i-2}}\ \overline{c_i}\ c_{i+1})}\ \overline{(\overline{c_{i-1}}\ \overline{c_i}\ c_{i+1})}\ \overline{(c_{i-1}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}\ \overline{(c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2})}} \tag{3.37}$$

Las expresiones algebraicas de las reglas 45 y 166 son:

$$\boldsymbol{R45} = (1 + c_{i-1} + c_{i+1} + c_i c_{i+1}) MOD2 \tag{3.38}$$

$$\boldsymbol{R166} = (c_i + c_{i-1} c_i + c_{i+1}) MOD2 \tag{3.39}$$

Y la expresión algebraica resultante de la composición es:

$$[(1 + c_{i-1} + c_{i+1} + c_i c_{i+1}) MOD2 + [(1 + c_{i-2} + c_i + c_{i-1} c_i) MOD2 * (1 + c_{i-1} + c_{i+1} + c_i c_{i+1}) MOD2] + (1 + c_i + c_{i+2} + c_{i+1} c_{i+2}) MOD2] MOD2 \tag{3.40}$$

la que puede reducirse al aplicar las propiedades de la aritmética modular:

$$(1 + c_{i-2} + c_{i+2} + c_{i-2} c_{i-1} + c_{i-2} c_{i+1} + c_{i-1} c_i + c_{i+1} c_{i+2} + c_{i-2} c_i c_{i+1}) MOD2 \tag{3.41}$$

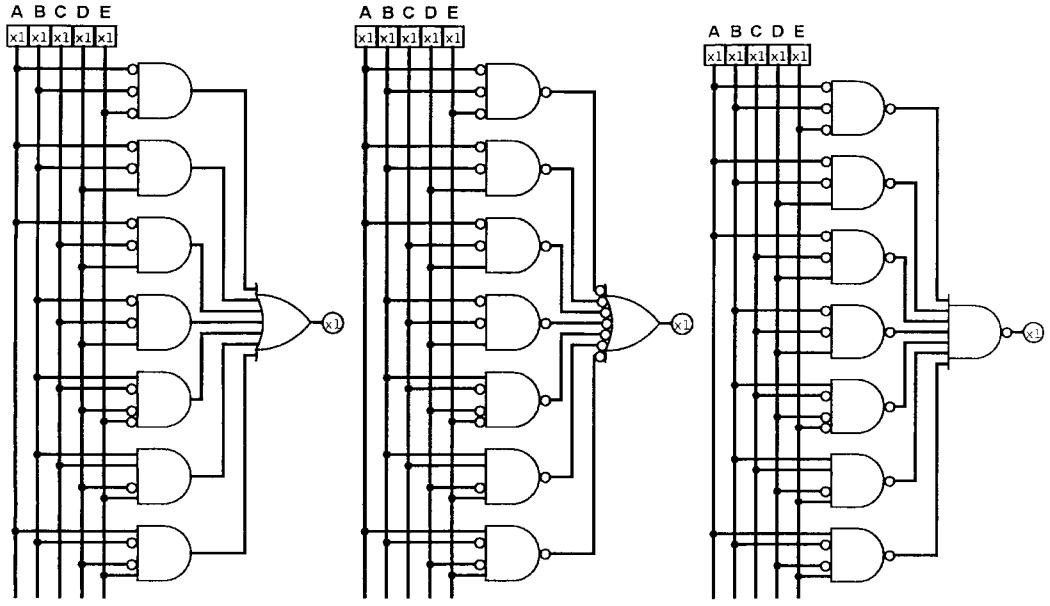


Figura 3.15: Circuitos lógicos equivalentes de la regla $R166 \circ R45$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para una condición inicial aleatoria:

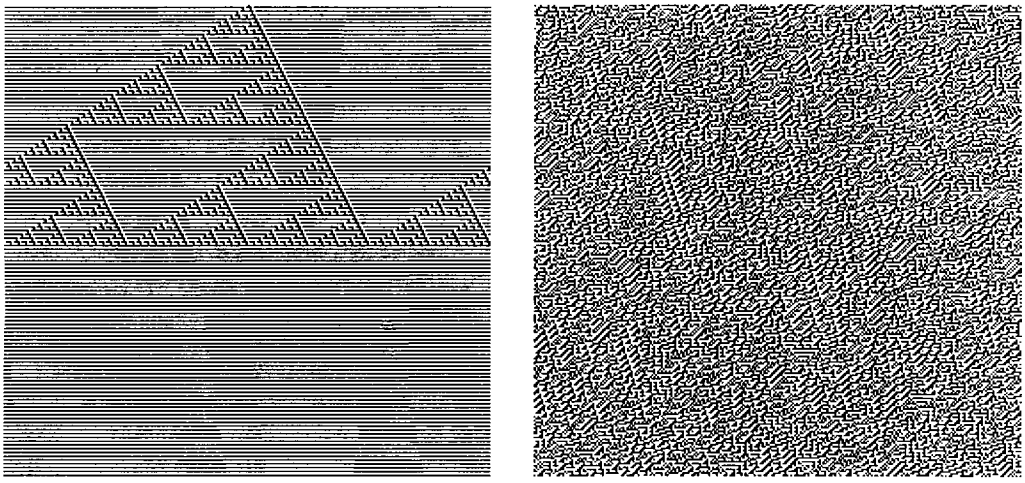


Figura 3.16: Diagramas espacio-temporales para la Regla $166 \circ 45$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El estudio de entropía realizado para esta regla (y con una estructura

de 15 celdas) se muestra en la figura D.6 del anexo D, donde se observa que el autómata mantiene el estado de desorganización, de forma similar a la regla 45.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.6:

# Celdas	μ	σ
6	5,485	3,107
7	31,358	14,3852
8	11,53	7,93095
9	99,447	38,7043
10	443,408	275,284
11	225,4	259,001
12	112,112	173,582
13	3363,53	1829,27
14	639,399	193,736
15	5627,91	2072,18

# Celdas	μ	σ
16	369,169	253,343
17	40755,7	17396,7
18	3854,84	2340,42
19	46384,2	10068,7
20	3274,51	2046,64
21	48655,9	6515,16
22	38977,9	10347,7
23	50000,2	1,04105
24	28263	17447,8
25	49754,6	2579,28

Cuadro 3.12: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R166 \circ R45$ con diferentes cantidades de celdas

3.5. Regla $106 \circ 85$

Esta regla se compone aplicando las reglas de ECA 85 y 106 (en ese orden) por lo cual se denominará $106 \circ 85$. Su valor numérico decimal es 1.448.498.774, y su equivalente en base32hex es **APB5CLG**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	0

Cuadro 3.13: Tabla de verdad para CA $R106 \circ R85$

A continuación se puede observar la evolución de cada uno de los vecindarios existentes:

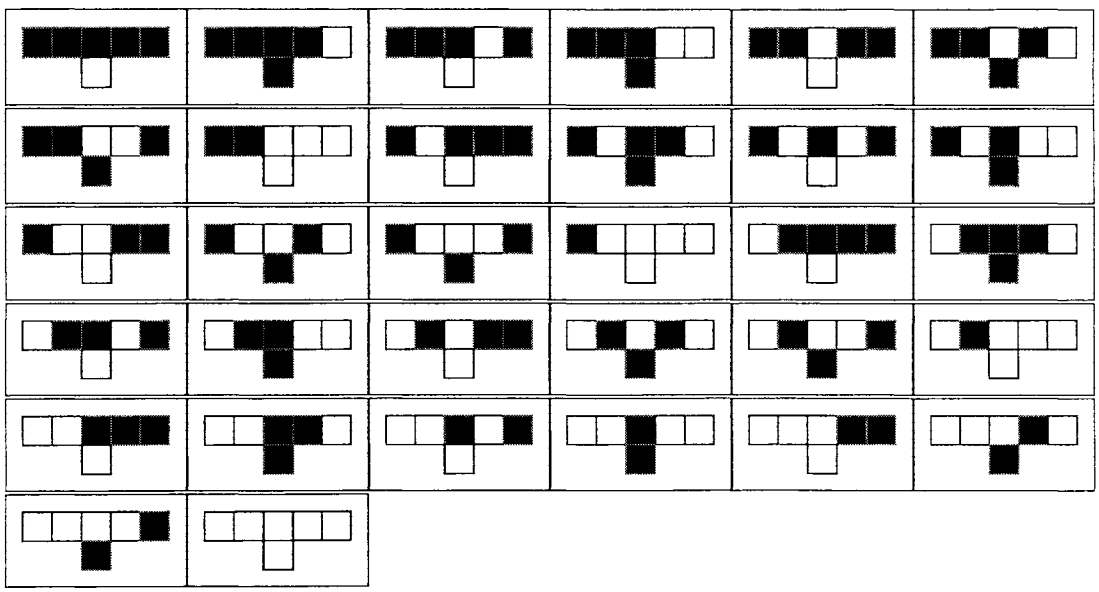


Figura 3.17: Regla $R106 \circ R85$

Observando la figura 3.17, el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.42}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

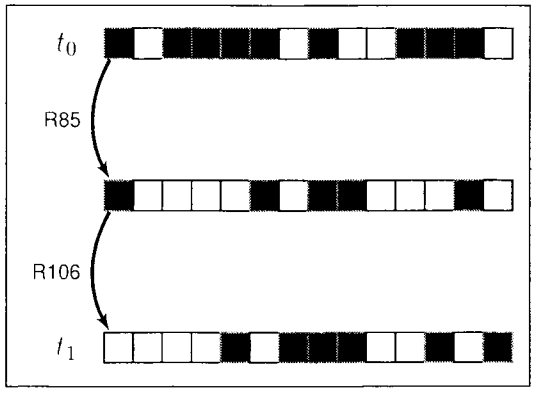


Figura 3.18: Evolución CA $R106 \circ R85$

Esta regla de evolución es la composición de las reglas 85 y 106, respectivamente, cuyas expresiones booleanas son:

$$\boldsymbol{R85} = \overline{c_{i+1}} \tag{3.43}$$

$$\boldsymbol{R106} = (c_{i-1} \wedge c_i) \veebar c_{i+1} \tag{3.44}$$

La composición de ambas reglas ($R106 \circ R85$) es equivalente a la expresión booleana siguiente:

$$(\overline{c_i} \wedge \overline{c_{i+1}}) \veebar \overline{c_{i+2}} \tag{3.45}$$

La anterior expresión posee una expresión AND-OR mínima equivalente:

$$(c_{i+1}\ \overline{c_{i+2}}) + (c_i\ \overline{c_{i+2}}) + (\overline{c_i}\ \overline{c_{i+1}}\ c_{i+2}) \tag{3.46}$$

Que además se puede implementar con compuertas universales NAND:

$$\overline{\overline{(c_{i+1}\ \overline{c_{i+2}})}\ \overline{(c_i\ \overline{c_{i+2}})}\ \overline{(\overline{c_i}\ \overline{c_{i+1}}\ c_{i+2})}} \tag{3.47}$$

Las expresiones algebraicas de las reglas 85 y 106 son:

$$\boldsymbol{R85} = (1 + c_{i+1}) MOD2 \tag{3.48}$$

$$\boldsymbol{R106} = (c_{i-1}c_i + c_{i+1}) MOD2 \tag{3.49}$$

Y la expresión algebraica resultante de la composición es:

$$\{[(1 + c_i) MOD2 \times (1 + c_{i+1}) MOD2] + (1 + c_{i+2}) MOD2\} MOD2 \tag{3.50}$$

la cual puede reducirse al aplicar las propiedades de la aritmética mo-

dular:

$$(c_i + c_{i+1} + c_{i+2} + c_i c_{i+1}) MOD2 \tag{3.51}$$

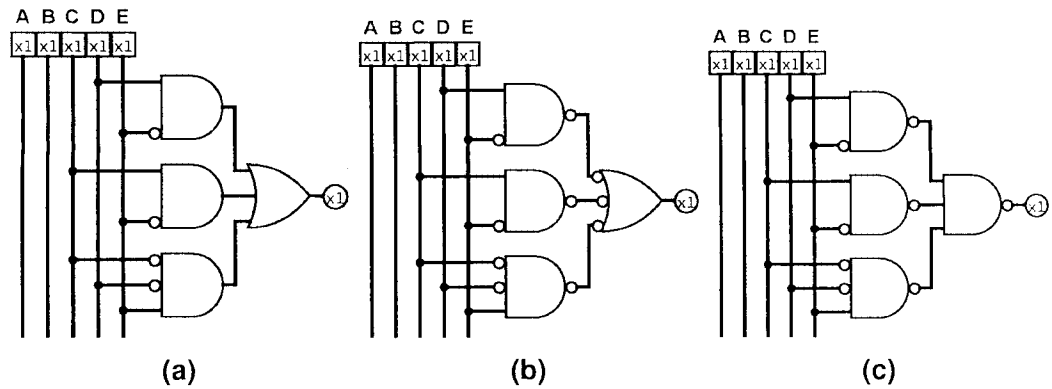
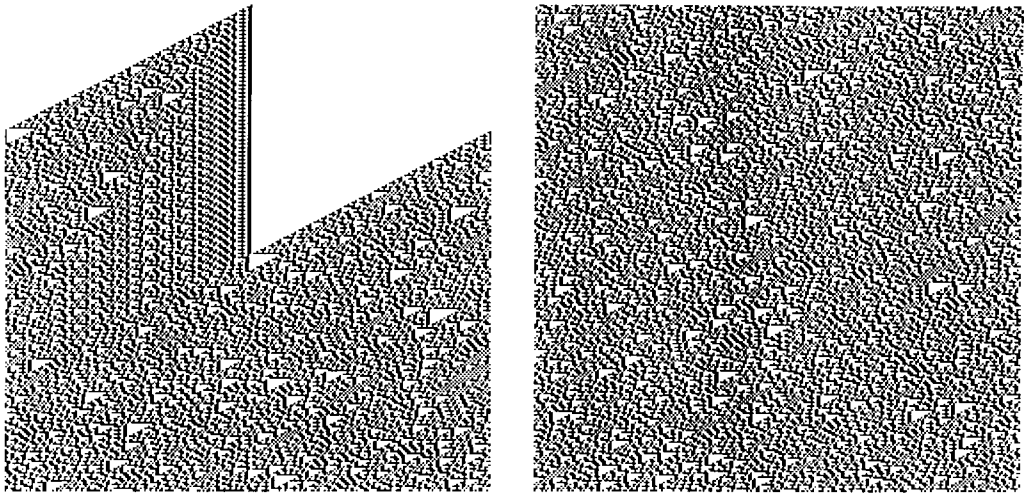


Figura 3.19: Circuitos lógicos equivalentes de la regla $R106 \circ R85$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

Dicha expresión algebraica es similar a la correspondiente a la regla 86 de Wolfram [5]:

$$(c_{i-1} + c_i + c_{i+1} + c_{i-1} c_i) MOD2 \tag{3.52}$$

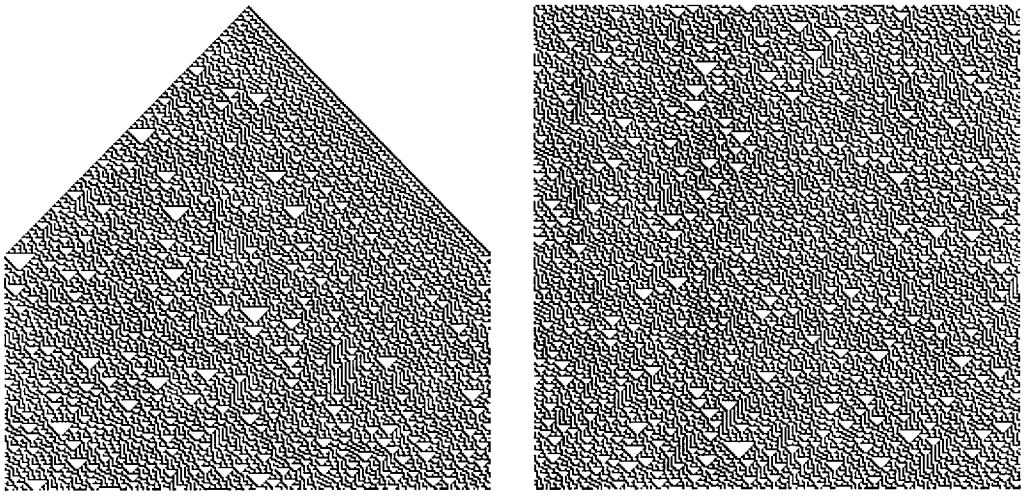
Como se puede apreciar, comparando ambas expresiones, la estructura es la misma pero involucrando celdas distintas lo cual produce un efecto de "inclinación" en la evolución. A continuación se muestran los diagramas espacio-temporales tanto de la regla $106 \circ 85$ y luego los de la regla 86:



(a) (b)

Figura 3.20: Diagramas espacio-temporales para la Regla $106 \circ 85$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

Los diagramas espacio-temporales de la regla 86:



(a) (b)

Figura 3.21: Diagramas espacio-temporales para la Regla 86. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El ECA R86, es la versión conjugada (las celdas en 0 son puestas a 1 y viceversa) y reflejada (reflexión respecto de la celda central) del ECA R30. Por ello en la tabla 3.3, λ_{max} de la R30 y λ_{max} de $R106 \circ R85$ son similares.

El estudio de entropía realizado para esta regla muestra un comportamiento similar al de la regla 30, como de esperarse. Comparar las figuras D.7 y D.1 del anexo D.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente

tabla y su respectivo gráfico se encuentra en la figura B.7:

# Celdas	μ	σ
6	5,366	2,5196
7	15,313	10,0039
8	8,981	5,25721
9	143,412	60,5387
10	44,327	17,485
11	70,387	69,0859
12	237,082	75,1671
13	445,984	259,513
14	1251,15	513,451
15	572,899	181,416

# Celdas	μ	σ
16	3544,29	2598,04
17	10400,7	2212,35
18	779,473	319,352
19	1695,02	1413,18
20	8469,25	5557,83
21	3281,67	1235,98
22	5562,76	1815,57
23	40896,9	1410,68
24	46844,8	11975,4
25	49903,4	1766,24

Cuadro 3.14: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R106 \circ R85$ con diferentes cantidades de celdas

3.6. Regla $154 \circ 105$

Esta regla se compone aplicando las reglas de ECA 105 y 154 (en ese orden) por lo cual se denominará $154 \circ 105$. Su valor numérico decimal es 1.784.260.965, y su equivalente en base32hex es **D9CQIP8**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	0
0	1	1	1	1	1

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	0

Cuadro 3.15: Tabla de verdad para CA $R154 \circ R105$

Se muestra a continuación la evolución para cada uno de los posibles vecindarios:

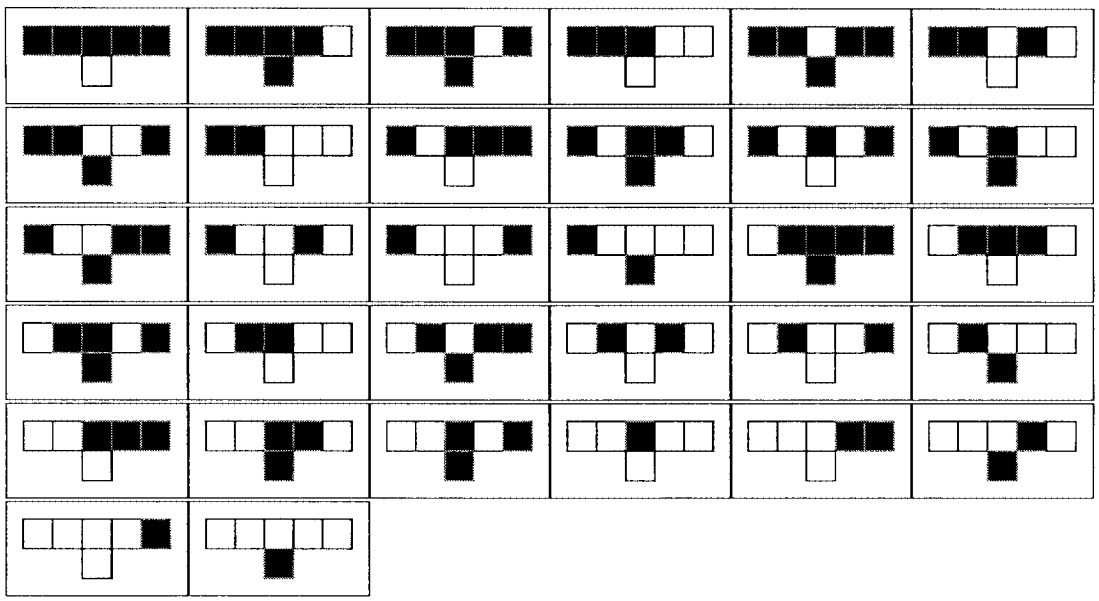


Figura 3.22: Regla $R154 \circ R105$

Como se puede apreciar en la figura 3.22, el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.53}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

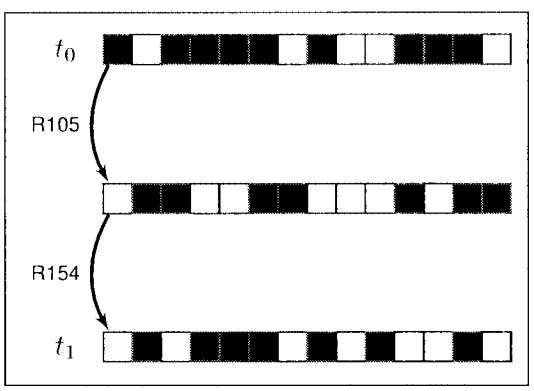


Figura 3.23: Evolución CA $R154 \circ R105$

Esta regla de evolución es la composición de las reglas 105 y 154, respectivamente, cuyas expresiones booleanas son:

$$\boldsymbol{R105} = c_{i-1} \veebar c_i \veebar \overline{c_{i+1}} \tag{3.54}$$

$$\boldsymbol{R154} = c_{i-1} \veebar (c_{i+1} \wedge c_i) \veebar c_{i+1} \tag{3.55}$$

La composición de ambas reglas ($R154 \circ R105$) es equivalente a la expresión booleana siguiente:

$$\begin{gathered}(c_{i-2} \veebar c_{i-1} \veebar \overline{c_i}) \veebar [(c_{i-2} \veebar c_{i-1} \veebar \overline{c_i}) \wedge (c_{i-1} \veebar c_i \veebar \overline{c_{i+1}})] \veebar \\ (c_i \veebar c_{i+1} \veebar \overline{c_{i+2}})\end{gathered} \tag{3.56}$$

Luego de minimizar la expresión anterior en una SOP, esta queda de expresada de la siguiente forma:

$$\begin{gathered}(\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + \\ (\overline{c_{i-2}}\ c_{i-1}\ c_{i+1}\ c_{i+2}) + (c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + \\ (c_{i-2}\ \overline{c_i}\ c_{i+1}\ c_{i+2}) + (c_{i-2}\ c_i\ c_{i+1}\ \overline{c_{i+2}}) + (c_{i-2}\ c_{i-1}\overline{c_i}\ c_{i+2})\end{gathered} \tag{3.57}$$

La expresión booleana (3.57), tiene una representación mínima equivalente con compuertas universales NAND:

$$\begin{gathered}\overline{\overline{(\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ c_{i+2})}} \\ \overline{\overline{(\overline{c_{i-2}}\ c_{i-1}\ c_{i+1}\ c_{i+2})}\ \overline{(c_{i-1}\ c_i\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}} \\ \overline{\overline{(c_{i-2}\ \overline{c_i}\ c_{i+1}\ c_{i+2})}\ \overline{(c_{i-2}\ c_i\ c_{i+1}\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ c_{i-1}\overline{c_i}\ c_{i+2})}}\end{gathered} \tag{3.58}$$

Las expresiones algebraicas de las reglas 105 y 154 son:

$$\boldsymbol{R105} = (1 + c_{i-1} + c_i + c_{i+1}) MOD2 \tag{3.59}$$

$$\boldsymbol{R154} = (c_{i-1} + c_{i-1}c_i + c_{i+1}) MOD2 \tag{3.60}$$

Y la expresión algebraica resultante de la composición es:

$$\begin{gathered}\{(1 + c_{i-2} + c_{i-1} + c_i) MOD2 + [(1 + c_{i-2} + c_{i-1} + c_i) MOD2 \times \\ (1 + c_{i-1} + c_i + c_{i+1}) MOD2] + (1 + c_i + c_{i+1} + c_{i+2}) MOD2\} MOD2\end{gathered} \tag{3.61}$$

la cual puede reducirse al aplicar las propiedades de la aritmética modular:

$$(1 + c_i + c_{i+2} + c_{i-2}c_{i-1} + c_{i-2}c_i + c_{i-2}c_{i+1} + c_{i-1}c_{i+1} + c_ic_{i+1}) MOD2 \tag{3.62}$$

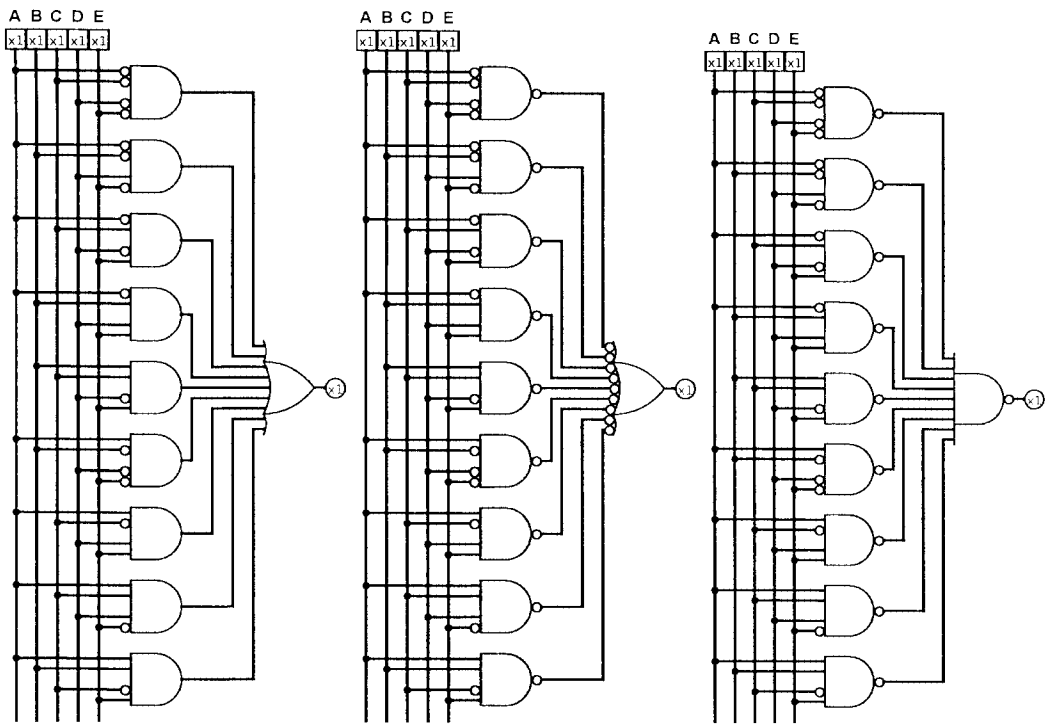


Figura 3.24: Circuitos lógicos equivalentes de la regla $R154 \circ R105$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para una condición inicial aleatoria:

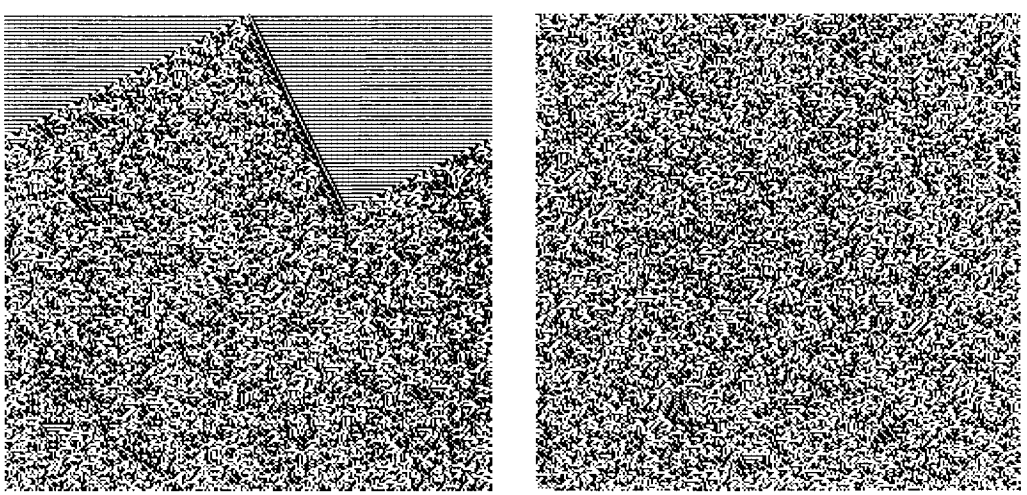


Figura 3.25: Diagramas espacio-temporales para la Regla $154 \circ 105$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

Analizando la entropía para esta regla sucede nuevamente, de igual forma que para las anteriores reglas que incluyen a la 105, que el autómata rápidamente adquiere mayor organización interna, con la particularidad que esta regla es la que mayor organización adquiere (de las anteriormente mencionadas).

