Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

AUTÓMATAS CELULARES DESDE LA TEORÍA DE NÚMEROS

T E S I S

que para obtener el título de:

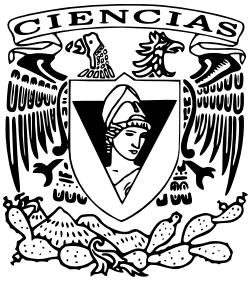
MATEMÁTICO

presenta:

ANDRÉS CASILLAS GARCÍA DE PRESNO

Director de tesis:

MAT. J. CÉSAR GUEVARA BRAVO



Ciudad Universitaria, Ciudad de México, 2023

A la educación pública, gratuita, laica, universal y desinteresada.

Esta página ha sido intencionalmente dejada en blanco
(y ahora ya no está en blanco).

Primero
Que nada:
Me complace
Enormísimamente
Ser
Un buen
Poeta
De segunda
Del
Tercer
Mundo

Efraín Huerta

Sólo existen dos tipos de personas que pueden hacer matemáticas: los masoquistas y los enamorados.

Dr. Carlos Torres Alcaraz

It is perhaps a little humbling to discover that we as humans are in effect computationally no more capable than cellular automata with very simple rules. But the Principle of Computational Equivalence also implies that the same is ultimately true of our whole universe. [...]
And indeed in the end the Principle of Computational Equivalence encapsulates both the ultimate power and the ultimate weakness of science. For it implies that all the wonders of our universe can in effect be captured by simple rules, yet it shows that there can be no way to know all the consequences of these rules, except in effect just to watch and see how they unfold.

Stephen Wolfram, A new kind of science.

Nota Técnica

La versión digital de este trabajo contiene animaciones que pueden ser visualizadas en formato PDF. Sin embargo, es necesario contar con un lector moderno, como por ejemplo Adobe Acrobat Reader 64 bits.

En su defecto, la versión impresa contiene las imágenes sucesivas que conforman cada animación.

Agradecimientos

Aprovecho este espacio para externar mis sinceros agradecimientos a las personas e instituciones que me han acompañado a lo largo de esta travesía, y que en gran parte la han hecho posible.

Agradezco, en primer lugar, a mi familia y a Brenda por brindarme la alegría y compañerismo que me han acompañado durante toda mi vida. Les agradezco infinitamente.

A mi ex novia Ana, por acompañarme a lo largo de estos años y hacer de la vida un lugar más cálido, diverso, incluyente, pacífico y armonioso. Por mostrarme otras formas de vivir, de vencer la adversidad y de querer.

A mis queridos amigos Huesos, Gasch, Martín, Ro, Fercho, Pablo, entre muchos otros, con quienes crecí y sigo creciendo, con quienes tengo los mejores recuerdos de mi infancia y adolescencia e incontables recuerdos de muchísimos años de amistad fraterna. Difícilmente les puedo expresar mi gratitud con palabras. También, por supuesto, a los amigos que conocí en mi querida Facultad de Ciencias: Dieguillo, Mike, Pablito, Paoli, René, Nico, Tony, Isaac, Urda, Diego, Iñigo, Sof con quienes no sólo compartí la pasión por las matemáticas sino también los mejores años de mi vida. Gracias por acompañarme a lo largo de estos años con tanto cariño, respeto y aprecio. No saben cuánto los admiro y quiero.

A la Facultad de Ciencias, por supuesto, y a la UNAM en general, por brindarme no sólo una extraordinaria educación a cambio de nada, sino también por abrirme los ojos ante la realidad de mi querido y lastimado país. Siempre las recordaré como los lugares en los que he sido más feliz y en donde más he crecido y más me he realizado. Estoy en deuda con ustedes.

A mis queridos profesores: César Guevara, por guiarme a lo largo de esta tesis y haberme enamorado de la teoría de números y de la historia de las matemáticas. A Javier Fernández García, por recibirme con los brazos abiertos y haberme enseñado la paciencia y dedicación que se necesita para ser un gran matemático. A la Dra. Diana Avella Alaminos, por haberme dado la oportunidad de ser su ayudante con tanta paciencia y compasión. A los integrantes del Seminario de Foliaciones Holomorfas (Laura, Jessy, Gil, Jesús, Oziel, etc.) por haberme aceptado dentro de su círculo como uno de lo suyos, siempre incluyéndome en sus discusiones y esforzándose en sus explicaciones. En particular quiero destacar el apoyo que recibí de la Dra. Jessie Pontigo al aceptar ser mi tutora en

el IMATE: le agradezco muchísimo. También, por supuesto, a Gil Bruno, quien siempre ha sido un guía para mí y al cual reconozco como uno de los mejores matemáticos que he conocido. Al Dr. José Antonio de la Peña, con quien tuve contacto por azares de la vida, y quien siempre nos recibió, a los integrantes de la "Materturlia", con tanto gusto para platicar de matemáticas. Como parte fundamental de la "Matertulia"se encuentra el Dr. Jesús Arturo Jiménez González; muchas gracias, Jesús, por tu tiempo y dedicación al mando del seminario y por tomarte el tiempo de revisar este trabajo. Al M. en C. Renato Leriche Vázquez, por inculcarme el amor a la programación matemática a través de los sistemas dinámicos, misma que ha moldeado enormemente mi vocación científica. Al Dr. Carlos Torres Alcaraz, por las clases más divertidas que tuve y por haberme explicado los teoremas más fascinantes de toda la carrera: los teoremas de Gödel. A mis demás profesores, que recuerdo con cariño y admiración y a quienes les debo mucho: Dr. Vinicio Gomez, Dra. Ana Meda Guardiola, Dr. Omar Antolín Camarena, Dr. Max Neumann, Dr. Javier Bracho, Dr. Jorge Chávez Carlos, Dr. Ernesto Rosales, Dr. Carlos Prieto, Dra. Mónica Alicia Clapp, M. en C. Luz Arely Carrillo, entre otros, quienes a pesar de una pandemia y la precariedad laboral universitaria, supieron transmitirme el valor y amor por la ciencia. Agradezco también a todos mis sinodales por sus atinadas observaciones y correcciones.

Quisiera expresarle un agradecimiento muy especial al Dr. Francisco Antonio Godínez Rojano, con quien realicé mi servicio social -mismo que culminó con la publicación de mi primer *paper* y la invitación a mi primer congreso- y con quien entablé una sólida amistad. Muchas gracias por creer en mí y darme una introducción tan grata al mundo de la investigación científica.

Asimismo quisiera recalcar el apoyo que recibí por parte de la Dra. Diana Avella Alaminos a lo largo del año y medio que tuve el privilegio de ser su ayudante. Le agradezco infinitamente el haberme transmitido el amor a la docencia. Asimismo, su apoyo para mis futuros estudios de posgrado fueron fundamentales. Muchas gracias Dra.

Le debo un agradecimiento, también, a aquellos matemáticos y científicos cuyas obras han influido enormemente en mi pensamiento y desarrollo como matemático. En particular a Stephen Wolfram, por haber escrito el libro de ciencia más maravilloso e influyente que he leído.

Por supuesto que el listado anterior no es ni exhaustivo ni exclusivo. Sería imposible agradecerle a todas las personas que merecen una mención en este espacio, ya que, este trabajo, como muchas cosas en la vida hechas con amor y dedicación, no es únicamente obra mía sino de todos ellos.

Introducción

A pesar de su aparente sencillez, nada similar a un autómata celular parece haber existido antes de 1950. Lo anterior obedece principalmente a dos razones íntimamente relacionadas entre sí: la primera es que nuestra experiencia cotidiana en ciencia e ingeniería jamás hubiera podido sugerir que un sistema podía exhibir capacidades sofisticadas partiendo de configuraciones iniciales y reglas deterministas sencillas; la segunda es que la intuición anterior sólo hubiera podido haberse modificado -como de hecho sucedió- con el desarrollo y la realización de las computadoras electrónicas. Es interesante notar que un impedimento ideológico para el desarrollo de los autómatas celulares -como es el caso de la primera razón expuesta- fue transformado por el desarrollo material de un invento tecnológico tan revolucionario, como lo fueron las computadoras modernas.

Ideas similares y precursoras de la de autómata celular se encuentran por primera vez en los trabajos de John Von Neumann y Stanislaw Ulam en la década de los 50, quienes estaban principalmente motivados por dar una prueba matemática de la posibilidad de auto-replicación de los organismos. En sus primeros trabajos (ver [2]) Von Neumann da una construcción de un autómata celular de 20000 celdas, 29 estados posibles para cada una de ellas y reglas de evolución complicadas. Esto puso de manifiesto la creencia que existía en los primeros años de la revolución informática -y que en muchos aspectos sigue existiendo actualmente-, de que era necesario partir de gran complejidad para poder obtener gran complejidad. A partir del trabajo de Von Neumann y Ulam distintos matemáticos -algunos nombres incluyen a Gustav Hedlund, Morton Curtis, Roger Lyndon y Alyn Ray Smith- se dieron a la tarea de entender las bases matemáticas de los sistemas auto replicables (después formalizados como autómatas celulares) principalmente desde la óptica de los sistemas dinámicos discretos, la dinámica simbólica y la computabilidad, ésta última basada en las ideas de Alan Turing. Más adelante, en la década de los 70, el célebre matemático John Conway inventó el autómata celular denominado *Juego de la Vida* (ver capítulo 1) popularizado por Martin Gardner. Debido a la diversidad del comportamiento de distintas configuraciones iniciales, el *Juego de la Vida* logra romper con la vieja intuición acerca de la aparición de comportamiento complejo y trae consigo una enorme cantidad de estudio e interés alrededor de los autómatas celulares, misma que se traduce en investigaciones fructíferas principalmente conducidas por Stephen Wolfram en la década de los 80.

Wolfram centra su atención en el estudio de autómatas celulares elementales con la sospecha de que el mecanismo que le permite a los autómatas celulares generar comportamiento complejo a partir de configuraciones iniciales sencillas debe de ser el mismo que utiliza la

naturaleza para generar su enorme riqueza. Así, con el propósito de cambiar el paradigma de la física -cuyos esfuerzos culminan con la publicación de su libro *A Project to Find the Fundamental Theory of Physics* en 2020-, Wolfram formula los conceptos de aleatoriedad intrínseca y reducibilidad computacional, considerando a los autómatas celulares como los objetos ideales para su estudio. A raíz de dichos conceptos, diversos matemáticos -como Jarkko Kari y Mathew Cook- centran su investigación en los aspectos computacionales y algorítmicos de los autómatas celulares, demostrando importantes teoremas como la indecidibilidad del problema del Jardín del Edén de dimensión mayor a 1 (ver capítulo 1) y la universalidad de la Regla 110 (ver [18]). Wolfram resume sus esfuerzos en su prominente obra *A New Kind of Science* publicada en 2002, misma que constituye, en gran medida, la espina dorsal del presente trabajo.

Dicho lo anterior, no es difícil darse cuenta que sobran motivos y perspectivas para abordar el tema de autómatas celulares. Sin embargo, la motivación inicial del presente trabajo -que no necesariamente implica que sea la principal- fueron las curiosidades que observamos en el Triángulo de Pascal desde la teoría de números (principalmente los teoremas 2.5 y 2.6) y su potencial estudio desde la teoría de los autómatas celulares, apoyándonos en la teoría desarrollada por Wolfram en torno a polinomios característicos y dipolinomios de evolución (ver definición 2.2). A lo largo de todo el trabajo pretendemos establecer puentes entre ambas ramas de las matemáticas, intentando responder simultáneamente a las preguntas: ¿qué nos puede decir la teoría de números acerca de los autómatas celulares? y ¿qué nos pueden decir los autómatas celulares acerca de la teoría de números? El capítulo 2 es un bello ejemplo de cómo la teoría de números nos puede dar respuestas acerca del comportamiento de los autómatas celulares, mientras que el capítulo 3 muestra una extensión del concepto de autómata celular motivada fuertemente por la teoría de números.

