

2134



Universidad Nacional Autónoma de México

Facultad de Ciencias

AUTOMATAS CELULARES UNIDIMENSIONALES

T E S I S

Que para obtener el título de
F I S I C O
p r e s e n t a

MANUEL CRISTOBAL LOPEZ MICHELONE

México, D. F. 1988



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Índice

1. **Introducción**
2. **Autómatas Celulares Unidimensionales**
 - Clases de autómatas
 - Reglas locales de evolución
 - Notación y formalismo matemático
 - Autómatas de Wolfram. Un ejemplo
 - Reglas ilegales
3. **Propiedades Locales de los Autómatas Celulares**
 - Definiciones
4. **Propiedades Globales de los Autómatas Celulares**
5. **El Juego de la Vida**
 - Algunas configuraciones sencillas
 - Conclusiones
6. **La Imposibilidad de Demostrar la Conjetura de Collatz**
 - Introducción
 - La conjetura de Collatz
 - Una breve historia
 - Sistemas computacionales irreducibles
 - Relación de los Autómatas Celulares con la conjetura de Collatz
7. **Los Programas**
 - Graficación de un autómata celular unidimensional
 - Prueba de la conjetura de Collatz

Introducción

En 1948 Von Neumann se embarcó en un proyecto muy ambicioso: Mostrar que fenómenos tan complejos como la vida —supervivencia, reproducción, y evolución de formas de organización complejas— podían ser reducidas en principio a la dinámica de fenómenos idénticos más primitivos, capaces de interactuar y mantener su identidad. En otras palabras, Von Neumann intentó probar la posibilidad de que una máquina fuese autoreproducible. Tal máquina, dadas ciertas instrucciones, debía construir un duplicado exacto de sí misma. A su vez, estas dos máquinas debían poder autoreproducirse de la misma manera, creando un duplicado exacto de ellas mismas. Y así sucesivamente. De esta manera es como empieza la historia de la teoría de los Autómatas Celulares.

Por su parte, Stanislaw M. Ulam (amigo de Von Neumann), mostró la posibilidad de representar estas máquinas de una manera mucho más elegante y abstracta. La representación usada para la demostración de Von Neumann consiste simplemente de lo que actualmente se llama un *espacio celular uniforme*, el cual es equivalente a un gran tablero de ajedrez, en donde cada celda (o escaque del tablero, para seguir con la analogía) puede tener un número finito de *estados* o valores, incluyendo un valor indefinido (vacío), y un número finito de celdas adyacentes que pueden influir sobre el estado de otras celdas. El patrón de cambios en los estados de cada celda está dado por una regla de transición local, es decir, que afecta sólo a cierta vecindad de celdas, la cual va evolucionando a través del tiempo (en unidades discretas de éste). Esta regla se aplica a cada una de las celdas.

Las celdas simbolizan la parte básica de un autómata de estados finitos y una configuración dada de células vivas representa un modelo de la máquina autoreproducible.

En muchos casos, para hacer el modelo manejable, se considera que cada celda puede tener sólo dos posibles valores: 0 (cero) y 1 (uno). Otra restricción es la de trabajar en una sola dimensión, reduciendo un poco la complejidad de las teorías, sin embargo, estas restricciones no tienen porqué generalizarse para toda la teoría de Autómatas Celulares.

Un concepto equivalente de la máquina autoreproducible es la máquina de Turing, nombrada así por su inventor, el matemático británico A. M. Turing, la cual consiste en una máquina capaz de ejecutar cualquier cálculo. Introduciendo este concepto, Von Neumann fue capaz de producir el constructor universal.

Los autómatas celulares han sido aplicados a muchos fenómenos, entre los que se encuentran por ejemplo, teorías sobre la duplicación del ADN, el famoso *juego de la vida* inventado por John Conway,¹ el cual refleja en muchos sentidos el origen motivador de los autómatas en la biología. Así entonces, las similitudes entre el comportamiento de los autómatas celulares y muchos sistemas físicos, químicos y biológicos es altamente sugerente en sus posibles aplicaciones directas.

La intención de este trabajo es estudiar en detalle los autómatas celulares unidimensionales, dando una teoría general sobre éstos. Hecho lo anterior, proponer su aplicación inmediata a distintos fenómenos naturales.

¹ *Mathematical Games*. Martin Gardner. Scientific American, octubre 1971 a febrero 1972

Autómatas celulares unidimensionales

Los autómatas celulares en una dimensión son un modelo matemático para representar sistemas naturales dinámicos complejos, conteniendo un gran número de componentes idénticos en donde exista interacción entre ellos. Esto puede representarse como una matriz de sitios o celdas en una dimensión, cada celda con la posibilidad de tener un conjunto finito de posibles valores. Los valores en cada uno de estos sitios (o celdas) se desenvuelven sincrónicamente en pasos discretos de tiempo de acuerdo a cierta regla de transición, la cual se aplica a cada sitio de la matriz unidimensional. La regla de transición está dada por los valores previos de una vecindad de sitios alrededor del sitio que nos ocupa en ese momento. La regla de transición se aplica a cada sitio en paralelo.

Pueden verse entonces que los conceptos de espacio y tiempo en los autómatas celulares se discretizan, al igual que los valores que puede tener cada sitio se encuentra dentro de un conjunto de valores discreto. Los sistemas físicos que contienen muchos elementos discretos con interacciones locales son muy buenos candidatos para ser modelados por un autómata celular. Todo sistema físico que satisface una ecuación diferencial puede ser aproximado por un autómata celular introduciendo diferencias finitas y variables discretas.

Los autómatas celulares que se considerarán prácticamente en todo este trabajo son los autómatas en una dimensión, en donde cada sitio dentro del arreglo de éstos sólo pueden tomar dos posibles valores: 0 y 1 (base 2), y en donde una vecindad de sitios solamente pueden ser sitios que se encuentran a la izquierda y/o a la derecha

del sitio interés. A estos autómatas los llamaremos formalmente *Autómatas Celulares Elementales*.

Clases de autómatas

Existen diferentes clases de autómatas celulares, los cuales se han clasificado de acuerdo a su comportamiento. Según Wolfram² los autómatas celulares pueden clasificarse en cuatro grandes clases, de acuerdo a su nivel de predicción en su evolución.

Así entonces, los autómatas de clase 1 son aquellos en los que la evolución está determinada (con probabilidad 1), independiente de su estado base o configuración inicial. Los de clase 2 son aquellos en los que el valor de un sitio particular, a muchos pasos de tiempo está determinado por los valores iniciales de algunos sitios en una región limitada. En los autómatas de clase 3, el valor de un sitio particular depende de los valores de un número siempre creciente de sitios iniciales. En este caso, una configuración inicial dada al azar conduce a un comportamiento caótico. Por último, los de clase 4 son aquellos en los que un sitio en particular puede depender de muchos valores en la configuración inicial, y pueden aparentemente ser determinados solamente por un algoritmo, equivalente en complejidad a una simulación explícita de la evolución del autómata celular. Para estos autómatas no es posible predecir su evolución, ésta sólo puede determinarse por simulación explícita.

Reglas locales de evolución

Las reglas locales de evolución son las que originan el comportamiento de los autómatas celulares. Estas se basan en una regla general, la cual da como resultado posibles

² *Universality and Complexity in Cellular Automata*. Stephen Wolfram. *Physica* 10D (1984) 1-35. North Holland, Amsterdam

valores para los sitios en el arreglo unidimensional cuando es aplicada. Como un primer ejemplo, tomemos un arreglo de tres sitios, los cuales pueden tener como ya dijimos los valores 0 ó 1. Tendremos entonces 8 posibilidades, a saber:

000	001	010	011	100	101	110	111
↓	↓	↓	↓	↓	↓	↓	↓
0	1	0	1	1	0	1	0

FIGURA 1

donde el conjunto de reglas locales para la evolución en el tiempo están dadas en el segundo renglón de la figura 1. Las reglas locales se obtienen aplicando simultáneamente a cada sitio de la dimensión la regla general: *El valor de un sitio en un tiempo t particular está dado por la suma módulo 2 de su vecindad izquierda más su vecindad derecha.* Estas reglas locales pueden también verse como funciones booleanas de los sitios de interés en términos de sus vecindades, esto es, los sitios adyacentes.

Notación y formalismo matemático

La notación y el formalismo matemático que a continuación se describen no intenta ser un tratamiento riguroso. Considérese

$$a_i^{(t)}$$

como el valor del sitio i en un autómata celular unidimensional al tiempo t . Cada valor, en cada sitio está especificado por un entero que va de 0 a $k - 1$. Los valores de los sitios se desenvuelven por la iteración del mapeo

$$a_i^{(t)} = F[a_{i-r}^{(t-1)}, a_{i-r+1}^{(t-1)}, \dots, a_i^{(t-1)}, \dots, a_{i+r}^{(t-1)}] \quad (1)$$

donde F es una función arbitraria que especifica las reglas de transición locales del autómata. El parámetro r en esta última ecuación determina el rango de la regla: *El*

valor de un sitio dado depende de los últimos valores a en lo más $2r + 1$ sitios. La región afectada por un sitio crece a lo más en r sitios en cada dirección a cada instante de tiempo. Después de algunos pasos t de tiempo, una región de a lo más $1 + 2rt$ sitios serán afectados por un valor en un sitio dado.

Un autómata celular elemental puede entonces expresarse con k valores (dentro del conjunto definido, el cual es en muchos casos los enteros) y rango r , donde r se interpreta como el tamaño de la vecindad afectada. Por ejemplo, cuando $r = 1$ entonces sólo es afectada la vecindad inmediata, es decir, el sitio inmediatamente a la derecha y a la izquierda. Así entonces, una forma alternativa de la ecuación (1) es

$$a_i^{(t)} = f \left[\sum_{j=-r}^{j=r} \alpha_j a_{i+j}^{(t-1)} \right] \quad (2)$$

donde α_j son constantes enteras, y la función f toma un único argumento entero. Las reglas especificadas a partir de la ecuación (1) pueden ser reproducidas directamente tomando $\alpha_j = k^{r-j}$.

Definiremos las reglas *totalísticas* como aquellas en donde $\alpha_j = 1$ (en la ecuación (2)) y donde cada regla da el mismo peso a todos los sitios de la vecindad, e implican que el valor del sitio depende solamente sobre un total de todos los valores precedentes de los otros sitios.

Tenemos además que las reglas para los autómata celulares pueden ser combinadas por composición. De hecho, el conjunto de reglas de un autómata es cerrada bajo la composición, a pesar de que la composición involucra un incremento en el número de sitios de la vecindad. Por ejemplo, la composición de una regla consigo misma da patrones correspondientes a pasos de tiempos alternados de acuerdo a la regla dada. La composición de distintas reglas en general no conmuta. Sin embargo, si una composición de reglas $F_1 F_2$ genera una secuencia de configuración con período π , entonces la regla $F_2 F_1$ debe permitir una secuencia de configuración con período π . Esto, en otras palabras implica que las reglas $F_1 F_2$ y $F_2 F_1$ deben llevar a comportamientos de la misma clase.

La configuración $a_i = 0$ puede ser considerada como un estado especial *nulo* de configuración, al cual también se le llama *estado base*. El requerimiento de que esta configuración se mantenga invariante bajo la evolución del tiempo implica que

$$F[0, 0, \dots, 0] = 0 \quad (3)$$

y

$$f[0] = 0 \quad (4)$$

Por otra parte, es importante considerar las reglas simétricas para los autómatas, las cuales son

$$F[a_{i-r}, \dots, a_{i+r}] = F[a_{i+r}, \dots, a_{i-r}] \quad (5)$$

Una vez que un autómata con reglas simétricas ha desarrollado un estado simétrico, es decir, $a_{n+i} = a_{n-i}$ para alguna n y toda i , sólo puede generar estados simétricos (siempre y cuando se asuman condiciones de frontera simétricas). Cuando se cumplan las condiciones (3), (4) y (5) diremos que la regla de evolución es *legal*, y llamaremos a este tipo *autómatas de Wolfram*.³

Las ecuaciones (1) y (2) pueden ser consideradas como analogías discretas de ecuaciones diferenciales de orden a lo más $2r + 1$ en espacio y primer orden en tiempo. Los autómatas celulares de un orden mayor en tiempo pueden ser construidos permitiendo que el valor de un sitio en particular dependa de valores en una vecindad de sitios en un número s de pasos del tiempo anteriores.

La forma de la función F en la regla de evolución del tiempo (ecuación (1)) puede ser especificada por un *número de regla*

$$R_F = \sum_{a_{i-r}, a_{i+r}} F[a_{i-r}, \dots, a_{i+r}] k^{\sum_{j=-r}^r k^{r-j} a_{i+j}} \quad (6)$$

³ Debido a que Stephen Wolfram (del Instituto para estudios avanzados de Princeton en Nueva Jersey) los ha estudiado con cierta profundidad

o en otros términos, la función f en la ecuación (2) puede ser especificada de manera similar por un código numérico

$$C_f = \sum_{n=0}^{(2r+1)(k-1)} k^n f[n] \quad (7)$$

La condición (3) implica que tanto R_f y C_f son múltiplos de k .

En general, hay un total de $k^{k^{2r+1}}$ reglas posibles para un autómata celular de la forma (1) ó (2). De estas reglas, sólo son legales $k^{k^{r+1}(k^r+1)/2-1}$.