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.8:

# Celdas	μ	σ
6	2,766	0,793646
7	124,016	15,5667
8	73,698	55,3897
9	7,908	3,45101
10	151,395	56,2309
11	399,295	270,183
12	8,396	4,95598
13	3962,37	583,774
14	2643,8	303,366
15	84,689	27,1711

# Celdas	μ	σ
16	1852,45	701,841
17	39162,9	14148,2
18	341,082	85,3809
19	49418,6	5078,62
20	18904,1	18617,1
21	1707,21	1177,37
22	49852,5	2694,66
23	49905,8	1748,96
24	1264,41	453,87
25	49851,3	2240,76

Cuadro 3.16: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R154 \circ R105$ con diferentes cantidades de celdas

3.7. Regla $166 \circ 106$

Esta regla se compone aplicando las reglas de ECA 106 y 166 (en ese orden) por lo cual se denominará $166 \circ 106$. Su valor numérico decimal es 1.516.660.326, y su equivalente en base32hex es **B9J6CPG**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	0
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	0

Cuadro 3.17: Tabla de verdad para CA $R166 \circ R106$

Los siguientes gráficos muestran todos los vecindarios existentes junto a su correspondiente evolución:

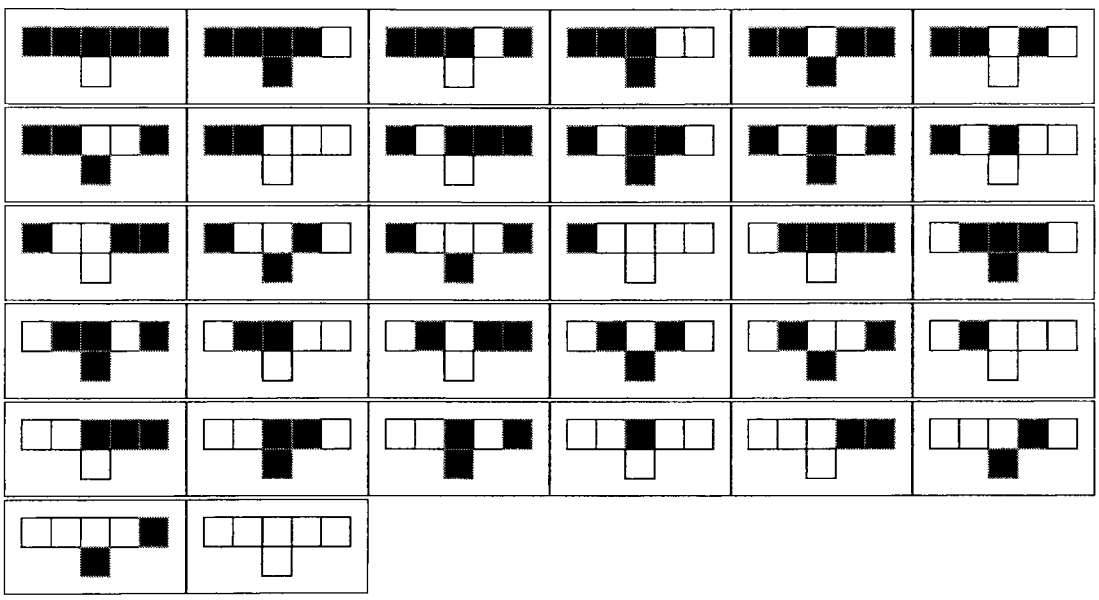


Figura 3.26: Regla $R166 \circ R106$

En la figura precedente, se pueden apreciar las 16 celdas inactivas y el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.63}$$

El gráfico que muestra la evolución de la regla en un sólo instante de

tiempo es el siguiente:

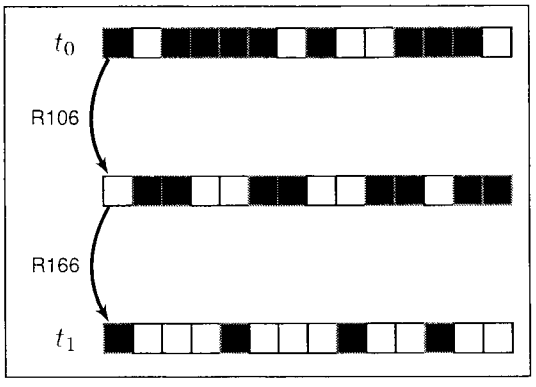


Figura 3.27: Evolución CA $R166 \circ R106$

Esta regla de evolución es la composición de las reglas 106 y 166, respectivamente, cuyas expresiones booleanas son:

$$\boldsymbol{R106} = (c_{i-1} \wedge c_i) \veebar c_{i+1} \tag{3.64}$$

$$\boldsymbol{R166} = (c_{i-1} \wedge c_i) \veebar c_i \veebar c_{i+1} \tag{3.65}$$

La composición de ambas reglas ($R166 \circ R106$) es equivalente a la expresión booleana siguiente:

$$\begin{aligned}\{[(c_{i-2} \wedge c_{i-1}) \veebar c_i] \wedge [(c_{i-1} \wedge c_i) \veebar c_{i+1}]\} \veebar [(c_{i-1} \wedge c_i) \veebar c_{i+1}] \veebar \\ [(c_i \wedge c_{i+1}) \veebar c_{i+2}]\end{aligned} \tag{3.66}$$

La cual se puede expresar con un circuito equivalente AND-OR de dos niveles:

$$\begin{aligned}(\overline{c_{i-2}}\ \overline{c_{i+1}}\ c_{i+2}) + (\overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2}) + (\overline{c_{i-2}}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}}) + \\ (c_{i-2}\ c_{i-1}\ \overline{c_i}\ c_{i+2}) + (c_{i-2}\ c_{i-1}\ c_i\ \overline{c_{i+2}})\end{aligned} \tag{3.67}$$

Y la anterior expresión booleana, posee una representación mínima equivalente con compuertas universales NAND (ver figura 3.28):

$$\begin{aligned}\overline{\overline{(\overline{c_{i-2}}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(\overline{c_{i-1}}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(\overline{c_{i-2}}\ c_{i+1}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}})}} \\ \overline{\overline{(c_{i-2}\ c_{i-1}\ \overline{c_i}\ c_{i+2})}\ \overline{(c_{i-2}\ c_{i-1}\ c_i\ \overline{c_{i+2}})}}\end{aligned} \tag{3.68}$$

Las expresiones algebraicas de las reglas 106 y 166 son:

$$\boldsymbol{R106} = (c_{i-1}c_i + c_{i+1})MOD2 \tag{3.69}$$

$$\boldsymbol{R166} = (c_i + c_{i-1}c_i + c_{i+1})MOD2 \tag{3.70}$$

Y la expresión algebraica resultante de la composición es:

$$\{(c_{i-1}c_i + c_{i+1})MOD2 + [(c_{i-2}c_{i-1} + c_i)MOD2 \times (c_{i-1}c_i + c_{i+1})MOD2] + (c_ic_{i+1} + c_{i+2})MOD2\}MOD2 \tag{3.71}$$

la cual puede reducirse al aplicar las propiedades de la aritmética modular:

$$(c_{i+1} + c_{i+2} + c_{i-2}c_{i-1}c_i + c_{i-2}c_{i-1}c_i)MOD2 \tag{3.72}$$

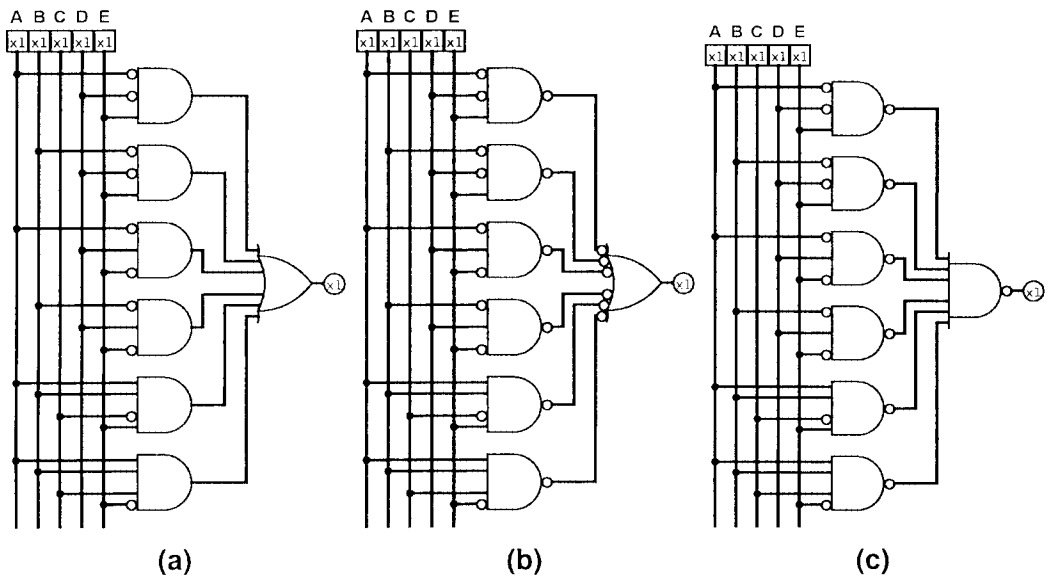
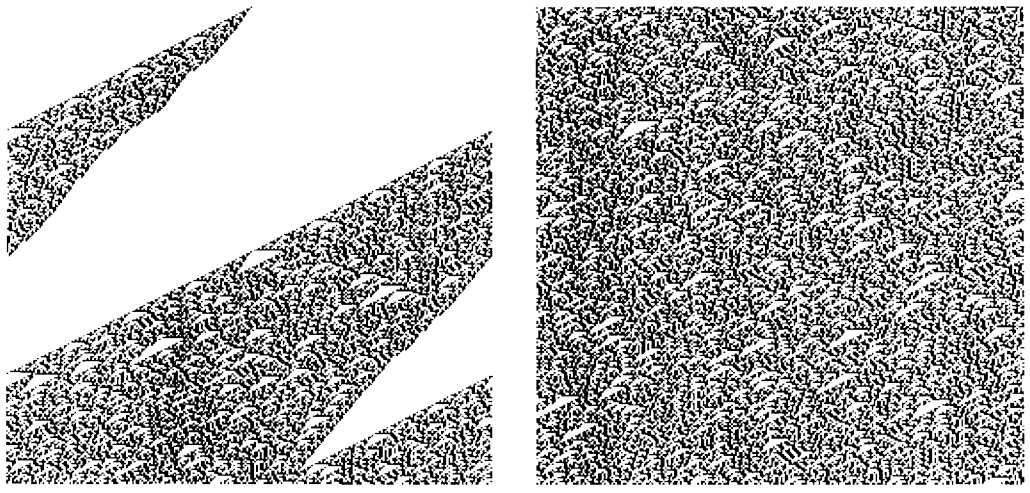


Figura 3.28: Circuitos lógicos equivalentes de la regla $R166 \circ R106$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para una condición inicial aleatoria:



(a) (b)

Figura 3.29: Diagramas espacio-temporales para la Regla $166 \circ 106$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

Esta regla adquiere, desde un estado de desorden, gradualmente orden, exhibiendo un comportamiento similar al de la regla 30, pero alcanzado al final mayor orden. Se puede observar en la figura D.9.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.9:

# Celdas	μ	σ
6	5,807	2,99177
7	4,007	2,24767
8	43,074	22,7973
9	144,996	17,6463
10	111,803	38,8783
11	95,388	13,4865
12	35,471	14,178
13	1052,64	356,897
14	78,971	28,5057
15	334,703	181,139

# Celdas	μ	σ
16	790,519	405,415
17	8683,2	319,792
18	2484,69	1598,53
19	5803,44	1715,77
20	3589,24	1982,37
21	30374,8	8735,88
22	14123,9	7299,13
23	47007,6	11006,1
24	13090,8	7160,1
25	35331,9	17135,3

Cuadro 3.18: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R166 \circ R106$ con diferentes cantidades de celdas

3.8. Regla $154 \circ 150$

Esta regla se compone aplicando las reglas de ECA 150 y 154 (en ese orden) por lo cual se denominará $\mathbf{154 \circ 150}$. Su valor numérico decimal es

2.794.822.230, y su equivalente en base32hex es **KQAPKLG**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	1	0	1
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	1
0	1	0	1	0	0
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	1

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	0	0
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1

Cuadro 3.19: Tabla de verdad para CA $R154 \circ R150$

Se exhibe en los siguientes gráficos todos los posibles vecindarios con su respectiva evolución:

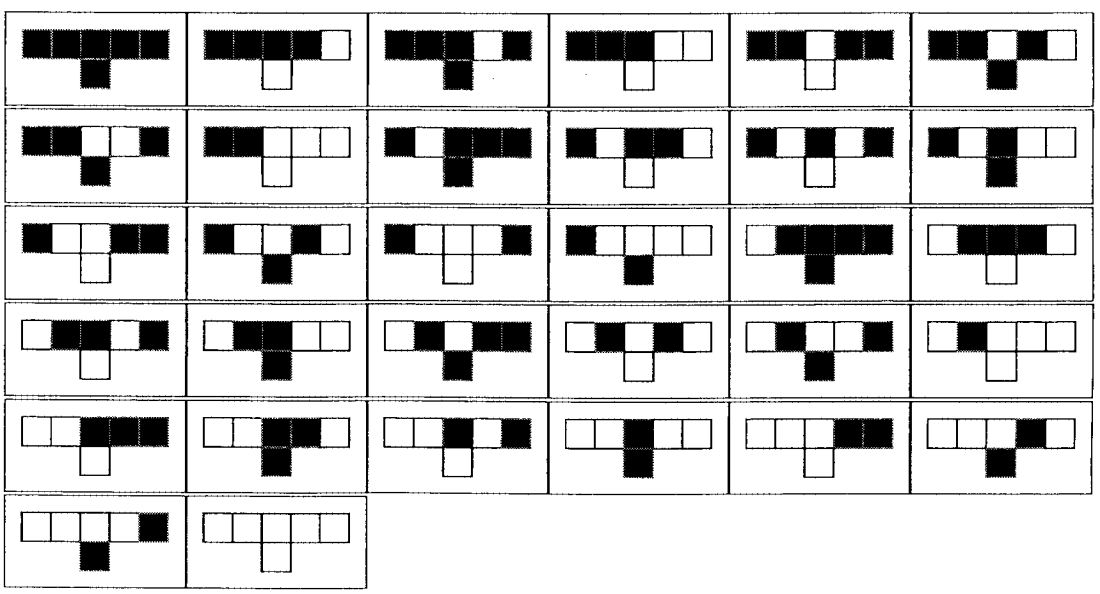


Figura 3.30: Regla $R154 \circ R150$

La figura 3.30 permite calcular el parámetro de Langton :

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0.5 \tag{3.73}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

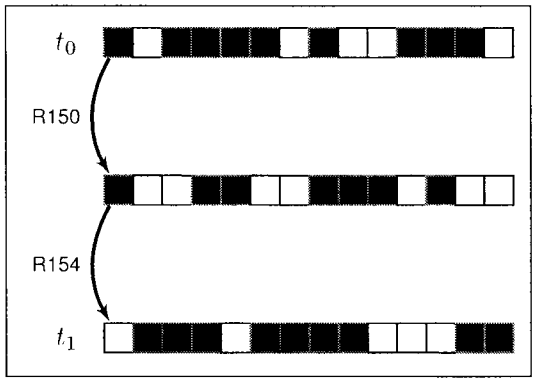


Figura 3.31: Evolución CA $R154 \circ R150$

Esta regla de evolución es la composición de las reglas 150 y 154, respectivamente, cuyas expresiones booleanas son:

$$\boldsymbol{R150} = c_{i-1} \veebar c_i \veebar c_{i+1} \tag{3.74}$$

$$\boldsymbol{R154} = c_{i-1} \veebar (c_{i+1} \wedge c_i) \veebar c_{i+1} \tag{3.75}$$

La composición de ambas reglas ($R154 \circ R150$) es equivalente a la expresión booleana siguiente:

$$\begin{aligned}(c_{i-2} \veebar c_{i-1} \veebar c_i) \veebar [(c_{i-2} \veebar c_{i-1} \veebar c_i) \wedge (c_{i-1} \veebar c_i \veebar c_{i+1})] \veebar \\ (c_i \veebar c_{i+1} \veebar c_{i+2})\end{aligned} \tag{3.76}$$

Utilizando la tabla de verdad mostrada en el cuadro 3.19 y diagramas de Karnaugh se halla la siguiente expresión SOP mínima:

$$\begin{aligned}&(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + (c_{i-2}\ c_{i-1}\ \overline{c_{i+1}}\ c_{i+2}) + (c_{i-2}\ c_i\ c_{i+1}\ c_{i+2}) + \\ &(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ c_{i-1}\ c_{i+1}\ c_{i+2}) + (\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}}) + \\ &(c_{i-2}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}}) + (\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2})\end{aligned} \tag{3.77}$$

La anterior expresión booleana, tiene una representación mínima equi-

valente con compuertas universales NAND (ver figura 3.32):

$$\begin{array}{c}\overline{\overline{(c_{i-2}\ \overline{c_{i-1}}\ \overline{c_{i+1}}\ \overline{c_{i+2}})}\ \overline{(c_{i-2}\ c_{i-1}\ \overline{c_{i+1}}\ c_{i+2})}\ \overline{(c_{i-2}\ c_i\ c_{i+1}\ c_{i+2})}} \\ \overline{\overline{(\overline{c_{i-2}}\ \overline{c_{i-1}}\ c_{i+1}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ c_{i-1}\ c_{i+1}\ c_{i+2})}\ \overline{(\overline{c_{i-2}}\ c_i\ \overline{c_{i+1}}\ \overline{c_{i+2}})}} \\ \overline{\overline{(c_{i-2}\ \overline{c_i}\ c_{i+1}\ \overline{c_{i+2}})}\ \overline{(\overline{c_{i-2}}\ \overline{c_i}\ \overline{c_{i+1}}\ c_{i+2})}}\end{array} \tag{3.78}$$

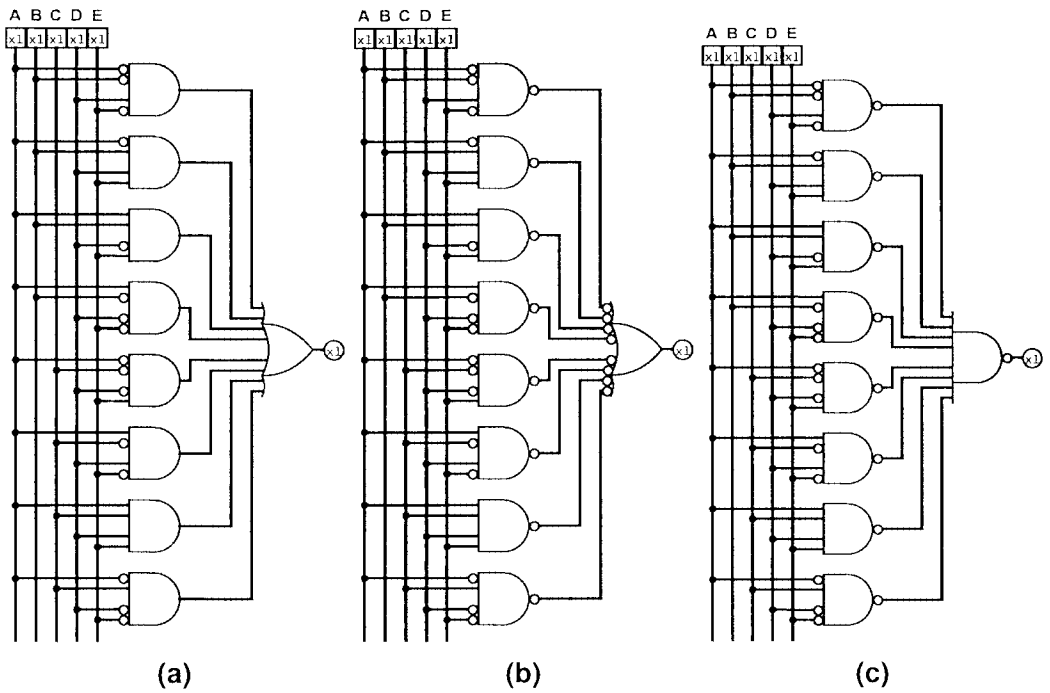


Figura 3.32: Circuitos lógicos equivalentes de la regla $R154 \circ R150$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

Las expresiones algebraicas de las reglas 150 y 154 son:

$$\boldsymbol{R150} = (c_{i-1} + c_i + c_{i+1})MOD2 \tag{3.79}$$

$$\boldsymbol{R154} = (c_{i-1} + c_{i-1}c_i + c_{i+1})MOD2 \tag{3.80}$$

Y la expresión algebraica resultante de la composición es:

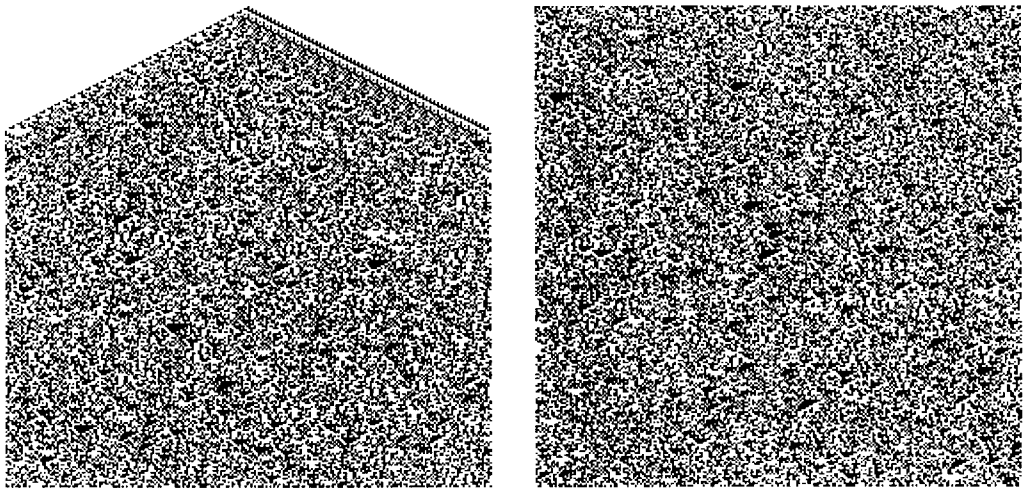
$$\begin{array}{l}\{(c_{i-2} + c_{i-1} + c_i)MOD2 + [(c_{i-2} + c_{i-1} + c_i)MOD2 \times \\ (c_{i-1} + c_i + c_{i+1})MOD2] + (c_i + c_{i+1} + c_{i+2})MOD2\}MOD2\end{array} \tag{3.81}$$

esta expresión puede reducirse al aplicar las propiedades de la aritméti-

ca modular:

$$(c_{i-2} + c_i + c_{i+1} + c_{i+2} + c_{i-2}c_{i-1} + c_{i-2}c_i + c_{i-2}c_{i+1} + c_{i-1}c_{i+1} + c_ic_{i+1})MOD2 \tag{3.82}$$

La caracterización de esta regla en diagramas espacio-temporales es la siguiente, tanto para la condición inicial más sencilla, como también para una condición inicial aleatoria:



(a) (b)

Figura 3.33: Diagramas espacio-temporales para la Regla $154 \circ 150$. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

En el caso del análisis de entropía de esta regla, se aprecia que el autómata rápidamente (en menos de 50 *time steps*) adquiere mayor organización. Esto se muestra en la figura D.10 del anexo D.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.10:

# Celdas	μ	σ
6	1,156	0,772179
7	83,211	29,9774
8	34,802	37,2792
9	31,008	19,8026
10	87,539	21,7607
11	1403,07	634,397
12	9,172	4,61321
13	6722,41	1943,71
14	222,62	69,8755
15	139,517	77,1776

# Celdas	μ	σ
16	672,101	412,359
17	37250,9	17895,8
18	701,668	287,286
19	45516,3	10653,3
20	16957,6	10579,3
21	1502,29	197,148
22	43870,4	14532
23	49856,4	1622,38
24	524,987	333,548
25	50000,2	1,01298

Cuadro 3.20: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R154 \circ R150$ con diferentes cantidades de celdas

3.9. Regla $154 \circ 15$

Esta regla se compone aplicando las reglas de ECA 15 y 154 (en ese orden) por lo cual se denominará $154 \circ 15$. Su valor numérico decimal es 252.702.735, y su equivalente en base32hex es **1S7V03O**.