En el capítulo 1 definimos algunas nociones básicas de autómatas celulares y exhibimos un panorama general de los teoremas más importantes de la teoría de autómatas celulares desarrollada hasta la actualidad.

En el capítulo 2 presentamos el importante concepto de autómata celular aditivo y restringimos nuestra atención al estudio de dos autómatas celulares aditivos estelares: la Regla 60 y la Regla 90. Los estudiamos tanto desde la teoría de números como desde la teoría de la computabilidad, para posteriormente establecer relaciones entre ambas.

En el capítulo 3 ampliamos nuestra noción de autómata celular mediante lo sugerido por los polinomios característicos de los autómatas celulares aditivos, y con ello damos una interpretación, dentro del universo de los autómatas celulares, del Teorema de los Números Pentagonales de Euler.

Cabe mencionar que -en el espíritu que muestra Stephen Wolfram en *A New Kind of Science*- hemos incluido los pseudocódigos que utilizamos para la elaboración del presente trabajo con la noble intención y el profundo deseo de que el lector realice sus propias exploraciones computacionales del universo de los autómatas celulares. Debemos advertirle al lector que aventurarse en dicho universo trae consigo extraordinarias sorpresas, a tal

grado que en varias ocaciones pensará que sus códigos están mal escritos o que presentan errores. Nuestra sugerencia es la siguiente: revise sus códigos y, una vez que verifique que no contienen errores, déjese maravillar por sus descubrimientos.

Finalmente, querido lector, espero que disfrute leer este trabajo tanto como yo disfruté escribirlo.

Índice general

Capítulo 1

Autómatas celulares

> "There is no use trying," said Alice;
> "one can't believe impossible things."
> "I dare say you haven't had much
> practice," said the Queen.
>
> *Lewis Carroll, Alice in Wonderland*

1.1. Érase una vez...

La historia de los autómatas celulares inició en la década de los años 60 con los intentos de John Von Neumann y Stanislaw Ulam para crear una máquina autoreplicadora. En el texto célebre *Theory of Self-Reproducing Automata*, Von Neumann estudia a los autómatas en la naturaleza -como el sistema nervioso humano- y explora las dificultades por enfrentar para poder crear un autómata artificial que pueda, de alguna forma, mimetizar el comportamiento de un autómata natural. Para la descripción de las características que debería de tener un autómata para adquirir la capacidad de autoreplicación, podemos ver la cita siguiente de Von Neumann:

> *The constructing automaton floats on a surface, surrounded by an unlimited supply of parts. The constructing automaton contains in its memory a description of the automaton to be constructed. Operating under the direction of this description, it picks up the parts it needs and assembles them into the desired automaton. To do this, it must contain a device which catches and identifies the parts that come in contact with it.* [2]

El lector podrá constatar que dicha descripción en efecto es muy parecida a la de los modelos de autómatas celulares con los que trabajamos en la actualidad. Por esta razón a Von Neumann y a Ulam se les considera los padres de los autómatas celulares, ellos los imaginaron y conceptualizaron por primera vez, pero no fue sino hasta la década de los años 80 que Stephen Wolfram haría un experimento que sugería que lo imaginado por Von Neumann y Ulam era apenas la punta de un iceberg sobre el cuál, según Wolfram, descansará la ciencia futura.

1.1.1. El experimento de Wolfram

Figura 1.1: Stephen Wolfram con Richard Feynman, 1984. [1]

En la década de los años 80, Stephen Wolfram realizó una serie de experimentos computacionales que dieron pie a gran parte de la teoría que se ha desarrollado en torno a los autómatas celulares. Wolfram conjeturó que, si la ciencia tradicional ha sido parcialmente exitosa en explicar una gran cantidad de fenómenos naturales, es porque dichos fenómenos deben de seguir una serie de reglas determinadas, y además sencillas. A lo largo de la historia reciente se ha supuesto que estas reglas pueden ser explicadas mediante métodos matemáticos tradicionales, mismos sobre los cuáles se ha edificado una gran cantidad de teoría científica -por ejemplo, la física y la química-. Sin embargo, -y ésta es una de las ideas claves del trabajo de Wolfram- podríamos pensar que las reglas que sigue la naturaleza no están limitadas a relaciones matemáticas clásicas, sino que forman parte de una categoría más general de relaciones. Pero, ¿cuál sería una categoría más general de relaciones que aquella descrita por las matemáticas tradicionales? Según Wolfram, dichas relaciones se encuentran en un universo al cual estamos cada vez más acostumbrados, pero que hubiera sido imposible de imaginar hace apenas unos cuantos lustros: el universo de los programas computacionales.

Típicamente, los programas computacionales a los que estamos acostumbrados están diseñados para realizar tareas específicas, y por lo tanto las reglas en las que se basan y la forma en la que se relacionan tienden a ser complicadas. Más aún, los programas computacionales que típicamente programamos están hechos para que podamos *predecir* su comportamiento, precisamente para que tengamos control sobre dichos sistemas y sepamos que pueden realizar las tareas para las cuáles los estamos implementando. Sin

[1] Wolfram, S. (2016, julio 11). *My time with Richard Feynman.* Wired. Consultada diciembre 7, 2022. https://www.wired.com/2016/07/my-time-with-richard-feynman/

embargo, Wolfram argumenta que, en principio, un programa computacional puede seguir esencialmente cualquier conjunto de reglas finitas, independientemente de si nosotros podamos predecir o no su comportamiento, o de si nos serán útiles para realizar tareas específicas presentes o planteadas anteriormente o no. Con estas ideas en la mente Wolfram empezó a explorar el mundo de los programas computacionales, inició con aquellos que siguen las reglas más sencillas imaginables -el estado de una celda cambiará en función de los estados de sus celdas vecinas- y que tienen la representación más sencilla imaginable -simplemente sucesiones finitas de 1's y 0's- (ver sección 1.6). Al realizar los experimentos correspondientes se dió cuenta de una característica contraintuitva que poseen esta clase de sistemas -a los que llama autómatas celulares elementales- y que resulta ser su principal atractivo: a pesar de ser regidos por reglas sencillas y condiciones iniciales sencillas, su comportamiento puede ser arbitrariamente complejo.

Wolfram escribe lo siguiente:

> *One might have thought -as at first I certainly did- that if the rules for a program were simple then this would mean that its behavior must also be correspondingly simple. For our everyday experience in building things tends to give us the intuition that creating complexity is somehow difficult, and requires rules or plans that are themselves complex. But the pivotal discovery that I made some eighteen years ago is that in the world of programs such intuition is not even close to correct.* [18]

Este descubrimiento es central no sólo para la teoría de autómatas celulares sino para la ciencia en general, pues pretende responder uno de los grandes misterios del universo: ¿cómo es que existe tanta complejidad en la naturaleza? Basta entonces, según Wolfram, con que la naturaleza se comporte esencialmente como un conjunto de programas sencillos.

Cabe entonces preguntarnos: ¿qué tan sencillas pueden ser las reglas de un sistema sin que su comportamiento sea trivial o, al menos, muy predecible? Mientras él realizaba los experimentos relacionados con autómatas celulares elementales, rápidamente se encontró con uno -al cuál llamó Regla 30- que, al comenzar únicamente por una celda con el valor 1, exhibía comportamiento complejo y en muchos aspectos aleatorio. Wolfram menciona que la Regla 30 (ver figuras 1.2 y 1.3) es el descubrimiento más importante y sorprendente que ha hecho en su vida. [18]

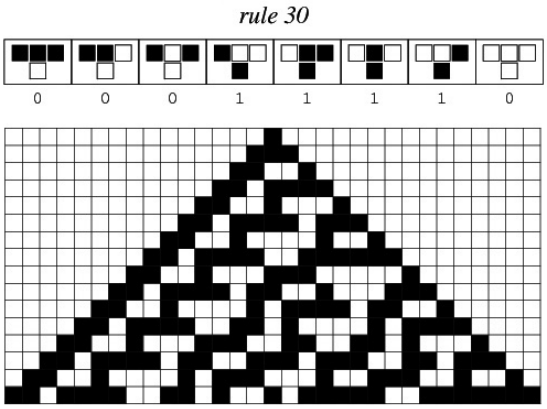


Figura 1.2: Regla 30. [2]

[2]Weisstein, Eric W. *Rule 30.* From MathWorld: A Wolfram Web Resource. https://mathworld.wolfram.com/Rule30.html

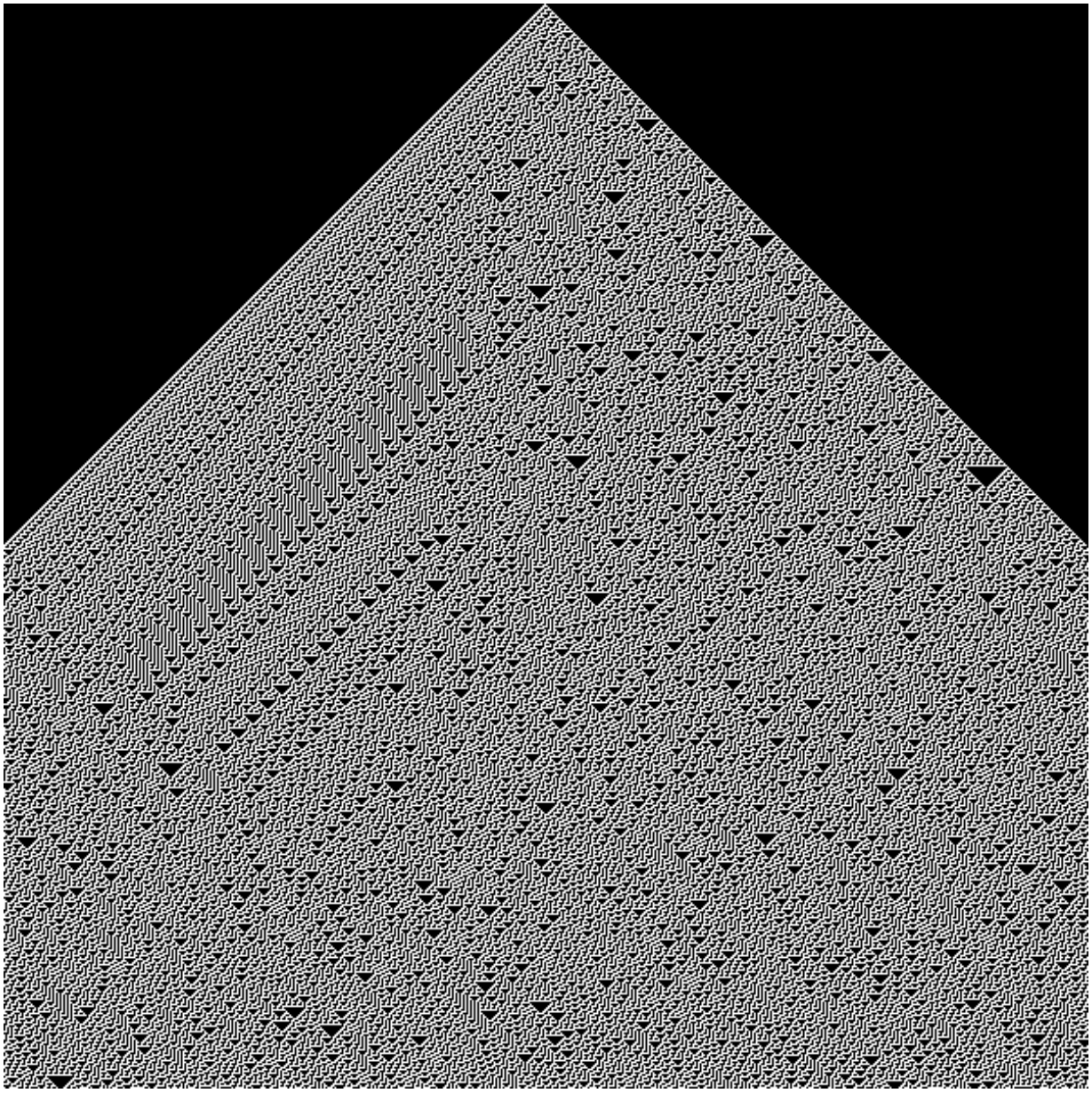


Figura 1.3: Regla 30. [3]

No deja de ser sorprendente cómo un sistema determinista, con la configuración inicial más sencilla imaginable, y con un conjunto de 8 reglas increíblemente sencillas, pueda producir un patrón que, hasta la fecha, parece ser completamente aleatorio.