Diffícilmente el comportamiento de un autómata celular puede ser deducido de propiedades de su regla de evolución, sin embargo, algunos resultados deben ser claros:

1. Una condición necesaria (pero no suficiente) para que una regla termine en un estado de crecimiento no acotado es

$$F[a_{i-r}, a_{i-r+1}, \dots, a_{i-1}, 0, 0, \dots, 0] \neq 0, \\ F[0, \dots, 0, a_{i+1}, \dots, a_{i+r}] \neq 0 \quad (8)$$

para algún conjunto de a_j . Si estas condiciones no se satisfacen, entonces las regiones conteniendo sitios distintos de cero rodeadas por sitios de valor cero nunca pueden crecer, y el autómata celular debe exhibir un comportamiento de clase 1 ó

2. Para las reglas totalísticas, la condición (8) se convierte en

$$f[n] \neq 0 \quad (9)$$

para alguna $n < r$.

2. Las reglas totalísticas para las cuales

$$f[n_1] \geq f[n_2] \quad (10)$$

para toda $n_1 > n_2$ exhiben un comportamiento de *no inhibición* al crecimiento y deben en consecuencia ser de clase 1 ó 2.

Los autómatas celulares discutidos aquí consisten en N sitios acomodados en un arreglo lineal, el cual puede ser circular (lo cual daría condiciones periódicas de frontera). Tales autómatas tienen un número finito de estados máximo de k^N . Después de un paso del tiempo suficientemente grande (aunque siempre menor que k^N), cualquier autómata finito debe entrar a un ciclo, en el cual, una secuencia de la configuración se repite. Estos ciclos representan *atractores* para la evolución del autómata celular.

Autómatas de Wolfram. Un ejemplo

Consideremos de nuevo la figura 1, la cual muestra el conjunto de reglas locales que se aplica a un autómata de Wolfram unidimensional de acuerdo con la evolución del tiempo. De acuerdo con el formalismo introducido, estamos usando la regla 90, (la cual es simplemente tomar el valor binario de los 8 unos en el segundo renglón de la figura 1 y convertir a su valor decimal). Debido a la regla general usada para encontrar el conjunto de reglas locales del autómata, tenemos que hay $2^8 = 256$ posibles reglas para una vecindad de tres sitios. Ya que las condiciones para que una regla sea *legal* deben cumplir con las ecuaciones (3), (4) y (5), entonces las reglas que rompen la simetría en el tratamiento de sitios de ceros y unos están prohibidas, al igual que las reglas que alteran el estado nulo, por lo que las que terminan en 1 están desechadas *per se*.⁴ Así pues, sólo son válidas las reglas de la forma $\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5\alpha_6\alpha_7\alpha_8$. Como se dijo antes, las reglas locales pueden ser expresadas como una función booleana de los sitios dentro de la vecindad. Sea entonces $s_n(m)$ el valor del sitio m al paso del tiempo n . Considérese la regla 90 (la de la figura 1). De acuerdo a esta regla, el valor de un sitio está dado por los dos valores de los sitios vecinos en el paso del tiempo previo. El equivalente booleano de esta regla es

$$s_{n+1}(m) = s_n(m-1) \oplus s_n(m+1) \quad (11)$$

o esquemáticamente,

$$s_+ = s^- \oplus s^+$$

⁴ Nótese que en el valor del ejemplo, 000 sólo puede dar como resultado un cero, sino se viola la ecuación (4).

donde \oplus denota la suma *módulo 2*. De esta manera, la regla 18 (00010010) es equivalente a $s_+ = s \vee s^- \oplus s^+$ (donde s denota $s_n(m)$). Estas representaciones como funciones booleanas tiene la ventaja de que es conveniente cuando se necesita implementarlas en una computadora digital.

Algunas de las reglas legales exhiben una importante característica de una *superposición aditiva* o simplemente *aditividad*. La evolución de acuerdo a estas reglas satisfacen el *principio de superposición*, tan conocido y usado en muchos fenómenos físicos. Este principio puede expresarse equivalentemente como

$$s_0 = t_0 \oplus u_0 \Leftrightarrow s_n = t_n \oplus u_n \quad (12)$$

Sin embargo, nótese que esta aditividad no implica linealidad en el mismo sentido que en los números, ya que aquí se está considerando un campo finito de valores enteros. De todo esto, podemos ver que el autómata celular satisface el principio de superposición si y sólo si la regla es de la forma $\alpha_1\alpha_20\alpha_3\alpha_2\alpha_1\alpha_30$ con $\alpha_3 = \alpha_1 \oplus \alpha_2$. Las que cumplen con esta condición son las reglas 0 (00000000), 90 (01011010), 150 (10010110) y 204 (11011010). Es evidente que las reglas 204 y 0 son casos triviales (en el sentido matemático). La regla 0 borra cualquier configuración inicial dada, mientras que la regla 204 mantiene su configuración sin cambios a través del tiempo. Por su parte, las reglas 150 y 90 son similares en su comportamiento.

La representación booleana de algunas reglas locales las refleja como *periféricas*, en el sentido que los valores de un sitio en particular depende de los valores de sus vecindades más cercanas en el paso de tiempo anterior, pero no en el valor mismo del sitio de interés. Las reglas 0, 90, 160 y 250 son de la forma $\alpha_1\alpha_2\alpha_1\alpha_20\alpha_20$ y exhiben esta propiedad.

Habiendo analizado el comportamiento de las reglas legales (32 en este caso), podemos ver la evolución de todas las reglas válidas para una configuración inicial de un solo sitio con valor 1 (véase figura 2), donde esto es análogo al crecimiento de un *crystal* dada una *semilla* microscópica. La evolución crece de arriba hacia abajo y está dada para a lo más 198 generaciones. Puede verse que cierto tipo de comportamiento es evidente. En una clase, la semilla inicial es borrada en la generación siguiente (como

en las reglas 0 y 160), o bien se mantiene sin cambio (como en las reglas 4 y 36). Las reglas de esta clase se distinguen por la presencia de reglas locales $100 \rightarrow 0$ y $001 \rightarrow 0$, las cuales previenen cualquier propagación de la semilla inicial 1. Una segunda clase de reglas (por ejemplo, la regla 50 ó 122) copian la semilla inicial y generan una estructura uniforme, la cual se expande un sitio en cada dirección a cada paso del tiempo. Estas dos clases de reglas se denominan *simples*. Un tercer tipo de reglas, denominado *complejas* (ejemplificadas por las reglas locales 18, 22 y 90), generan patrones no triviales.

Reglas Ilegales

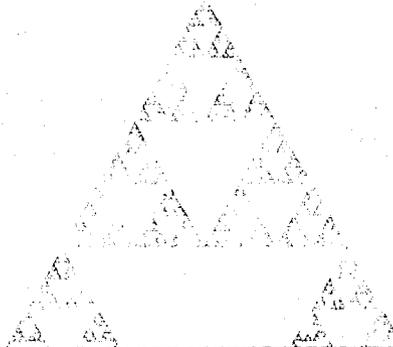
Hasta ahora hemos discutido la evolución de los autómatas de Wolfram para las reglas locales legales.⁵ Sin embargo, no hemos dicho nada de las llamadas *ilegales*. Estas reglas también pueden ser analizadas de manera análoga a las reglas legales. Por ejemplo, cuando el criterio de simetría es violado, la evolución del autómata tiende a un corrimiento hacia la izquierda o derecha de generación en generación (de acuerdo a la regla usada). También es posible encontrar patrones similares a los de las reglas legales, como en el caso de la regla 30 (00011110). En esencia, se puede decir que el estudio de las reglas ilegales no introduce ninguna característica nueva o no encontrada en el estudio de las reglas legales.

⁵ Recuérdese que una regla se considera ilegal si viola el principio de reflexión (o de simetría), el cual requiere valores idénticos para 100 y 001 y para 110 y 011, o si violan el principio que requiere que el estado inicial conteniendo sólo 0 se mantenga sin cambio.

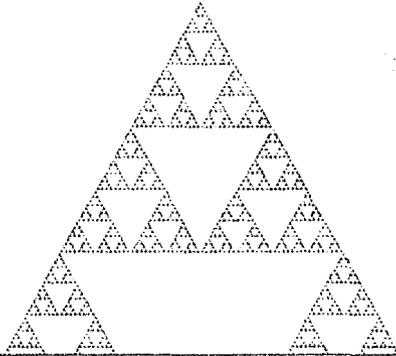
Regla: 0000000 (0)

Regla: 0000100 (4)

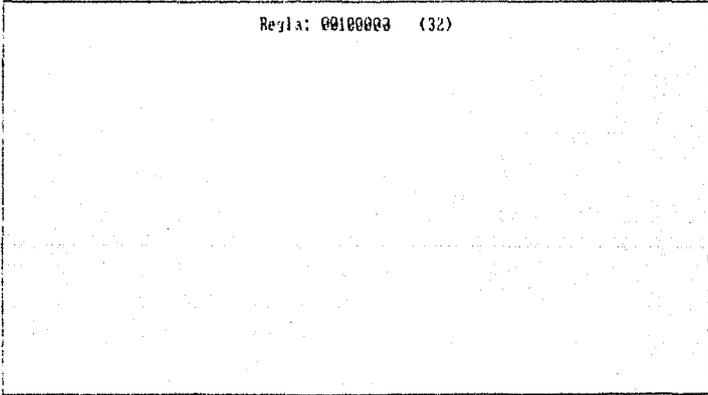
Regla: 00010010 (10)



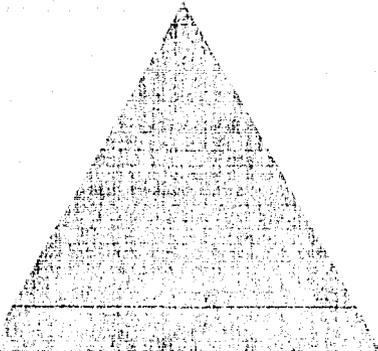
Regla: 000!0!10 (22)



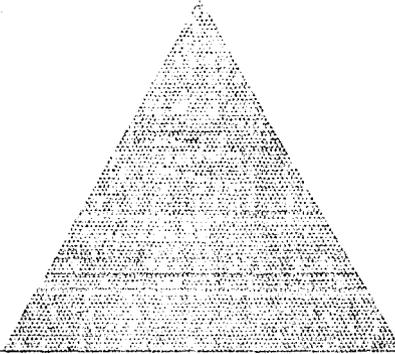
Regla: 00100000 (32)



Regla: 00110010 (50)



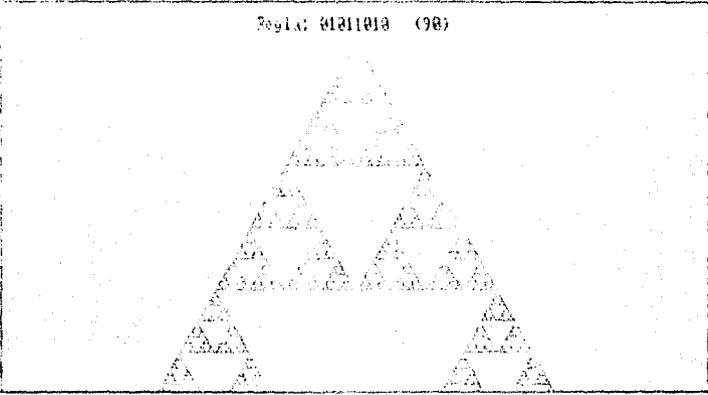
Regla: 0010110 (54)



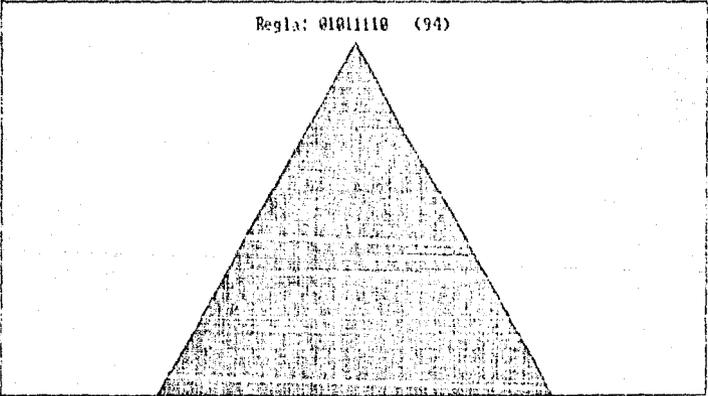
Regla: 01001000 (72)

Regla: 01001100 (76)

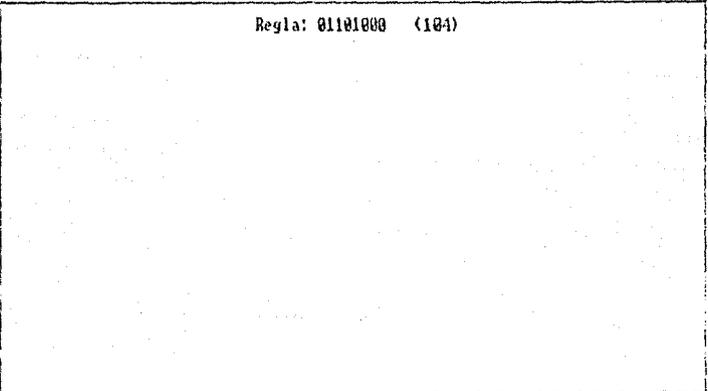
Regla: 01011010 (90)