La tabla de verdad que representa a esta regla puede apreciarse a continuación:

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1

c_{i-2}^t	c_{i-1}^t	c_i^t	c_{i+1}^t	c_{i+2}^t	c_i^{t+1}
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	0	1	0
1	1	1	1	0	0
1	1	1	1	1	0

Cuadro 3.21: Tabla de verdad para CA $R154 \circ R15$

Las imágenes presentadas a continuación exhiben la evolución de los 32 posibles vecindarios:

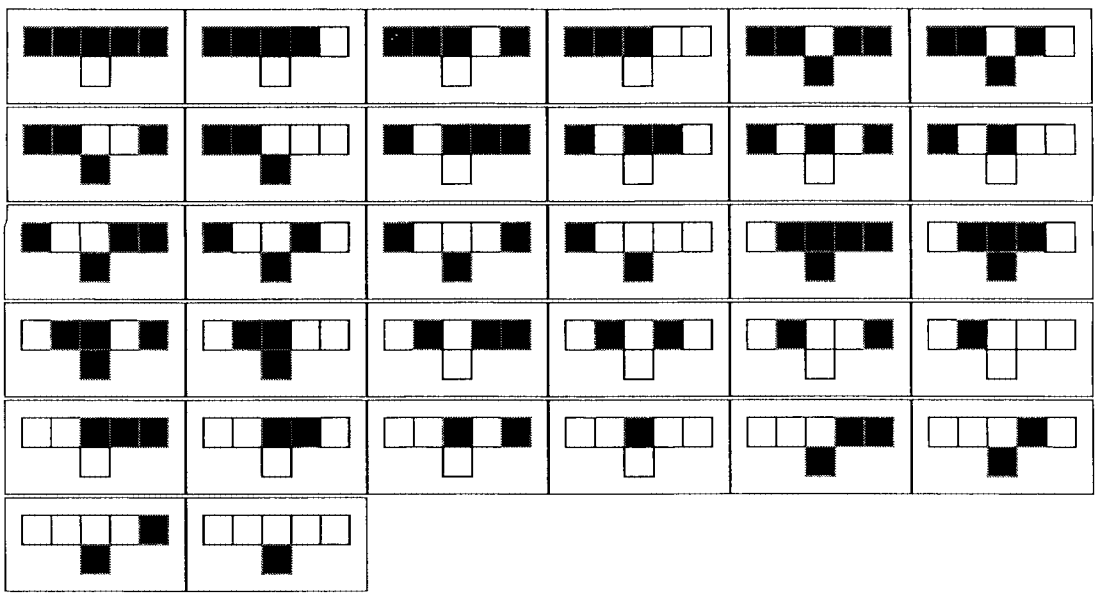


Figura 3.34: Regla $R154 \circ R15$

En la figura 3.34 se pueden apreciar las 16 celdas inactivas y el parámetro de Langton se calcula como:

$$\lambda = \frac{k^p - n_q}{k^p} = \frac{2^5 - 16}{2^5} = 0{,}5 \tag{3.83}$$

El gráfico que muestra la evolución de la regla en un sólo instante de tiempo es el siguiente:

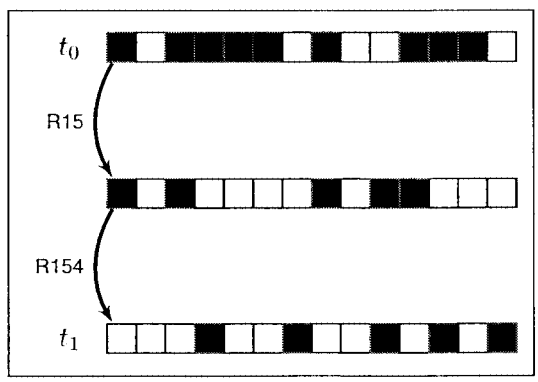


Figura 3.35: Evolución CA $R154 \circ R15$

Esta regla de evolución es la composición de las reglas 15 y 154, respectivamente, cuyas expresiones booleanas son:

$$\boldsymbol{R15} = \overline{c_{i-1}} \tag{3.84}$$

$$\boldsymbol{R154} = c_{i-1} \veebar (c_{i+1} \wedge c_i) \veebar c_{i+1} \tag{3.85}$$

La composición de ambas reglas ($R154 \circ R15$) es equivalente a la ex-

presión booleana siguiente:

$$\overline{c_{i-2}} \veebar (\overline{c_{i-2}} \wedge \overline{c_{i-1}}) \veebar \overline{c_i} \tag{3.86}$$

A partir de la tabla de verdad mostrada en el cuadro 3.21 y utilizando los diagramas de Karnaugh, como se ha visto en la sección 3.1, se construye la expresión de dos niveles AND-OR mínima:

$$(\overline{c_{i-1}}\ \overline{c_i}) + (c_{i-2}\ \overline{c_i}) + (\overline{c_{i-2}}\ c_{i-1}\ c_i) \tag{3.87}$$

Y esa expresión booleana (3.87) tiene una representación mínima equivalente utilizando compuertas universales NAND, la cual se puede apreciar a continuación. El detalle de la conversión, en diagramas de circuitos lógicos, se muestra en la figura 3.36.

$$\overline{\overline{(\overline{c_{i-1}}\ \overline{c_i})}\ \overline{(c_{i-2}\ \overline{c_i})}\ \overline{(\overline{c_{i-2}}\ c_{i-1}\ c_i)}} \tag{3.88}$$

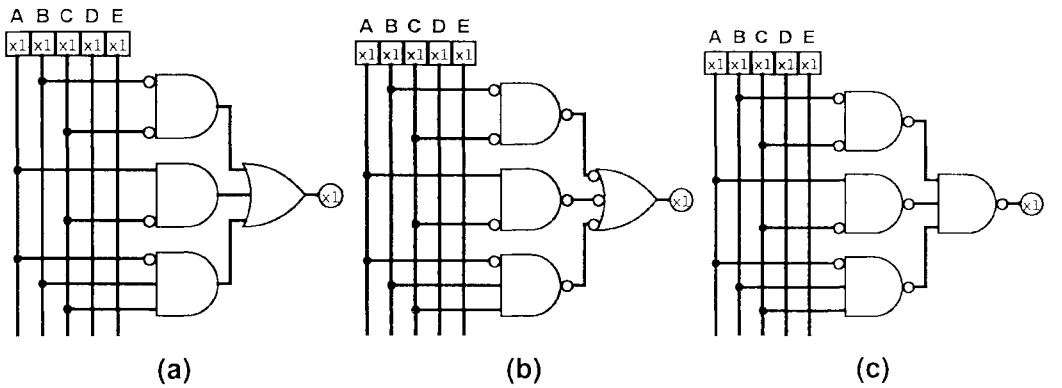


Figura 3.36: Circuitos lógicos equivalentes de la regla $R154 \circ R15$. (a) circuito AND-OR. En (b) se cambian las compuertas AND por compuertas NAND y para mantener la equivalencia se agregan burbujas de inversión en las entradas de la compuerta OR. (c) circuito NAND-NAND obtenido luego de aplicar la ley de DeMorgan. Las variables $c_{i-2}, c_{i-1}, c_i, c_{i+1}$ y c_{i+2} se reemplazan por A, B, C, D y E respectivamente

Las expresiones algebraicas de las reglas 15 y 154 son:

$$\boldsymbol{R15} = (1 + c_{i-1}) MOD2 \tag{3.89}$$

$$\boldsymbol{R154} = (c_{i-1} + c_{i-1}c_i + c_{i+1}) MOD2 \tag{3.90}$$

Y la expresión algebraica resultante de la composición es:

$$\{(1+c_{i-2})MOD2+[1+(c_{i-2})MOD2\times(1+c_{i-1})MOD2]+ \\ (1+c_i)MOD2\}MOD2 \tag{3.91}$$

la cual puede reducirse al aplicar las propiedades de la aritmética modular:

$$(1+c_{i-1}+c_i+c_{i-2}c_{i-1})MOD2 \tag{3.92}$$

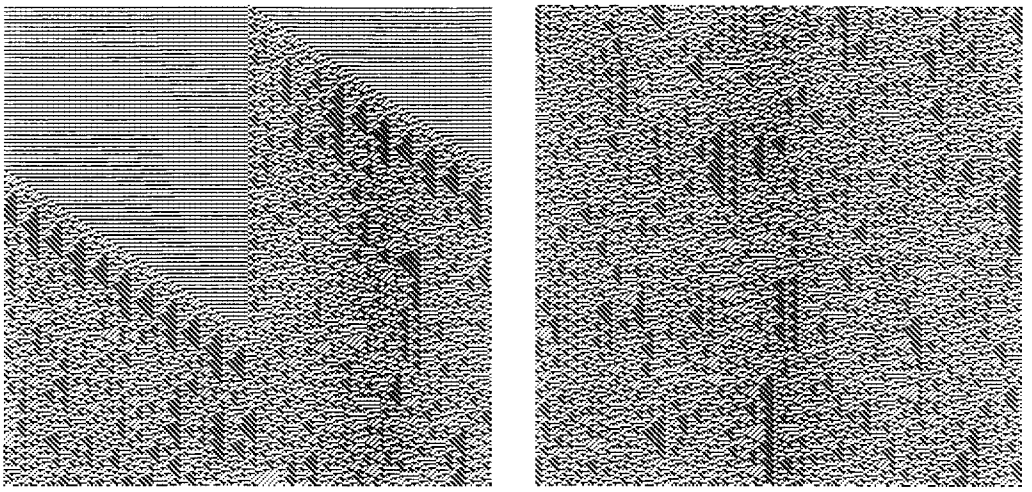
Dicha expresión es similar a la regla 89 (ver [5]), pero empleando otras celdas:

$$(1+c_i+c_{i+1}+c_{i-1}c_i)MOD2 \tag{3.93}$$

Producto de ese "corrimiento" en las celdas empleadas, cuando se compara la evolución de la regla $154 \circ 15$ con la de R89 se observará un corrimiento o inclinación. Lo cual se puede observar en los diagramas espacio-temporales de las figuras 3.37 y 3.38.

Sin embargo el lector habrá notado en el inicio de este capítulo, cuando se presentaron los exponentes de Lyapunov característicos, que λ_{max} de R45 y λ_{max} de $R154 \circ R15$ son iguales. Esto se debe a que la R89 es la versión conjugada y reflejada de la R45 (ver [4, pp. 517]).

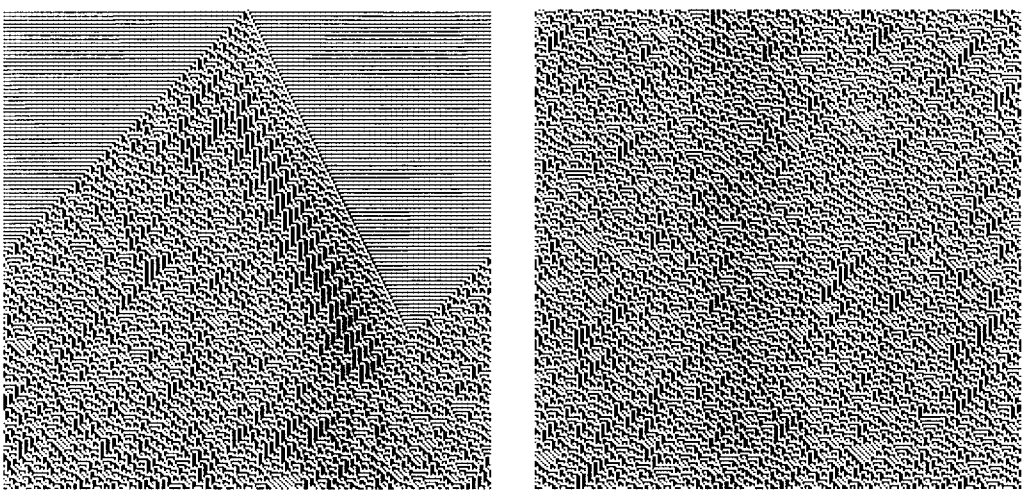
Este hecho resalta diferentes cosas: en primer lugar se destaca la utilidad de λ_{max} en la caracterización de las reglas de evolución y reglas similares poseen igual λ_{max}. En segundo término se observa la mayor utilidad de λ_{max} en comparación con λ_L y λ_R.



(a) (b)

Figura 3.37: Diagramas espacio-temporales para la Regla 154 o 15. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

Los diagramas espacio-temporales de la regla 89:



(a) (b)

Figura 3.38: Diagramas espacio-temporales para la Regla 89. En (a) la condición inicial corresponde a la celda central (la única) distinta de cero. En (b) la condición inicial es aleatoria.

El estudio de entropía realizado para esta regla muestra un comportamiento similar al de la regla 45, como de esperarse. Comparar las figuras D.11 y D.2 del anexo D.

La evaluación por medio del algoritmo de Berlekamp-Massey de la secuencia de bits generada por este autómata finito se presenta en la siguiente tabla y su respectivo gráfico se encuentra en la figura B.11:

# Celdas	μ	σ
6	8,58	3,19591
7	13,457	3,60682
8	11,95	8,81278
9	54,327	6,50172
10	160,928	90,3801
11	901,503	221,27
12	79,339	61,8724
13	4178,25	2368,07
14	1191,86	511,262
15	13514	8800,39

# Celdas	μ	σ
16	1059,1	857,876
17	30410,2	23486,9
18	10114,1	2722,74
19	47619,7	9924,4
20	9924,7	9292,06
21	46502,7	10845,9
22	49159,9	6394,88
23	49349,5	5486,32
24	35237,4	18191,8
25	49650,3	4169,6

Cuadro 3.22: Media y desvío estándar de la complejidad lineal de una secuencia de 100.000 bits, para la regla $R154 \circ R15$ con diferentes cantidades de celdas

Cualquiera que intente generar números aleatorios por medio de métodos deterministas, por supuesto, vive en un estado de pecado.[1]

John von Neumann (1951)

4

Conclusiones

Así como pequeñas variaciones en k o en r aumentan considerablemente la cantidad de posibles autómatas celulares a estudiar, cada autómata puede ser analizado desde diferentes perspectivas, produciéndose gran cantidad de datos para la elaboración de un análisis minucioso.

Algunos de estos datos, para las reglas de evolución estudiadas, fueron presentados en el capítulo 3. Otros se han consolidado y se muestran a través de gráficos en los anexos A a G.

La primera conclusión del presente trabajo indica que la composición de reglas de evolución de ECAs puede tener como resultado un autómata celular equivalente a otro ECA. Por tal motivo, no sería racional emplear reglas compuestas para obtener resultados que se pueden alcanzar con una estructura más simple. Sin perjuicio de ello, es interesante destacar que si bien los diagramas espacio temporales de las reglas compuestas equivalentes a otros ECA no son iguales, cosa que el lector habrá notado al comparar las figuras 3.29a y 3.29b de la regla $R106 \circ R85$ con las figuras 3.21a y 3.21b de la regla $R86$ o al comparar las figuras 3.37a y 3.37b de la regla $R154 \circ R15$ con las figuras 3.38a y 3.38b de la regla 89; la complejidad lineal se comporta en

[1] Texto original: *Anyone who attempts to generate random numbers by deterministic means is, of course, living in a state of sin.*. Tomado de [23, p.640].

forma idéntica entre cada par de reglas respectivamente. Este último hecho se puede apreciar comparando las figuras B.1 y B.7 y las figuras B.2 y B.11 respectivamente.

Otra conclusión a la que se arriba, es la similitud existente entre las reglas $R105 \circ R45$ y $R154 \circ R150$. Este hecho se puede apreciar al rever los exponentes de Lyapunov en la tabla 3.3, en el estudio de ciclos de cada una de estas reglas: comparar los gráficos de ciclo máximo (figuras A.5 y A.10), comparar la longitud promedio de los ciclos (figuras A.27 y A.32), comparar la cantidad de ciclos generados por cada regla (figuras A.16 y A.21) y la fracción de estados globales que integran ciclos (figuras A.38 y A.43). También se observa, entre ambas reglas de evolución, similar complejidad lineal de las secuencias de bits por ellas generadas (ver figuras B.5 y B.10). La curva que representa la fracción de entropía de medida máxima se comporta de la misma forma en ambas reglas (figuras D.5 y D.10). Inclusive, los diagramas de transición de estados son idénticos (a excepción de los diagramas que representan el autómata compuesto por seis celdas, el cual es muy similar, pero no igual) como se puede apreciar comparando los STDs correspondientes a la regla $R105 \circ R45$ (figuras G.25 a G.30) con los STDs de la regla $R154 \circ R150$ (figuras G.55 a G.60). Al revisar el estudio de distancia de Hamming de cada uno de estas reglas se puede observar un comportamiento similar (con diferencias mínimas). Por todo ello y dado que los resultados producidos por los autómatas con una u otra de estas reglas de evolución son similares, se deberá elegir aquel cuya implementación sea más económica (en cantidad de componentes o compuertas lógicas). Por ejemplo, si se comparan las expresiones booleanas mínimas, ver las expresiones (3.25) y (3.77), o su representación en los circuitos lógicos de las figuras 3.11 y 3.32 respectivamente, se puede concluir que la regla $R154 \circ R150$ es más compacta (9 compuertas NAND y 16 inversores NOT) que la regla $R105 \circ R45$ (10 compuertas NAND y 17 inversores NOT). Si se emplea una menor cantidad de elementos, se redunda en un circuito más pequeño y además (de acuerdo a la disposición o conexión entre esos elementos) se logra menor tiempo de compuerta. De esta forma se puede obtener mejor rendimiento en términos de tasa de bits (*bit rate*) generados.

En este punto, las reglas estudiadas con posibilidades de ser candidatas son: $R45 \circ R30$, $R105 \circ R30$, $R166 \circ R45$, $R154 \circ R105$, $R154 \circ R150$ y $R166 \circ R106$. Al considerar los resultados obtenidos de exponentes de Lyapunov, estudios de ciclos, complejidad lineal de la secuencia generada, fracción de la entropía de medida máxima, distancia de Hamming y dispersión de di-

ferencias (todos ellos presentados en este trabajo); para cada uno de esas reglas, podemos concluir que las reglas $R166 \circ R45$ y $R166 \circ R106$ son potenciales competidoras de la regla 30 propuesta por Wolfram. Estas dos reglas son serias candidatas para ser analizadas, en estudios posteriores, empleando diversas pruebas de aleatoriedad sobre las secuencias de bits por ellas generadas. Pudiéndose generar dichas secuencias, de diferentes formas (implementaciones particulares), por ejemplo: tomando un único bit de una celda determinada por instante de tiempo , tomando un bit de una celda en un instante de tiempo y luego de la celda en la posición
pm 1 respecto al instante de tiempo anterior, tomando una cantidad n de bits (con $n > 1$) en un instante de tiempo, etc.

Las restantes reglas de evolución ($R45 \circ R30$, $R105 \circ R30$, $R105 \circ R45$ y $R154 \circ R105$) si bien pueden tener alguna característica destacable, otras propiedades que poseen son desfavorables en la generación de secuencias aleatorias. Si se toma como ejemplo la regla $R105 \circ R45$ se observará que los ciclos máximos que poseen los diversos CAs son muy convenientes (ver figura A.5); pero cuando se analiza la complejidad lineal de la secuencia de bits generada por dicha regla, se puede apreciar que la misma posee gran irregularidad y no presenta un patrón consistente (ver figura B.5). Las demás reglas citadas al inicio del presente párrafo sufren deficiencias similares.

Anexos

A

Ciclos

En las gráficas que se presentan a continuación se muestran los resultados obtenidos, en función de la cantidad de sitios (o celdas) del CA bajo las reglas de evolución que se presentaron en el capítulo 3.

Se debe aclarar que el incremento en la cantidad de celdas acarrea un incremento exponencial en los recursos computacionales necesarios para llevar a cabo el estudio. Por tal motivo, el rango de celdas sobre el cual se realizó el análisis es acotado.

En las figuras A.1 a A.10 se exhibe el tamaño de ciclo máximo para cada regla en función de la cantidad de celdas que componen el CA.

A continuación se presentan (figuras A.12 a A.21) la cantidad de ciclos hallados en cada configuración. Cada uno de ellos formará la raíz (*trees rooted on cycles*, ver [24, p. 52]) de diferentes árboles en los diagramas de transición de estados **STD**(ver Anexo G).

Luego se procede a exhibir la longitud promedio de los ciclos, también en función de la cantidad de celdas del CA (figuras A.23 a A.32).

Finalmente en este anexo el lector podrá encontrar la proporción de estados globales que integran ciclos sobre la la cantidad total de estados globales del CA (figuras A.34 a A.43).