Wolfram nos confirma lo señalado en esta cita:

> *But where does this complexity come from? We certainly did not put it into the system in any direct way when we set it up. For we just used a simple cellular*

[3]Imagen tomada de [18] pp.30.

automaton rule, and just started from a simple initial condition containing a single black cell. [...]

Over and over again we will see the same kind of thing: that even though the underlying rules for a system are simple, and even though the system is started from a simple initial condition, the behavior that the system shows can nevertheless be highly complex. And I will argue that it is the basic phenomenon that is ultimately responsible for most of the complexity that we see in nature. [18]

Entonces, la respuesta a la pregunta planteada es que existen sistemas increíblemente sencillos que pueden dar lugar a un comportamiento increíblemente complejo. Y más aún, a lo largo de su obra, Wolfram argumenta que no sólo existen dichos sistemas, sino que práticamente cualquier sistema que tenga comportamiento no trivial alcanzará a producir comportamiento complejo [18]. Cabe mencionar que muchas de las preguntas relacionadas con el comportamiento de la Regla 30 continúan siendo preguntas abiertas, y el propio Wolfram ha ofrecido $\$30,000$ USD a quien responda alguna de las siguientes preguntas:

1. ¿La columna del centro siempre permanece no-periódica?
2. ¿En promedio, existe la misma cantidad de 1's y 0's en la columna del centro?
3. ¿Computar la n-ésima celda de la columna del centro requiere al menos un esfuerzo computacional de $O(n)$?

Las preguntas anteriores son de relevancia central no sólo para el estudio de autómatas celulares elementales, también son fundamentalmente para entender muchos de los procesos que observamos en la naturaleza e incluso dentro de las propias teorías científicas. La intuición que nos da la matemática clásica no siempre es capaz de sugerir que el tipo de objetos que estudiaremos en este trabajo son relevantes, pero así como en su época el microscopio hizo posible el desarollo de la biología moderna, así la computadora hará posible el desarollo de *una nueva ciencia*.

1.2. Formalización

Intuitivamente, podemos plantear que un autómata celular es un sistema dinámico discreto que consiste de un conjunto de celdas que cambian de estado en función del estado de sus vecinos, según una regla de evolución dada. Todas las celdas cambian de estado al mismo tiempo y el proceso se repite en intervalos discretos de tiempo [7].

Una propiedad atractiva de los autómatas celulares es que, a partir de una regla de evolución sencilla y una condición inicial sencilla, se puede alcanzar un comportamiento extremadamente complejo (de hecho, en muchos casos, tan complejo como una Máquina de Turing). [18] A lo largo de este trabajo abreviaremos "autómata celular" como AC.

Para formalizar esta idea, introduzcamos los siguientes conceptos:

Definición 1.1. *Sea $d \in \mathbb{Z}^+$. Por **espacio celular d-dimensional** entenderemos $\mathbb{Z}^d$. A los elementos de dicho espacio celular les llamaremos **celdas**.*

A lo largo de este trabajo se considerará $d = 1$, de tal manera que el espacio celular unidimensional pueda ser representado únicamente por una cinta de celdas indexadas por $\mathbb{Z}$.

Figura 1.4: Espacio celular 1-dimensional

Definición 1.2. *Un **conjunto finito de estados** es un anillo conmutativo finito con unidad, que denotaremos por S.*

Comúnmente trabajaremos con $S = \mathbb{Z}_2$ o $S = \mathbb{Z}_p$ con p primo.

Definición 1.3. *Sea S un conjunto finito de estados. Una **configuración** de un autómata celular d-dimensional con conjunto de estados S es una función*

$$c : \mathbb{Z}^d \to S$$

que le asigna un estado a cada celda. El estado de la celda $n \in \mathbb{Z}^d$ lo denotamos por $c(n)$.

Una configuración debe de ser entendida como una descripción de todos los estados del autómata celular en un tiempo determinado. Es decir, una sucesión de estados que determinan al autómata en un tiempo dado.

Definición 1.4. *Sea $m \in \mathbb{Z}^+$. Una **m-vecindad** d-dimensional es un vector*

$$N = (n_1, \ldots, n_m)$$

*donde $n_i \in \mathbb{Z}^d$ y $n_i \neq n_j$ para toda $i \neq j$. A los elementos de una vecindad los llamaremos **vecinos**. Una vez establecido m, hablaremos únicamente de la **vecindad**.*

Definición 1.5. *Una **regla de evolución** con conjunto de estados S y tamaño $m \in \mathbb{Z}^+$ es una función*

$$f : S^m \to S$$

que especifica el nuevo estado de una celda de acuerdo a los estados anteriores de sus vecinos. Si los vecinos de una celda n_i tienen estados $s_1, s_2, \ldots, s_m$ entonces el nuevo estado de la celda será $f(s_1, s_2, \ldots, s_m)$.

Dado que usaremos la misma regla de evolución aplicada a la vez para todas las celdas del AC, podemos pensar en el cambio global de estados. La configuración c cambia a la configuración c' donde, para cada $n \in \mathbb{Z}^d$,

$$c'(n) = f[c(n + n_1), c(n + n_2), \ldots, c(n + n_m)]$$

Definición 1.6. *A la transformación $c \mapsto c'$ la llamaremos la* ***función de transición global*** *del autómata celular. Dicha función es*

$$G : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$$

donde $S^{\mathbb{Z}^d}$ denota al conjunto de funciones de $\mathbb{Z}^d$ en S.

Comúnmente, iteraremos a la función G, obteniendo una sucesión

$$c \mapsto G(c) \mapsto G^2(c) \mapsto G^3(c) \mapsto \dots$$

del sistema, donde c es la configuración inicial del autómata. Así, de forma usual, denotaremos a la órbita de una configuración c como

$$orb(c) = c, G(c), G^2(c), \dots$$

En este contexto, el tiempo se referirá al número de aplicaciones de la función G, de forma tal que $G^t(c)$ es la configuración del autómata al tiempo $t \in \mathbb{N}$.

Con las definiciones anteriores ya estamos en condiciones de dar la siguiente definición.

Definición 1.7. *Un* ***autómata celular*** *es una tupla $A = (d, S, N, f)$ donde $d \in \mathbb{Z}^+$, S denota un conjunto de estados, $N \in (\mathbb{Z}^d)^m$ denota una vecindad y f denota una función de evolución.*

Denotaremos a la función de trasición global inducida por el conjunto anterior como $G[A]$ o simplemente G cuando sea claro quién es el autómata en cuestión.

Comúnmente identificamos a un autómata celular A con su función de transición global, en tanto que hablamos del autómata celular G. Sin embargo, estrictamente, la misma función G puede ser inducida por dos autómatas (tuplas) distintos. Decimos que dos autómatas A y B son iguales si $G[A] = G[B]$.

Ejemplo 1.1. *Sea $d = 1, S = \mathbb{Z}_2, N = (-1, 0, 1), f : \mathbb{Z}_2^3 \to \mathbb{Z}_2$ dada por*

$$f(x, y, z) = x + y + z \qquad (mód\ 2).$$

Podemos pensar al espacio celular como una línea de celdas, indexadas por $\mathbb{Z}$. Cada celda cambia de estado sumando los estados de sus vecinos y de sí misma módulo 2. En la numeración de Wolfram este autómata corresponde a la Regla 150.

Consideremos la configuración inicial c_0 donde $c_0(0) = 1$ y $c_0(i) = 0$ para toda $i \neq 0$. Luego, $c_1 = G(c_0)$ es tal que $c_1(-1) = c_1(0) = c_1(1) = 1$. Procediendo de forma análoga, obtenemos $c_2 = G(c_1), c_3 = G(c_2), \dots$

Podemos representar cada configuración de forma horizontal, siendo la configuración inicial c_0 el primer renglón. Después añadimos cada configuración sucesiva debajo de la anterior. Así, podemos obtener una representación visual de las iteraciones de la función G (ver figuras 1.5 y 1.6).

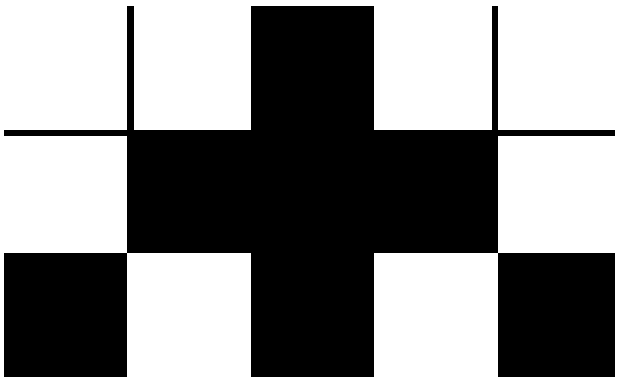


Figura 1.5: Representación de $G^n(c_0)$ para $n = 0, 1, 2$.

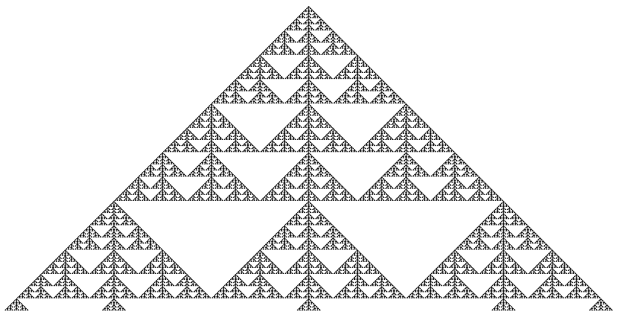


Figura 1.6: Representación de $G^n(c_0)$ para $n = 0, 1, \ldots, 400$.