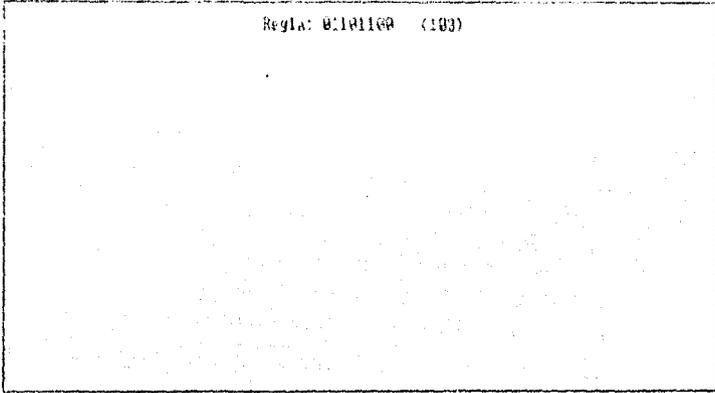
Regla: 01011110 (94)



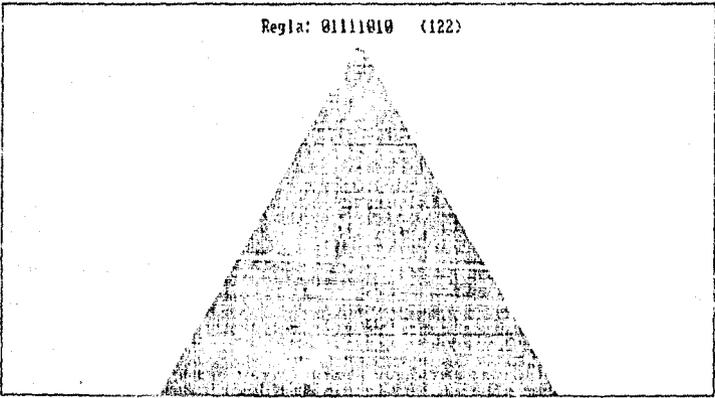
Regla: 01101000 (104)



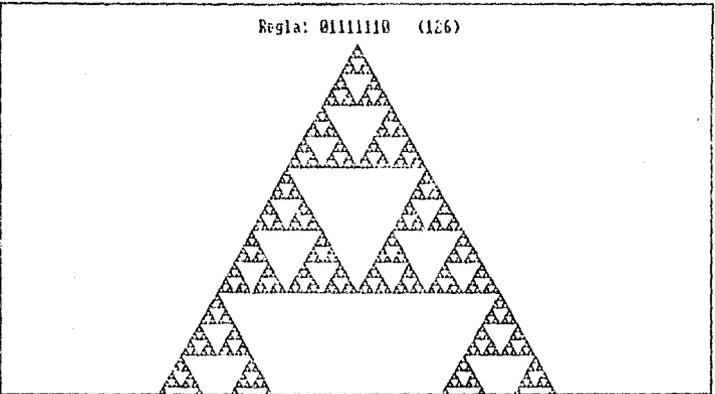
Regla: 01101100 (100)



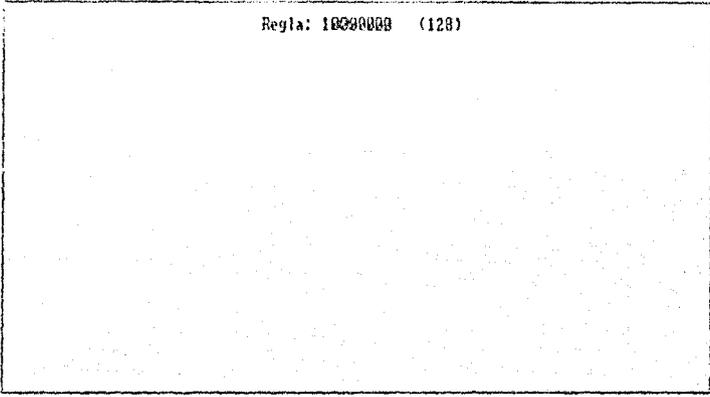
Regla: 01111010 (122)



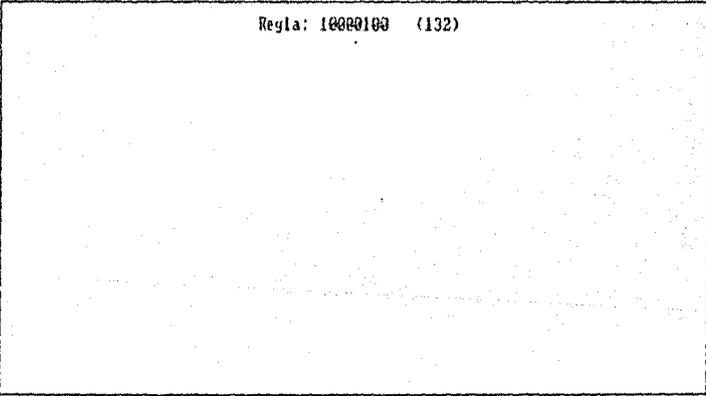
Regla: 01111110 (126)



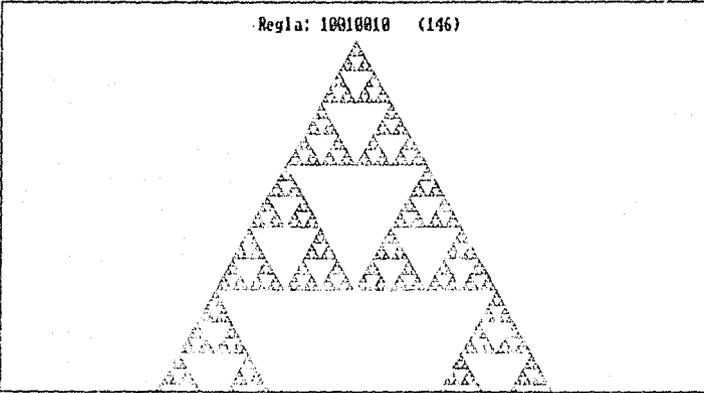
Regla: 10000000 (128)



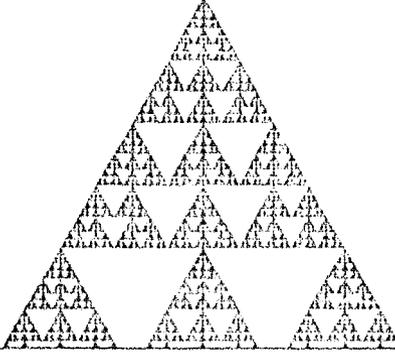
Regla: 10000100 (132)



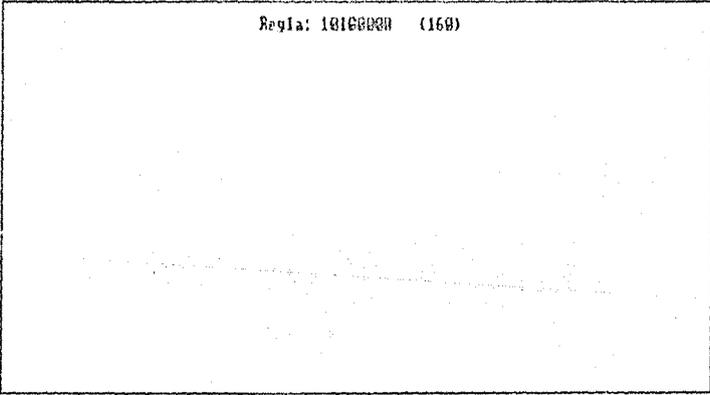
Regla: 10010010 (146)



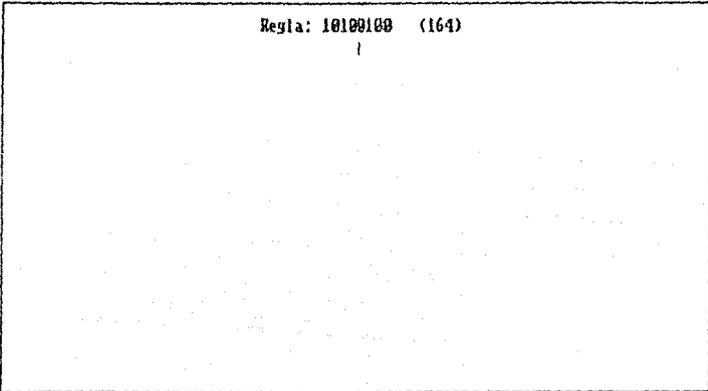
Regla: 10010110 (150)



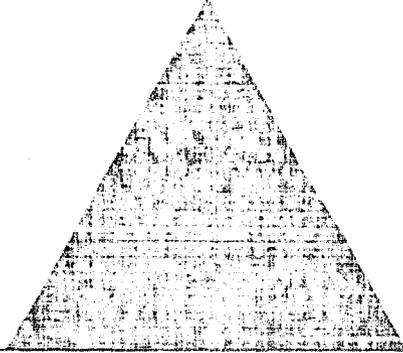
Regla: 10100000 (160)



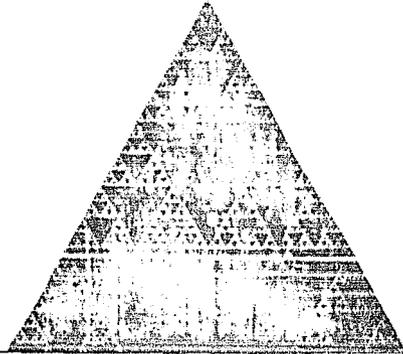
Regla: 10100100 (164)



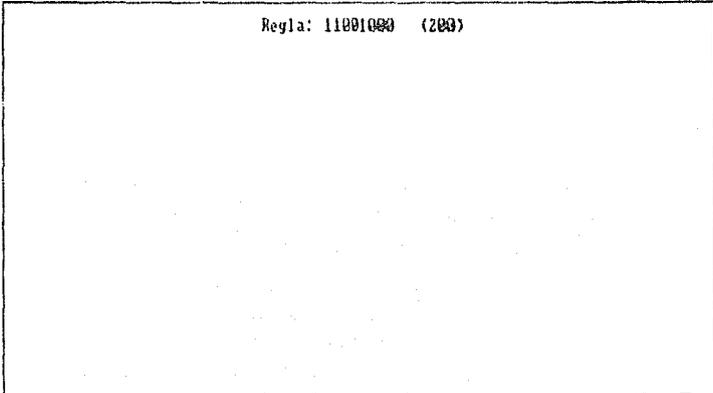
Regla: 10010010 (173)



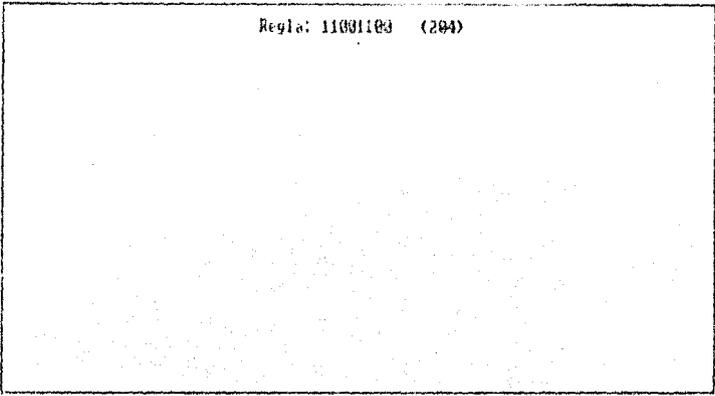
Regla: 10010010 (162)



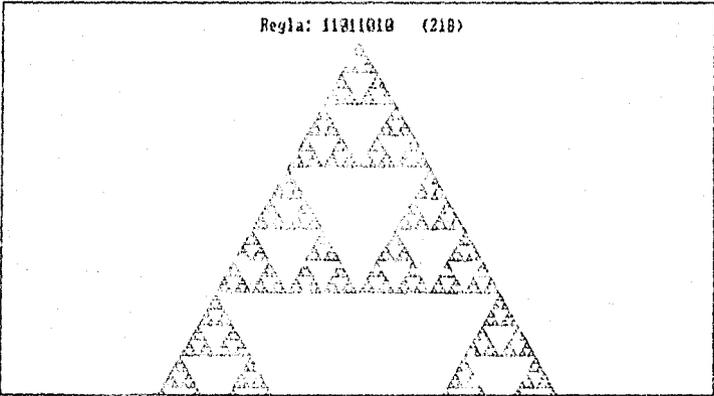
Regla: 11001000 (200)



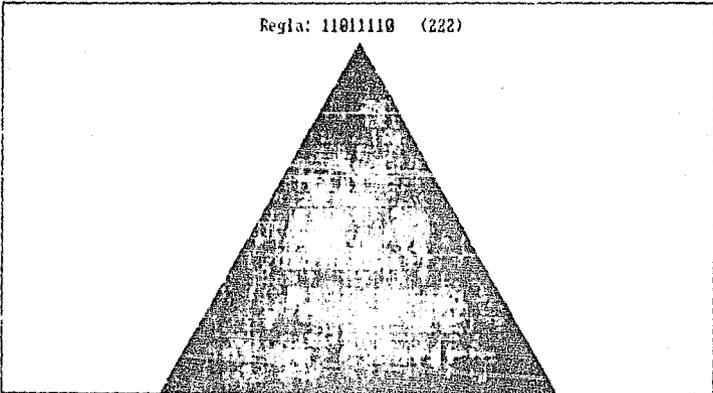
Regla: 11001100 (204)



Regla: 11011010 (218)



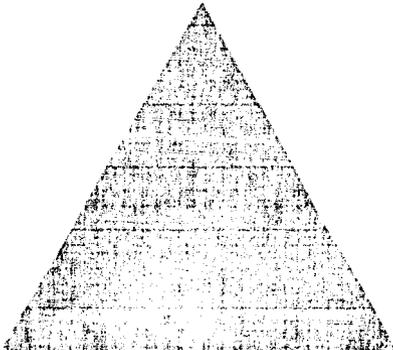
Regla: 11011110 (222)



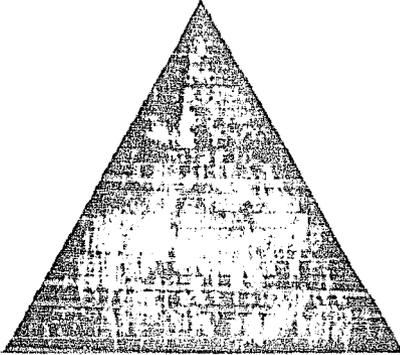
Regla: 11101000 (232)

Regla: 11101100 (236)

Regla: 11111010 (250)



Regla: 11111110 (254)



Capítulo 3

Propiedades Locales de los Automatas Celulares

En este capítulo examinaremos el análisis estadístico de las configuraciones generadas por la evolución del tiempo de los autómatas celulares elementales (ilustrados en las gráficas que acompañan el capítulo anterior). Aquí serán consideradas las propiedades estadísticas individuales de dichas configuraciones. El propósito principal es obtener una caracterización cuantitativa de la *auto-organización* evidente de tales autómatas.