A.1. Ciclos máximos

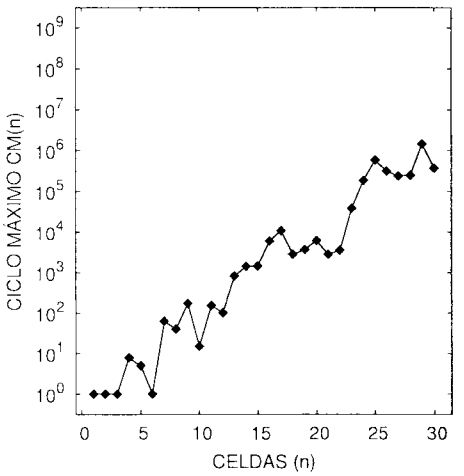


Figura A.1: Regla 30

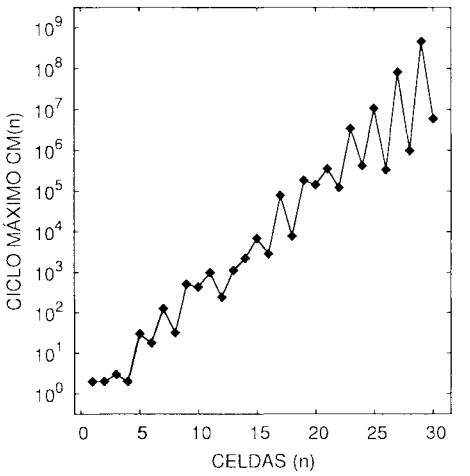


Figura A.2: Regla 45

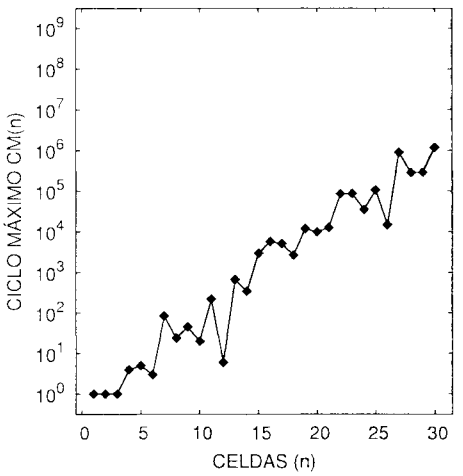


Figura A.3: Regla $45 \circ 30$

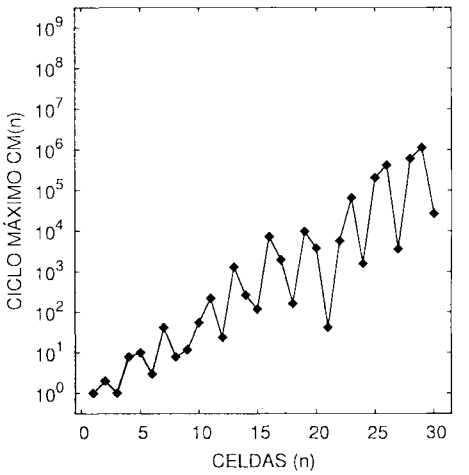


Figura A.4: Regla $105 \circ 30$

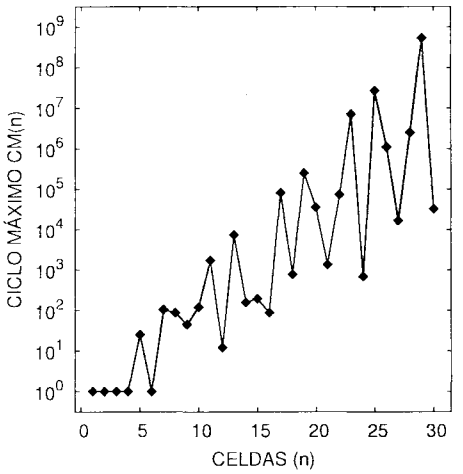


Figura A.5: Regla $105 \circ 45$

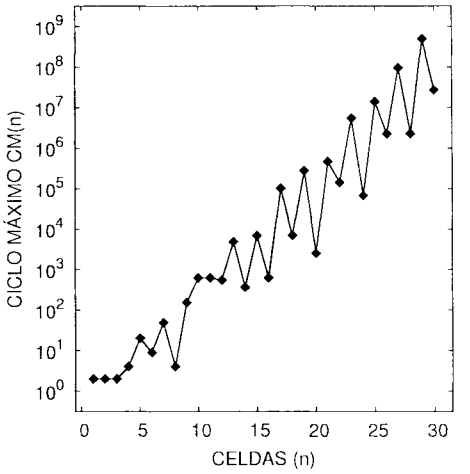


Figura A.6: Regla $166 \circ 45$

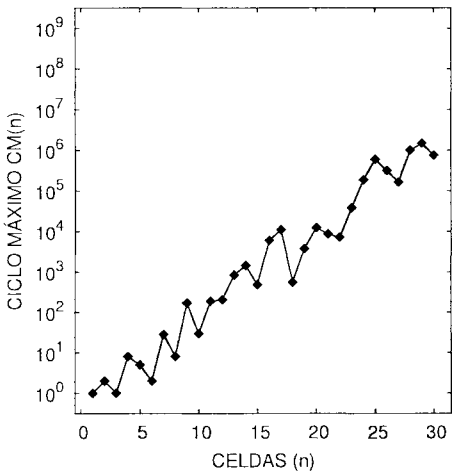


Figura A.7: Regla $106 \circ 85$

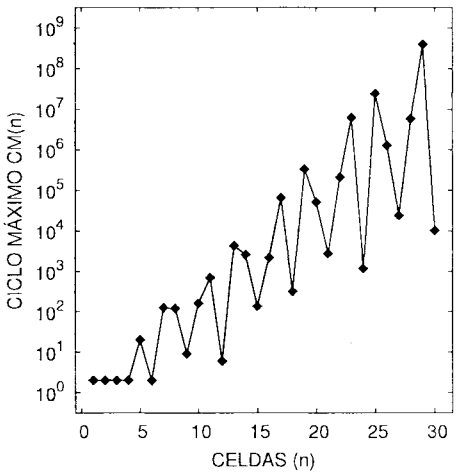


Figura A.8: Regla $154 \circ 105$

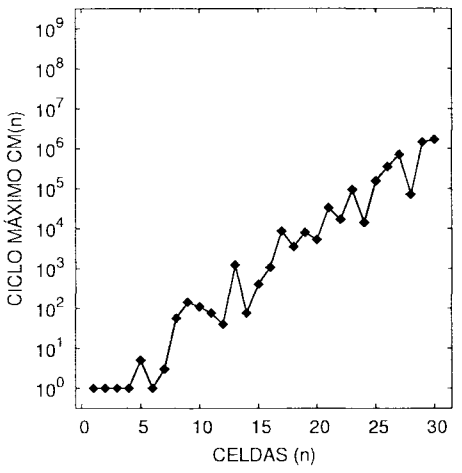


Figura A.9: Regla $166 \circ 106$

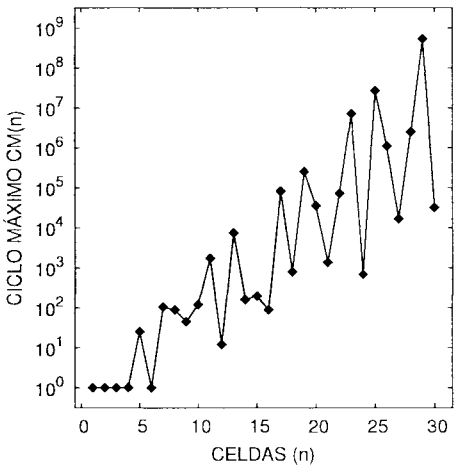


Figura A.10: Regla $154 \circ 150$

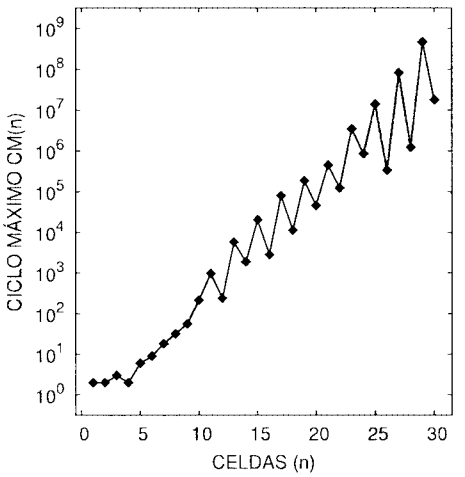


Figura A.11: Regla 154 o 15

A.2. Cantidad de ciclos

Los gráficos que se muestran a continuación se encuentran en escala semi-logarítmica.

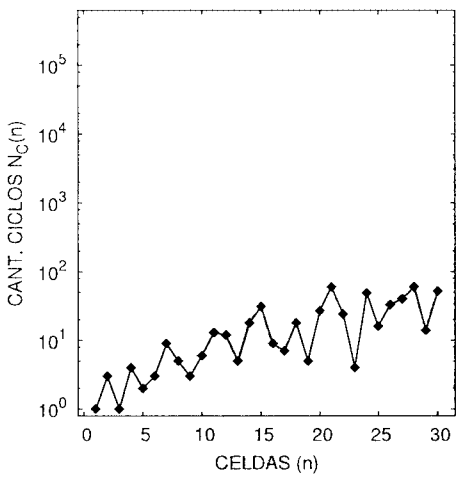


Figura A.12: Regla 30

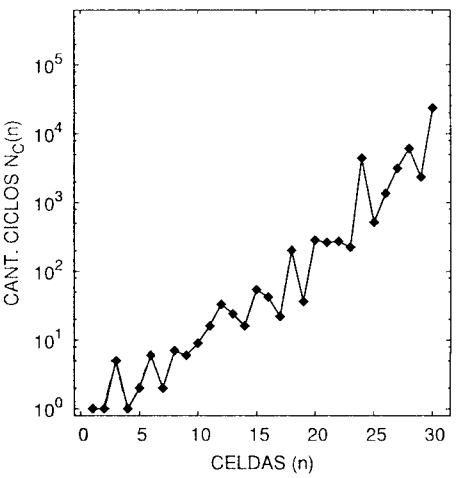


Figura A.13: Regla 45

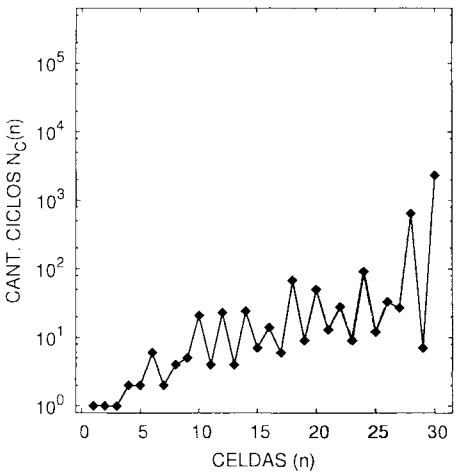


Figura A.14: Regla 45 o 30

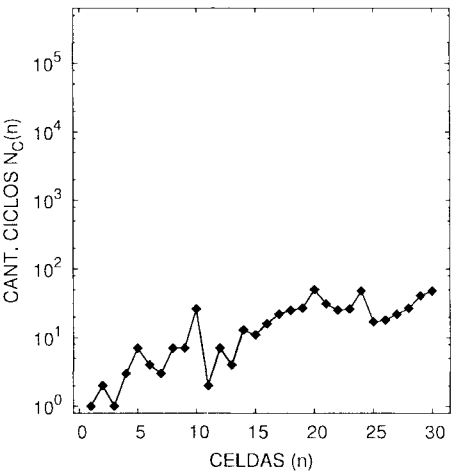


Figura A.15: Regla 105 o 30

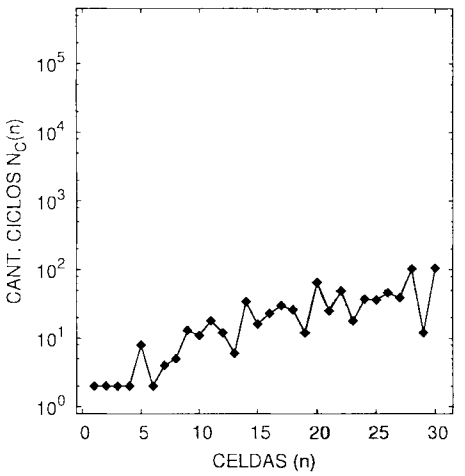


Figura A.16: Regla 105 o 45

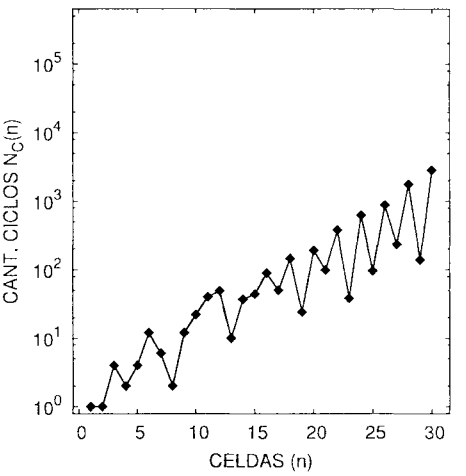


Figura A.17: Regla 166 o 45

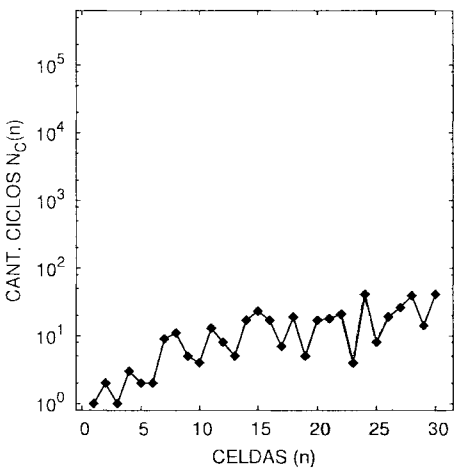


Figura A.18: Regla 106 o 85

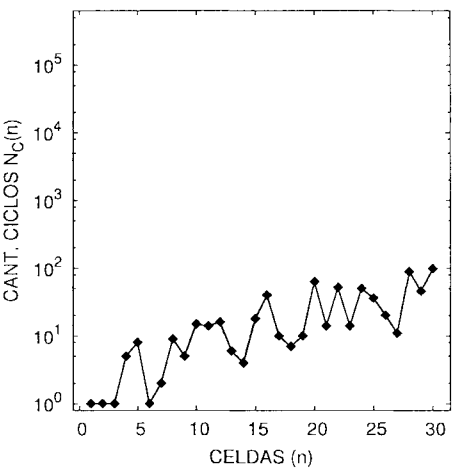


Figura A.19: Regla 154 o 105

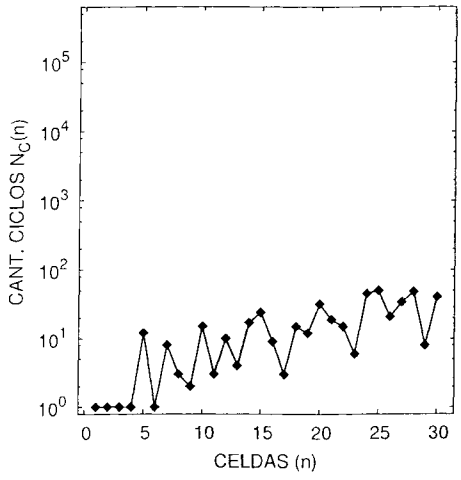


Figura A.20: Regla 166 o 106

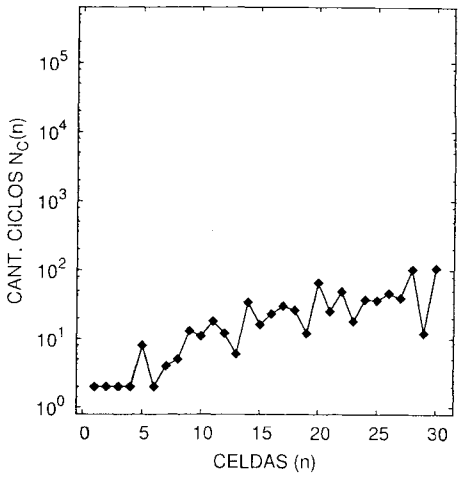


Figura A.21: Regla 154 o 150

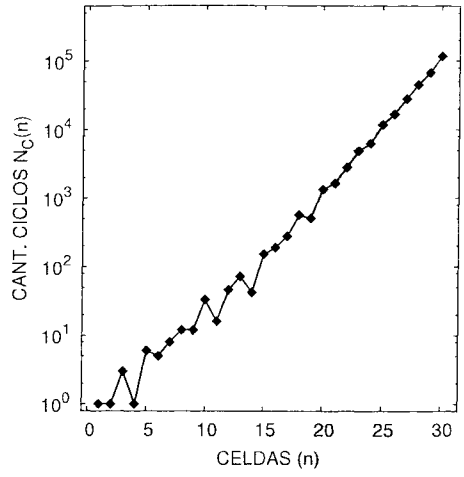


Figura A.22: Regla 154 o 15

A.3. Longitud promedio de los ciclos

Los gráficos que se muestran a continuación se encuentran en escala semi-logarítmica.

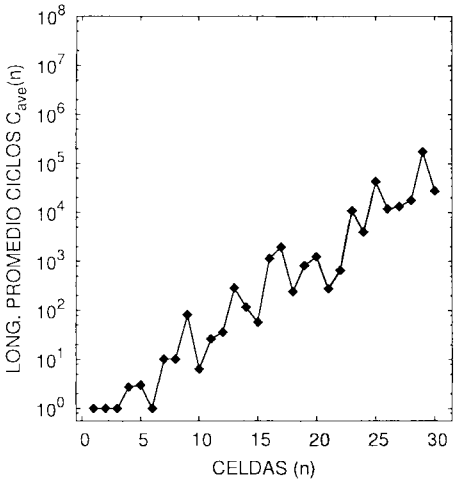


Figura A.23: Regla 30

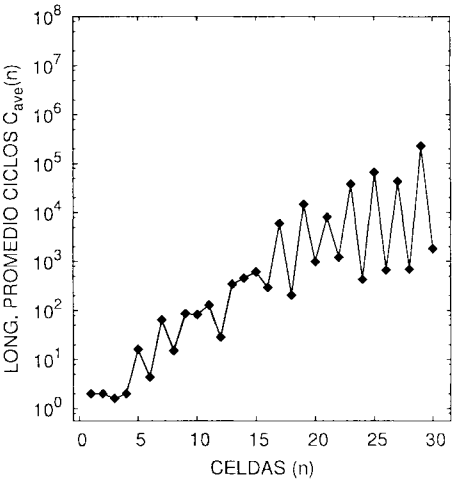


Figura A.24: Regla 45

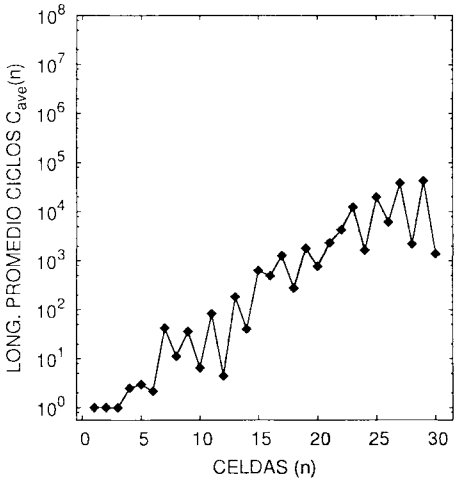


Figura A.25: Regla $45 \circ 30$

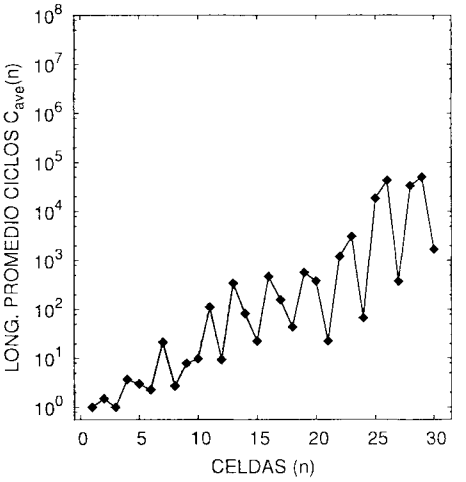


Figura A.26: Regla $105 \circ 30$

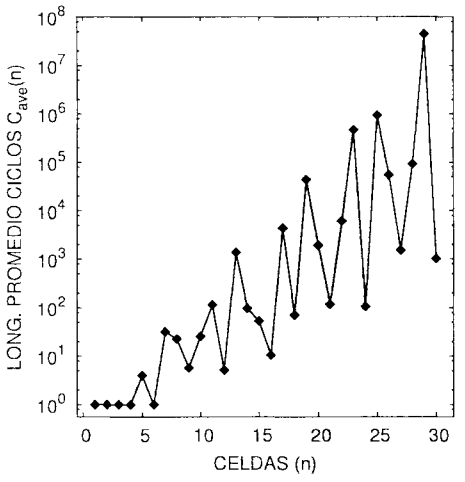


Figura A.27: Regla $105 \circ 45$

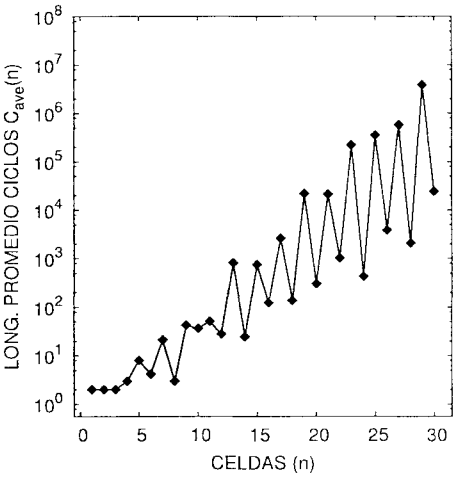


Figura A.28: Regla $166 \circ 45$

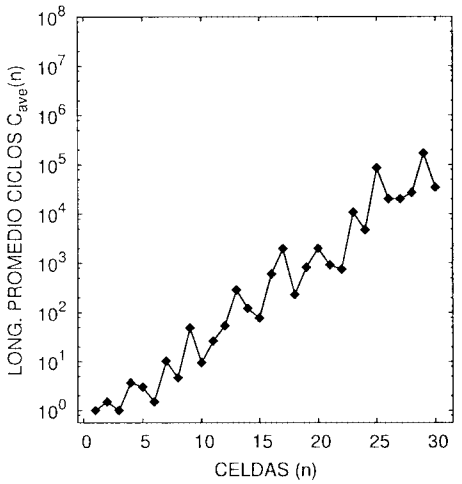


Figura A.29: Regla 106 o 85

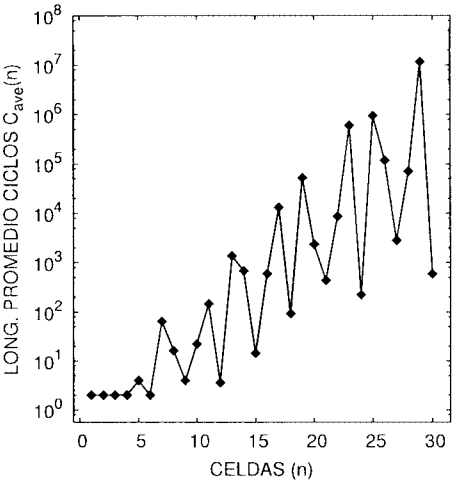


Figura A.30: Regla 154 o 105

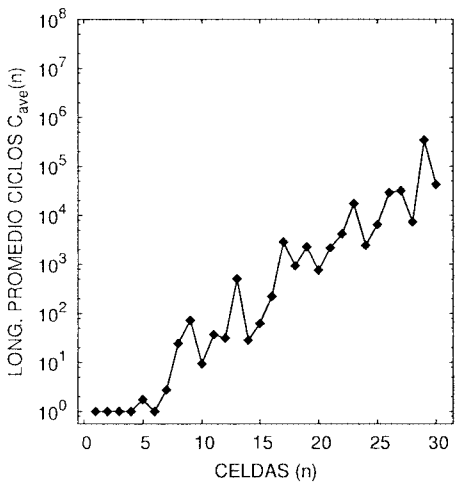


Figura A.31: Regla 166 o 106

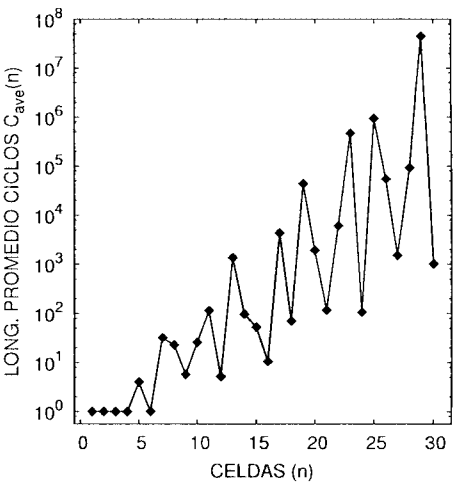


Figura A.32: Regla 154 o 150

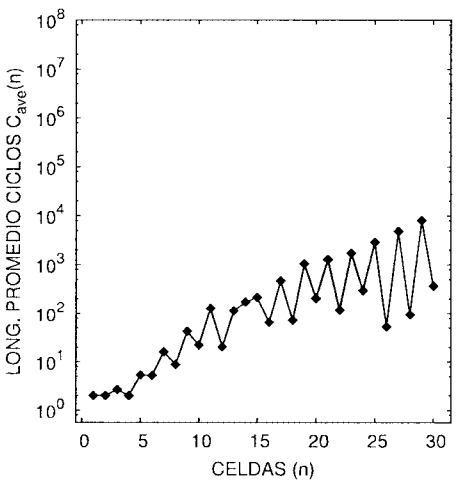


Figura A.33: Regla 154 o 15

A.4. Proporción de estados en ciclos

Los gráficos que se muestran a continuación se encuentran en escala semi-logarítmica.