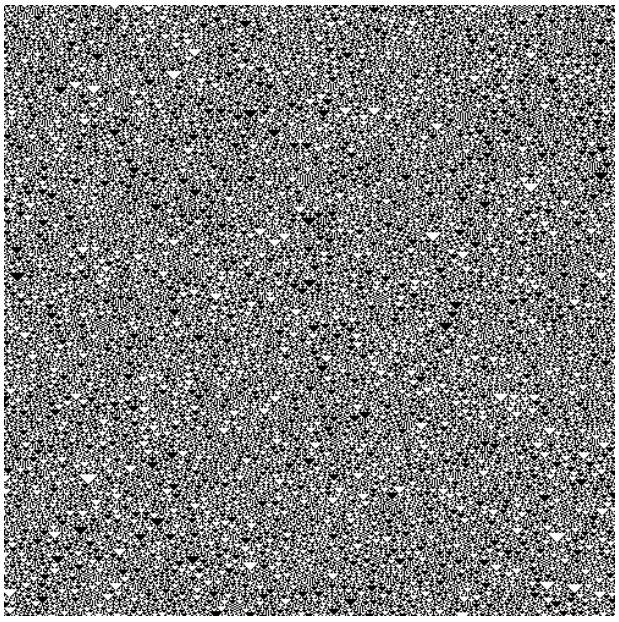


Figura 1.7: Representación de $G^n(c_0)$ para $n = 0, 1, \ldots, 500$. y $c_0(i) = X$ con X una variable aleatoria Bernoulli con $p = 1/2$ para toda $i = 1, \ldots 500$.

Ejemplo 1.2. *Sea* $d = 2, S = \mathbb{Z}_2, N = ((-1,-1),(0,-1),(1,-1),(-1,0),(0,1),(-1,1),(0,1),(1,1))$, $f : \mathbb{Z}_2^8 \to \mathbb{Z}_2$. *Para definir a la función* f, *sea* $h : \mathbb{Z}^2 \to \mathbb{N}$ *la función que a cada celda le asocia el número de vecinos en su vecindad. Es decir,*

$$h(a,b) = (\sum_{i=-1}^{1} \sum_{j=-1}^{1} s_{a+i,b+j}) - s_{a,b}$$

donde $s_{k,l}$ *denota el estado de la celda* (k,l).

Así, la función f *está dada por*

$$f(x_1,x_2,x_3,x_4,(a,b),x_6,x_7,x_8) = \begin{cases} 1 & si \quad 2 \leq h(a,b) \leq 3 \quad \& \quad s_{a,b} = 1 \\ 1 & si \quad h(a,b) = 3 \quad \& \quad s_{a,b} = 0 \\ 0 & e.o.c \end{cases}$$

Podemos pensar al espacio celular como una retícula de celdas indexadas por $\mathbb{Z}^2$. *La función* f *anteriormente definida puede ser interpretada de la siguiente manera:*

- *Toda celda viva con dos o tres vecinos vivos permanece viva.*
- *Toda celda muerta con tres vecinos vivos se convierte en una celda viva.*
- *En el resto de los casos, las celdas mueren (o permanecen muertas).*

El autómata anterior, ideado por John Conway (1937-2020) es conocido como "El Juego de la Vida". Es uno de los autómatas que tiene la propiedad de ser Turing Universal [18]. A continuación se muestran algunas de las configuraciones -junto con sus respectivas evoluciones- que puede alcanzar dicho autómata.

Figura 1.8: Configuración periódica.

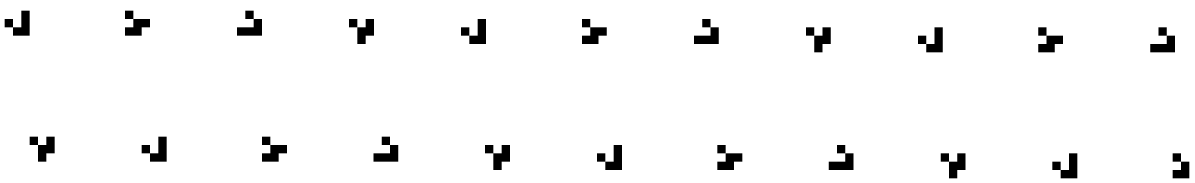


Figura 1.9: Configuración conocida como Glider. $G^i(c_0)$ para i = 1,. . . ,22.

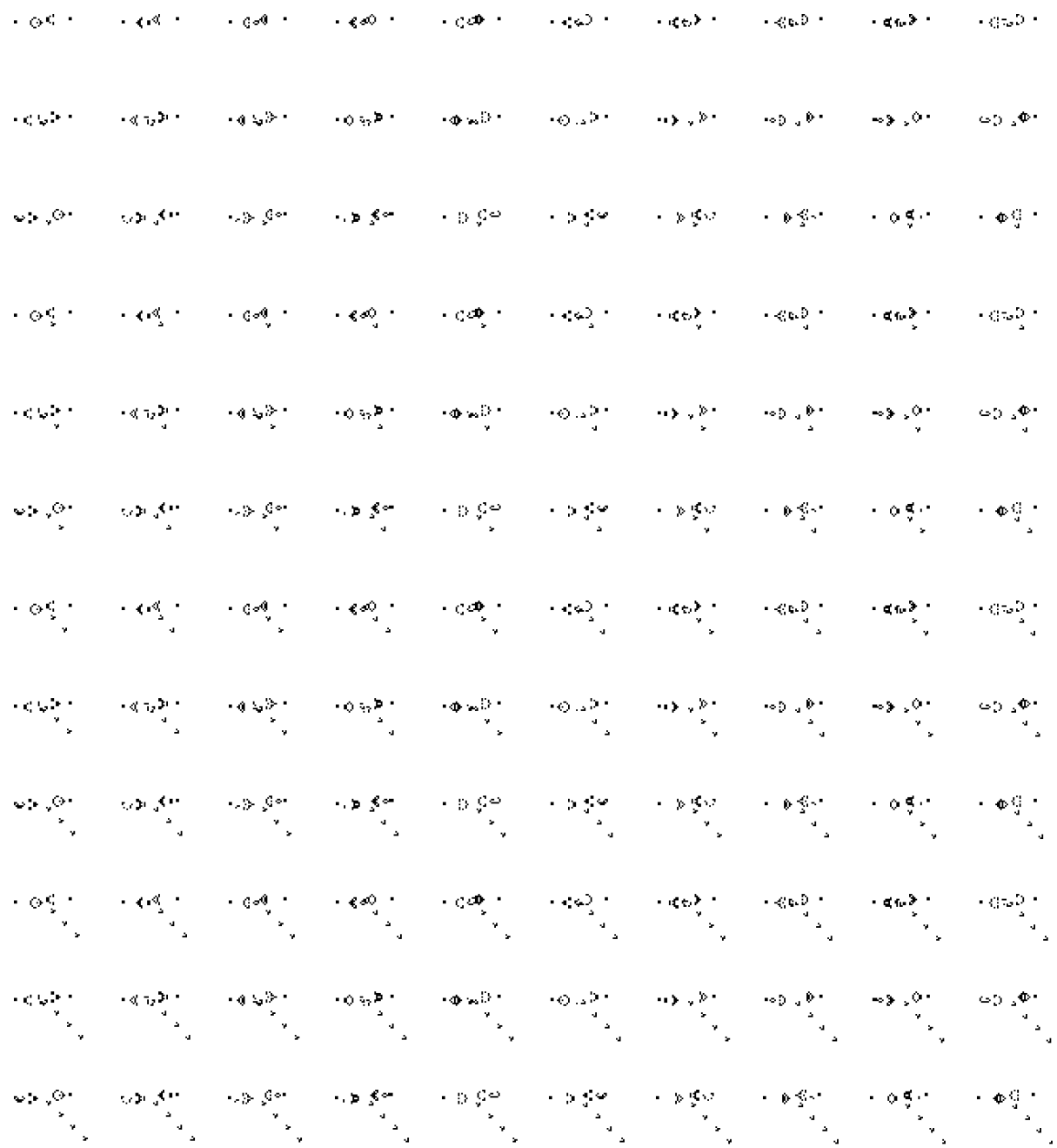


Figura 1.10: Configuración conocida como Glider Cannon. $G^i(c_0)$ para i = 1,...,120.

1.3. Propiedades topológicas

Una primera característica de los autómatas celulares es el hecho de que la función de transición global se aplique uniformemente en el espacio. Con base en la exposición desarrollada en [7] por el Dr. Jarkko Kari, encapsulamos la idea anterior en el siguiente resultado

Teorema 1.1. *Sea G una función de transición global arbitraria y τ una traslación. Entonces G y τ conmutan,* i.e., *$G \circ \tau = \tau \circ G$.*

$$\begin{array}{ccc} S^{\mathbb{Z}^d} & \xrightarrow{G} & S^{\mathbb{Z}^d} \\ \downarrow{\scriptstyle \tau} & & \downarrow{\scriptstyle \tau} \\ S^{\mathbb{Z}^d} & \xrightarrow{G} & S^{\mathbb{Z}^d} \end{array}$$

Demostración. Sean τ la traslación determinada por $r \in \mathbb{Z}^d$, G la función de transición global del autómata $A = (d, S, N, f)$, donde $N = (n_1, \ldots, n_m)$, $c \in S^{\mathbb{Z}^d}$ y $n \in \mathbb{Z}^d$ arbitrarios. Entonces

$$\begin{aligned} \tau(G(c))(n) = G(c)(n+r) &= f[c(n+r+n_1), \ldots, c(n+r+n_m)] \\ &= f[\tau(c)(n+n_1), \ldots, \tau(c)(n+n_m)] = G(\tau(c))(n). \end{aligned}$$

Luego, $\tau(G(c)) = G(\tau(c))$ de forma que $G \circ \tau = \tau \circ G$. □

Observación 1.1. *El hecho de que G conmute con traslaciones significa que la aplicación de la regla de evolución es la misma para todas las celdas.*

El teorema anterior motiva la siguiente definición:

Definición 1.8. *Una configuración $c \in S^{\mathbb{Z}^d}$ es r-periódica si*

$$c(n) = c(n+r) \quad \forall n \in \mathbb{Z}^d,$$

es decir, c es invariante bajo una r-traslación.

Nuestro objetivo es estudiar las configuraciones que pueden o no ser alcanzadas por un autómata celular dado. Para esto, necesitamos dotar al espacio de configuraciones $S^{\mathbb{Z}^d}$ de una métrica. Lo haremos como lo sugiere el autor en [16].

Intuitivamente queremos que dos configuraciones que toman valores parecidos estén cerca, y dos configuraciones que difieran mucho en sus valores estén lejos. Siendo más precisos, una vez escogida arbitrariamente una celda que indexaremos con el $0_{\mathbb{Z}^d}$, queremos que dos configuraciones estén cerca si sus valores alrededor de la celda $0_{\mathbb{Z}^d}$ coinciden y, en otro caso, queremos que estén lejos.

Procedemos a dar la siguiente definición y un teorema.

Definición 1.9. *Sean $c_1, c_2 \in S^{\mathbb{Z}^d}$ dos configuraciones de un autómata celular d-dimensional. La **distancia** entre c_1 y c_2 está dada por*

$$d(c_1, c_2) = \begin{cases} 0 & para \quad c_1 = c_2 \\ 2^{-mín\{\|x\|_\infty | c(x) \neq d(x)\}} & para \quad c_1 \neq c_2 \end{cases} \tag{1.1}$$

donde $x \in \mathbb{Z}^d$ *y*

$$\|(x_1, \ldots, x_d)\|_\infty = máx\{|x_1|, \ldots, |x_d|\}$$

Teorema 1.2. *La función d definida en 1.1 es una métrica.*

Demostración. Las primeras dos propiedades de métrica son claras a partir de la definición. Basta entonces con probar la desigualdad del triángulo.

Basta con ver que $d(c, b) \leq \text{máx}\{d(c, e), d(b, e)\}$, pues máx $\{d(c, e), d(b, e)\} \leq d(c, e) + d(b, e)$.