Definiciones

Una configuración se dice que es o está desordenada, (también llamada esencialmente *aleatoria*), si los valores en diferentes sitios están estadísticamente no correlacionados. O dicho con otras palabras, su comportamiento es análogo a *variables aleatorias independientes*. Tales configuraciones representan una forma discreta de *ruido blanco*. Las desviaciones de las mediciones estadísticas para las configuraciones de autómatas celulares de sus valores correspondientes a configuraciones desordenada indican cierto orden y señalan la presencia de correlaciones entre distintos valores en diferentes sitios. Una configuración desordenada (infinita) está especificada por un solo parámetro: la probabilidad independiente p para cada sitio con valor 1. La descripción de una configuración ordenada requiere de más parámetros. En otras palabras, es posible dar una configuración inicial para un autómata celular definiendo el parámetro p , esto es, definiendo la probabilidad de que ocurra el valor 1 en la configuración inicial de dicho autómata. Las figuras al final del capítulo señalan tres ejemplos de configuraciones:

Con probabilidad (a) $p = 0.25$, (b) $p = 0.50$ y (c) $p = 0.75$. Si se comparan las configuraciones obtenidas por evoluciones aleatorias con distintas probabilidades p podrá verse una fuerte tendencia de los autómatas celulares a una configuración más ordenada, exhibiendo una forma muy simple de auto-organización.

La cantidad estadística más simple con la que podemos caracterizar a una configuración de un autómata celular es su fracción promedio (*densidad*) de sitios con valor 1, denotado por ρ . Por ejemplo, para una configuración desordenada, ρ está dada por la probabilidad independiente p para cada sitio con valor 1.

Consideremos primeramente la densidad ρ_1 obtenida de una configuración desordenada para la evolución de un autómata celular para un paso del tiempo. Cuando $p = \rho = 0.5$, una configuración desordenada contiene los ocho posibles valores para una vecindad de tres sitios. Esto es, todas las posibles combinaciones de tres sitios, desde 111 hasta 000 están representadas en la configuraciones con igual probabilidad. Aplicando una regla binaria R al autómata, (como por ejemplo la regla 90) a este estado inicial para un paso del tiempo

$$\tau = 1$$

nos debe llevar a una configuración en la cual la fracción de sitios con valores 1 esté dada por una fracción de las ocho posibles respuestas (de acuerdo a la regla), que den como resultado el valor 1. Esta fracción está dada por:

$$\rho_1 = \#_1(R) / (\#_0(R) + \#_1(R)) = \#_1(R) / 8 \quad (1)$$

donde $\#_d(S)$ denota el número de ocurrencias del dígito d en la representación binaria de S . Aquí por ejemplo, $\#_1(10110110) = \#_1(182) = 5$ y $\#_0(10110110) = \#_0(182) = 3$. Usando la regla 182, la densidad ρ después del primer paso del tiempo es $5/8$ si se incluye un número infinito de sitios. El resultado de la ecuación 1 puede ser generalizado a un estado inicial con $p \neq 1/2$ usando las probabilidades $p(\sigma) = p^{\#_1(\sigma)}(1-p)^{\#_0(\sigma)}$, para cada uno de las posibles vecindades σ (como por ejemplo 110), mostrado en la regla 90, y añadiendo las probabilidades para aquellas en donde σ dé el valor 1 en la aplicación de la regla del autómata celular. Considérese el comportamiento de la densidad ρ_τ obtenida después de τ pasos de tiempo para el límite cuando τ es muy

grande. Cuando $\tau > 1$, la correlación inducida por la evolución del autómata celular invalida la aproximación usada en la ecuación 1.

Las gráficas encontradas sugieren que con algunas reglas sencillas (tales como 0, 32 o 72), cualquier configuración inicial se desarrolla en última instancia al estado nulo ($\rho = 0$). Para la regla 0 es claro que $\rho = 0$ para toda $\tau > 0$. De la misma manera, para la regla 72, $\rho = 0$ para $\tau > 1$. Para la regla 32, la probabilidad de que un sitio particular distinto de cero sobreviva para τ pasos del tiempo dada una configuración inicial desordenada con $\rho = 1/2$ es de probabilidad $(1 - \rho_0)^{2^{-3(2^{\tau+1})}}$. La regla 254 da $\rho_\infty = 1$, con probabilidad $(1 - \rho_0)^{2^{\tau+1}}$ para una longitud $\geq \tau$. La regla 204 es en sí la *identidad*, la cual propaga cualquier configuración inicial sin cambio y por tanto $\rho_\infty = \rho_0$. El principio de *superposición disyuntiva* para la regla 250 (discutida en el capítulo pasado) implica que $\rho_\infty = 1$. Para la regla 50, el principio de *superposición conjuntiva* da por tanto $\rho_\infty = 1/2$.

Otras reglas *simples* sirven como *filtros* para secuencias específicas, dando densidades finales proporcionales a la densidad inicial de la secuencia a ser seleccionada. Para la regla 4, la densidad final es igual a la densidad inicial de las secuencias 101, de tal manera que $\rho_\infty = \rho_0^2(1 - \rho_0)$. Para la regla 36, ρ_∞ está determinada por la densidad inicial de secuencias de 00100 y ...1010101..., y es aproximadamente 1/16 para $\rho_0 = 1/2$. Resultados exactos para el comportamiento de ρ_τ dada por la regla 90, módulo 2 puede ser derivadas usando la propiedad aditiva del principio de superposición discutida anteriormente.

Considérese ahora el número de sitios $N_r^{(1)}$ con valor 1 obtenido por la evolución de acuerdo a la regla 90 desde un estado inicial conteniendo un solo sitio con valor 1, (como se ilustra en las figuras del capítulo pasado). Consideraciones geométricas y otras fuentes⁶ dan como resultado

$$N_r^{(1)} = 2^{\#_1(\tau)} \quad (2)$$

donde la función $\#_1(\tau)$ dados el número de ocurrencias del dígito en la representación binaria del entero τ , como se definió en la ecuación anterior. Esta ecuación muestra

⁶ Roberts, J. B., 1957, On Binomial Coefficient Residues, Can. J. Math. 9, 363.

que el promedio de la densidad sobre una región de sitios distintos s de cero, también llamado *cono de luz* en la evolución de la regla 90, está dada por $\rho_r = N_r^{(1)}/(2r + 1)$ y no tiene a un límite definido para una r muy grande. Sin embargo⁷, la densidad de tiempo promedio

$$\bar{\rho}_T = (1/T) \sum_{r=0}^{r=T} \rho_r$$

tiende a cero como $T^{1.0923} 2^{-2T-0.42}$. Los resultados obtenidos para estados iniciales con valor 1 pueden ser obtenidos por superposición aditiva. Si la configuración inicial es una en la cual se va a alcanzar por la evolución de un solo sitio, digamos, τ_0 pasos del tiempo, en donde entonces la densidad resultante está dada por la ecuación 2 reemplazando $\tau \rightarrow (\tau - \tau_0)$. Sólo una muy pequeña fracción de las configuraciones iniciales puede ser tratada de esta forma, ya que la evolución de un solo sitio genera sólo una de las 2^k posibles configuraciones en las cuales la máxima separación entre sitios distintos de cero es k . Para configuraciones iniciales pequeñas o altamente regulares, resultados análogos a la ecuación 2 pueden ser encontrados o derivados. La ecuación implica que que después de exactamente $\tau = 2^j$ pasos de tiempo, un estado inicial conteniendo un solo sitio distinto deriva en una configuración con solamente dos sitios distintos de cero. En este punto, el valor de un sitio particular en la posición n es simplemente la suma módulo dos de los valores iniciales de sitios en las posiciones $n - \tau$ y $n + \tau$. Si empezamos de un estado de configuración desordenado, la densidad a cada paso del tiempo será de $\rho_{\tau=2^j} = 2\rho_0(1 - \rho_0)$. Así entonces, en general, el valor de un sitio al tiempo τ es una suma módulo dos de los valores iniciales de $N_r^{(1)} = 2^{\#1}(r)$ sitios, los cuales tienen como valor 1 con probabilidad ρ . Entonces la probabilidad de que la suma de los valores de los sitios sea impar⁸ es

$$\sum_{i_{\text{impar}}}^k \binom{k}{i} \rho^i (1 - \rho)^{k-i} = 1/2 [1 - (1 - 2\rho)^k].$$

Así entonces, la densidad de sitios con valor 1 obtenida por evolución para τ pasos del tiempo desde un estado inicial con densidad ρ_0 de acuerdo a la regla 90, estará dada

⁷ Según Wolfram esto es estrictamente correcto sólo para $T = 2^k$. Para $T = 2^k(1 + \delta)$, donde δ es un factor de corrección de aproximadamente $(1 + \delta^{1.0923})/(1 + \delta)^{1.0923}$, el cual cae entre 0.86 y 1, con una $\delta = 0.3$.

⁸ igual a 1 módulo dos

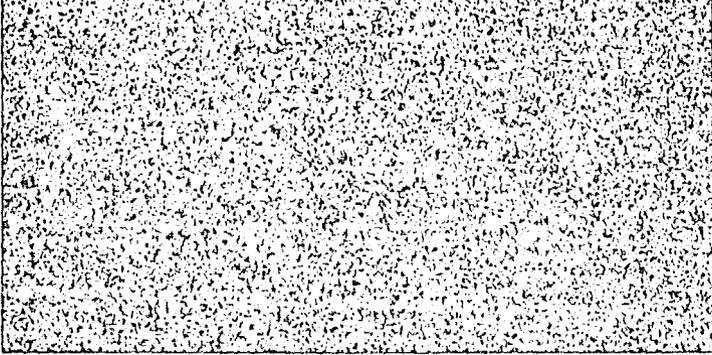
por:

$$\rho_r = 1/2[1 - (1 - 2\rho_0)^{2^r} 1^{(r)}].$$

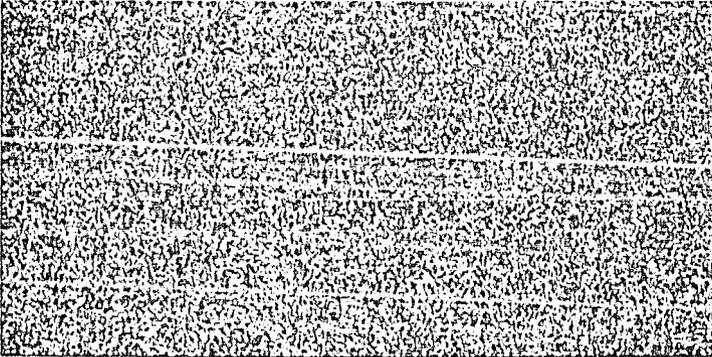
La regla 150 comparte con la regla 90 la propiedad de la superposición aditiva. Inspeccionando con detalle la regla 150 (véanse las figuras del capítulo anterior), indica que el valor de un sitio particular depende de los valores de al menos tres sitios iniciales, de tal forma que $|\rho_r - 1/2| \leq [(1 - 2\rho_0)]^{2^r}$. Algunos aspectos del comportamiento a largo plazo de un autómata puede ser encontrada analizando la correspondencia entre reglas aditivas y no aditivas. Ciertas configuraciones específicas en autómatas con reglas de evolución no aditivas terminan por evolucionar de acuerdo a reglas aditivas. Por ejemplo, tomemos la regla 18 y una configuración en la cual digamos, todos los sitios pares tienen valor cero. Al primer "pulso" del reloj, los sitios impares estarán en uno y los sitios pares con la suma módulo 2 de sus vecindades impares (en el paso previo de la evolución de la generación, sino todos serían cero), tal como en la regla de evolución 90.

Hemos visto pues, que los autómatas celulares usualmente evolucionan de acuerdo a reglas locales determinísticas muy bien definidas. En el problema de modelar sistemas físicos o biológicos a veces es mejor considerar autómatas con reglas locales que incluyen elementos probabilísticos de ruido. El procedimiento más sencillo es prescribir que a cada paso del tiempo, el valor obtenido de acuerdo a la regla determinística en cada sitio se transforme invirtiendo su valor con una probabilidad k .