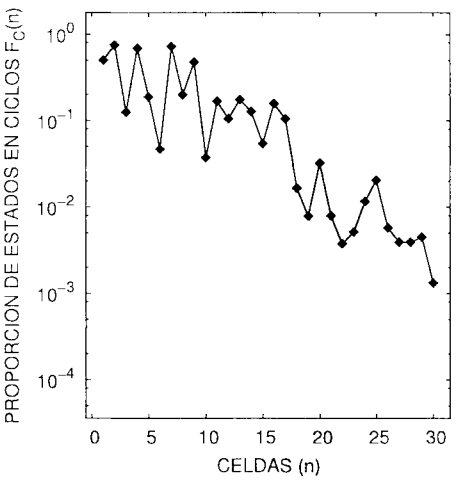


Figura A.34: Regla 30

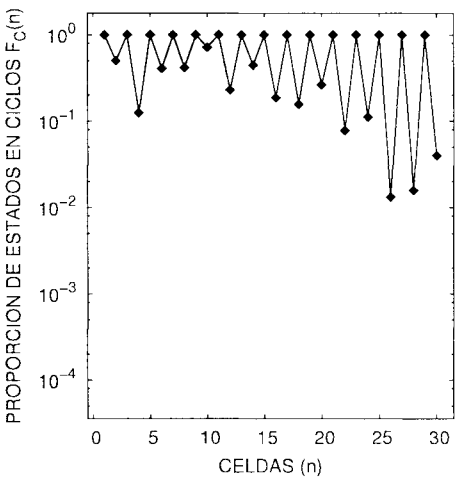


Figura A.35: Regla 45

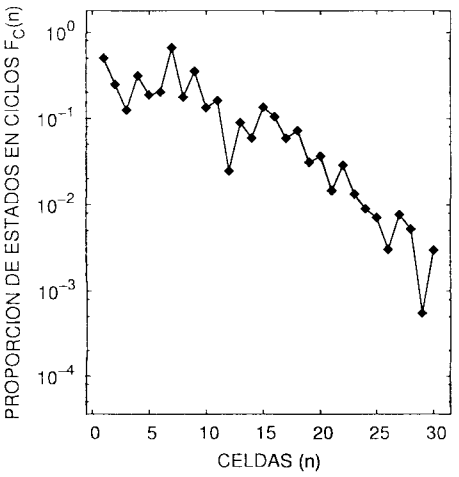


Figura A.36: Regla $45 \circ 30$

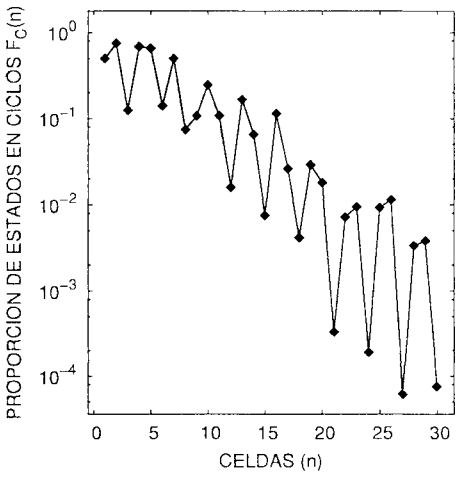


Figura A.37: Regla $105 \circ 30$

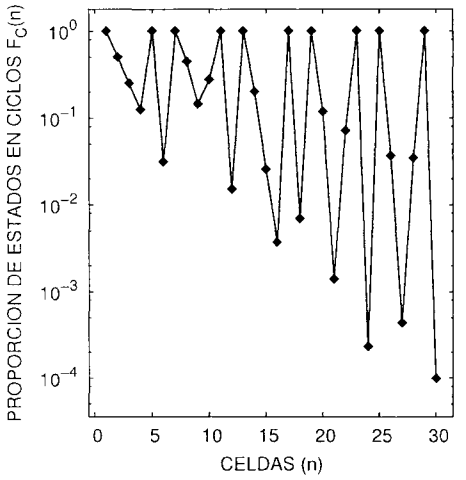


Figura A.38: Regla $105 \circ 45$

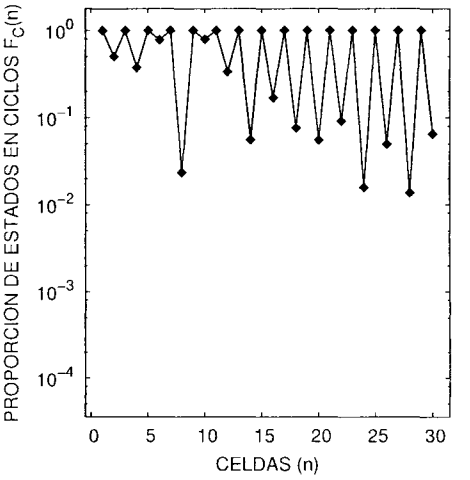


Figura A.39: Regla $166 \circ 45$

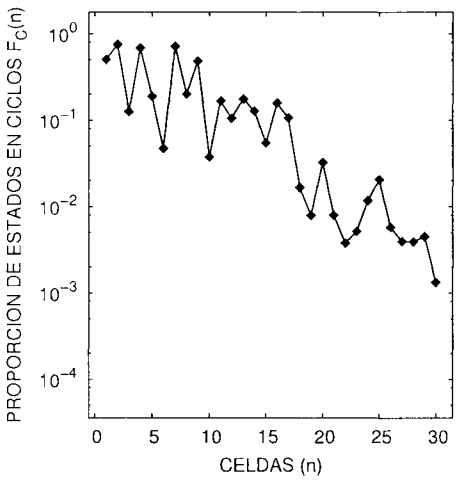


Figura A.40: Regla $106 \circ 85$

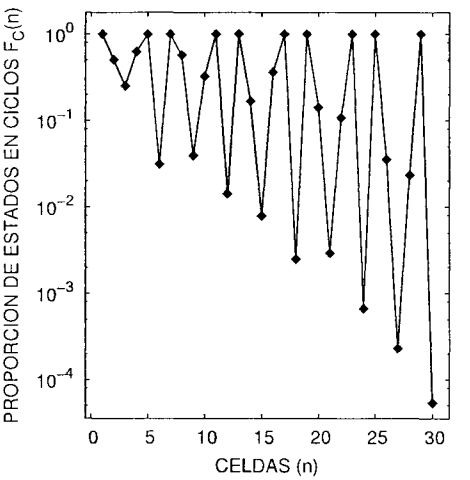


Figura A.41: Regla $154 \circ 105$

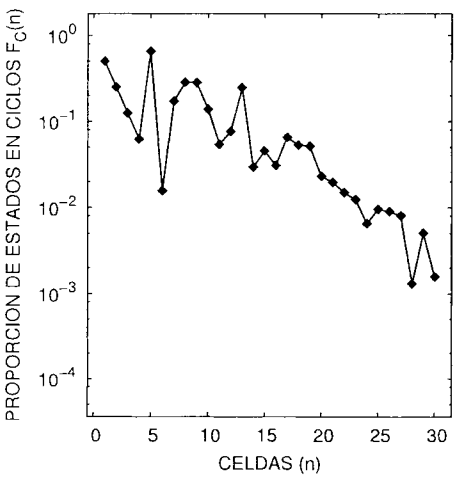


Figura A.42: Regla $166 \circ 106$

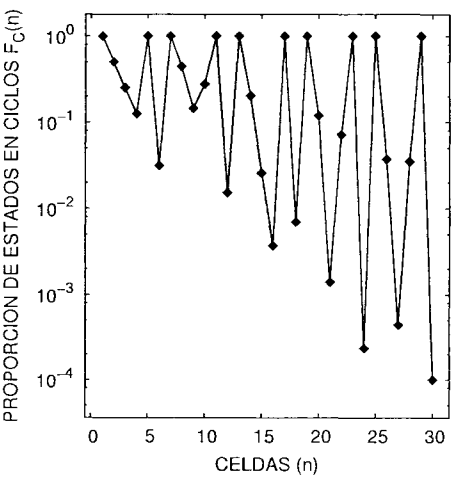


Figura A.43: Regla $154 \circ 150$

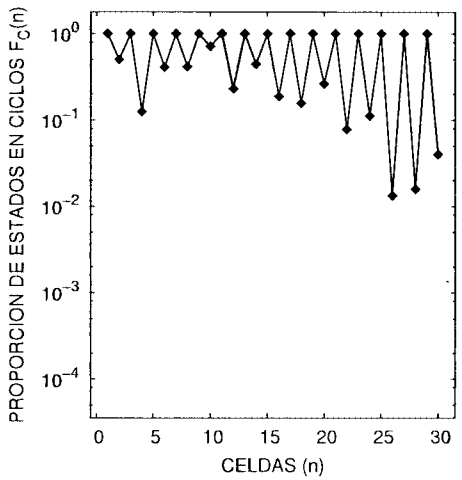


Figura A.44: Regla $154 \circ 15$

B

Complejidad Lineal

En este anexo se muestra la complejidad lineal (*L = linear complexity profile*) de las secuencias de bits obtenidas a partir de diferentes configuraciones de CAs. Los valores estadísticas se han extraído a partir de 1.000 secuencias, de 100.000 bits, generadas por el CA (su celda central). Luego las secuencias han sido analizadas utilizando el algoritmo de Berlekamp-Massey (implementado a partir de [1, p. 201] [2, p. 80]). La complejidad lineal esperada para una secuencia aleatoria de longitud N es: $L = N/2$ [1, p. 199].

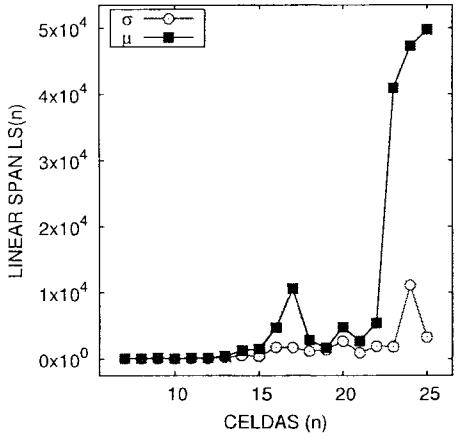


Figura B.1: Regla 30.

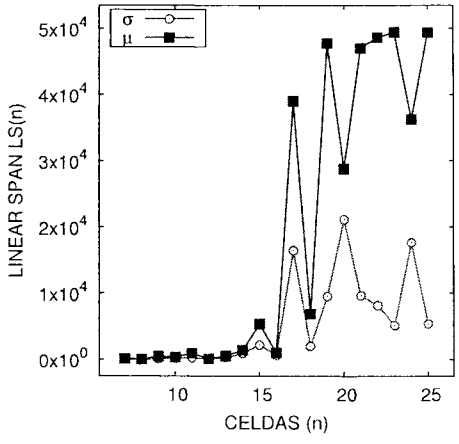


Figura B.2: Regla 45.

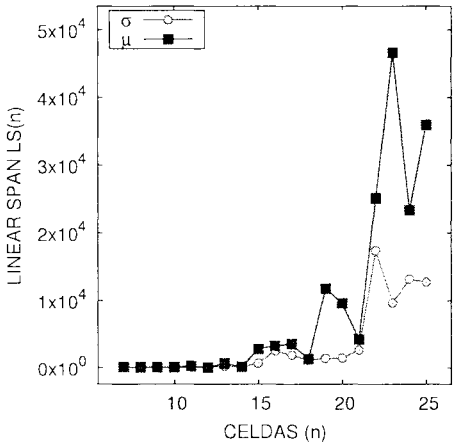


Figura B.3: Regla 45 o 30.

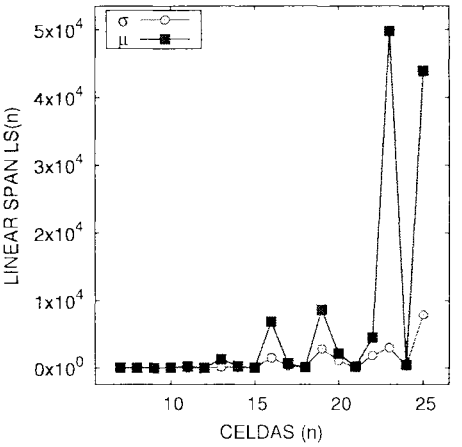


Figura B.4: Regla 105 o 30.

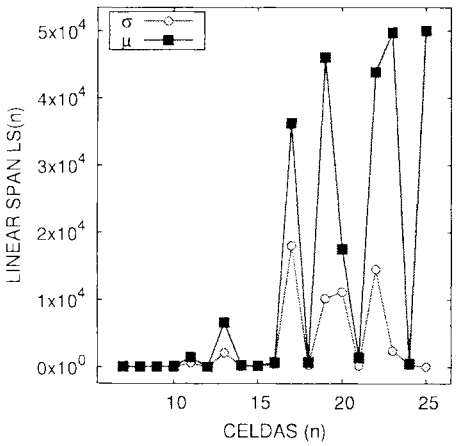


Figura B.5: Regla 105 o 45.

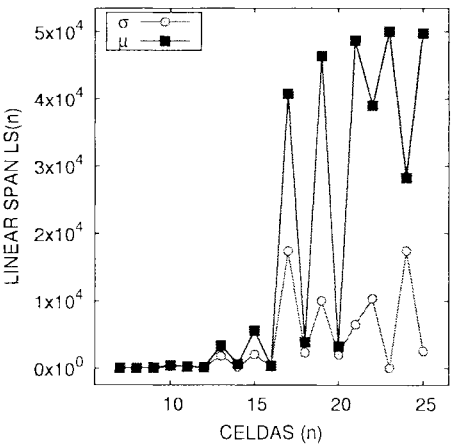


Figura B.6: Regla 166 o 45.

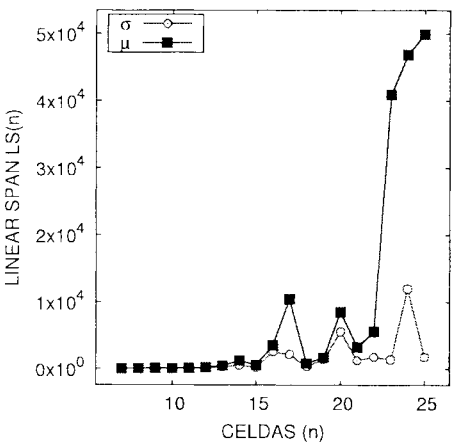


Figura B.7: Regla 106 o 85.

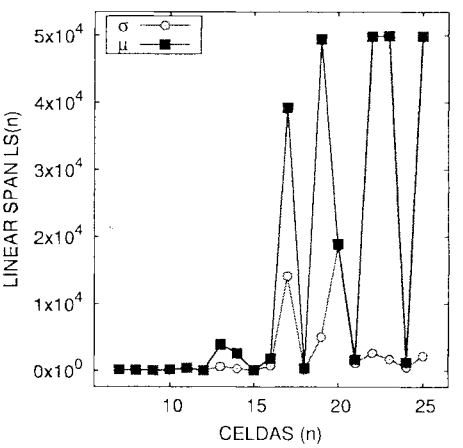


Figura B.8: Regla 154 o 105.

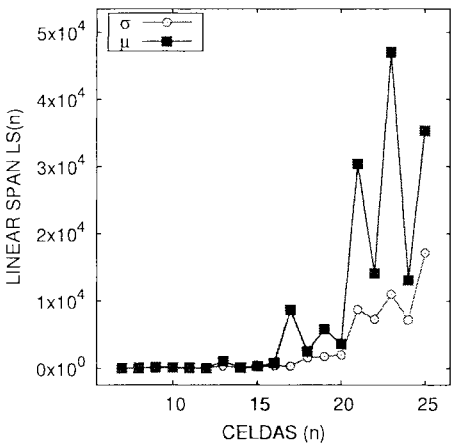


Figura B.9: Regla 166 o 106.

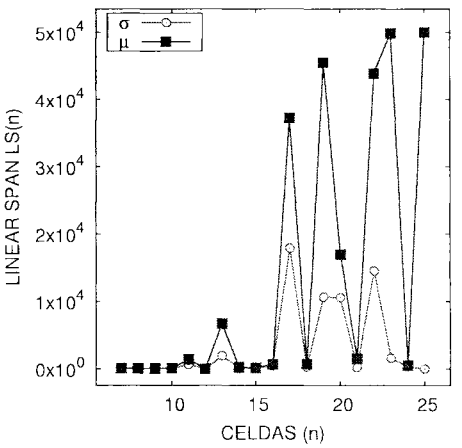


Figura B.10: Regla 154 o 150.

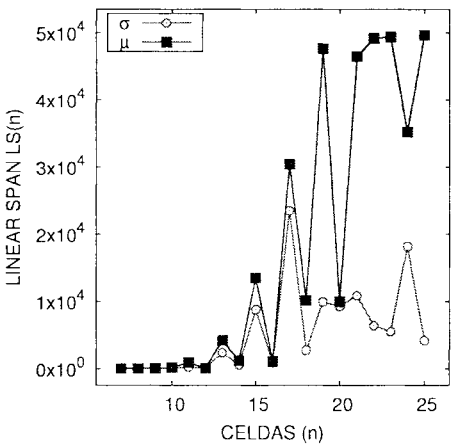


Figura B.11: Regla 154 o 15.

C

Mapeo de estados sobre sí mismos

En las imágenes que se muestran en este anexo observa el mapeo de los N estados sobre sí mismos, para la evolución de 1 instante de tiempo (*1 time step*). Cada punto en las imágenes representa la evolución desde un estado de origen en $t = 0$ (eje horizontal) hasta el estado de destino en $t = 1$ (eje vertical); para un CA de 10 celdas y $k = 2$ (1024 estados posibles). Comparar con *State space as Cantor set* en [5].

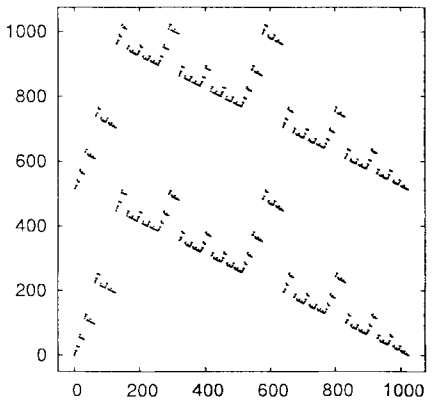


Figura C.1: Regla 30

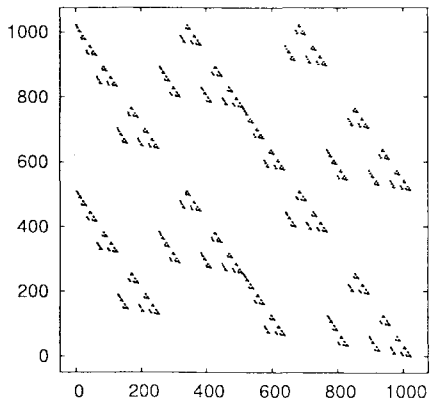


Figura C.2: Regla 45

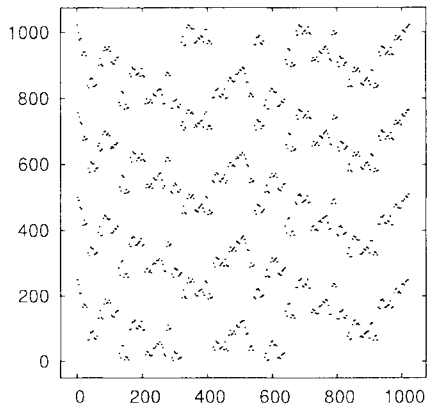


Figura C.3: Regla $45 \circ 30$

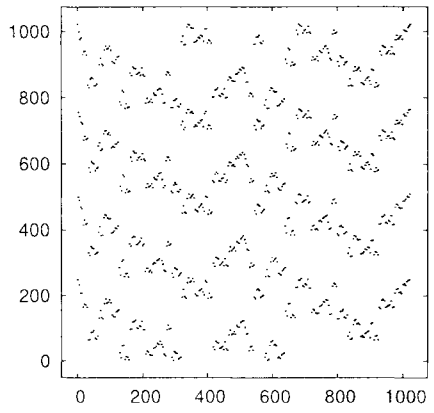


Figura C.4: Regla $105 \circ 30$

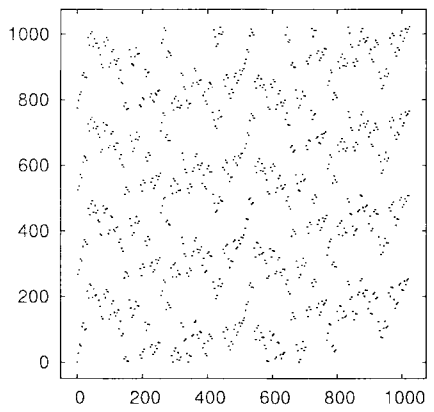


Figura C.5: Regla $105 \circ 45$

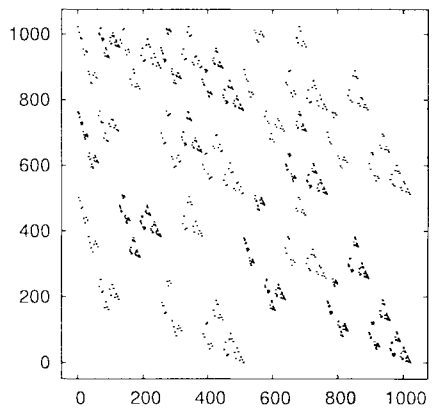


Figura C.6: Regla $166 \circ 45$

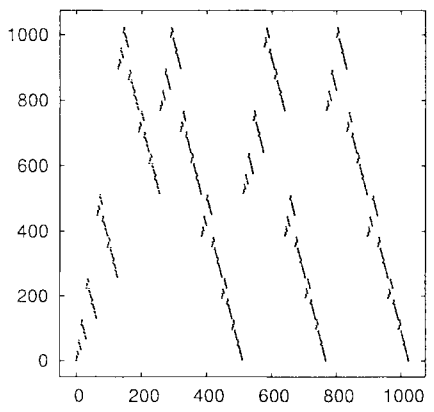


Figura C.7: Regla $106 \circ 85$

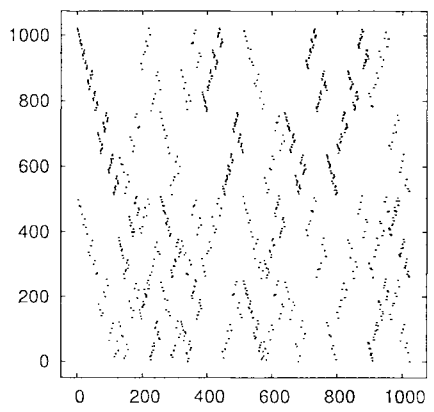


Figura C.8: Regla $154 \circ 105$

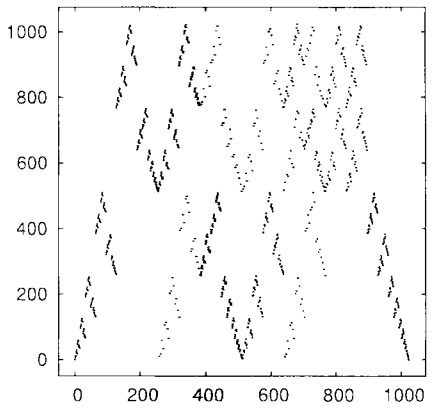


Figura C.9: Regla $166 \circ 106$

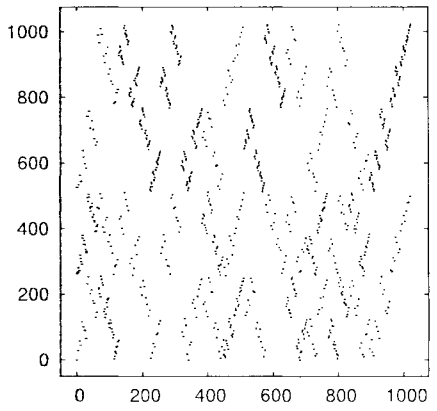


Figura C.10: Regla $154 \circ 150$

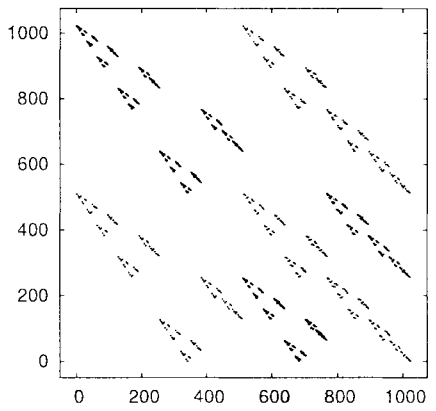


Figura C.11: Regla $154 \circ 15$

D

Entropía

La entropía, concepto introducido por Claude Shannon en 1948, es una medida matemática de cantidad de información o la incertidumbre de una variable aleatoria [25, p.54]. La entropía $H(X)$ se define como [1, p.56]:

$$H(X) = -\sum_{i=1}^{n} p_i log(p_i) \tag{D.1}$$

donde p_i es la probabilidad de ocurrencia cd cada uno de los x_i valores que puede adoptar la variable aleatoria X.

Existen diferentes entropías asociadas a los autómatas celulares, ya sean estas espaciales, temporales, topológicas o de medida. La entropías topológicas reflejan los estados globales (configuraciones) posibles y las entropías de medida sólo aquellos estados posibles, es decir, no son afectadas por los eventos con cero probabilidad de ocurrencia. Los autómatas clase I tienen entropía espacial y temporal de medida 0 (cero), los autómatas clase 2 tienen entropía temporal de medida 0 (cero) y los autómatas clase III tienen entropía espacial y temporal de medida positivas [4, pp. 460-462].

En $t = 0$ los autómatas pueden estar en cualquiera de los k^N estados iniciales, por lo tanto su entropía espacial de medida es máxima ($S(0) = 1$). En cada evolución la cantidad de estados posibles puede reducirse y algunos

estados son más probables que otros (leer capítulo 2), mientras otros estados ya no serán alcanzables. Esto reduce la entropía espacial de medida a una fracción del máximo ($0 < S(t) < 1$ para $t \geq 1$) y refleja la "compresión de estados" o irreversibilidad del autómata.

La entropía espacial de medida, corresponde a un instante de tiempo determinado. Las imágenes que se exhiben a continuación muestran la evolución de la entropía espacial de medida en función del tiempo y corresponden a los autómatas celulares estudiados con una estructura de 15 celdas. Se ha empleando la fórmula : $S(t) = -\frac{1}{N}\sum_{i=1}^{2^N} P_i^t log_2(P_i^t)$ (donde P_i^t es la probabilidad de ocurrencia del estado global i en el instante de tiempo t) [4, p.83].