Para hacer más clara la demostración usaremos la siguiente notación:

$$B := \{\|x\|_\infty : b(x) \neq e(x)\}$$
$$C := \{\|x\|_\infty : c(x) \neq e(x)\}$$

Sean $c, b, e \in S^{\mathbb{Z}^d}$ configuraciones distintas por pares.

Notemos que para toda $x \in \mathbb{Z}^d$ tal que $c(x) \neq b(x)$ se tiene que $c(x) \neq e(x)$ o bien $b(x) \neq e(x)$ (pues si suponemos que $c(x) = e(x)$ y $b(x) = e(x)$ deducimos que $c(x) = b(x)$).

Así

$$\{x \in \mathbb{Z}^d : c(x) \neq b(x)\} \subseteq \{x \in \mathbb{Z}^d : c(x) \neq e(x)\} \cup \{x \in \mathbb{Z}^d : b(x) \neq e(x)\}$$

y por lo tanto

$$\{\|x\|_\infty : c(x) \neq b(x)\} \subseteq C \cup B$$

.

Luego

$$\text{mín}\{\|x\|_\infty : c(x) \neq b(x)\} \geq \text{mín}(C \cup B)$$

y como $\text{mín}(C \cup B) = \text{mín}\{\text{mín } C, \text{ mín } B\}$ entonces

$$\text{mín}\{\|x\|_\infty : c(x) \neq b(x)\} \geq \text{mín}\{\text{mín}C, \text{mín}B\}$$

de forma que $\text{mín}\{\|x\|_\infty : c(x) \neq b(x)\} \geq \text{mín}C$ o bien $\text{mín}\{\|x\|_\infty : c(x) \neq b(x)\} \geq \text{mín}B$. Así $d(c, b) \leq d(c, e)$ o bien $d(c, b) \leq d(b, e)$ y por lo tanto $d(c, b) \leq \text{máx}\{d(c, e), d(b, e)\}$. □

Observación 1.2. *Para el caso* $\mathbb{Z}^d = \mathbb{Z}$ *tenemos que* $\|x\|_\infty = |x|$ *de forma que 1.1 se simplifica como*

$$d(c, d) = \begin{cases} 0 & para \quad c = d \\ 2^{-mín\{|x| : c(x) \neq d(x)\}} & para \quad c \neq d \end{cases} \tag{1.2}$$

Ejemplo 1.3. *Las siguientes configuraciones c, d son tales que $d(c,d) = 2^{-2}$ ya que la primera celda en la que sus valores difieren es la celda 2, es decir, mín$\{|x| : c(x) \neq d(x)\} = 2$.*

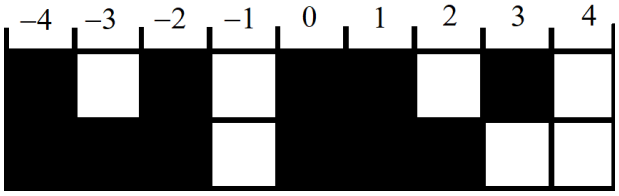


Figura 1.11: Distancia entre dos configuraciones

Teorema 1.3. *Una sucesión de configuraciones $(c_k)_{k\in\mathbb{N}}$ converge a $c \in S^{\mathbb{Z}^d}$ si y sólo si para toda $x \in \mathbb{Z}^d$ existe $k \in \mathbb{N}$ tal que $c_i(x) = c(x)$ para toda $i \geq k$.*

Demostración. Supongamos primero que para toda $x \in \mathbb{Z}^d$ existe $k \in \mathbb{N}$ tal que $c_i(x) = c(x)$ para toda $i \geq k$, y sea $\epsilon > 0$. Por el principio arquimedeano existe una $N \in \mathbb{N}^+$ tal que $\frac{1}{2^N} < \epsilon$. Consideremos el conjunto $X := \{x : \quad \|x\|_\infty \leq N+1\}$ y su correspondiente conjunto de índices $\{k : \exists x \in X, c_i(x) \geq c(x) \quad \forall i \geq k\}$. Dicho conjunto es finito, por lo que definimos $\hat{k} :=$ máx$\{k : \exists x \in X, c_i(x) \geq c(x) \quad \forall i \geq k\}$.

Así, por elección de $\hat{k}$, se tiene que $\forall x \in \mathbb{Z}^d$ tal que $\|x\|_\infty \leq N+1 \quad c_i(x) = c(x) \quad \forall i \geq \hat{k}$, de forma que mín$\{\|x\|_\infty : c_i(x) \neq c(x)\} > N \quad \forall i > \hat{k} \quad$ y así $d(c_i, c) < \frac{1}{2^N} < \epsilon$.

Conversamente, supongamos que $(c_k)_{k\in\mathbb{N}} \longrightarrow c$.

Consideramos la N tal que $\forall i \geq N$ se tiene que $\|x\|_\infty <$ mín$\{\|x\|_\infty : c_i(x) \neq c(x)\}$. Dicha N existe pues por hipótesis para toda $\epsilon > 0$ existe una N_ϵ tal que $\forall i > N_\epsilon, \quad log_2(\frac{1}{\epsilon}) <$ mín$\{\|x\|_\infty : c_i(x) \neq c(x)\}$. Por elección de N se cumple que $c_i(x) = c(x) \quad \forall i \geq N$. □

Observación 1.3. *El teorema anterior nos dice que si nos fijamos en el recorrido de una celda a través de la sucesión convergente $c_1, c_2, \ldots$, a partir de algún momento siempre veremos el mismo estado en esa celda. En términos gráficos, a partir de algún momento siempre la veremos del mismo color.*

Observación 1.4. *Los límites de sucesiones convergentes en $S^{\mathbb{Z}^d}$ son únicos.*

Demostración. Sabemos que todo espacio métrico es de Hausdorff [14], por lo que los límites de sucesiones convergentes son únicos. □

Una enorme ventaja de dotar al espacio de configuraciones con la topología anterior es que todas las funciones de transición global G de un autómata celular resultan ser funciones continuas. Replicaremos la demostración presentada en [7].

Teorema 1.4. *Sea $G : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$ la función de transición global del autómata $A = (d, S, Nf)$. Entonces G es continua.*

Demostración. Probaremos que para cualquier sucesión convergente de configuraciones $c_1, c_2, \ldots$ tales que $lim_{n\to\infty} c_n = c$ se cumple que la sucesión $G(c_1), G(c_2), \ldots$ converge a $G(c)$.

Sea G la función de transición global del autómata $A = (d, S, Nf)$ donde $N = (n_1, \ldots, n_m)$ y sea $n \in \mathbb{Z}^d$. Dado que $lim_{k\to\infty} c_k = c$, para cada $j = 1, \ldots, m$ existe $k_j \in \mathbb{Z}^+$ tal que

$$c_i(n + n_j) = c(n + n_j) \quad \forall i \geq k_j$$

Sea $k = \text{máx}\{k_1, \ldots, k_m\}$. Entonces, para toda $i \geq k$ se tiene que

$$\begin{aligned} G(c_i)(n) &= f[c_i(n + n_1), \ldots, c_i(n + n_m)] \\ &= f[c(n + n_1), \ldots, c(n + n_m)] \\ &= G(c)(n) \end{aligned}$$

Así, dado que $n \in \mathbb{Z}^d$ fue arbitrario, se tiene que $G(c_1), G(c_2), \ldots$ converge a $G(c)$.

□

Observación 1.5. *Por [14] sabemos que*

Una sucesión (x_n) de puntos del espacio producto $\Pi_{\alpha\in J} X_\alpha$ converge a x si y sólo si $\pi_\alpha(x_n)$ converge a $\pi_\alpha(x)$ para cada $\alpha \in J$.

Dado que el teorema 1.3 prueba que la convergencia en el espacio de funciones $S^{\mathbb{Z}^d}$ es la convergencia puntual, nos sugiere ver a dicho espacio como un espacio producto.

Teorema 1.5. *Consideremos el espacio de configuraciones $S = \{0,1\}^{\mathbb{Z}^d}$ y sea $(\{0,1\}, \tau)$ un espacio topológico, donde $\tau = \mathcal{P}(\{0,1\})$ es la topología discreta. Consideremos el espacio producto*

$$\prod_{i\in\mathbb{Z}^d} \{0,1\} := \{0,1\}^{\mathbb{Z}^d}$$

con la topología producto τ_π. Entonces τ_π coincide con la topología inducida por la métrica 1.1 cuando $S = \mathbb{Z}_2$.

Demostración. Dado que los conjuntos $\prod_{i\in\mathbb{Z}^d}\{0,1\}$ y $\{0,1\}^{\mathbb{Z}^d}$ son numerables, entonces $(\{0,1\}^{\mathbb{Z}^d}, d) \cong (\prod_{i\in\mathbb{Z}^d}\{0,1\}, d)$ [14]. Ahora bien, el espacio $(\prod_{i\in\mathbb{Z}^d}\{0,1\}, \tau_\pi)$ con la topología producto es un espacio metrizable, pues es producto numerable de metrizables (teorema de Nagata-Smirnov). Más aún, la métrica que induce la topología producto está dada por

$$d(x,y) = \sum_{i=0}^{\infty} 2^{-i} \frac{d_i(x_i, y_i)}{1 + d_i(x_i, y_i)}$$

donde $d_i(x_i, y_i)$ es la métrica discreta. Luego, $d(x,y)$ también es la métrica discreta. Así, por el teorema 1.3 los espacios $(\prod_{i\in\mathbb{Z}^d}\{0,1\}, d)$ y $(\prod_{i\in\mathbb{Z}^d}\{0,1\}, d(x,y))$ son dos espacios

métricos con las mismas sucesiones convergentes, por lo que su topología es la misma, en particular $(\prod_{i\in\mathbb{Z}^d}\{0,1\},d) \cong (\prod_{i\in\mathbb{Z}^d}\{0,1\},\tau_\pi)$. Concluimos entonces que $(\{0,1\}^{\mathbb{Z}^d},d) \cong (\prod_{i\in\mathbb{Z}^d}\{0,1\},\tau_\pi)$. □

El teorema anterior también es cierto para $S^{\mathbb{Z}^d}$ arbitrario pero finito (ver [16]). La prueba anterior, para el caso de $S = \{0,1\}^{\mathbb{Z}^d}$, es relevante porque son los autómatas con los que estaremos trabajando.

Daremos ahora una base para la topología τ_π que nos será útil más adelante. Sabemos que los básicos de τ_π son los subconjuntos de $S^{\mathbb{Z}}$ de la forma

$$\prod_{i\in\mathbb{Z}} B_i$$

tales que

1. $B_i \in \tau$ para toda $i \in \mathbb{Z}^d$
2. $B_i = S$ para toda $i \in \mathbb{Z}^d$ salvo para un número finito de ellas. [14]

Sea $c \in S^{\mathbb{Z}^d}$ y $D \subset \mathbb{Z}^d$ un conjunto finito. Definimo entonces el **cilindro** como

$$Cil(c,D) = \{e \in S^{\mathbb{Z}^d} | \quad e(x) = c(x) \quad \forall x \in D\}$$

Notemos que el cilindro determinado por c y D es el conjunto de configuraciones que coinciden con c dentro del dominio D.

Por lo dicho anteriormente, el conjunto de cilindros es una base para la topología τ_π.

De ahora en adelante denotaremos al espacio $(S^{\mathbb{Z}^d},\tau_\pi)$ simplemente como $S^{\mathbb{Z}^d}$, salvo cuando sea relevante destacar la topología. Los siguientes lemas nos serán útiles más adelante.