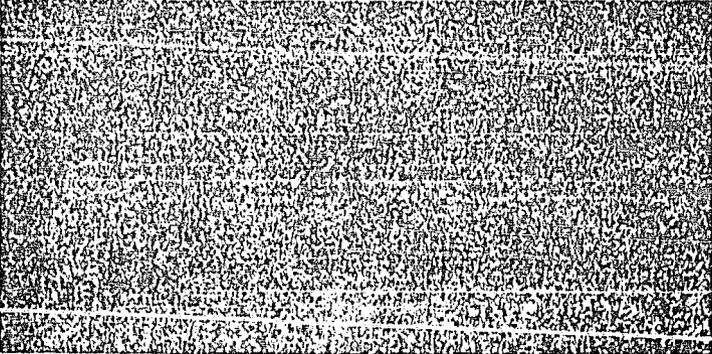
Porcentaje-25



Porcentaje-50



Porcentaje-75



Propiedades Globales de los Automatas Celulares

En la sección anterior se analizó el comportamiento de los automatas celulares considerando las propiedades estadísticas del conjunto de los valores de los sitios en una configuración típica de un automata. Aquí se intentará tomar otro camino de análisis, el cual consiste en considerar las propiedades estadísticas de un conjunto de valores —también llamado *ensemble*, para hacer una analogía con la mecánica cuántica— comprimiendo todas las posibles configuraciones de los automatas celulares. Este enfoque se conecta directamente con la teoría de los sistemas dinámicos y la teoría formal de la computación⁹, lo cual nos lleva a una visión de un fenómeno de auto-organización que complementa lo visto en la sección sobre propiedades locales de los automatas. Consideraremos aquí las reglas de evolución de un automata celular como una forma de *dinámica simbólica*, en donde los grados de libertad de un sistema son genuinamente discretos, en lugar de ser continuos.

Así entonces, considérese la imagen de un automata celular finito, esto es, conteniendo sólo un número finito N de sitios. Hay entonces 2^N posibles configuraciones de dicho automata. Cada configuración es especificada únicamente por la longitud del entero (dado en forma binaria) N , en donde los dígitos binarios corresponden —de manera biunívoca— a los valores de cada sitio en la configuración del automata. Para un automata infinito, Wolfram sugiere la correspondencia a un número real (escrito en su forma binaria), sin embargo, esta concepción no resulta la más acertada. Más bien, para un automata celular infinito, podemos pensar en la correspondencia con un

⁹ Computation: Finite and Infinite Machines, Marvin Minski, Prentice Hall, 1967

número trascendente o bien en términos de una serie infinita, de tal suerte que en el límite no converja.

La evolución de un autómata celular finito dependerá entonces de las condiciones de frontera aplicadas. Usualmente asumimos condiciones de frontera periódicas, en las cuales el primer y último sitio del autómata son identificadas como si los sitios se encontraran bajo un círculo de perímetro N . Dicho de otra manera, las condiciones de frontera periódicas definen el primero y último sitio en un arreglo de sitios que se cierra sobre si mismo, esto es, el primer sitio con el último. Uno entonces puede alternativamente tomar una secuencia infinita de sitios, pero aquellos que se encuentran fuera de la región de longitud N se toman con valor cero¹⁰. Los resultados obtenidos con estas condiciones de frontera en los sitios, no difieren cualitativamente de aquellos donde las condiciones de frontera no se dan de manera periódica, esto es, donde el primero y último sitio no se conectan entre sí. De hecho, la mayoría de los resultados obtenidos no son sensibles a la forma de las condiciones de frontera establecidas, sino más bien, al valor de N que se tome.

Básicamente, la regla de evolución de un autómata celular define una transformación de una secuencia de números binarios a otra. Las reglas proveen entonces un mapeo de un conjunto de números binarios de longitud N hacia si mismos. Para el caso trivial de la regla 0, todos los números binarios se mapean a cero¹¹.

Una medida de la distancia conveniente en el espacio de configuración de un autómata celular es la llamada *distancia de Hamming* $H(s_1, s_2)$, definida como el número de dígitos (bits) los cuales difieren entre las secuencias s_1 y s_2 . Así entonces, es su forma booleana tenemos $H(s_1, s_2) = \#_1(s_1 \oplus s_2)$. Bajo la evolución de un autómata celular, cada configuración inicial traza una trayectoria en el tiempo. Si la evolución del autómata es *estocástica*, es decir, que la relación existente entre la generación anterior no es función de la siguiente, aunque tampoco son independientes, entonces las trayectorias de puntos cercanos debe diverger exponencialmente con el tiempo. Este resultado no es trivial. Considérese primero el caso de dos configuraciones S_1 y S_2 , las cuales difieren

¹⁰ Esto no es más que hablar de una analogía con la delta de Dirac, de manera conceptual por supuesto

¹¹ Esto es $1 \rightarrow 0$ y $0 \rightarrow 0$

por un cambio en el valor de un sitio. Así entonces, estarán separados por la unidad en la distancia de Hamming. Después de r pasos del tiempo en la evolución del autómata, esta diferencia inicial podrá afectar a los valores de a lo más $2r$ sitios (entonces H será menor o igual a $2r$). Sin embargo, para las reglas simples de un autómata, las diferencias se localizan a un par de sitios, por lo que la distancia de Hamming tiende rápidamente a un valor constante pequeño. El comportamiento para reglas más complejas (con superposición aditiva, como las reglas 150 y 90) difieren radicalmente de las reglas no aditivas. Para las reglas aditivas la diferencia obtenida después de r pasos del tiempo está dada por simplemente por la evolución de la diferencia inicial (en este caso un solo sitio distinto de cero) para r pasos. La distancia de Hamming al paso del tiempo r está dada entonces por el número de sitios distintos de cero en la configuración obtenida por la evolución de un solo sitio, y para la regla 90 en particular tenemos que $H_r = 2_1^{\#}(r)$. Para reglas no aditivas, la diferencia entre configuraciones obtenidas a través de la evolución del autómata no depende sólo de la diferencia entre la configuración inicial, sino que un cambio en el valor de un pequeño número de sitios se amplifica al evolucionar el autómata. Lo cual nos da configuraciones con un número de sitios cada vez mayor (linealmente) de sitios no correlacionados. Algunas veces, los cambios en los sitios pueden ser erradicados después de un solo paso del tiempo r . Este comportamiento esencialmente se aplica a la regla 18, pero siempre está ausente cuando más de un sitio adyacente es distinta.

Uno puede especificar un ensemble estadístico de estados para un autómata celular finito, dando la probabilidad para cada una de las 2^N configuraciones posibles. En una colección de muchos estados desordenados con densidad $\rho = 1/2$, cada configuración posible crece asintóticamente con igual probabilidad. Tal colección de estados se denomina un *ensemble equiprobable* y puede ser considerado como *completamente desorganizado*. La evolución de los autómatas celulares modifican la probabilidad para los estados de algún ensemble, generando por lo tanto cierta *organización*. La evolución del autómata modifica las probabilidades para diferentes configuraciones, reduciendo las probabilidades para algunas a cero y dejando ciertos *gaps* (véase figura 3). En el ensemble inicial, a todas las configuraciones se les asigna *a priori* igual probabilidad. Después de la evolución del autómata para unos cuantos pasos del tiempo, se ve que ciertas configuraciones cambian sus probabilidades de acuerdo a una distribución de-

finida. Las propiedades de las configuraciones más probables dominan los promedios estadísticos sobre el ensemble, dando a una característica de promedio local para ciertas configuraciones de equilibrio, de acuerdo a las características locales de los autómatas, vistas en la sección anterior.

Una característica importante de los autómatas celulares elementales considerados aquí y en la sección anterior es su *irreversibilidad local*. Las reglas de los autómatas celulares pueden transformar diversas configuraciones iniciales distintas en la misma configuración final¹². Tenemos además que una configuración particular tiene descendientes únicos, aunque no necesariamente ancestros únicos. Un ejemplo trivial de esto puede verse usando la regla 0, la cual a cualquier configuración inicial, esta se desarrolla a la configuración nula (después de una sola generación). En un sistema reversible, cada estado tiene descendientes únicos y ancestros únicos, y por lo tanto, el número total de configuraciones debe mantenerse constante en el tiempo. (A esto se le conoce como teorema de Liouville). Sin embargo, en un sistema irreversible, el número de configuraciones posibles puede disminuir con el paso del tiempo. Esto provoca como consecuencia fundamental, que la auto-organización sea posible, permitiendo que algunas configuraciones sean mucho más probables que otras.

Una consecuencia de la irreversibilidad local es que algunas configuraciones de autómatas celulares puedan aparecer en las condiciones iniciales, mas nunca como descendientes de otras configuraciones a través de la evolución del tiempo. Esto hace imposible la existencia de los llamados *Jardines del Edén*. En el caso trivial de la regla 0, sólo el estado nulo puede ser alcanzado cuando el tiempo evoluciona. Todas las demás configuraciones son inalcanzables. La regla 4, por ejemplo, genera sólo aquellas configuraciones en las cuales no dos sitios adyacentes pueden tener el mismo valor. La fracción de las 2^N posibles configuraciones las cuales satisfacen este criterio tienden a cero en la medida que N tiende a infinito, de tal forma, que en este límite, sólo una pequeñísima fracción de las configuraciones es alcanzada. La regla 204 nos da la transformación identidad, y en este caso es única en el sentido que permite alcanzar todas las configuraciones posibles. Puede verse que esta regla además es trivialmente

¹² Un caso sorprendentemente en donde esto se cumple es aquel en el que se analiza la conjetura de Collatz.

reversible. Si se asumen condiciones de frontera periódicas, puede encontrarse que para N impar, la regla 90 genera sólo configuraciones en las cuales un número par de sitios tienen valores uno, y eso permite alcanzar la mitad de las 2^N configuraciones posibles. Para N par, sólo es posible alcanzar $1/4$ de las configuraciones totales posibles. Así entonces, una fracción finita de todas las posibles configuraciones es alcanzada cuando $N \rightarrow \infty$.

El comportamiento irreversible en un autómata celular puede ser analizado considerando el comportamiento de su *entropía* S o bien *el contenido de la información* $-S$. Usualmente se define la entropía como el logaritmo (aquí usaremos log base 2) del promedio del número de posibles estados de un sistema, esto es:

$$S = - \sum_i p_i \log_2 p_i$$

donde p_i es la probabilidad del estado i . Dicho con otras palabras, la entropía puede ser equivalente a considerar el número promedio de bits (binarios) para especificar un estado en un ensemble de posibles estados. La entropía total de un sistema es la suma de las entropías de subsistemas estadísticamente independientes. La entropía puede entonces tomar su valor máximo de un bit por sitio para un ensemble equiprobable. Para sistemas reversibles, la evolución del tiempo siempre lleva a un incremento en la entropía. Sin embargo, para sistemas irreversibles, tales como los autómatas celulares¹³, la entropía puede decrecer con el tiempo.

¹³ Nótese que la irreversibilidad no es necesariamente una característica de los autómatas celulares, como por ejemplo, el caso de la regla local de evolución $0 \rightarrow 1$ y $1 \rightarrow 0$. Aquí no sólo hay reversibilidad, sino también un comportamiento cíclico.

El Juego de la Vida

John Horton Conway es un distinguido matemático de la Universidad de Cambridge, el cual inventó un *juego solitario* que denominó *Juego de la Vida*, a causa de sus semejanzas con el surgimiento, decadencia y alteraciones que experimentan las sociedades de seres vivos. Básicamente este juego pertenece a los llamados *juegos de simulación*, que remedan o recuerdan procesos de la vida real.

Desde el punto de vista de la teoría de autómatas celulares, el juego de Conway puede definirse como un autómata celular en dos dimensiones. Las reglas de evolución de dicho autómata fueron cuidadosamente elegidas que satisficieran tres requisitos:

1. No ha de haber ninguna configuración inicial para la que pueda demostrarse fácilmente que la población crecerá ilimitadamente.
2. Deben existir configuraciones iniciales que *aparentemente* crezcan sin límite.
3. Han de existir configuraciones iniciales sencillas que crezcan y cambien durante períodos de tiempo considerables, antes de acabar de una de estas tres posibles formas: (a) extinguirse completamente (ya sea por superpoblación o por excesivo enrarecimiento), (b) adoptar una configuración estable, invariable en lo sucesivo, o (c) entrar en fase oscilatoria, donde se repiten sin fin, cíclicamente dos o más estados.

Dicho en pocas palabras, las reglas han de ser tales que la conducta de la población resulte a un tiempo interesante e impredecible. Nótese que esto nos lleva al problema de irreductibilidad computacional, el cual en la teoría de autómatas es fundamental.

Las reglas de evolución de Conway (llamadas por él *Reglas Genéticas*) son de una sencillez deliciosa. Tomemos un plano (o malla) cuadrículado de dimensiones infinitas. Así entonces, cada sitio (o escaque del plano) tiene ocho casillas vecinas: cuatro ortogonalmente adyacentes y cuatro adyacentes en diagonal. En cada sitio es posible poner un valor (o ficha). Consideraremos solamente dos valores: 0 y 1. Las reglas son:

1. Supervivencia: Cada ficha (o sitio con valor 1) que tenga dos o tres fichas vecinas sobrevive y pasa a la generación siguiente.
2. Fallecimiento: Cada ficha que tenga cuatro o más vecinas muere, y es retirada del tablero (el valor del sitio se vuelve 0), por superpoblación. Las fichas con sólo una, o ninguna vecina, fallecen por aislamiento.
3. Nacimientos: Cada casilla vacía, adyacente a exactamente tres fichas vecinas –tres, ni más, ni menos– es casilla generatriz. En la jugada siguiente, habrá de colocarse ese sitio con un 1.