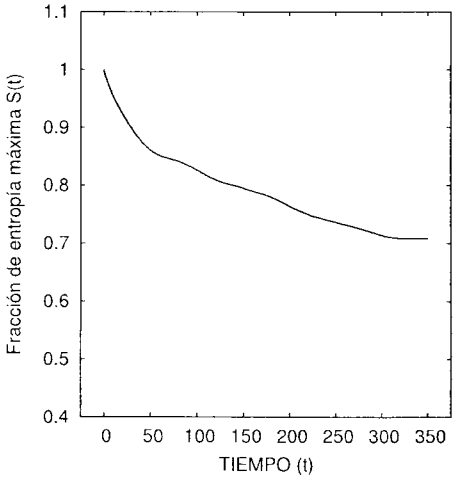


Figura D.1: Regla 30

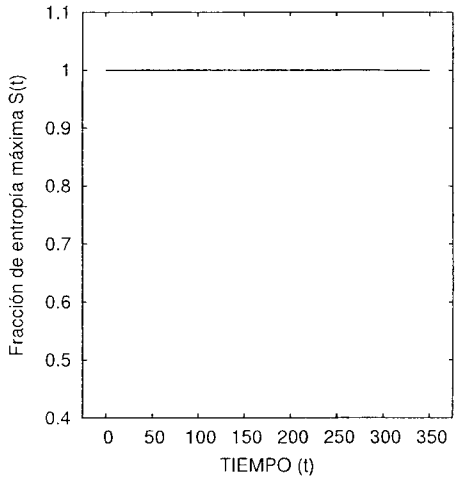


Figura D.2: Regla 45

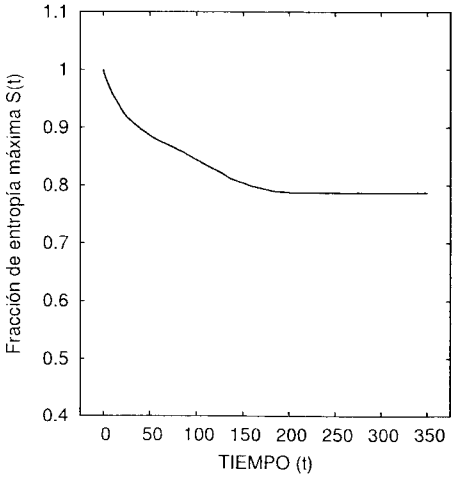


Figura D.3: Regla 45 o 30

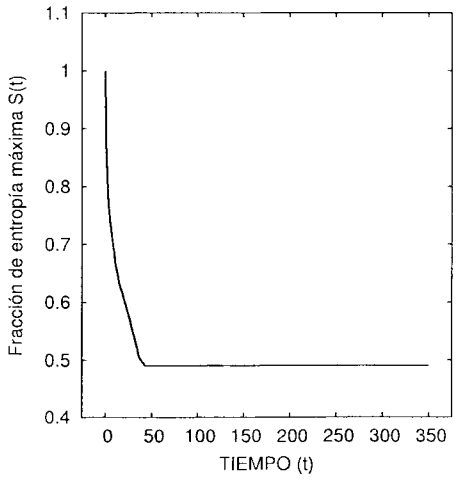


Figura D.4: Regla 105 o 30

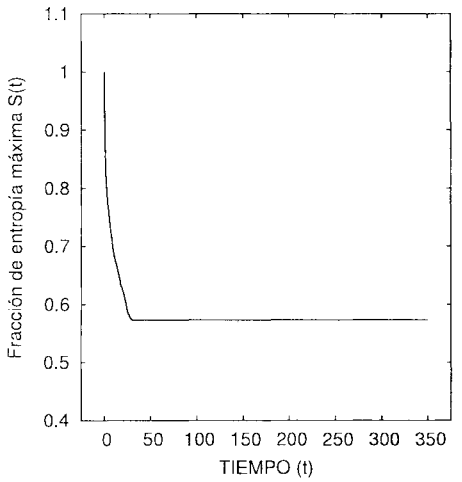


Figura D.5: Regla 105 o 45

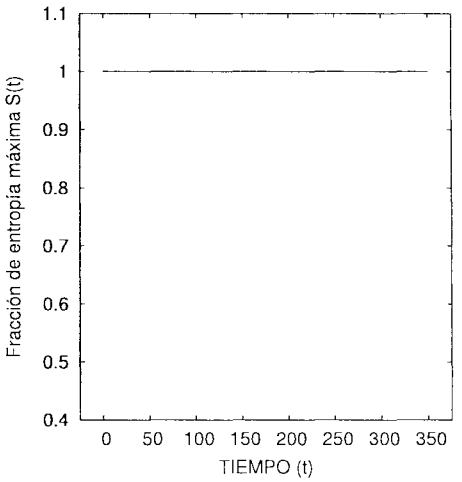


Figura D.6: Regla 166 o 45

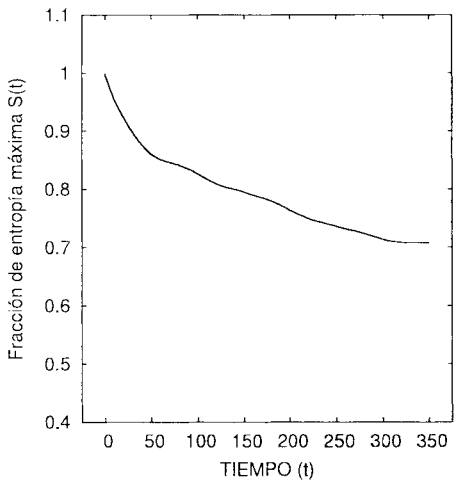


Figura D.7: Regla 106 o 85

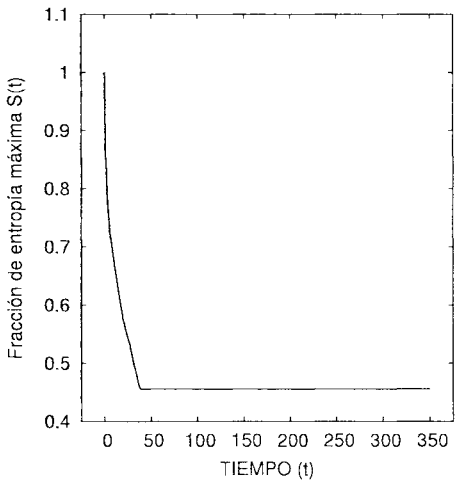


Figura D.8: Regla 154 o 105

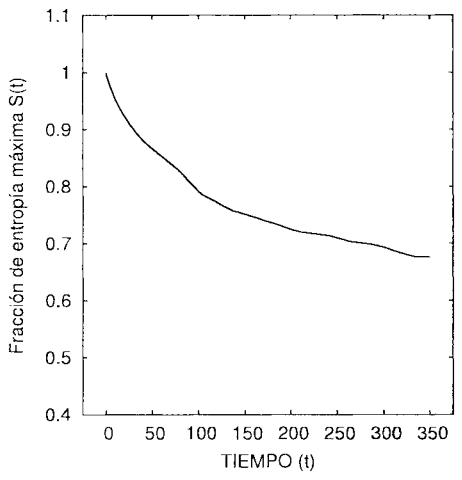


Figura D.9: Regla 166 o 106

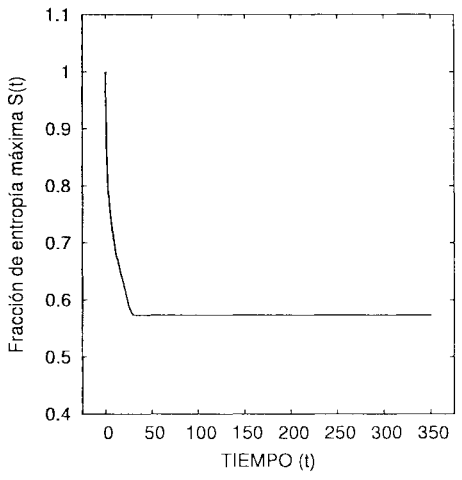


Figura D.10: Regla 154 o 150

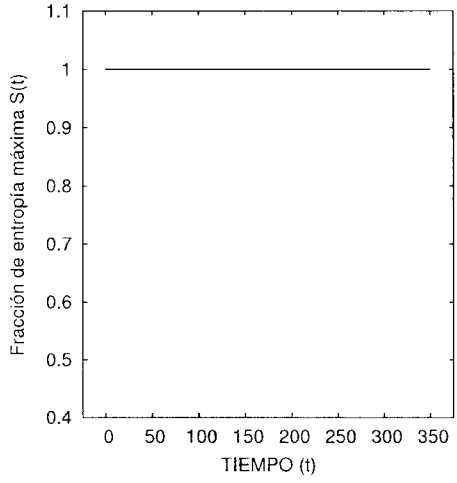


Figura D.11: Regla $154 \circ 15$

E

Distancia de Hamming

Dadas dos cadenas de bits (en general, símbolos de un alfabeto), S_1 y S_2, la distancia de Hamming entre ambas cadenas corresponde a la cantidad de bits en que difieren [26, p. 15(37)], al realizar la comparación bit a bit. En el caso de los autómatas celulares, la distancia de Hamming corresponderá a la cantidad de celdas en que difieren dos autómatas, en un instante determinado. Imágenes que representan la evolución de la distancia de Hamming en función del tiempo, entre dos configuraciones iniciales aleatorias de 1024 celdas partiendo de $H(0) = 1$. H(t) se encuentra normalizada, habiéndose analizado 1000 condiciones iniciales distintas.

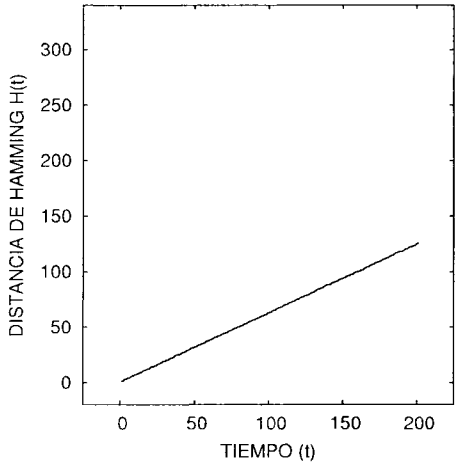


Figura E.1: Regla 30

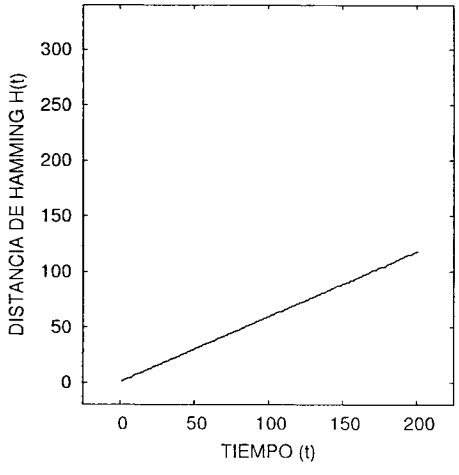


Figura E.2: Regla 45

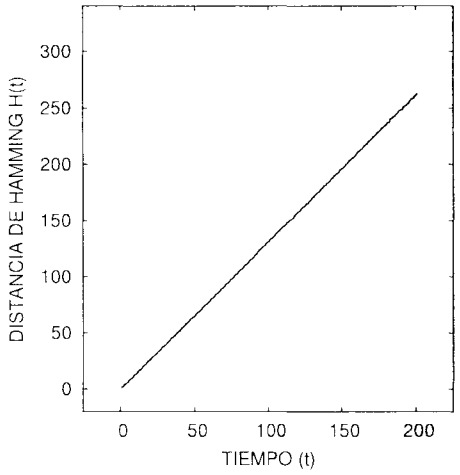


Figura E.3: Regla $45 \circ 30$

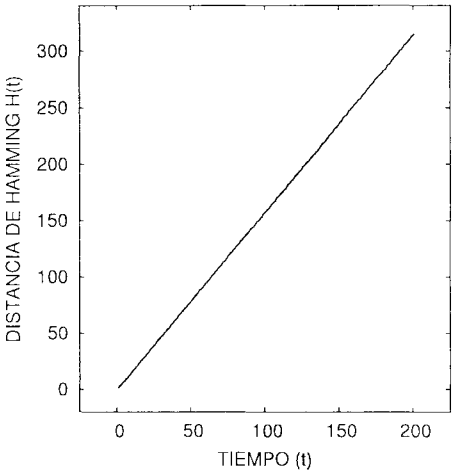


Figura E.4: Regla $105 \circ 30$

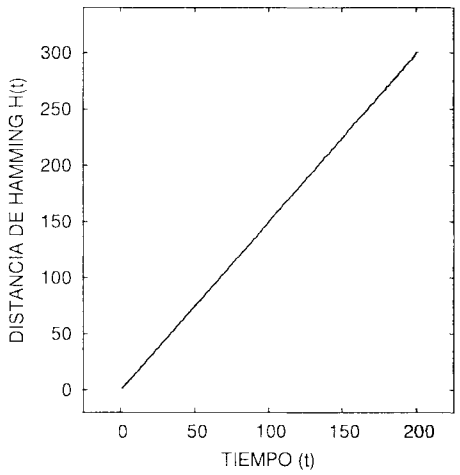


Figura E.5: Regla $105 \circ 45$

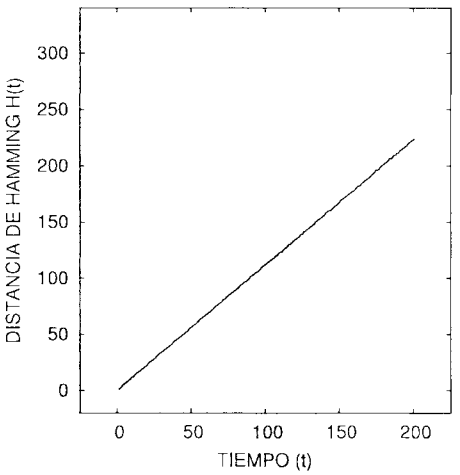


Figura E.6: Regla $166 \circ 45$

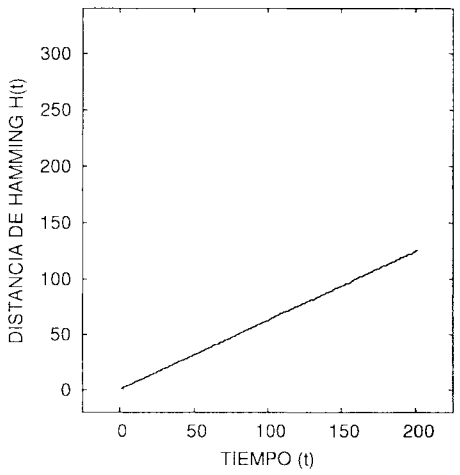


Figura E.7: Regla $106 \circ 85$

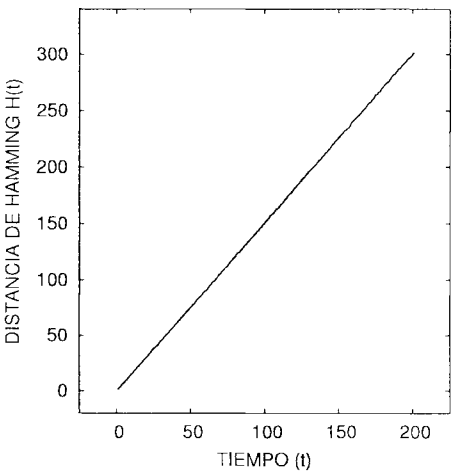


Figura E.8: Regla $154 \circ 105$

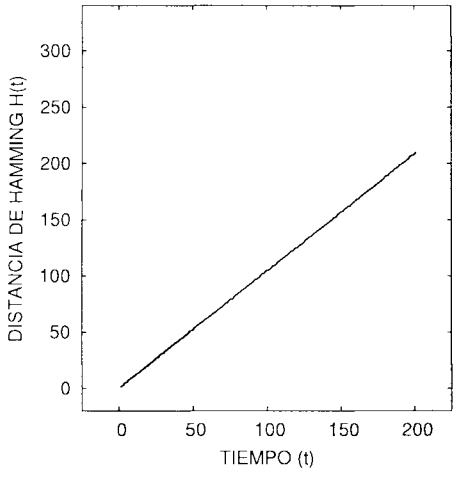


Figura E.9: Regla $166 \circ 106$

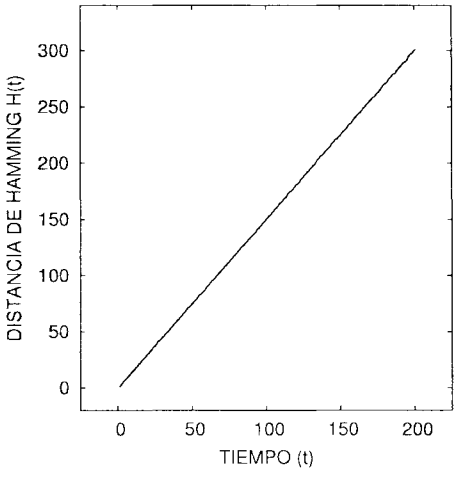


Figura E.10: Regla $154 \circ 150$

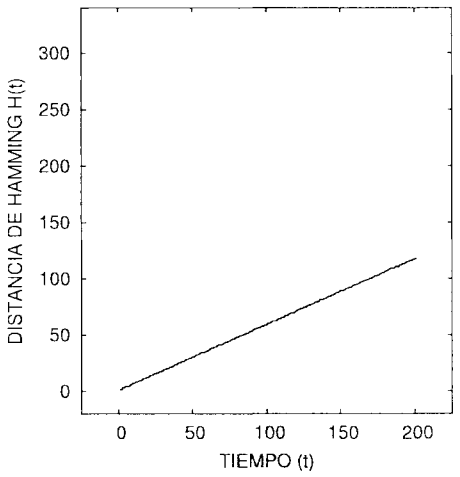


Figura E.11: Regla $154 \circ 15$

F

Diagramas dispersión de diferencias

Cuando dos CA con configuraciones iniciales que difieren mínimamente (una única celda), evolucionan aplicando la misma regla de evolución a ambas, puede ocurrir que esa diferencia se mantenga en el mínimo, o desaparezca o que se expanda. Este último caso es precisamente lo que ocurre con las reglas de evolución propuestas en este trabajo, como se aprecian en las figuras que suceden a este texto. Este tipo de diagrama se obtiene (para CA 1D con $k = 2$) calculando en cada instante de tiempo t la diferencia entre ambos CA, esto se realiza aplicando la operación lógica "øexclusivo" (XOR) celda a celda.

Esta propiedad, estabilidad ante la mínima perturbación, puede emplearse para clasificar autómatas celulares. Mientras que las perturbaciones desaparecen rápidamente en los CAs clase I y permanecen acotadas a una región localizada en la clase II; en los autómatas celulares clase III la perturbación se expande a una velocidad o tasa constante, la cual está relacionada con los exponentes de Lyapunov derecho e izquierdo [4, p.461]. Como se puede apreciar en las imágenes que se presentan en este anexo, todos los CAs considerados expanden, en forma constante, la diferencia; lo cual muestra que poseen características de la clase III. Asimismo consultando el cuadro 3.3 se podrá percibir la relación de la velocidad de propagación con λ_L y λ_R.

Este tipo de diagrama es conocido como patrón o dispersión de diferencias (*difference pattern*). Este tipo de gráficos permite visualizar como, para algunas reglas de evolución, un mínimo "error" se propaga en el tiempo y en el espacio, ampliándose la distancia (de Hamming) entre los dos autómatas. También conocido como propagación de daños (*damage spreading*)[20].

En particular las imágenes expuestas a continuación representan la expansión de las diferencias para los diferentes CA estudiados, constituidos por 300 celdas y para 75 generaciones (*time steps*).

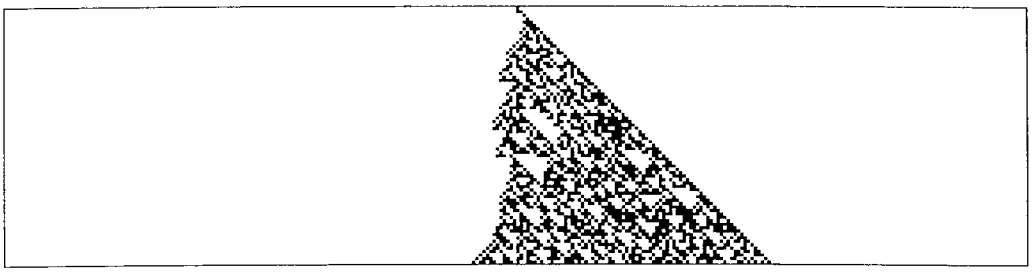


Figura F.1: Regla 30

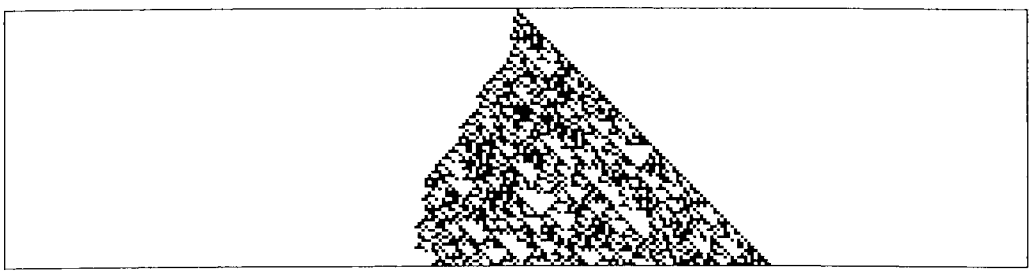


Figura F.2: Regla 45

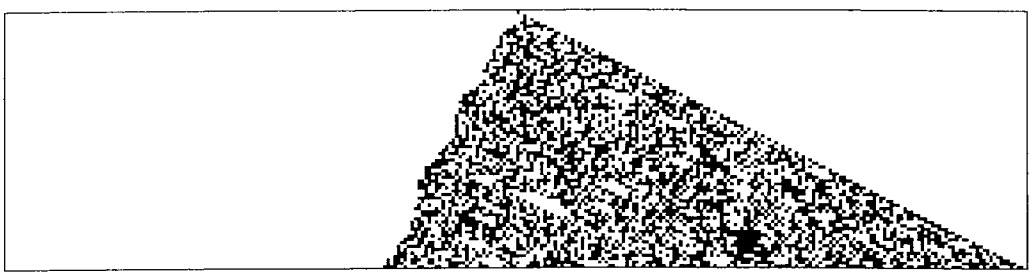


Figura F.3: Regla 45 ∘ 30

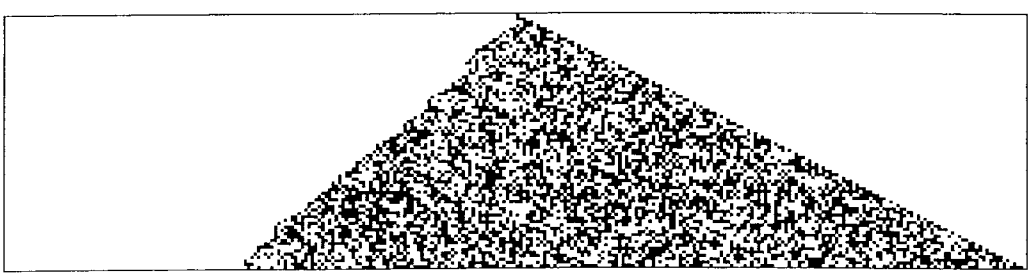


Figura F.4: Regla 105 ∘ 30

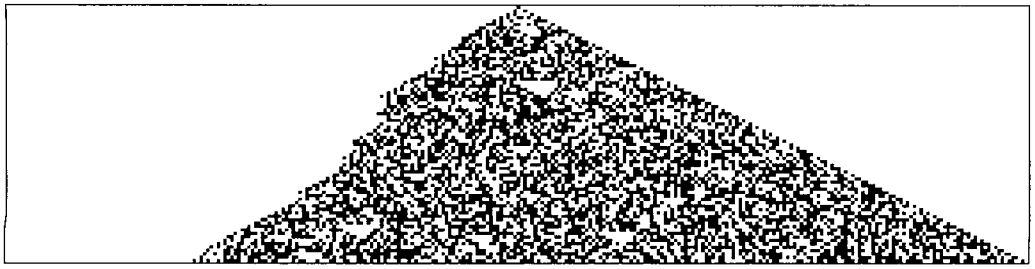


Figura F.5: Regla $105 \circ 45$

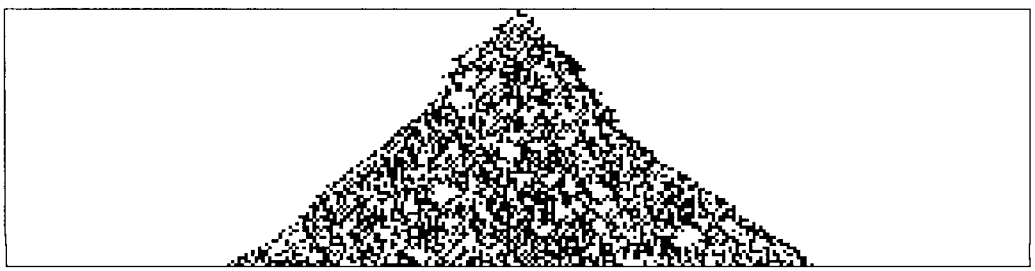


Figura F.6: Regla $166 \circ 45$

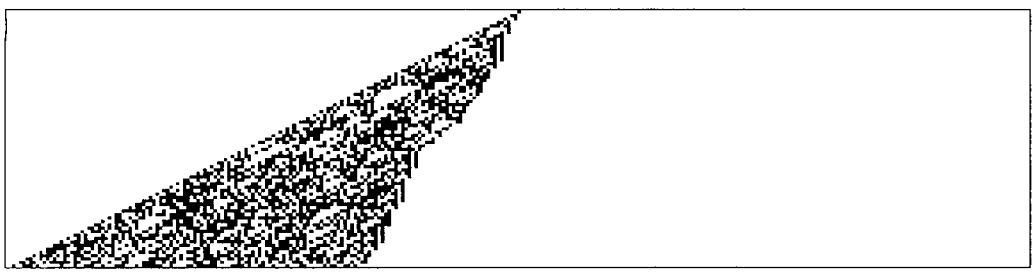


Figura F.7: Regla $106 \circ 85$

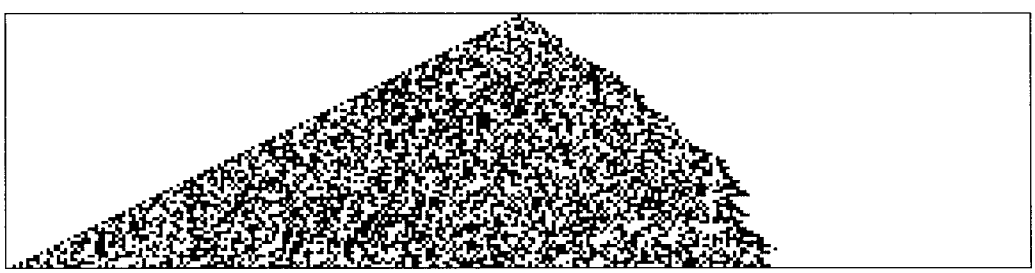


Figura F.8: Regla $154 \circ 105$

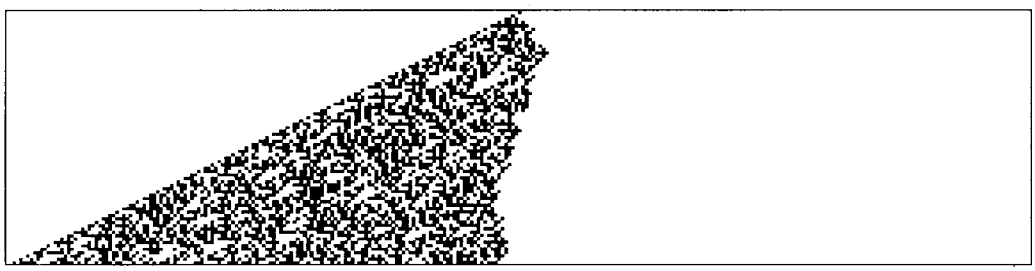


Figura F.9: Regla $166 \circ 106$

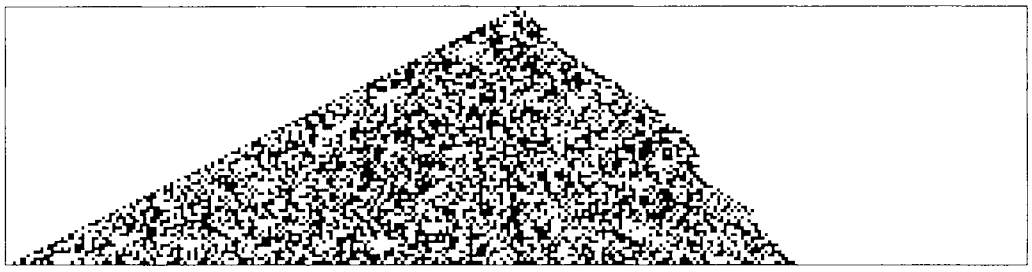


Figura F.10: Regla 154 o 150

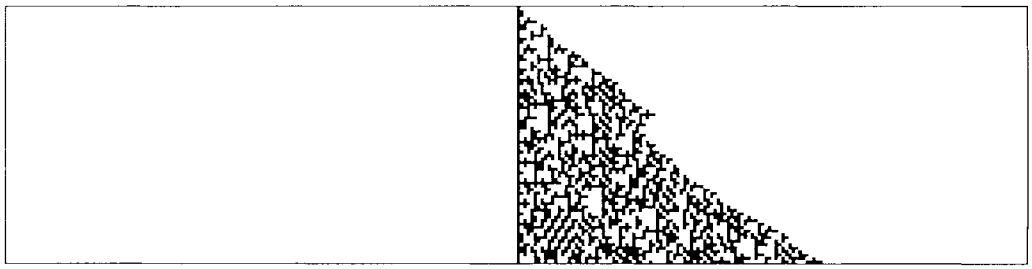


Figura F.11: Regla 154 o 15

Diagramas de Transición de Estados (STD)

Un CA de N celdas, las cuales puede estar en uno de k estados posibles, en cada instante de tiempo se encuentra en uno de los k^N estados globales del autómata. Cada vez que se aplica la regla de evolución el autómata adopta otro estado (también es posible que un estado global puede evolucionar sobre sí mismo, formando un bucle). Esta transición entre estados puede presentarse gráficamente mediante la utilización de grafos dirigidos, donde los estados se muestran como nodos y la evolución entre ellos se muestra con aristas que unen a los primeros [4].