Lema 1.1. *Sean $(X,\tau_X),(Y,\tau_Y)$ espacios topológicos, $f : X \longrightarrow Y$ una función y S una subbase de la topología τ_Y. Entonces f es continua si y sólo si $f^{-1}(S_i) \in \tau_X$ para todo $S_i \in S$.*

Demostración. Si f es continua, dado que cada $S_i \in \tau_Y$ entonces $f^{-1}(S_i) \in \tau_X$.

Conversamente, dado $U \in \tau_X$, como S es subbase entonces $U = \cup_{j\in J} \cap_{i=1}^n S_i$ para algunos S_i de forma que $f^{-1}(U) = f^{-1}(\cup_{j\in J} \cap_{i=1}^n S_i) = \cup_{j\in J} \cap_{i=1}^n f^{-1}(S_i)$. Dado que por hipótesis $f^{-1}(S_i) \in \tau_X$ entonces f es continua. □

Lema 1.2. *Sean (X,τ_X) un espacio topolgico y Y,Z conjuntos. Si existe una función $f : Y \longrightarrow Z$ entonces existe una función $f^* : (X^Z,\tau_\pi) \longrightarrow (X^Y,\tau_\pi)$ continua dada por $f^*(\hat{z}) = \hat{z} \circ f$.*

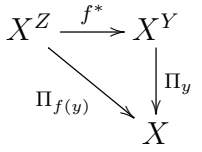
Demostración. Veamos que $f^* : (X^Z, \tau_\pi) \longrightarrow (X^Y, \tau_\pi)$ dada por

$$f^*(\hat{z}) = \hat{z} \circ f$$

es continua. Por el lema 1.1 basta con ver que la preimagen de subbásicos es abierta.

Sea S_i subbásico de τ_X en X^Y. Por definición de topología producto, $S_i = \Pi_y^{-1}(U_i)$ donde Π_y denota a la proyección y-ésima y $U_i \in \tau_X$. Así, $(f^*)^{-1}(\Pi_y^{-1}(U_i)) = (\Pi_y \circ f^*)^{-1}(U_i)$.

Veamos ahora que $\Pi_y \circ f^* = \Pi_{f(y)}$.



Sea $\hat{z} \in X^Z$ una función de Z en X. Entonces por definición de proyección $\Pi_{f(y)}(\hat{z}) = \hat{z}(f(y)) = (\hat{z} \circ f)y$.

Por otro lado, por definición de f^*, $\Pi_y \circ f^*(\hat{z}) = \Pi_y(\hat{z} \circ f)$ y por definición de proyección $\Pi_y(\hat{z} \circ f) = (\hat{z} \circ f)(y)$.

Así, $\Pi_y \circ f^* = \Pi_{f(y)}$, de forma que $(\Pi_y \circ f^*)^{-1}(U_i) = \Pi_{f(y)}^{-1}(U_i)$ es un abierto pues es la preimagen de un aberto bajo una proyección con la topología producto. Luego f^* es continua. □

Lema 1.3. *Sean X, Y, Z conjuntos. Si existe una función $f : Y \longrightarrow Z$ biyectiva entonces existe una función $f^* : X^Z \longrightarrow X^Y$ biyectiva dada por $f^*(\hat{z}) = \hat{z} \circ f$ y tal que $(f^*)^{-1} = (f^{-1})^*$.*

Demostración. $f^* : X^Z \longrightarrow X^Y$ dada por $f^*(\hat{z}) = \hat{z} \circ f$. Veamos que $(f^*)^{-1} : X^Y \longrightarrow X^Z$ dada por $(f^*)^{-1}(\hat{y}) = \hat{y} \circ f^{-1}$ es su inversa.

$$f^* \circ (f^*)^{-1}(\hat{y}) = f^*(\hat{y} \circ f^{-1}) = \hat{y} \circ f^{-1} \circ f = \hat{y}$$

$$(f^*)^{-1} \circ f^*(\hat{z}) = (f^*)^{-1}(\hat{z} \circ f) = \hat{z} \circ f \circ f^{-1} = \hat{z}$$

Así, f^* y $(f^*)^{-1}$ son inversas.

Además, notemos que $(f^{-1})^*(\hat{y}) = \hat{y} \circ f^{-1} = (f^*)^{-1}(\hat{y})$, de tal manera que $(f^{-1})^* = (f^*)^{-1}$.

□

Corolario 1.1. *Sean V, W conjuntos. Sea una función $g : V \longrightarrow W$ biyectiva, entonces existe una función $g^* : (X^W, \tau_\pi) \cong (X^V, \tau_\pi)$ que es un homeomorfismo.*

Demostración. Como $g : V \longrightarrow W$ es biyectiva entonces, por 1.2 g^* es continua, y por 1.3 g^* es biyectiva y su inversa es $(g^{-1})^*$.

Dado que $g^{-1} : V \longleftarrow W$ es biyectiva, entonces por 1.2 $(g^{-1})^*$ es continua, y por 1.3 $(g^{-1})^*$ es biyectiva.

$$X^W \underset{(g^{-1})^*}{\overset{g^*}{\rightleftarrows}} X^V$$

Así, $g^* : X^W \longrightarrow X^V$ es continua y biyectiva con inversa continua, de forma que es homeomorfismo *i.e.* $X^W \cong X^V$.

□

Teorema 1.6.

1. *El espacio* $\{0,1\}^{\mathbb{Z}^d}$ *es un espacio de Cantor.*
2. *El espacio* $\{0,1\}^{\mathbb{Z}^d}$ *es compacto, completo, totalmente acotado, perfecto y totalmente disconexo.*

Demostración. 1. Veamos que $(\{0,1\}^{\mathbb{N}}, \tau_\pi) \cong (C, \tau_{\mathbb{R}})$ donde $C \subset \mathbb{R}$ es el conjunto de Cantor clásico. Sea $h : (\{0,1\}^{\mathbb{N}}, \tau_\pi) \to (C, \tau_{\mathbb{R}})$ dada por

$$h((a_i)_{i\in\mathbb{N}}) = \sum_{n=0}^{\infty} \frac{2a_n}{3^{n+1}}.$$

Por la unicidad de la representación en base 3 de cualquier número h es inyectiva. Dado que el conjunto de Cantor son los elementos del intervalo $[0,1]$ tales que no tienen al 1 en su representación ternaria, entonces h es biyectiva.

Veamos ahora que h es continua. Por el lema 1.1, basta con ver que las preimágenes de los subbásicos son abiertos. Así, sea $B_{1/n}(x) \cap C \in \tau_R$ p.a. $n \in \mathbb{Z}^+, x \in C$, entonces $h^{-1}(B_{1/n}(x) \cap C) = \{(a_i)_{i\in\mathbb{N}} : a_i = x_i \forall i = 1, \ldots, n+1\} \in \tau_\pi$.

Veamos ahora que h^{-1} es continua. Sea $U \in \tau_\pi$ un abierto, de forma que $U = \{(a_i) : i \in I$ está fijo para un número finito de índices$\}$. Para cada i, sea $\{i_1, \ldots, i_n\} \subset \{0,1\}^n$ el conjunto de índices fijos en a_i. Entonces $(h^{-1})^{-1}(U) = \cup_{i\in I}(\cap_{j=1}^{n} B_{\frac{1}{3^{i_j}}}(a_i))^c \in \tau_{\mathbb{R}}$.

Así, $(\{0,1\}^{\mathbb{N}}, \tau_\pi) \cong (C, \tau_{\mathbb{R}})$. Dado que $|\mathbb{N}| = |\mathbb{Z}^d|$ entonces, por el lema 1.3, $|\{0,1\}^{\mathbb{N}}| = |\{0,1\}^{\mathbb{Z}^d}|$. Luego, por el corolario 1.1, se tiene que $(\{0,1\}^{\mathbb{N}}, \tau_\pi) \cong (\{0,1\}^{\mathbb{Z}^d}, \tau_\pi)$, y dado que $(\{0,1\}^{\mathbb{N}}, \tau_\pi) \cong (C, \tau_{\mathbb{R}})$ entonces $(\{0,1\}^{\mathbb{Z}^d}, \tau_\pi) \cong (C, \tau_{\mathbb{R}})$, lo cual demuestra el resultado.

2. Dado que $(\{0,1\}^{\mathbb{Z}^d}, \tau_\pi) \cong (C, \tau_{\mathbb{R}})$ y C es compacto, completo, totalmente acotado, perfecto y totalmente disconexo, entonces $\{0,1\}^{\mathbb{Z}^d}$ también lo es.

□

Observación 1.6. *Dado que el teorema 1.5 es cierto para $S^{\mathbb{Z}^d}$ con S arbitrario (finito), todos los corolarios anteriores pueden ser generalizados a $S^{\mathbb{Z}^d}$ (ver [16]).*

Por el teorema 1.4 sabemos que la función de transición global G es continua, y la observación anterior junto con el teorema 1.6 nos aseguran que $S^{\mathbb{Z}^d}$ es compacto para todo conjunto finito de estados S. Esto motiva la siguiente

Definición 1.10. *Sea X un conjunto compacto y $F : X \longrightarrow X$ una función continua. Entonces el par (X, F) es un sistema dinámico.*

Observación 1.7. *El par $(S^{\mathbb{Z}^d}, G)$ es un sistema dinámico.*

1.4. Jardines del Edén

Para conocer las capacidades y los límites de un autómata celular, nos interesa estudiar las configuraciones que puede o no puede alcanzar. Como veremos más adelante, a las configuraciones que no se alcanzan las llamaremos **Jardines del Edén**. Procederemos a exponer los conceptos como en [16].

Definición 1.11. *Un **autómata celular es inyectivo, sobreyectivo o biyectivo** si su función de transición global G es inyectiva, sobreyectiva o biyectiva respectivamente.*

Tenemos que asegurarnos que la inversa de una función de transición global G no sólo existe, sino que corresponde a la función de transición global de algún autómata celular. Para esto necesitamos una caracterización de las funciones de transición global de autómatas celulares, como la dada en el teorema que sigue de Curtis-Hedlund-Lyndon.