Es importante hacer notar que todos los natalicios y fallecimientos ocurren *simultáneamente*, y constituyen en su conjunto una generación particular, al paso del tiempo τ , también llamado *tic del reloj*. Se descubrirá que para ciertas configuraciones iniciales de sitios distintos de 0, las generaciones subsiguientes van experimentando cambios constantemente, algunos insólitos, unos muy bellos, otros inesperados. En unos cuantos casos, la sociedad termina por extinguirse (al quedar eliminadas todos los sitios iguales a 1), aunque esto puede acontecer después de un gran número de generaciones o pasos del reloj. Casi todas las configuraciones terminan por alcanzar figuras estables –que Conway bautizó como *Naturalezas Muertas*– incapaces de cambio, o formaciones que oscilan por siempre. Puede verse que la diferencia fundamental entre los autómatas de una dimensión con los del juego de Conway consiste fundamentalmente que en el

último caso, sólo es posible ver la última generación de la evolución. Las anteriores generaciones desaparecen frente a nosotros. En el caso unidimensional, tenemos la historia completa en la evolución del autómata en cuestión.

En un principio, Conway conjeturó que ningún patrón inicial podría crecer ilimitadamente. Dicho con otras palabras, ninguna configuración compuesta por un número finito de sitios puede crecer hasta rebasar un límite superior finito, que limita el número de fichas que puede contener el campo de juego. Seguramente esta sea la cuestión más difícil y profunda que planteé este juego. Conway ofreció un premio de 50 dólares a la primera persona capaz de probar o de refutar la conjetura antes de finalizar el año de 1970. Una forma de refutarla sería dar con configuraciones que, generación tras generación, añadan más piezas al terreno de juego: un *cañón* (es decir, una configuración que repetidamente dispara objetos en movimiento), o bien el *tren puf-puf* (configuración que al paso del tiempo τ avanza dejando tras de sí una estela de "humo"). La conjetura de Conway se refutó en noviembre de 1970. Un grupo integrado en el proyecto de inteligencia artificial del M.I.T., comandado por William Gosper halló un cañón lanza-deslizadores, el cual genera un deslizador cada 30 pulsos de reloj. La existencia de tal cañón suscita la apasionante posibilidad de que el juego de Conway simule una máquina de Turing, la cual es capaz de hacer en principio, cualquier computación. Si el juego permite esta alternativa, entonces la siguiente pregunta a resolver sería si permite crear un constructor universal. De lo cual, se encontraría una máquina con capacidad de autorreproducción no trivial. Desgraciadamente, a la fecha ésto no ha podido conseguirse. El carácter universal del juego de la vida significa, en principio, es posible utilizar deslizadores para llevar a cabo cualquier cómputo que pueda efectuarse con la más potente de las computadoras digitales. Mediante la disposición de cañones lanza deslizadores y otras formas de "vida", es posible calcular π , e , la raíz cuadrada de 2, o de cualquier otro número real, con cualquier número de cifras decimales. Conway de hecho se vale del último teorema de Fermat para ilustrar la capacidad computacional del juego de la vida como sus limitaciones. Puede contruirse una máquina en el seno del juego que vaya sistemáticamente tanteando los valores de las cuatro incógnitas de la famosa fórmula de Fermat. Podría diseñarse el programa de modo tal que se detuviera, desapareciendo, por ejemplo, si encontrara algún contraejemplo de la conjetura de Fermat. Por otra parte, si la conjetura es verdadera, la máquina

de cálculo construída bajo "vida" tendría que continuar tanteando eternamente, en búsqueda de la combinación correcta de valores. Por la teoría de indecidibilidad sabemos que no hay manera de saber por adelantado si un problema cualquiera dado es soluble o no, y por consiguiente, no hay forma de saber de manera anticipada si en el juego de la vida, dada una configuración cualquiera, continuará cambiando, o si alcanzará un final estable. Conway mismo, en una carta a Martin Gardner comenta: "...si los deslizadores pueden formar por colisión un pentadecatlón, entonces puede producir máquinas autorreplicantes; la cuestión de si una máquina dada es autorreplicante es indecidible". Es sin embargo un hecho comprobado que los deslizadores pueden crear pentadecatlones al colisionar, por lo que en el espacio de configuración del juego es posible construir máquinas autorreplicantes, es decir, máquinas que construyan copias exactas de sí mismas. La máquina primitiva puede permanecer en el espacio, o bien ser programada para que se autodestruya para cuando haya sacado una copia de sí. Hasta ahora no se sabe de nadie que haya construído una máquina de este tipo, pero si Conway está en lo cierto, entonces es posible construirlas.

Algunas Configuraciones Sencillas

Existen una gran cantidad de configuraciones posibles. Lo más sensato es analizar desde las formas de "vida" más elementales, hasta las más complejas posibles. Así entonces, una configuración inicial que consista en un solo sitio, se extinguirá al primer *tic* del reloj. Puede verse que la orientación de la configuración en el plano es irrelevante de acuerdo a las reglas de evolución planteadas anteriormente. Una formación de tres sitios se extinguirá de inmediato a menos que por lo menos un sitio tenga a un par de vecinos distintos de 0. Un bloque de tres sitios colineales se convierte en un intermitente (con ese nombre bautizó Conway a esta configuración), la cual tiene período 2, y también se les denomina *biestables* o *flip-flops*. Se tienen también las configuraciones de cinco sitios denominadas *pentominós*, las cuales han sido estudiadas profusamente. El llamado pentominó *R* es el único que no se extingue, se estabiliza o cae en fase oscilatoria. Como Conway comenta después de haber seguido la evolución

de esta configuración hasta el paso de 460 generaciones: *"Ha dejado por ahí a un montón de residuos y detritus, y tiene tan solo unas cuantas regiones activas, por lo que no es evidente que vaya a sobrevivir indefinidamente"*. Uno de los descubrimientos más llamativos de los pentominós fue el que se representa en la figura 1. Al cabo de dos pasos del tiempo (en la evolución), esta configuración se ha desplazado un poco y ha quedado reflejado respecto a una recta diagonal. Este es el llamado deslizador que crea lo que en geometría se llama una simetría con desplazamiento.

Conclusiones

Técnicamente hablando, el Juego de la Vida es un ejemplo clásico de una clase especial de autómatas celulares totalísticos, en los cuales el valor de un sitio depende sólo de la suma de los valores de sus vecindades en el tiempo t anterior, y no de los valores individuales de cada sitio.

Ahora bien, muchas configuraciones que exhiben propiedades particulares han sido encontradas. La más sencilla, invariante a través del tiempo es el *cuadrado o bloque o "block"*, que consiste en cuatro sitios adyacentes vivos. También existe el *hexágono o colmena*, el cual contiene seis sitios con vida. Existen las configuraciones que *oscilan* a lo largo de dos o más pasos del tiempo; la más sencilla es la configuración *intermitente o "blinker"*, la cual consiste de tres sitios vivos, con ciclos de período de dos pasos del tiempo. También se conocen osciladores de 3, 5 y 7 pasos del tiempo, aunque otros períodos pueden encontrarse por composición. Existen, por otra parte, configuraciones que se "mueven" uniformemente a través de la malla, ejecutando un par de ciclos internos. El ejemplo más sencillo es el *deslizador o "glider"*, el cual consiste de cinco sitios vivos, y después de un ciclo de 2, llega a su estado inicial. Han sido halladas configuraciones que generan un infinito número de deslizadores, los llamados *cañones deslizadores*, dando como conclusión una población más abundante en cada paso del tiempo. No obstante, estas configuraciones en algunos casos no existen a menos que se den como una configuración inicial, por lo que Conway las denominó como *jardines del Edén*, ya que no pueden ser alcanzadas al paso de ninguna generación posterior.

El juego de Conway en realidad plantea a un complicado autómatas celular que sólo puede ser simulado expresamente para encontrar como evoluciona a través del tiempo. Sin embargo, nadie ha podido demostrar que éste no sea reducible computacionalmente.

Todo esto nos lleva a pensar en la posibilidad de que el universo sea un vasto autómatas celular (lo siento, imposible describir sus reglas de evolución en este trabajo), que funciona de acuerdo con los movimientos de partículas primordiales (quizás aún no descubiertas), de conformidad con ciertas reglas de transición. Los físicos están buscando una Gran Teoría de Unificación (GTU) que articule y reúna todas las fuerzas de la naturaleza en una teoría unificada. Hacer corresponder dicha teoría conlleva implícitamente que las partículas fundamentales del universo (las piezas), las leyes fundamentales (las reglas de transición) y el espacio-tiempo (el tablero), no son necesidades lógicas. Sencillamente nos son dadas. Está fuera de toda discusión preguntarse por qué son así. Lo mismo que en el ajedrez, los físicos deben aceptar el juego tal y como es, y disfrutar de su interminable tarea de conjeturar cómo se juega, y no desperdiciar energía, en cambio, especulando por qué está el juego diseñado como está.

	1	2	3	4	5	6	7	8	9	10	11	12	
1													1
2						*							2
3							*						3
4					*	*	*						4
5													5
6													6

Generación Inicial (0)

	1	2	3	4	5	6	7	8	9	10	11	12	
1													1
2													2
3					*		*						3
4						*	*						4
5						*							5
6													6

Generación (1)

	1	2	3	4	5	6	7	8	9	10	11	12	
1													1
2													2
3							*						3
4					*		*						4
5						*	*						5
6													6

Generación (2)

	1	2	3	4	5	6	7	8	9	10	11	12	
1													
2													
3					*								
4						*	*						
5					*	*							
6													

Generación [3]

	1	2	3	4	5	6	7	8	9	10	11	12	
1													
2													
3						*							
4							*						
5					*	*	*						
6													

Generación [4]

Capítulo 6

La Imposibilidad de Demostrar la Conjetura de Collatz

Introducción

Los autómatas celulares pueden ser no sólo una herramienta útil en la investigación de problemas físicos y biológicos. También pueden servir como un elemento de peso para ciertas demostraciones en matemáticas, algunas en particular, en teoría de números.

En este apéndice se intentará demostrar que la conjetura de Collatz (explicada más adelante) es indemostrable. Dicho de otra manera, esta conjetura no puede reducirse a una demostración analítica, sino que para demostrarla, es necesario realizarla explícitamente para cada caso.

La Conjetura de Collatz

Hay una insólita clase de problemas aritméticos los cuales pueden ser asociados a un problema computacional de iteraciones, "bucles" o "loops". Se genera una serie de enteros de acuerdo a cierta regla. Se pregunta entonces uno, si la serie acabará entrando en uno o más bucles, en los que un conjunto de enteros se va repitiendo periódicamente.

La conjetura que nos ocupa se define de la siguiente manera: Tómesese algún número entero positivo. Divídase entre dos si es par; si es impar, multiplíquese por tres y al producto súmese uno. Si aplicamos este procedimiento repetidamente eventualmente llegaremos al número 1, entonces diremos que el número (con el que empezamos el procedimiento) es maravilloso.

Por ejemplo, consideremos el número 12. Como es par, dividámoslo entre 2, el resultado es 6. Como éste es par, dividámoslo de nuevo entre 2, esto nos da como resultado 3. Este último entero es impar, así que multipliquemos por 3 y sumemos 1. Esto nos da 10, el cual es par. De nuevo, dividamos entre 2. Esto nos lleva al 5, el cual es impar, por lo cual, multiplicamos por 3 y le sumamos uno. Esto nos da como resultado 16, el cual es par, entonces dividimos entre 2, lo cual nos entrega un 8, el cual es par y se vuelve a dividir entre 2, el cual da 4, que a su vez, dividido entre 2 resulta 2 y éste último dividido entre 2 nos da finalmente 1. Por lo tanto, 12 es un número maravilloso¹⁴. (Podríamos seguir con este procedimiento y nos encontraríamos entrampados en una serie cíclica 2, 1, 4, 2, 1, 4, ...) La pregunta fundamental de esta conjetura es la siguiente: Dado cualquier número entero positivo y utilizando el procedimiento anteriormente descrito, ¿se caerá en todos los casos en la serie 2, 1, 4? Nadie hasta ahora ha demostrado que esto ocurra forzosamente. Nadie ha podido hallar un contraejemplo.

Una Breve Historia

A este problema, también denominado el problema $3X + 1$, todavía se resiste a los esfuerzos por resolverlo. Según Richard Guy, fue propuesto antes de la segunda guerra mundial por Lothar Collatz, hoy matemático de la Universidad de Hamburgo, cuando era estudiante. En una conferencia que dió en 1970 H.S.M. Coxeter ofreció 50 dólares por una demostración que él pudiera entender, y 100 dólares por un contraejemplo. Tal ha sido el diluvio de falsas demostraciones que le ha llovido, que Coxeter no está dispuesto a evaluarlas. Parece, en efecto, que es tan fácil cometer sutiles errores en

¹⁴ Yo preferiría llamarle número de Collatz, pues fue él quien propuso el problema

las demostraciones de este tipo como en las del último teorema de Fermat (en el cual abundan las demostraciones equivocadas). De hecho, hasta se han publicado algunas demostraciones falsas¹⁵. En 1982 Paul Erdős expresó su opinión (¿y cuál más calificada que la suya?) de que si la conjetura es verdadera, la teoría de números carece hoy de instrumental para demostrarla.

Un grupo de investigadores del laboratorio de inteligencia artificial del MIT ha puesto a prueba, mediante una computadora, todos los enteros positivos hasta el 60 000 000 sin encontrar una sola excepción. Descubrieron también que si la regla $3n + 1$, utilizada cuando es impar, se reemplaza por $3n - 1$, el resultado, en valores absolutos, es el mismo que si se comenzase con un entero negativo y se siguiera la antigua regla. En este caso, se descubrió que todos los enteros negativos hasta -100 000 000 caían en uno de los tres bucles, con los siguientes valores absolutos:

(1) 2, 1, 2, ...