Estos grafos, generalmente conocidos como diagrams de transición de estados (**STD** por su nombre en inglés *State Transition Diagram*), en términos de sistemas dinámicos representan el plano de fases (*phase space*)del CA y en algunos casos reciben el nombre de centros de atracción (*basins of attraction*) [27].

A continuación se presentan algunos STD para las reglas de evolución tratadas en el presente trabajo con diferente cantidad de celdas.

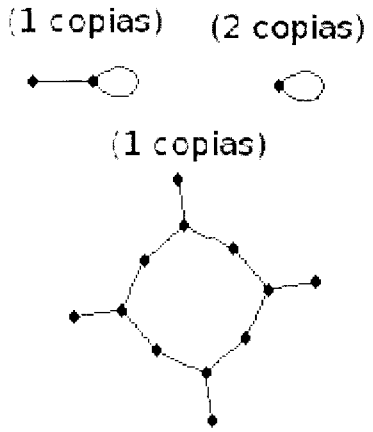


Figura G.1: Regla 30 4 celdas

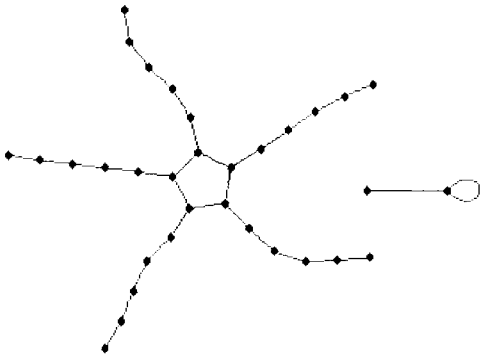


Figura G.2: Regla 30 5 celdas

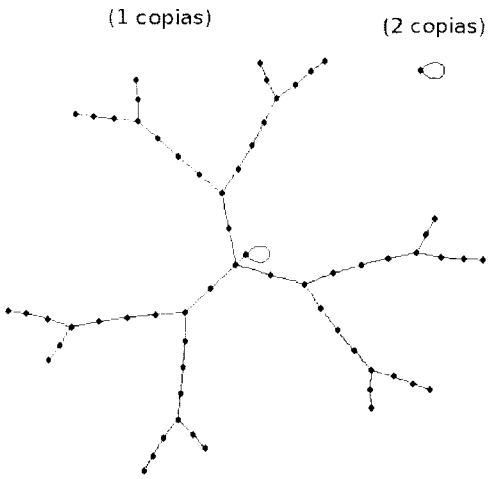


Figura G.3: Regla 30 6 celdas

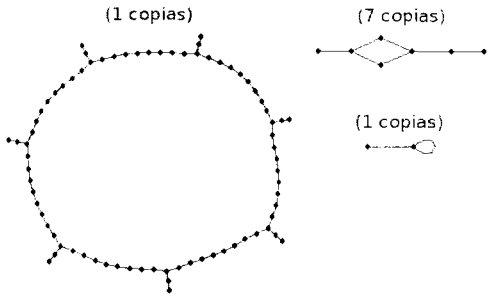


Figura G.4: Regla 30 7 celdas

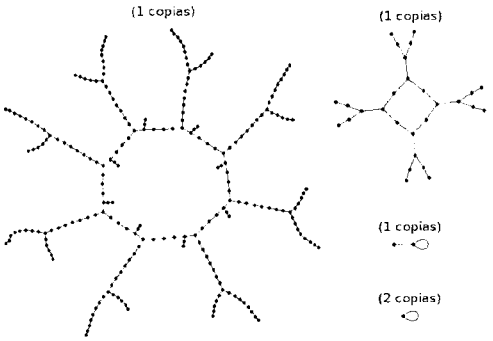


Figura G.5: Regla 30 8 celdas

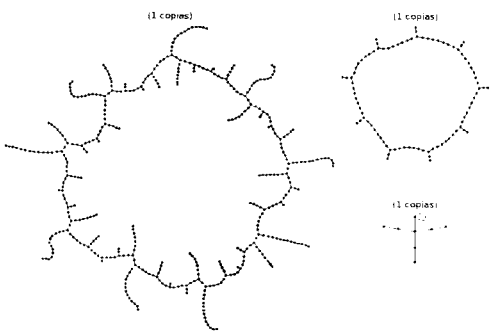


Figura G.6: Regla 30 9 celdas

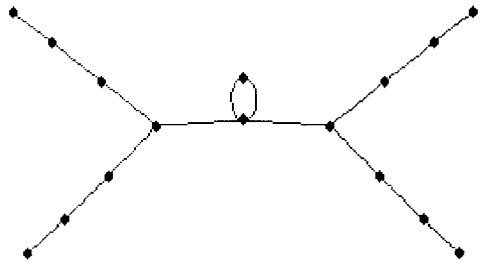


Figura G.7: Regla 45 4 celdas

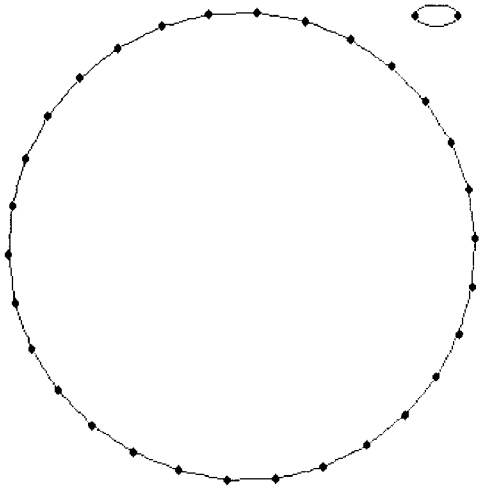


Figura G.8: Regla 45 5 celdas

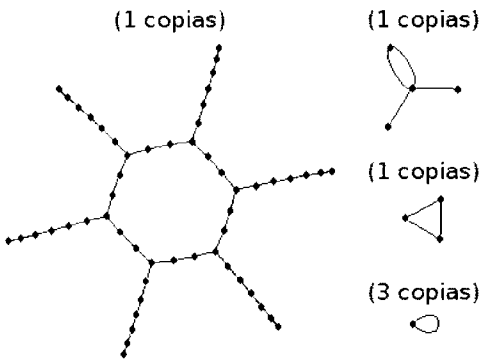


Figura G.9: Regla 45 6 celdas

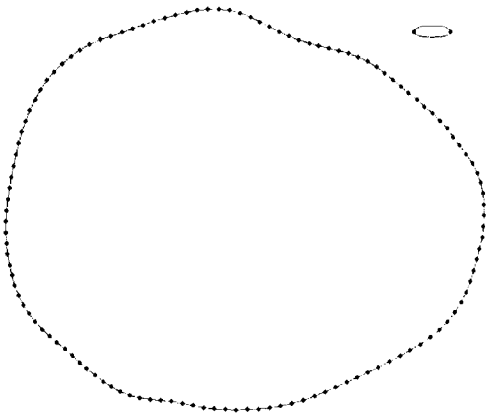


Figura G.10: Regla 45 7 celdas

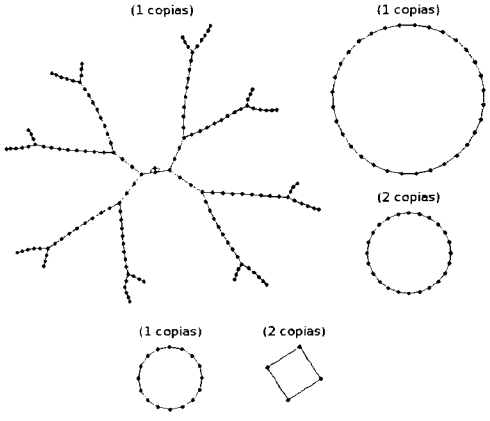


Figura G.11: Regla 45 8 celdas

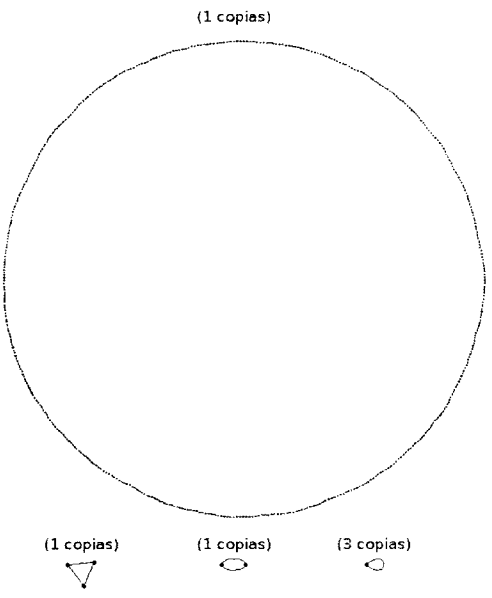


Figura G.12: Regla 45 9 celdas

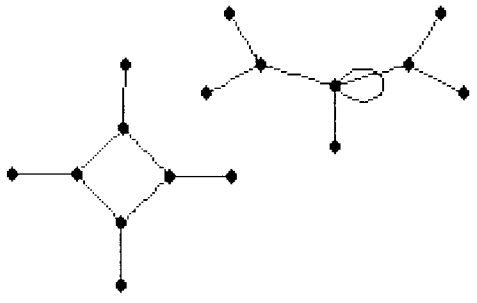


Figura G.13: Regla 45 o 30 4 celdas

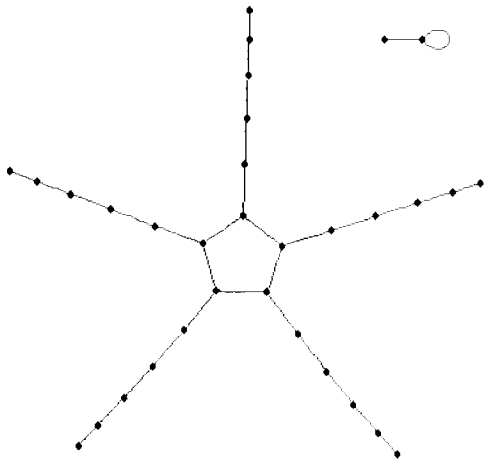


Figura G.14: Regla 45 o 30 5 celdas

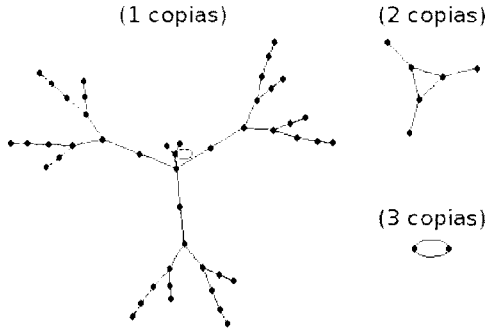


Figura G.15: Regla 45 o 30 6 celdas

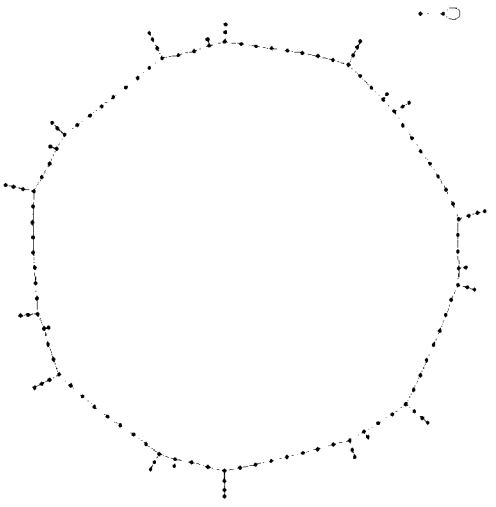


Figura G.16: Regla 45 o 30 7 celdas

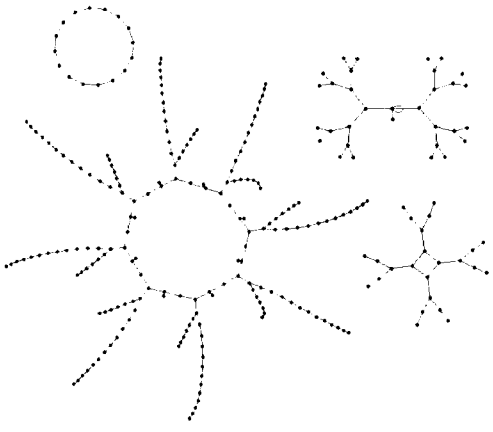


Figura G.17: Regla 45 o 30 8 celdas

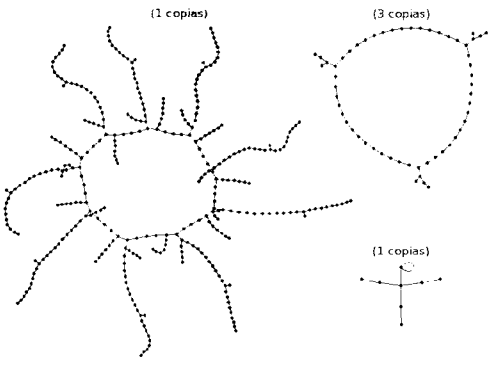


Figura G.18: Regla 45 o 30 9 celdas

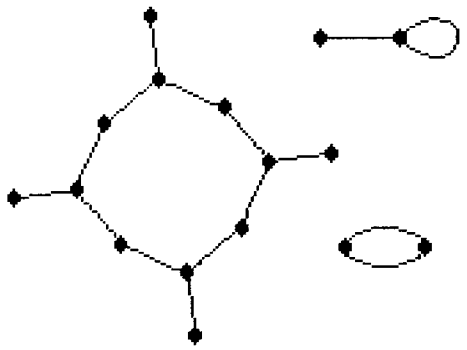


Figura G.19: Regla 105 o 30 4 celdas

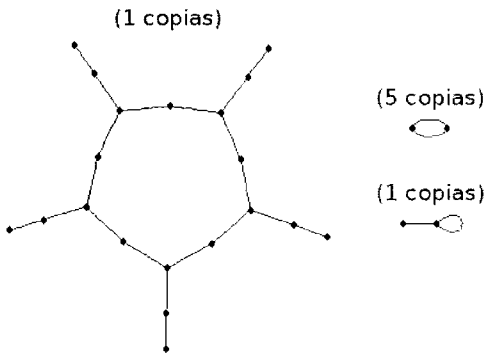


Figura G.20: Regla 105 o 30 5 celdas

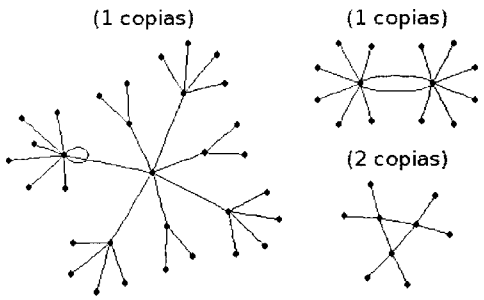


Figura G.21: Regla 105 o 30 6 celdas

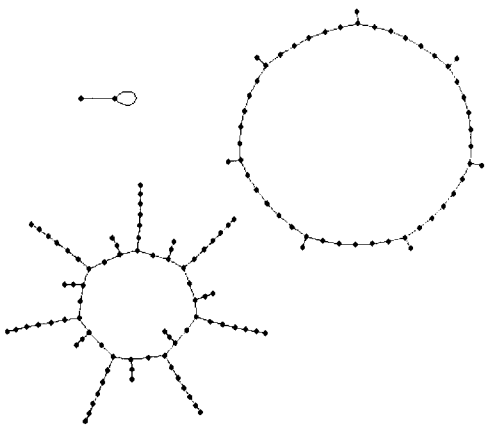


Figura G.22: Regla 105 o 30 7 celdas

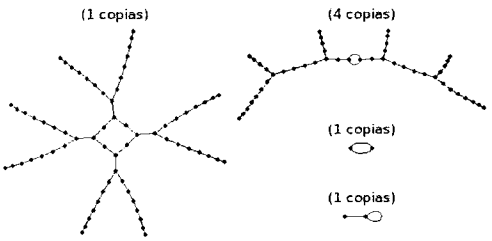


Figura G.23: Regla 105 o 30 8 celdas

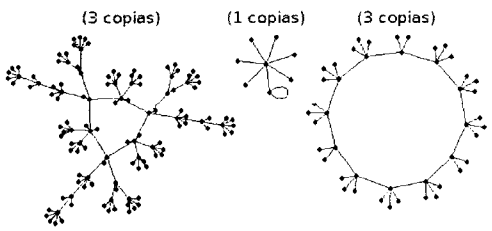


Figura G.24: Regla 105 o 30 9 celdas

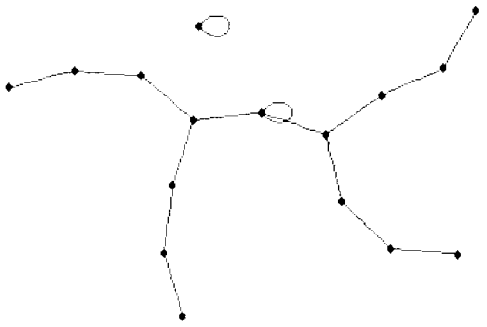


Figura G.25: Regla $105 \circ 45$ 4 celdas

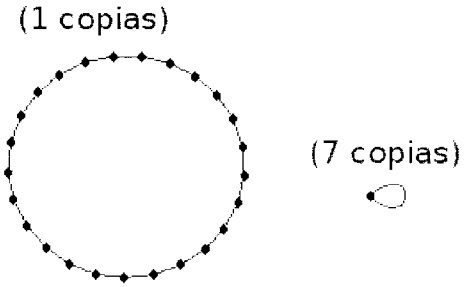


Figura G.26: Regla $105 \circ 45$ 5 celdas

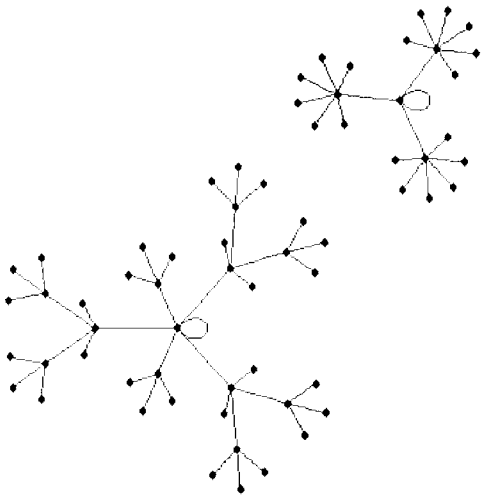


Figura G.27: Regla $105 \circ 45$ 6 celdas

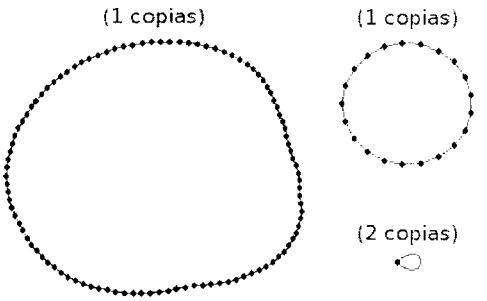


Figura G.28: Regla $105 \circ 45$ 7 celdas

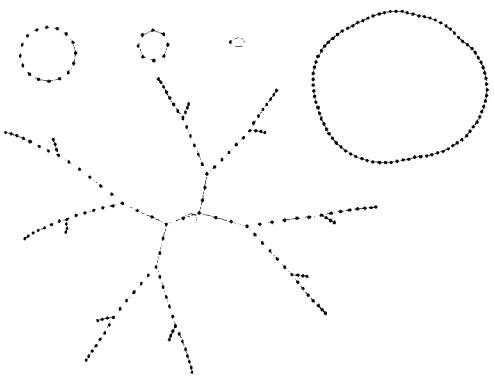


Figura G.29: Regla $105 \circ 45$ 8 celdas

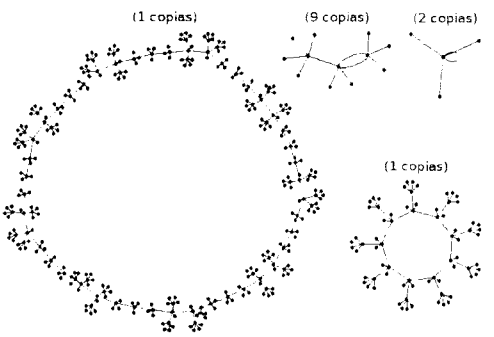


Figura G.30: Regla $105 \circ 45$ 9 celdas

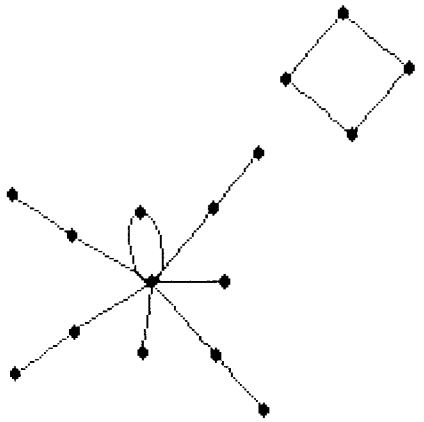


Figura G.31: Regla $166 \circ 45$ 4 celdas

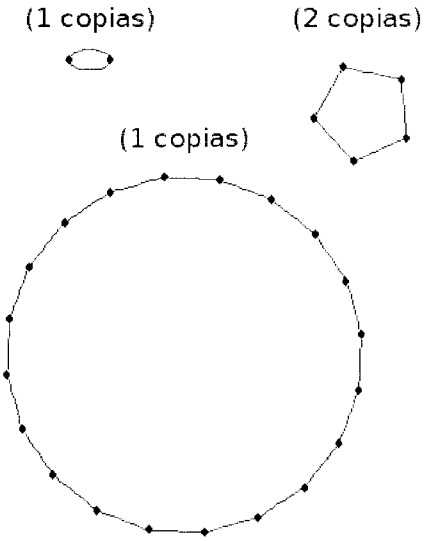


Figura G.32: Regla $166 \circ 45$ 5 celdas

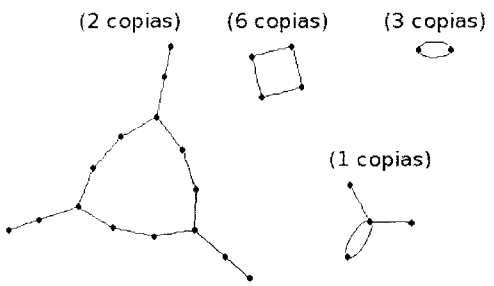


Figura G.33: Regla $166 \circ 45$ 6 celdas

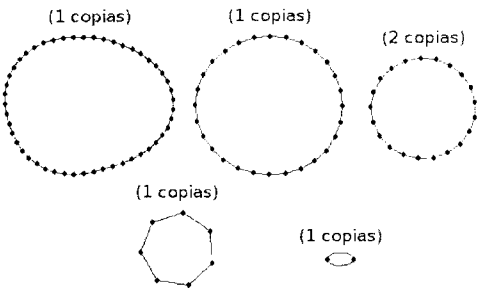


Figura G.34: Regla $166 \circ 45$ 7 celdas

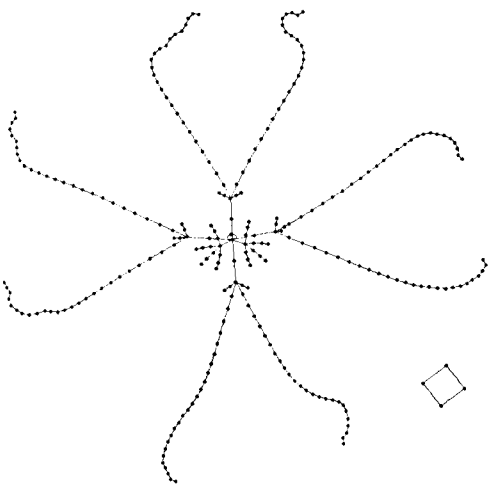


Figura G.35: Regla $166 \circ 45$ 8 celdas

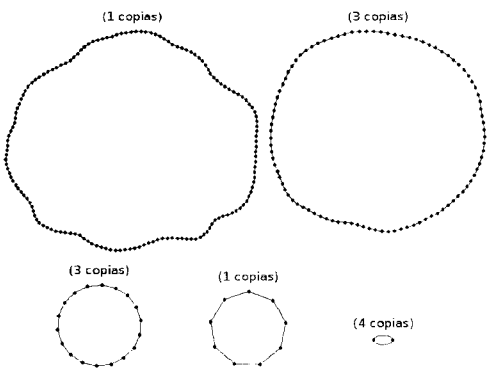


Figura G.36: Regla $166 \circ 45$ 9 celdas

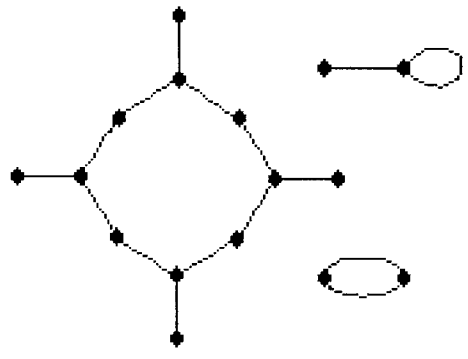


Figura G.37: Regla 106 o 85 4 celdas

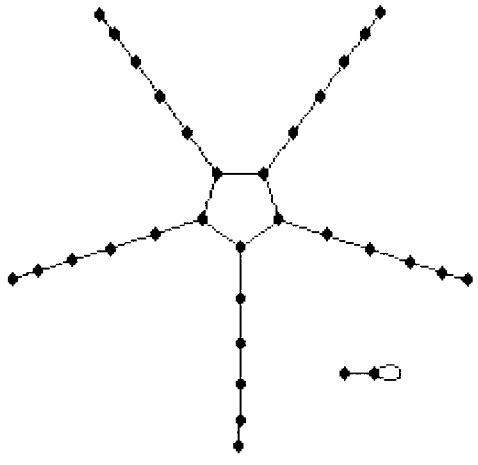


Figura G.38: Regla 106 o 85 5 celdas

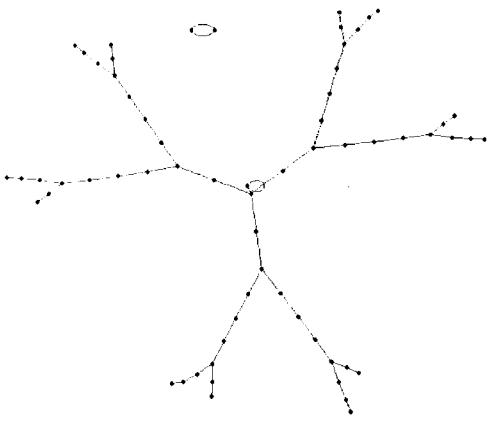


Figura G.39: Regla 106 o 85 6 celdas

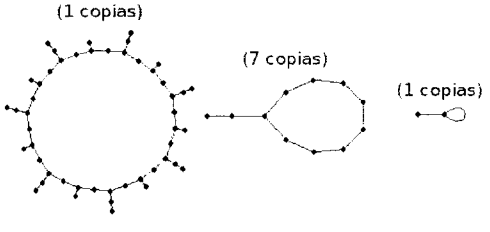


Figura G.40: Regla 106 o 85 7 celdas

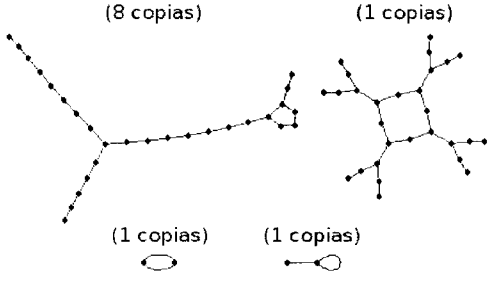


Figura G.41: Regla 106 o 85 8 celdas

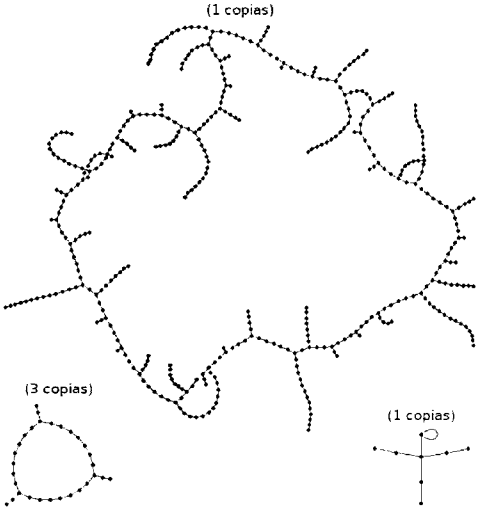
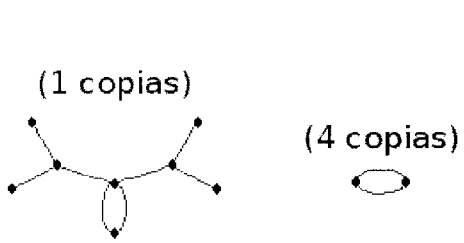


Figura G.42: Regla 106 o 85 9 celdas



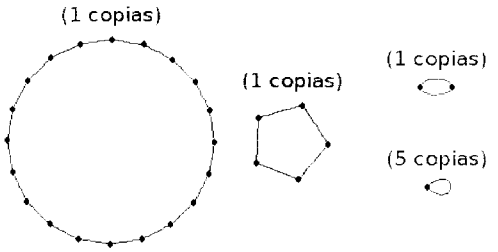
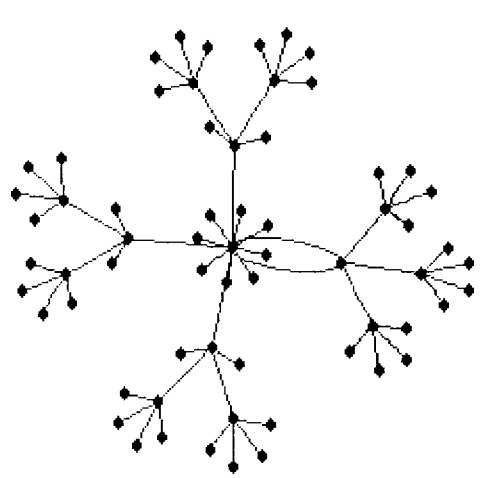


Figura G.43: Regla 154 o 105 4 celdas **Figura G.44:** Regla 154 o 105 5 celdas



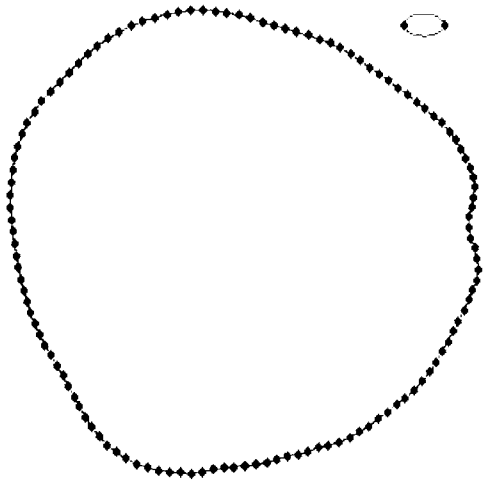
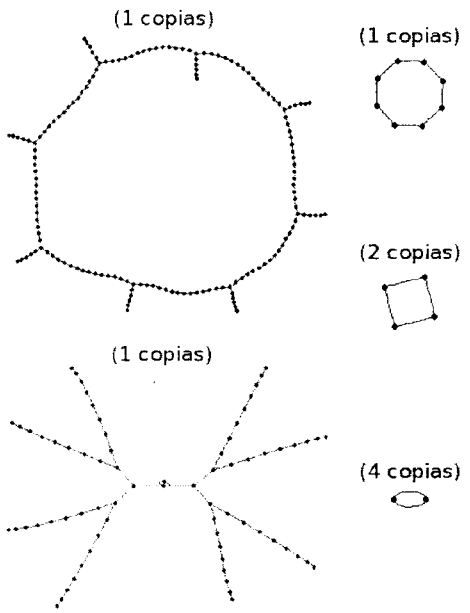


Figura G.45: Regla 154 o 105 6 celdas **Figura G.46:** Regla 154 o 105 7 celdas



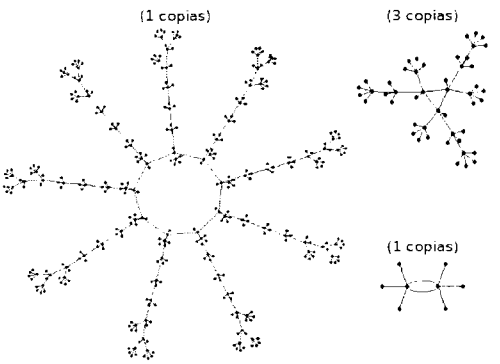


Figura G.47: Regla 154 o 105 8 celdas **Figura G.48:** Regla 154 o 105 9 celdas

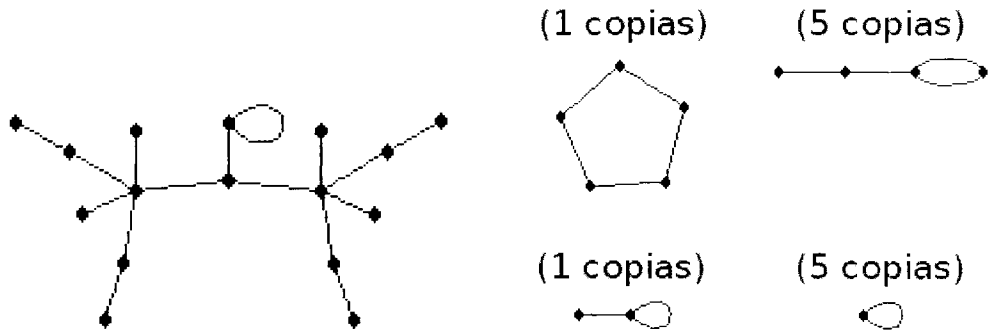


Figura G.49: Regla 166 o 106 4 celdas **Figura G.50:** Regla 166 o 106 5 celdas

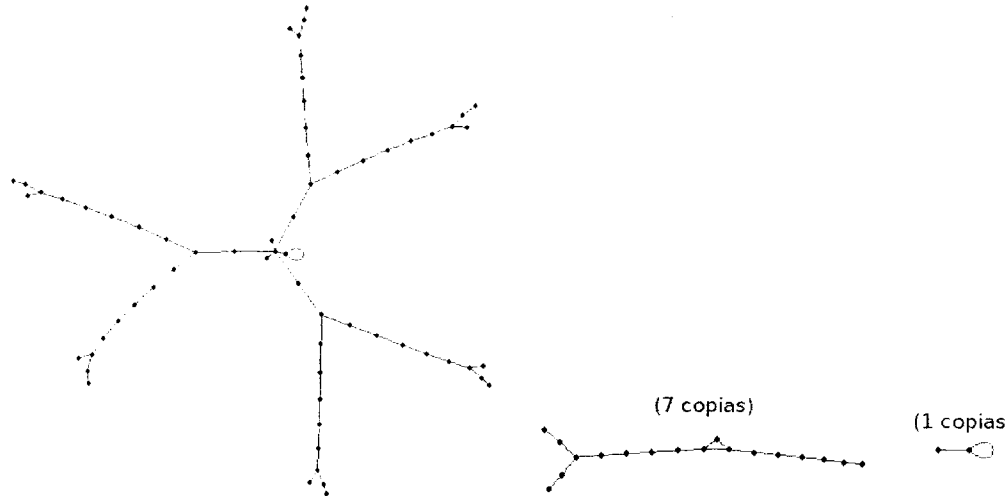


Figura G.51: Regla 166 o 106 6 celdas **Figura G.52:** Regla 166 o 106 7 celdas

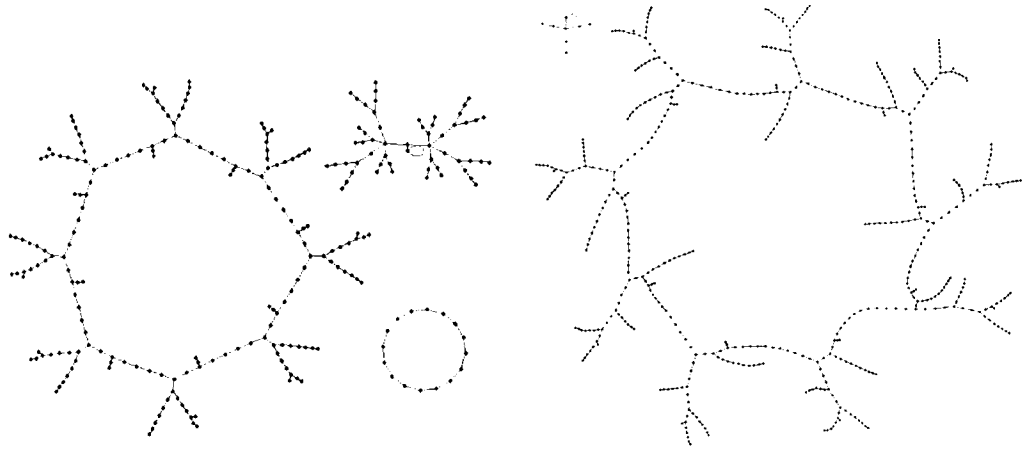


Figura G.53: Regla 166 o 106 8 celdas **Figura G.54:** Regla 166 o 106 9 celdas

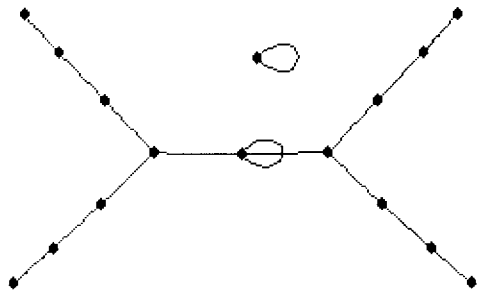


Figura G.55: Regla $154 \circ 150$ 4 celdas

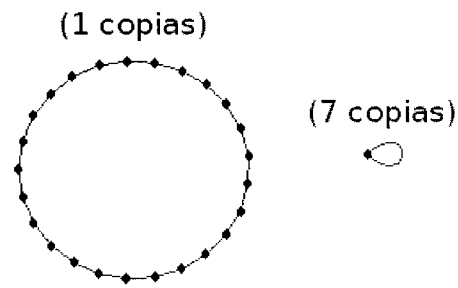


Figura G.56: Regla $154 \circ 150$ 5 celdas

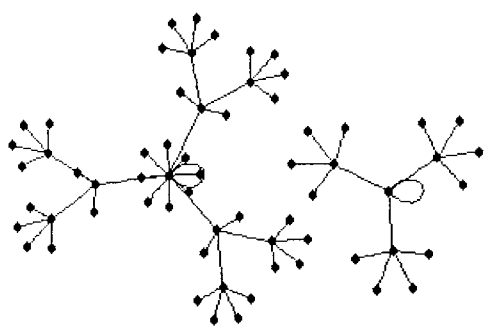


Figura G.57: Regla $154 \circ 150$ 6 celdas

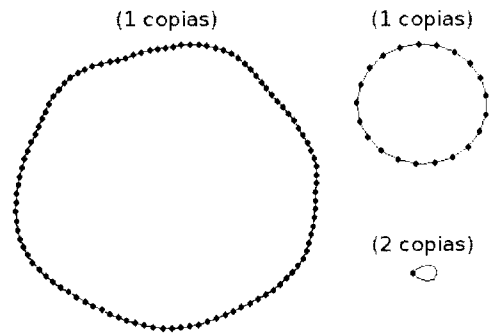


Figura G.58: Regla $154 \circ 150$ 7 celdas

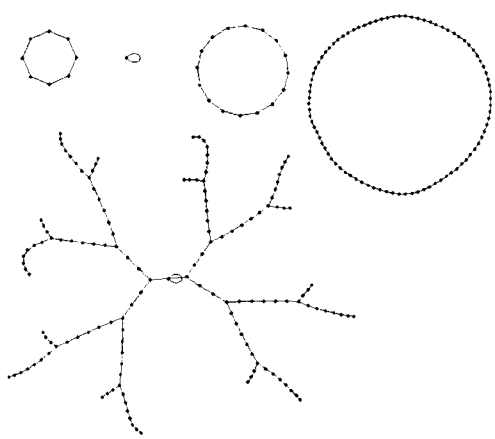


Figura G.59: Regla $154 \circ 150$ 8 celdas

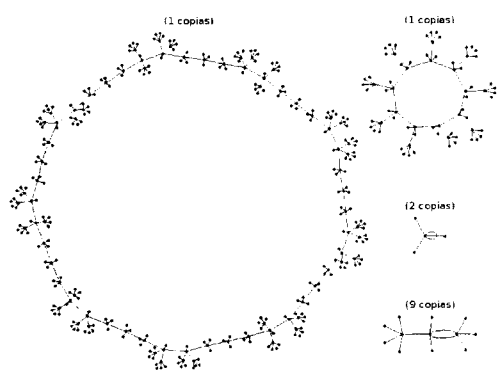


Figura G.60: Regla $154 \circ 150$ 9 celdas

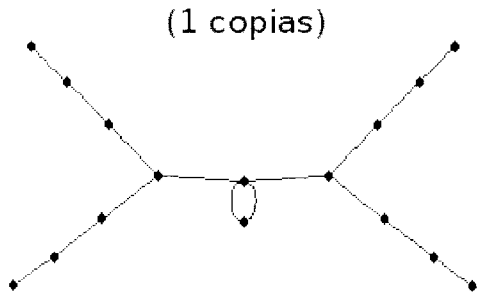


Figura G.61: Regla 154 o 15 4 celdas

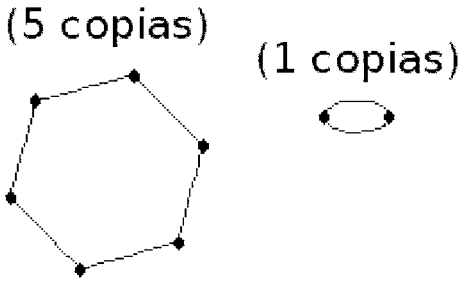


Figura G.62: Regla 151 o 15 5 celdas

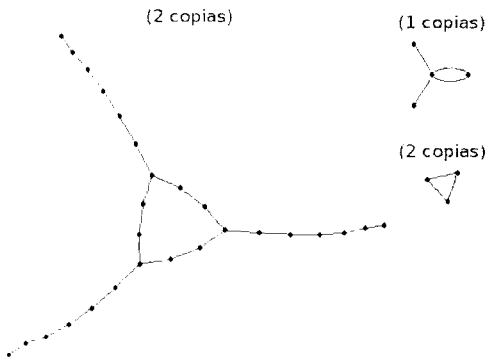


Figura G.63: Regla 154 o 15 6 celdas

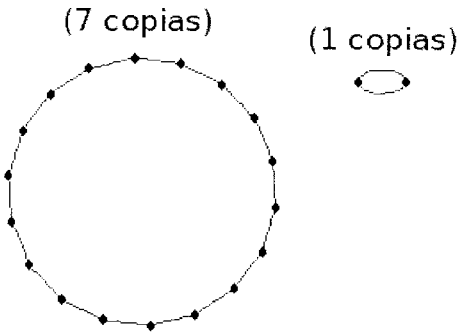


Figura G.64: Regla 154 o 15 7 celdas

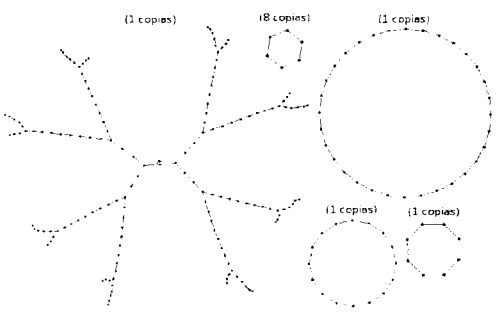


Figura G.65: Regla 154 o 15 8 celdas

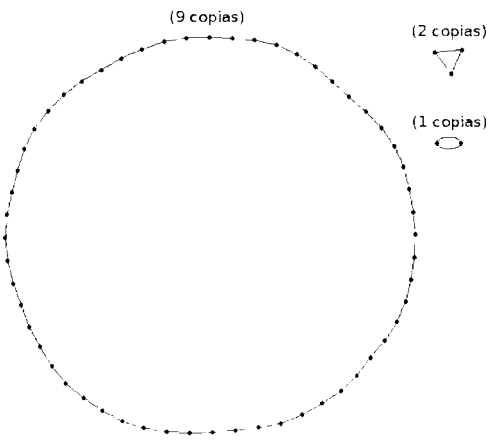


Figura G.66: Regla 154 o 15 9 celdas

Herramientas utilizadas