Teorema 1.7. *(Hedlund 1969) Una función $G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ es la función de transición global de algún autómata celular si y sólo si G es continua y conmuta con traslaciones.*

Demostración. La demostración puede consultarse en [6]. □

Corolario 1.2. *(Hedlund 1969) La función de transición global G de un autómata celular es reversible si y sólo si es biyectiva.*

Demostración. Sea G la función de transición global de un autómata celular $A = (d, S, N, f)$. Por definición, si G es reversible entonces es biyectiva. Conversamente, si G es biyectiva, dado que $S^{\mathbb{Z}^d}$ es compacto y G es continua (1.4, 1.6) entonces G^{-1} es continua. Dado que G conmuta con traslaciones (1.7) entonces:

$$\begin{aligned}\tau \circ G^{-1} &= (G^{-1} \circ G) \circ \tau \circ G^{-1} = G^{-1} \circ (G \circ \tau) \circ G^{-1} \\ &= G^{-1} \circ (\tau \circ G) \circ G^{-1} = G^{-1} \circ \tau \circ (G \circ G^{-1}) \\ &= G^{-1} \circ \tau\end{aligned}$$

Por el teorema 1.7 se tiene que G^{-1} es la función de transición global de un autómata celular. □

1.4.1. El teorema del balance

Definición 1.12. *Sea $A = (d, S, N, f)$ un autómata celular. Una configuración c es un* ***Jardín del Edén*** *si no tiene preimágenes, es decir si $G^{-1}(c) = \emptyset$.*

Observación 1.8. *Un autómata celular $A = (d, S, N, f)$ tiene un Jardín del Edén si y sólo si A no es sobreyectivo.*

Nuestro objetivo es estudiar las condiciones bajo las cuáles se dan Jardines del Edén. Un primer teorema afirma que un desbalance en la regla de evolución de un autómata celular implica la existencia de Jardines del Edén. Veamos un ejemplo procediendo como en [16]

Ejemplo 1.4. *Consideremos el autómata celular* 126 *dado por*

$$f(a_{i-1}, a_i, a_{i+1}) = \begin{cases} 0 & a_{i-1} = a_i = a_{i+1} \\ 1 & e.o.c \end{cases}$$

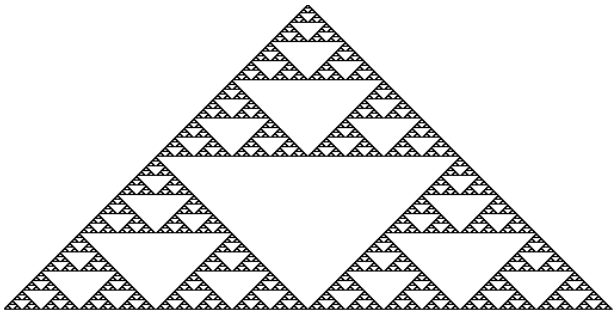


Figura 1.12: Regla126

Notemos que de las ocho posibles combinaciones con tres vecindades, dos de ellas van a dar al 0 *y seis de ellas van a dar al* 1*. Veamos cómo este desbalance implica la existencia de Jardines del Edén.*

Sea $k \in \mathbb{Z}^+$ y sea $c \in \{0,1\}^{\mathbb{Z}}$ una configuración tal que

$$c(3) = c(6) = \cdots = c(3k) = 0$$

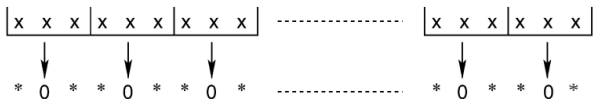


Figura 1.13: La configuración c debajo de su preimagen.[4]

[4]Imagen tomada de [16] pp.10.

El total de opciones para los estados faltantes (marcados con un asterisco en la imagen) es $2^{2k} = 4^k$. Dado que $|f^{-1}(0)| = 2$, entonces el total de preimágenes e (marcado con "X" en la imagen) tales que $f(e(3k-1), e(3k), e(3k+1)) = 0$ es 2^k. Como claramente $2^k < 4^k$ se tiene que no todas las elecciones de c tienen una preimagen bajo la Regla 126, es decir que no es sobreyectiva.
Así, cualquier configuración que contenga a un patrón que no tenga preimagen será un Jardín del Edén, por lo que existen Jardines del Edén en la Regla 126.

El ejemplo anterior muestra también que basta con encontrar un arreglo finito de celdas sin preimagen bajo G para que exista un Jardín del Edén. Introduzcamos entonces el concepto de patrón, seguido del de huérfano.

Definición 1.13. *Un **patrón***

$$p = (D, g)$$

es una configuración parcial donde $D \subset \mathbb{Z}^d$ y $g : D \longrightarrow S$ es una función que a cada elemento de D le asigna un estado.

Observación 1.9. *Todo cilindro $Cil(c, D) = \{c \in S^{\mathbb{Z}^d} : c(x) = g(x) \quad \forall x \in D\}$ está determinado por un patrón finito $p = (D, g)$.*

Podemos pensar a los patrones como un conjunto finito de bloques de celdas de una configuración.

Definición 1.14. *Sea $A = (d, S, N, f)$ un autómata celular con $N = (x_1, \ldots, x_n)$ y $D \subset \mathbb{Z}^d$ un conjunto finito. La **vecindad del conjunto D** es el conjunto*

$$N(D) := \{x + x_i \quad | \quad x \in D, i = 1, 2, \ldots, n\}$$

de vecinos de elementos de D.

Sea $D \subset \mathbb{Z}^d$, $N = (x_1, \ldots, x_n)$ un vector de vecindad y $q = (N(D), h)$ un patrón. Al aplicar la regla local f al patrón q obtendremos un nuevo patrón $p = (D, g)$ donde

$$g(x) = f[h(x + x_1), \ldots, h(x + x_n)]$$

Al mapeo $q \mapsto p$ lo denotaremos por $G^{N(D) \longrightarrow D}$. Notemos que con esta notación la función de transición global G es $G^{\mathbb{Z}^d \longrightarrow \mathbb{Z}^d}$.

Definición 1.15. *Sean $A = (d, S, N, f)$ un autómata celular y G su función de transición global. Un patrón finito $p = (D, g)$ es un **huérfano** si no existe un patrón finito $q = (N(D), h)$ tal que $G^{N(D) \longrightarrow D}(q) = p$.*

Observación 1.10. *Un huérfano es un patrón finito que no puede aparecer en ninguna configuración al iterar la función G.*

Teorema 1.8. *Sean $A = (d, S, N, f)$ un autómata celular y G su función de transición global. Entonces A tiene un huérfano si y sólo si G no es sobreyectiva.*

Demostración. Si existe un huérfano entonces G claramente no es sobreyectiva, pues cualquier configuración que lo contenga será inalcanzable por G. Ahora bien, supongamos que G no es sobreyectiva. Dado que G es continua y $S^{\mathbb{Z}^d}$ es compacto entonces $G(S^{\mathbb{Z}^d})$ es compacto, y como $S^{\mathbb{Z}^d}$ es un espacio métrico entonces $G(S^{\mathbb{Z}^d})$ es cerrado, de forma que su complemento (el conjunto de jardines del Edén de A) es un abierto no vacío (pues $G(S^{\mathbb{Z}^d}) \neq S^{\mathbb{Z}^d}$). Dado que los cilindros son una base de la topología entonces existe un cilindro cuyos elementos son jardines del Edén. Así, el patrón que determina a ese cilindro es un huérfano. □

El teorema anterior nos lleva a un corolario importante.

Corolario 1.3. *Sea A un autómata celular. A tiene un Jardín del Edén si y sólo si A tiene un huérfano.*

Demostración. Es consecuencia inmediata de la observación 1.8 y el teorema 1.8. □

Ya estamos en condiciones para generalizar el ejemplo 1.4 mediante el teorema del balance. Dicho teorema afirma que en un autómata celular sobreyectivo todos los patrones finitos del mismo dominio deben tener el mismo número de preimágenes.

Teorema 1.9. ***(del balance)****[Maruoka, Kimura 1976]*
Sea $A = (d, S, N, f)$ un autómata celular sobreyectivo, y sea $D \subset \mathbb{Z}^d$ un conjunto finito. Entonces para todo patrón $p = (D, g)$ el número de patrones $q = (N(D), h)$ tales que

$$G^{N(D)\longrightarrow D}(q) = p$$

es $|S|^{|N(D)|-|D|}$.

Demostración. La demostración es análoga al ejemplo 1.4 y, dado que escapa a los objetivos del presente texto, referimos al lector a [7]. □

Para el caso de $d = 1, N = (x_1, x_2, x_3), S = \{0, 1\}$ tenemos el siguiente corolario.

Corolario 1.4. *Sea $A = (1, \{0, 1\}, N, f)$ un autómata celular sobreyectivo con $N = (x_1, x_2, x_3)$, y sea $D = \{x_i\} \subset \mathbb{Z}^d$. Entonces la regla de evolución f debe de estar balanceada, es decir, $|f^{-1}(0)| = |f^{-1}(1)| = 4$.*

Demostración. Basta con sustituir los valores de $|S| = 2, |N(D)| = 3, |D| = 1$ en el teorema anterior. □

Observación 1.11. *Notemos que la condición de balance sobre la regla de evolución f es necesaria más no suficiente para tener sobreyectividad. Por ejemplo, el autómata celular unidimensional 233 dado por $f(a, b, c) = 1$ si y sólo si $a + b + c \geq 2$, tiene una regla de evolución balanceada. Sin embargo, el teorema del balance exige que esto se cumpla para cualquier conjunto finito D, y en este caso no se cumple para patrones más extensos. Cualquier palabra de longitud 4 que contiene al menos un 1 es mapeada al 00, por lo que 00 tiene al menos cinco preimágenes ($\{0000, 1000, 0100, 0010, 0001\} \subset f^{-1}(00)$), pero el teorema del balance nos dice que cada preimagen debe de tener cardinalidad $2^2 = 4$, por lo que tiene un huérfano.*

Según [16], el teorema 1.9 implica que la medida uniforme de Bernoulli sobre el espacio de configraciones es invariante bajo aplicaciones de un autómata celular sobreyectivo. Es decir, que si escogemos de forma aleatoria y uniforme una configuración inicial c_0 y G es sobreyectiva, entonces $G^t(c_0)$ serán configuraciones aleatorias uniformes para toda $t \in \mathbb{Z}^+$.

1.4.2. El teorema del Jardín del Edén

Para probar el resultado estrella de esta sección, necesitamos lo siguiente:

Definición 1.16. *Sea A un autómata celular con regla de evolución f. Dos patrones finitos distintos $p = (N(D), f)$ y $q = (N(D'), f)$ con $p \neq q$ son* ***gemelos*** *si $G^{N(D)\longrightarrow D}(p) = G^{N(D')\longrightarrow D'}(q)$.*

Es decir, dos patrones son gemelos si podemos sustituir uno por el otro sin alterar la evolución del autómata.

Definición 1.17. *Un autómata celular es* ***localmente inyectivo*** *si y sólo si no tiene gemelos.*

Para los fines de este trabajo, presentaremos una demostración del famoso teorema del Jardín del Edén para autómatas celulares elementales. La prueba para el caso general es análoga, y referimos al lector a [13] y [15].

Teorema 1.10. ***(Jardín del Edén) [Moore y Myhill 1963]***
Sea A un autómata celular elemental. Entonces A tiene un Jardín del Edén si y sólo si A tiene gemelos.

Demostración. Supongamos primero que A tiene un Jardín del Edén. Por el corolario 1.3 A tiene un huérfano de longitud $m \in \mathbb{Z}^+$. Sea $k \in \mathbb{Z}^+$ arbitrario y consideremos todas las configuraciones cuyo soporte (celdas con estados distintos de 0) son de longitud $mk - 2$ (ver figura 1.14).

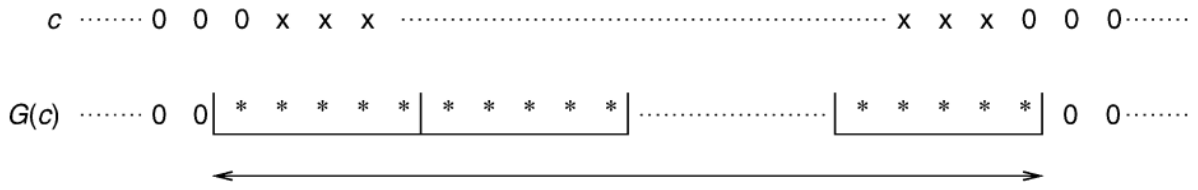


Figura 1.14: Configuraciones c con soporte contenido en un segmento de longitud $mk - 2$ y sus respectivas imágenes $G(c)$. [6]

Hay exactamente $2^{mk-2} = \frac{1}{4}2^{mk}$ configuraciones de ese tipo. Por otro lado, sabemos que $G(c)$ no puede ser el huérfano en cuestión, por lo que hay $(2^m - 1)^k$ distintas posibilidades para $G(c)$.

[6] Imagen tomada de [16] pp.13.