(2) 5, 14, 7, 20, 10, 5, ...

(3) 17, 50, 25, 74, 37, 110, 55, 164, 82, 41, 122, 61, 162, 81, 272, 136, 68, 34, 17, ...

Michael Deemer, William Gosper y Rich Schroepfel dan estos resultados en HA-KMEM (abreviatura de Hacker's Memo), Memo 239, Artificial Intelligence Laboratory, MIT, 1972. Parece ser que a nadie se le ha ocurrido una idea feliz que permita establecer el caso general para todos los enteros no nulos. (El cero, evidentemente pertenece al bucle 0, 0, 0, ...) Nadie sabe tampoco, si hay enteros que generen sucesiones divergentes hacia infinito carentes de bucle.

Si se busca un contraejemplo, éste tendría que ser un número que, o bien, fuese generando números siempre mayores, sin repetir jamás ninguno, o bien, cayese en un bucle distinto de 4, 2, 1. De existir algún contraejemplo, tendría que ser superlativamente grande, porque según Guy, la conjetura ha sido verificada hasta 7×10^{11} . Al poco de empezar este juego se descubrió que no era preciso ensayar los números pares, ni tampoco los impares de la forma $4k + 1$, $16k + 3$ o $128k + 1$. De este modo, los programas de computadora se abrevian mucho. Evidentemente, tan pronto como

¹⁵ como en Fibonacci Quarterly, vol. 18, 1980, pp 231-242

una sucesión tropieza con una potencia de 2, muchas veces, tras una caótica serie de altibajos, se desploma vertiginosamente hacia 4, 2, 1. La potencia de 2 hacia la que más sucesiones convergen es 16.

Entre los números menores de 50, el de peor comportamiento es el 27. Tras 77 pasos alcanza una cima de 9232. Bastan después 34 pasos para reducirlo a 1. Cuando John H. Conway (véase el capítulo sobre el juego de la vida en este mismo trabajo) presenta en sus lecciones la conjetura del $3x + 1$, le gusta ir a la pizarra, y decir: "Tomemos un número al azar, el 27, por ejemplo, y veamos que sucede."

Como podrá verse después de todo lo dicho anteriormente, la conjetura de Collatz no ha podido aún resolverse. Sin embargo, aquí intentaremos demostrar que el problema del $3x + 1$ es computacionalmente irreducible, lo cual significa en otras palabras que sólo mediante la ejecución explícita para cada número, es posible ver si el número en cuestión es maravilloso o no. Lo que equivaldría a decir que no puede existir una demostración analítica del problema en cuestión.

Sistemas Computacionalmente Irreducibles

Hemos visto anteriormente que la simulación por computadora es el único método utilizado actualmente para investigar muchos sistemas físicos y biológicos. Sin embargo habría que preguntarse si la simulación constituye el procedimiento posible más eficiente o si hay una fórmula matemática que pudiera llevar más directamente a los resultados. Para acotar la discusión, hemos de ahondar más en la correspondencia entre los procesos físicos y los procesos computacionales.

Sabemos que todo proceso físico puede expresarse (o describirse) a través de un algoritmo, y que, por lo tanto, cualquier sistema físico puede representarse como un proceso computacional. Se ha de determinar hasta dónde llega la complejidad de éste último. En el caso de los autómatas celulares, la correspondencia del proceso físico y el proceso por computadora es clara. El autómata celular puede contemplarse a la

luz de un modelo para un sistema físico, pero también puede considerarse un sistema computacional análogo a un ordenador digital ordinario. En un autómata celular, la secuencia de valores iniciales de las células puede interpretarse como dato abstracto o como información, de forma muy parecida a la secuencia de dígitos binarios de una computadora digital. A lo largo de la evolución del autómata se procesa esta información; los valores de las células se modifican según unas reglas predefinidas. De forma similar, los dígitos almacenados se modifican según las reglas contenidas en la unidad central del procesador de la computadora.

Así entonces, la evolución de un autómata celular a partir de su configuración inicial se asimila a una computación que procesa la información contenida en la configuración. Para los autómatas que muestran un comportamiento simple, la computación es sencilla. Por ejemplo, puede tratarse de detectar las secuencias de tres células consecutivas cuyo valor inicial sea igual a 1. Por otro lado, a la evolución de un autómata celular que muestra un comportamiento complicado puede corresponder una computación asimismo compleja.

Mediante una simulación explícita de cada paso, podemos determinar el resultado de un número dado en la evolución de un autómata celular. El problema estriba en saber si existe o no un procedimiento más eficaz. Dicho con otras palabras, ¿Existirá alguna forma para acortar la simulación paso a paso, esto es, un algoritmo que dé el resultado de muchos pasos en la evolución de un autómata celular sin necesidad de tener que seguirlos todos? La computadora podría ejecutar este algoritmo y sería posible predecir la evolución de un autómata celular sin simularlo explícitamente. Esto sería análogo a encontrar una fórmula que fuese expresada en función del paso del tiempo que se desea observar. El fundamento de esta operación consistiría que la computadora pudiese llevar a cabo una computación más refinada que el autómata celular, y alcanzar así los mismos resultados con menos pasos. Tal atajo es sólo posible si la computadora logra realizar cálculos intrínsecamente más complicados que los involucrados en la evolución del autómata celular mismo.

Podemos definir una clase de problemas, llamados problemas calculables, que son los que admiten solución en un tiempo finito siguiendo ciertos algoritmos determi-

nados. Una computadora simple, digamos una máquina sumadora puede resolver un subconjunto de estos problemas. Sin embargo, existen máquinas universales capaces de resolver cualquier problema calculable. Las computadoras digitales son máquinas de este tipo. Las instrucciones que puede ejecutar el procesador central son suficientes como para servir de elementos a un programa que pueda incorporar cualquier algoritmo. Además de las computadoras digitales, cierto número de sistemas se han mostrado como factibles de una computación universal, entre otros, algunos sistemas de autómatas celulares. En este sentido, se ha comprobado la capacidad de computación universal de un autómata celular bidimensional con sólo dos valores (por ejemplo 0 y 1) en cada sitio. Otras argumentaciones incluso inducen a pensar que varios autómatas celulares unidimensionales (de clase 4) son también máquinas universales. Los candidatos más simples tienen tres posibles valores para cada célula y reglas de evolución que toman en cuenta solamente las células más cercanas.

Así entonces, los autómatas celulares capaces de computación universal imitan el comportamiento de cualquier máquina calculable. Y dado que cualquier proceso físico puede representarse como un proceso computacional, pueden también imitar la acción de cualquier sistema físico posible. Si hubiera un algoritmo capaz de seguir el comportamiento de estos autómatas celulares con una celeridad mayor que la propia evolución de los autómatas celulares, permitiría acelerar cualquier computación. Puesto que esta conclusión lleva a una contradicción lógica, se deduce que no puede haber un atajo válido general para predecir la evolución de un autómata celular arbitrario. Los cálculos correspondientes a la evolución son irreducibles, esto es, el resultado puede obtenerse solamente mediante la simulación explícita de la evolución. De hecho, esta simulación directa constituye el mejor método para determinar el comportamiento de ciertos autómatas celulares. No hay manera de predecir su evolución; sólo queda observar lo que pasa.

No se sabe todavía cuán extendido se halla el fenómeno de la irreducibilidad computacional entre los autómatas celulares, ni entre los sistemas físicos en general. A pesar de lo cual, resulta claro que los elementos de un sistema no necesitan ser muy complicados para que la evolución global del mismo sea computacionalmente irreducible. Cabe incluso mencionar que la irreducibilidad computacional se da casi

siempre en sistemas con comportamiento caótico o complejo. No se conocen fórmulas generales que describan el comportamiento global de estos fenómenos; quizás no lleguen a encontrarse nunca fórmulas así. En tal caso, la simulación explícita es el único medio de investigación disponible.

Tradicionalmente la física se ha centrado (en su mayor parte), sobre el estudio de fenómenos computacionalmente reducibles, los cuales admiten una descripción simple y global. En los sistemas físicos reales, sin embargo, la reducción computacional es más la excepción que la regla. Podemos considerar como ejemplo la turbulencia de los flúidos como un caso típico de irreducción computacional. En los sistemas biológicos, la extensión de estos fenómenos irreducibles puede ser mayor todavía: quizás ocurriera que la forma de un organismo biológico pudiera determinarse a partir de su código genético, sin más que seguir, paso a paso, su desarrollo.

De la irreducción computacional se deduce que hay preguntas que pueden hacerse sobre el comportamiento fundamental de un sistema, pero a las cuales no se puede contestar en forma general mediante un proceso matemático o computacionalmente finito. Tales cuestiones han de catalogarse, entonces, como indecidibles. A manera de ilustración, supongamos que tenemos un autómata celular ha de extinguirse o no al paso de la evolución (esto es, a cada paso del tiempo). Esta cuestión puede ser muy simple de contestar para un número dado de pasos, digamos 1000. Esto implica simplemente simular 1000 pasos en la evolución del autómata. No obstante, a fin de conocer la respuesta para cualquier número de pasos, se ha de simular la evolución del autómata para un número de pasos potencialmente infinita. Si el autómata celular es irreducible, no hay alternativa real a esta simulación directa.

La conclusión a todo esto es que no hay ningún proceso de cálculo de longitud fija que pueda determinar categóricamente si un modelo se extinguirá al final. Se puede encontrar el destino de un modelo particular siguiendo sólo unos pocos pasos de su evolución, pero no hay forma general de conocer de antemano cuántos pasos se necesitarán. La forma final de un modelo es el resultado de un número infinito de pasos, correspondiente a una computación infinita; a menos que la evolución del modelo sea

computacionalmente reducible, sus consecuencias no pueden reproducirse con ningún proceso computacional o matemático finito.

La posibilidad de cuestiones indecidibles en los modelos matemáticos de los sistemas físicos puede verse como una manifestación del teorema de Gödel sobre la indecidibilidad en matemáticas, demostrado por Kurt Gödel en 1931. El teorema establece básicamente que en todos los sistemas matemáticos, hasta en los más simples, caben proposiciones que no pueden ni probarse ni refutarse con un proceso matemático o lógico. La prueba de una proposición puede requerir un número de pasos lógicos indefinidamente grande. Incluso, proposiciones de formulación concisa pueden exigir una prueba arbitrariamente larga. En la práctica hemos visto muchos teoremas matemáticos simples para los cuales las únicas demostraciones conocidas son muy largas. Además, los casos que han de examinarse para probar o refutar conjeturas, son a menudo muy complicados. En la teoría de números, por ejemplo, hay muchos casos en que el número más pequeño que posee una propiedad especial dada es extremadamente grande; muchas veces este número sólo puede encontrarse tanteando de uno en uno.

Relación de los Automatas Celulares con la Conjetura de Collatz

Consideremos un autómata celular unidimensional de longitud finita. Pensemos en una línea de sitios, en los cuales se pueden colocar las células correspondientes. Tomemos cada uno de los enteros (del 0 al 9) como un tipo de célula. Así entonces, en cada sitio del autómata podemos poner un valor correspondiente a uno de los diez posibles valores. Ahora bien, la regla de evolución de dicho autómata puede ser expresada en términos de la conjetura de Collatz. Si el autómata es par, divídase entre dos. Si el autómata es impar, multiplíquese por tres y al producto súmese uno.

Lo anterior define la conjetura de Collatz como un autómata celular unidimensional con $k = 9$, esto es, diez posibles valores para cada sitio. Donde la regla de evolución está dada por un proceso sobre cada sitio, el cual puede o no depender de los sitios vecinos al que nos ocupa. Esto último puede causar algunos inconvenientes para la simulación del

proceso de evolución de dicho autómata. Resulta que en muchos casos, al multiplicar cada sitio por 3 (en el caso que el autómata sea impar), nos encontraremos que el sitio vecino (de la derecha) hay que sumarle uno si la multiplicación por tres excede el valor de diez. Esto puede ir además en cascada, lo cual complica un poco más la simulación explícita del autómata. De lo cual, puede deducirse que la regla de evolución se puede dar a lo más en $n - 1$ sitios, donde n es el número total de sitios en toda la línea del autómata. Así entonces, r va desde 0 hasta $n - 1$. Esto puede hacer ver de manera clara que no existe forma de saber cuántos sitios involucrados hay en cada paso del autómata en un tiempo dado (es decir, cuál es la r), todo depende del número inicialmente elegido y su comportamiento ulterior. Así pues, la r es irreducible computacionalmente en la evolución del autómata. Si se considera que un autómata celular con $k = 9$ es muy difícil de simular, considérese un autómata con $k = 1$, en donde cada entero tiene una expresión en binario, lo cual puede simplificar en algún sentido las cosas, -como por ejemplo, la división entre dos es simplemente el recorrer cada sitio del autómata a la derecha (recuérdase que estamos trabajando en la representación binaria del entero propuesto inicialmente) - mas no en su simulación explícita.

Si la r es efectivamente irreducible computacionalmente, estamos frente a un caso de irreductibilidad computacional, lo cual acarrea como primera consecuencia la indecidibilidad del fenómeno descrito, el cual sólo puede ser expresado realizando su simulación. En otras palabras, la conjetura de Collatz se ha transformado simplemente en averiguar si el autómata que la describe puede ser computacionalmente irreducible o no. Desgraciadamente no existe un procedimiento para discernir si un autómata es irreducible o no. Hay que analizar los hechos que se presentan en la simulación del fenómeno. En el caso de la conjetura de Collatz, puede verse que dado un número al azar, el comportamiento en el autómata descrito puede en algunos casos ser caótico (como en el caso del 27) o bien elementalmente descrito (para un número par potencia de 2) en donde la conjetura sí es reductible computacionalmente de manera trivial. Esto es, para un número de la forma 2^n se requieren de $n - 1$ pasos para llegar a 1. Sin embargo, para ciertos números impares o pares distintos a potencias de dos, el problema parece ser indecidible.