Para el estudio e investigación de las reglas de los autómatas celulares en los tópicos presentados en este trabajo se han empleado las siguientes herramientas:

GCC: GNU C Compiler con las bibliotecas: multiprecisión aritmética GMP (The GNU Multiple Precision Arithmetic Library: `https://gmplib.org/`) y gestión de gráficos GD Graphics Library (`http://www.boutell.com/gd/`). Empleado con los programas desarrollados por el autor en lenguaje de programación C que han servido para cálculos sobre: ciclos, algoritmo de Berlekamp-Massey en el estudio de la complejidad lineal, entropía, distancia de Hamming, diagramas de transición de estados, cálculo de los exponentes de Lyapunov (por derecha, por izquierda y máximo) y compresión de estados (Cantor set).

Python: lenguaje de programación Python con el paquete NetworkX (`https://networkx.github.io/`), para la clasificación de subgrafos isomorfos en los diagramas de transición de estados.

Graphviz: software de código abierto para la visualización de estructuras gráficas. Empleado para la creación de los diagramas de transición de estados (`http://www.graphviz.org/`).

AWK: lenguaje de programación interpretado para procesamiento y extracción

de datos (`https://www.gnu.org/software/gawk/`). Empleado para los cálculos de la media y el desvío estándar en el estudio de complejidad lineal, proporción de estados en ciclos, longitud promedio de ciclos y normalización del exponente de Lyapunov máximo.

GIMP: programa de manipulación de imágenes de GNU (`https://www.gimp.org/`). Empleado en el ajuste los diagramas de transición de estados.

GnuPlot: herramienta para creación de gráficos desde la linea de comandos (`http://www.gnuplot.info/`). Empleado en la creación de los gráficos de compresión de estados, entropía, distancia de Hamming, ciclos y complejidad lineal.

WolframAlpha: herramientas para la resolución online de problemas computacionales (`http://www.wolframalpha.com/`). Empleado en la caracterización de las reglas, en la búsqueda de las expresiones booleanas mínimas, para generar el circuito lógico equivalente y para obtener el número de operador booleano (método de Wolfram para nombrar a las reglas de evolución ver2).

C# .NET 2010: empleado para generar los diagramas espacio-temporales.

Logisim: herramienta de libre distribución de diseño y simulación de circuitos lógicos digitales. Empleado en la creación de los diagramás de circuitos y para validar los mismos (`http://www.cburch.com/logisim/`)

Bibliografía

[1] Menezes, A. J., van Oorschot, P. C., and Vanstone S. A., *Handbook of Applied Cryptography.* CRC Press, 1996.

[2] Donida Labati R. and Scotti F., *Encyclopedia of Cryptography and Security (2nd ed.).* Springer, 2011.

[3] Wolfram, S, *New Kind of Science.* Wolfram Media, 2002.

[4] Wolfram, S., *Cellular Automata And Complexity: Collected Papers.* Westview Press, 1994.

[5] Wolfram, S., "The Wolfram Atlas of Simple Programs." http://atlas.wolfram.com/ (Consultada el 22/02/2015).

[6] Wolfram, S., "Cellular Automaton." `http://mathworld.wolfram.com/CellularAutomaton.html`. (Consultada el 12/03/2016).

[7] Schiff, J. L., *Cellular Automata: A Discrete View of the World.* John Wiley and Sons, Inc., 2008.

[8] Ilachinski, A., *Cellular Automata: A Discrete Universe.* World Scientific Publishing Co. Pte. Ltd., 2001.

[9] Baetens, J. M. and De Baets, B., "Towards generalized measures grasping ca dynamics," *9th International Conference on Cellular Automata for Research and Industry*, pp. 177–187, 2010.

[10] Cvitanović, P., Artuso R., Mainieri,R., Tanner, G., and Vattay, G., "Chaos: Classical and quantum." `http://chaosbook.org/`. Niels Bohr Institute, Copenhagen 2016 (Consultada el 12/03/2016).

[11] Cook, M., "Universality in elementary cellular automata," *Complex Systems*, vol. 15-1, pp. 1–40, 2004.

[12] Agapie, A., Andreica, A., and Giuclea, M., "Probabilistic cellular automata," *Journal of Computation Biology*, vol. 21, pp. 699–708, 2014.

[13] Hartman, H. and Vichniac, G.Y., "Inhomogeneous cellular automata (inca)," *Disordered Systems and Biological Organization*, 1986.

[14] Horowitz, J., "An Introduction to Quantum Cellular Automata," 2008. http://www.mit.edu/ joshuah/projects/qca.pdf (Consultada el 22/01/2016).

[15] Yanofsky, N.S. and Mannucci, M.A., *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.

[16] Sakurai, J.J. and Napolitano, J., *Modern Quantum Mechanics, 2nd Edition*. Addison-Wesley, 2010.

[17] Nielsen, M.A. and Chuang, I.L., *Quantum Computation and Quantum Information, 10th Anniversary Edition*. Cambridge University Press, 2010.

[18] Gleick, J., *Chaos: making a new science*. Viking Press, 1987.

[19] Bagnoli, F. and Ruffo, S., "Maximal lyapunov exponent for 1d boolean cellular automata," *Cellular Automata and Cooperative Systems*, pp. 19–28, 1993.

[20] Bagnoli, F., Rechtman, R., and Ruffo, S., "Damage spreading and lyapunov exponents in cellular automata," *Physics Letters A*, vol. 172, pp. 34–38, 1992.

[21] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings," RFC 4648, RFC Editor, October 2006. (Consultada el 27 de Marzo de 2016).

[22] Hill, F. J. and Peterson, G. R, *Teoría de Conmutación y Diseño Lógico*. Limusa, 1ra ed., 1980.

[23] Cariolaro, G., *Quantum Communications (Signals and Communication Technology)*. Springer - 2015 Edition, 2015.

[24] McIntosh, H. V., *One Dimensional Cellular Automata*. Luniver Press, 2009.

[25] Stinson, D. R., *Cryptography: Theory and Practice*. Chapman and Hall/CRC, 3rd ed., 2005.

[26] Gonzalez, T., Díaz-Herrera, J., and Tucker, A., eds., *Computing Handbook: Computer Science and Software Engineering*. CRC Press, third ed., 2014.

[27] Sahoo, S. and Pal Choudhury, P., "Issues on drawing the State Transition Diagram for arbitrary Cellular Automata," *ArXiv e-prints*, Nov. 2008.