En conclusión, la conjetura de Collatz puede representarse como un autómata celular unidimensional elemental, donde aparentemente la regla de evolución de dicho autómata no es reducible computacionalmente (para ciertos casos) y en consecuencia, el teorema de Gödel apoya la indecidibilidad de la conjetura, la cual en principio sólo puede ser simulada explícitamente. La consecuencia directa de esto último es que no puede existir una demostración algebraica o analítica del problema en cuestión. Esto implica que si en realidad, la conjetura de Collatz es computacionalmente irreducible, entonces se puede afirmar (por el teorema de Gödel) que la conjetura es verdadera, esto es, que todos los números tienen la propiedad de ser maravillosos. Si no fuese así, simplemente dando un contraejemplo se demostraría que la conjetura es falsa. No obstante, aún nadie ha encontrado uno.

Capítulo 7

Los Programas

Los programas presentados en el siguiente capítulo permiten calcular: (1) autómatas celulares unidimensionales y (2) el autómata celular de la conjetura de Collatz. Nótese que en principio puede verse el desarrollo caótico de algunos números, lo cual nos indica una fuerte correlación con las argumentaciones expuestas sobre la irreducibilidad computacional de los sistemas físicos y matemáticos. Se utilizan dos técnicas distintas en Pascal, una es usando arreglos y la otra es mediante apuntadores. Ambas simulan explícitamente la conjetura de Collatz.

```
program Automata7;
```

```
{
```

Este programa describe el comportamiento de un autómata de Wolfram tal y como aparece en el artículo: *Universality and Complexity in Cellular Automata*, *Physica* 10D (1984) 1-35, North Holland, Amsterdam. Además, pone en pantalla la regla usada por el usuario.

Programa: Manuel López Michelone

Versión : 1.1

```
}
```

```
{ Incluye todo lo necesario para Turbo GRAPHIX }
```

```
{ $ i typedef.sys }
```

```
{ $ i graphix.sys }
```

```
{ $ i kernel.sys }
```

```
{ $ U+ } { Para poder poner CTRL-C }
```

```
const
```

```
    limX = 639; { la pantalla está mapeada por bits, 640 en el
                eje X }
```

```
var
```

```
    AutoCel : array [-10..639, 0..1] of integer;
```

```
    config : string[200]; { configuración inicial del AC }
```

```
    i : integer;
```

```
    renglon : integer; { renglón en donde se encuentra
                        dibujándose autómata celular }
```

```
    regla : array [1..8] of integer; { reglas
                                       locales de evolución }
```

```
    numero : integer; { número de regla }
```

```
    tope : integer; { número de generaciones a dibujar }
```

```
procedure LimpiaArreglo;
```

```
{ Este procedimiento limpia el arreglo de sitios del autómata antes de
usarlo }
```

```
var
```

```
    i, j : integer;
```

```

begin
  for i := -10 to limX do
    for j := 0 to 1 do
      AutoCel[i, j] := 0;
    end; { LimpiaArreglo }
  procedure LeeDatos;
    { Este procedimiento lee la configuración inicial del autómata (edo.
base) }
  begin
    ClrScr;
    config := '';
    write( 'Dame la configuración inicial: ');
    readln(config);
    writeln;
    writeln( 'Dame el resultado de la regla:');
    write( '111: '); readln(regla[1]);
    write( '110: '); readln(regla[2]);
    write( '101: '); readln(regla[3]);
    write( '100: '); readln(regla[4]);
    write( '011: '); readln(regla[5]);
    write( '010: '); readln(regla[6]);
    write( '001: '); readln(regla[7]);
    write( '000: '); readln(regla[8]);
    write( '¿Cuántas generaciones hago? (2..198) ');
    readln(topc);
  end; { LeeDatos }
  procedure NumeroRegla;
    { Esta rutina calcula el valor decimal de la regla de evolución del AC }
  var
    i      : integer;
    numero : integer;
  begin
    numero := (regla[1] * 128) + (regla[2] * 64) + (regla[3] * 32)

```

```

      + (regla[4] * 16) + (regla[5] * 8);
numero1 := (regla[6] * 4) + (regla[7] * 2) + (regla[8] * 1);
numero := numero + numero1;
end; { NumeroRegla }
procedure Inicializa;
{ Inicializa la línea 0 del arreglo AutoCel con la configuración dada
por el usuario en LeeDatos }
var
  i      : integer;
  temp   : string[80];
  temp1  : integer;
  codigo : integer; { Variable tonta que usa Turbo Pascal }
begin
  for i:=1 to length(config) do
    begin
      temp := copy(config, i, 1);
      val(temp, temp1, codigo);
      AutoCel[i + 320, 0] := temp1;
    end; { for }
  end; { Inicializa }
procedure tocaLa;
{ Avisa con un tono de 440 Hertz (LA) que ya acabó de graficar }
var
  ch      : char;
begin
  Sound(440);
  Delay(100);
  NoSound;
  read(kbd, ch);
end; { tocaLa }
procedure IniGrafica;
{ Este procedimiento inicializa todo el turbo Graphix system }
begin

```

ESTA TESIS NO DEBE
 SALIR DE LA BIBLIOTECA

```

  SetClippingOn; { Para dibujar siempre dentro de la pantalla }
  InitGraphic;
  DrawBorder;
end; { IniGrafica }
procedure DibujaBlanco;
{ Esta rutina dibuja un punto blanco sobre fondo negro }
begin
  SetColorWhite;
  DrawPoint(i, renglon);
end; { DibujaBlanco }
procedure DibujaPunto;
{ Este procedimiento manda cada valor del autómata unidimensional a la
  pantalla de gráficas }
begin
  for i := 1 to limX do
    begin
      if AutoCel[i, 1] = 1 then DibujaBlanco;
    end; { for }
  end; { DibujaPunto }
procedure DibujaVezPrimera;
{ Este procedimiento manda cada valor del autómata unidimensional a la
  pantalla de gráficas en la configuración inicial }
begin
  for i := 1 to limX do
    begin
      if AutoCel[i, 0] = 1 then DibujaBlanco;
    end; { for }
  end; { DibujaVezPrimera }
procedure EscribeRegla;
{ Esta rutina imprime el conjunto de reglas locales usados para la
  evolución del autómata celular de Wolfram }
begin
  gotoXY(30, 2);

```

```

write( 'Regla: ', regla[1], regla[2], regla[3], regla[4], regla[5],
       regla[6], regla[7], regla[8]);
write( ' (', numero, ')');
end; { EscribeRegla }
procedure CambiaLinea;
{ Esta rutina actualiza la línea unidimensional del autómata }
begin
  for i := 0 to limX do
    AutoCel[i, 0]:=AutoCel[i, 1];
end; { CambiaLinea }
procedure ProcesaLinea;
{ Esta rutina calcula la siguiente línea del Autómata mediante las
reglas locales, la evolución del Autómata }
var
  i, j      : integer;
  línea, a   : string[5];
begin
  for i := -10 to limX do
    begin
      línea := '';
      for j := -1 to 1 do
        begin
          str(AutoCel[i+j, 0], a);
          línea := línea + a;
        end; { for j }
      if línea = '111' then AutoCel[i, 1] := regla[1] else
      if línea = '110' then AutoCel[i, 1]:=regla[2] else
      if línea = '101' then AutoCel[i, 1]:=regla[3] else
      if línea = '100' then AutoCel[i, 1]:=regla[4] else
      if línea = '011' then AutoCel[i, 1]:=regla[5] else
      if línea = '010' then AutoCel[i, 1]:=regla[6] else
      if línea = '001' then AutoCel[i, 1]:=regla[7] else
      if línea = '000' then AutoCel[i, 1]:=regla[8];
    end;
  end;

```

```
    end; { for i }
end; { ProcesaLinea }
{ Programa Principal }
begin
  LimpiaArreglo;
  LecDatos;
  NumeroRegla; { Calcula el Número de regla }
  Inicializa;
  IniGrafica;
  EscribelRegla;
  renglon := 00;
  DibujaVezPrimera;
repeat
  ProcesaLinea;
  renglon := renglon + 1;
  DibujaPunto;
  CambiaLinea;
until renglon > tope;
  tocaLa;
  LeaveGraphic;
end.
```

program *NumerosMaravillosos*;

{El programa lee un número entero positivo de cualquier magnitud y determina si se trata de un número maravilloso, a partir de las siguientes condiciones:

- i) 1 es maravilloso.
- ii) Si el número es par, se considera maravilloso si el valor del número dividido entre 2 es maravilloso.
- iii) Si el número es impar, el número es maravilloso si al multiplicarlo por 3 y sumarle 1, este resultado es maravilloso.

Para poder manejar números de una magnitud superior a MAX.INT se utilizan apuntadores.

Programó: Manuel López Michelone

Versión 2.01

}

type

Rango = 0..9;

Apuntador = ^*Digito*;

Digito = record

Liga_Ant,

Liga_Pos : *Apuntador*;

Valor : *Rango*;

end; {record}

var

Primero,

Ultimo,

Actual : *Apuntador*;

Carry : integer;

{-----}

procedure *Inserta*(*Nodo_Ant*, *Nodo_Pos*: *Apuntador*; *Digito* : *Rango*);

begin

New(*Actual*);

with *Actual*^ **do**

begin

```

    Valor:=Digito;
    Liga_Pos:=nodo_Pos;
    Liga_Ant:=nodo_Ant;
    nodo_Ant^.Liga_Pos:=Actual;
    nodo_Pos^.Liga_Ant:=Actual;
end;
if nodo_Ant=nil
then Primero:=Actual;
if nodo_Pos=nil
then Ultimo:=Actual;
end; {Inserta}
{-----}
procedure Lectura;
var
    Entrada : char;
    Error,
    Digito : integer;
begin
    ClrScr;
    write( 'Dame un número -> ');
    repeat
        read(kbd, Entrada);
        val(Entrada, Digito, Error);
        if Error = 0 then
            begin
                write(Digito);
                Inserta(Ultimo, nil, Digito)
            end
        until eoln (kbd)
    end; {Lectura}
{-----}
procedure Multiplica(Ult.nodo : Apuntador; Factor : integer);
var

```

```

    Numero : integer;
begin
    Actual:=Ult_nodo;
    Carry:=1;
    while Actual<>nil do
    with Actual^ do
    begin
        Numero:=(Factor*Valor) + Carry;
        Valor:=Numero mod 10;
        Carry :=(Numero div 10);
        Actual:=Liga_Ant;
    end;
    if Carry<>0 then
        Inserta(nil, Primero, Carry);
    end; {Multiplica}
}
procedure Divide(Ult_nodo : Apuntador);
begin
    Multiplica(Ultimo, 5);
    Ultimo:=Ultimo^.Liga_Ant;
    Ultimo^.Liga_Pos:=nil
end; {Divide}
}
procedure Verifica (Ult_nodo : Apuntador);
var
    Residuo : integer;
begin
    Residuo:=Ult_nodo^.Valor mod 2;
    if Residuo = 0 then
        Divide(Ultimo)
    else
        Multiplica(Ultimo, 3);
    end; {Verifica}

```

```

{-----}
function Termina : Boolean;
begin
  Termina := (Ultimo = Primero) and (Ultimo^.Valor = 1)
end; {Termina}
{-----}
procedure Imprime;
begin
  Actual := Primero;
  writeln;
  while (Actual <> nil) do
  with Actual^ do
    begin
      write(Valor);
      Actual := Liga.Pos
    end;
end; {imprime}
{-----}
begin {Programa Principal}
  repeat
    Ultimo := nil;
    Primero := nil;
    Lectura;
  until Primero^.Valor <> 0;
  while not termina do
    begin
      Verifica(Ultimo);
      Imprime
    end
end.

```

Conclusión

Voy en busca de un gran quizás.
Bajad el telón,
la farsa ha terminado

Rabelais

Bibliografía

Mathematical Games

Martin Gardner

Scientific American

Octubre 1971 a Febrero 1972

Physics Like Models of Computation

Norman Margolus

Physica 10D (1984) pp. 81-95

North Holland, Amsterdam

Discrete Systems, Cell-Cell Interactions and Color Pattern of Animals (I) Conflicting Dynamics and Pattern Formation

G. Cocho, R. Pérez-Pascual, J.L. Rius

Universidad Nacional Autónoma de México

Discrete Systems, Cell-Cell Interactions and Color Pattern of Animals (II) Clonal Theory and Cellular Automata

G. Cocho, R. Pérez-Pascual, J.L. Rius, F. Soto

Universidad Nacional Autónoma de México

Cellular Automata as an alternative to differential equations in modeling physics

Tommaso Toffoli

Physica 10D (1984) pp. 117-127

North Holland, Amsterdam

Simulating Physics with Cellular Automata

Gérard Y. Vichniac

Physica 10D (1984) pp. 96-116

North Holland, Amsterdam

Universality and Complexity in Cellular Automata

Stephen Wolfram

Physica 10D (1984) 1-35

North Holland, Amsterdam

Statistical Mechanics of Cellular Automata

Stephen Wolfram

Reviews of Modern Physics

Vol 55, No. 3, Julio 1983

Essays on Cellular Automata

Edited by Arthur W. Burks

University of Illinois Press 1970

U.S.A.