Dado que

$$lim_{k\to\infty}\frac{1}{4}\frac{(2^m)^k}{(2^m-1)^k} = \frac{1}{4}lim_{k\to\infty}(\frac{2^m}{2^m-1})^k = \infty$$

entonces para k suficientemente grande se tiene que

$$\frac{1}{4}2^{mk} > (2^m-1)^k$$

es decir, hay más opciones para c que para $G(c)$, lo cual implica que A no es localmente inyectivo y por lo tanto tiene gemelos.

Conversamente, supongamos que A tiene gemelos, digamos p, q de longitud m. Sea $k \in \mathbb{Z}^+$ arbitrario y consideremos k segmentos de longitud m. Cualquier patrón de longitud $mk-2$ que tiene una preimagen de longitud mk también tiene una preimagen de longitud mk en donde ninguno de los k segmentos de longitud m contiene al patrón p (si esto llegara a aparecer, lo podemos sustituir por q sin alterar la evolución del autómata). Así, para las 2^{mk-2} configuraciones posibles, sólo puede haber $(2^m-1)^k$ preimágenes y dado que para k suficientemente grande se tiene que

$$\frac{1}{4}2^{mk} > (2^m-1)^k$$

entonces algunas configuraciones no tienen preimágenes. Se sigue entonces que existe un huérfano, y por el corolario 1.3, existe un Jardín del Edén. □

El teorema anterior resulta extremadamente útil dado que es mucho más sencillo encontrar gemelos que Jardines del Edén.

Ejemplo 1.5. *Sea A el autómata denominado "Juego de la vida" introducido en el ejemplo 1.2. Notemos que un bloque de celdas muertas de 5×5 y un bloque de celdas muertas de 5×5 con la celda del centro viva son gemelos, lo cual garantiza la existencia de Jardines del Edén. Es interesante notar que, a pesar de que dicho autómata es Turing completo, no puede alcanzar todas las configuraciones posibles.*

Corolario 1.5. *Todo autómata celular inyectivo es sobreyectivo.*

Demostración. Si A es inyectivo entonces (claramente) es localmente inyectivo. Por el teorema anterior sabemos que no tiene Jardines del Edén, por lo cual es sobreyectivo. □

1.5. Consideraciones algorítmicas

Sean $A = (d, S, N, f)$ un autómata celular unidimensional ($d = 1$) y G su función de transición global. En [1] se presenta el siguiente algoritmo para decidir si G es sobreyectiva o no. Nuestro algoritmo construirá una gráfica finita en la que cada nodo será un subconjunto de tuplas en S^n cuyas imágenes bajo f son las mismas. Si la gráfica puede ser construida completamente, en un sentido que definiremos más adelante, entonces G es sobreyectiva, en caso contrario no lo es (ver figura 1.15) .

1. Inspeccionar la tabla de f para ver si existe un símbolo en S tal que no sea la imagen de ninguna n-tupla bajo f. En caso de no existir, A no es sobreyectivo. En caso de existir, continuar con el algoritmo.

2. Seleccionar un elemento $b \in S$ y sea el conjunto de n-tuplas $a_1, \ldots, a_n$ tales que $f(a_1, \ldots, a_n) = b$, el *único* nodo en el nivel cero.

3. Para cada nodo N en el nivel $i \geq 0$, construimos para cada $a \in S$, un nodo N_a en el nivel $i+1$ de la siguiente manera. Si $a_1, \ldots, a_n$ es un elemento del conjunto correspondiente a N, el conjunto correspondiente a N_a será el conjunto de todas las n-tuplas $a_2, \ldots, a_n, d$, $d \in S$ tales que $f(a_2, \ldots, a_n, d) = a$. Una arista etiquetada con a será trazada del nodo N al nodo N_a.

4. Si para cada $a_1, \ldots, a_n$ en N no existe tal elemento d, entonces este nodo N_a no se incluye en la gráfica, y así decimos que el nodo N es *terminal*. Decimos también que el símbolo *a hizo a N terminal*.

5. Si durante la construcción de la gráfica aparece el mismo nodo (asociado con el mismo subconjunto de S^n) de forma repetida en el mismo o en diferentes niveles, entonces cada uno será un nodo distinto de la gráfica pero solamente uno de ellos, escogido aleatoriamente, será *extendido*; es decir, sólo uno de ellos tendrá aristas conectándolo con nodos de niveles posteriores. Los nodos que no fueron extendidos se llamarán *nodos frontera*.

Observación 1.12. *El algoritmo anterior siempre termina, pues hay un número finito de subconjuntos de* S^n.

Antes de explicar por qué dicho algoritmo en efecto decide si G es sobreyectiva, ilustrémoslo con un ejemplo.

Ejemplo 1.6. *Consideremos el autómata celular* 126 *dado por*

$$f(a_{i-1}, a_i, a_{i+1}) = \begin{cases} 0 & a_{i-1} = a_i = a_{i+1} \\ 1 & e.o.c \end{cases}$$

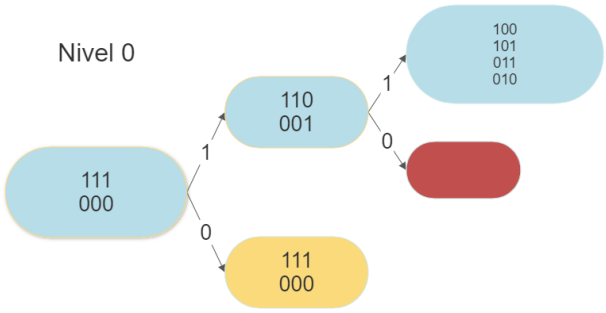


Figura 1.15: Gráfica construida a partir de la Regla 126. El nodo rojo es un nodo terminal, mientras que el nodo amarillo es un nodo frontera. Esto implica que G no es sobreyectiva y, más aún, que el patrón 010 es un huérfano.

La razón de ser del algoritmo está dada por el siguiente teorema.

Teorema 1.11. *Sean $A = (d, S, N, f)$ un autómata celular unidimensional ($d = 1$) y G su función de transición global. Entonces G es sobreyectiva si y sólo si la gráfica construida con el algoritmo anterior tiene un nodo terminal.*

Demostración. Supongamos que el nodo del nivel cero contienen a todos los subconjutos de n-tuplas tales que bajo f son mapeadas a $a \in S$, y que la gráfica tiene un nodo terminal hecho terminal por $b \in S$. Supongamos también que la secuencia finita $b_1, b_2, \ldots, b_n$ de símbolos conduce desde el nodo N_0 en el nivel cero hasta el nodo terminal N, es decir, existe una arista etiquetada con b_1 de N_0 a N_1 en el nivel 1, y otra etiquetada con b_2 de N_1 a N_2 en el nivel 2, $\ldots$, y otra etiquetada con b_n del nodo N_{n-1} al nodo terminal N. Por construcción de la gráfica y dado que b hizo terminal a N, el patrón $ab_1b_2, \ldots, b_n, b$ es un huérfano, y por el teorema 1.8 G no es sobreyectiva.

Conversamente, si G no es sobreyectiva, tiene un huérfano $ab_1b_2, \ldots, b_n, b$. Luego, por construcción de la gráfica, siguiendo las aristas etiquetadas con $b_1, \ldots, b_n$ llegaremos a un nodo terminal. □

En 1990 Jarkko Kari demostró mediante una reducción a un problema de teselaciones, que el problema de determinar si existe un Jardín del Edén en un autómata celular de dimensión mayor que 1 es indecidible *i.e.* no existe un algoritmo para determinar si existe o no un Jardín del Edén. Sin embargo, si se garantiza la existencia de un Jardín del Edén usando el teorema 1.10, se puede usar un algoritmo de búsqueda para encontrarlo. Al lector interesado lo referimos a [8].

1.6. Autómatas celulares elementales

Como ya hemos mencionado, con la finalidad de capturar las reglas fundamentales que rigen a los sistemas complejos, Wolfram estudió los programas más simples que existen: unidimensionales, con vecindades pequeñas y dos estados. A estos programas los llamó autómatas celulares elementales.

Definición 1.18. *Un* ***autómata celular elemental*** *es un autómata* $A = (d, S, N, f)$ *donde* $d = 1, S = \mathbb{Z}_2$ *y* $N = (-1, 0, 1)$.

Observación 1.13. *Para determinar* f*, basta con que especifiquemos* $f(x, y, z)$*, donde* $x, y, z \in \{0, 1\}$*. Así, basta con especificar la imagen de* 2^3 *elementos del dominio para determinar totalmente a* f*. Es común que dicha especificación se haga de manera pictórica: de derecha a izquierda, se enumeran todas las posibles combinaciones de una vecindad de* 3 *celdas y debajo de cada una de ellas se coloca su imagen bajo* f *(ver figura 1.16).*

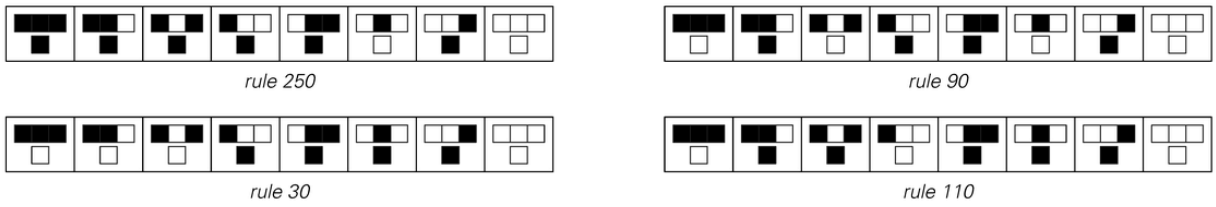


Figura 1.16: Esquematización de Wolfram. [7]

Una vez que fijamos el orden en el que vamos a escoger los elementos del dominio (de derecha a izquierda, enumerando todas las combinaciones en orden creciente en su representación en base 2*), podemos interpretar la sucesión de imágenes como un número en base* 2*, y referirnos al autómata con ese número.*

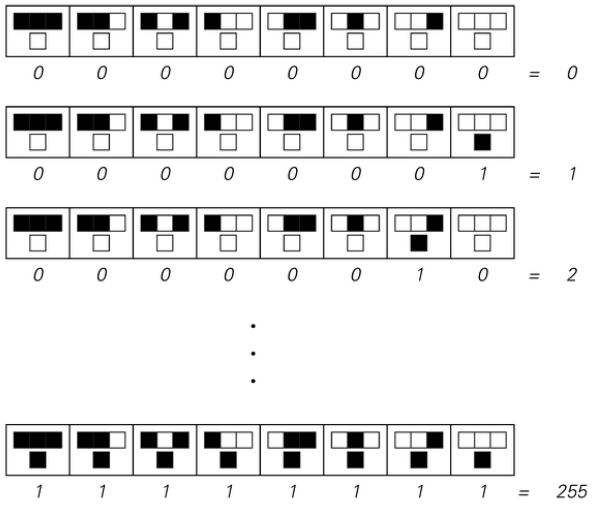


Figura 1.17: Etiquetado de Wolfram

Dicho etiquetado es conveniente ya que a partir de la descripción de f *podemos etiquetar al autómata con un único número entre* 0 *y* 2^8*, y dado un número en ese rango*

[7]Compilación de imágenes tomadas de [18].
[7]Compilación de imágenes tomadas de [18].

podemos recuperar toda la información referente a la función f.

Al autómata cuya función f le corresponde el número k lo llamaremos ***Regla*** *k.*

Ejemplo 1.7. *Dado que* $60_{10} = 00111100_2$*, entonces la esquematización de la Regla* 60 *es*

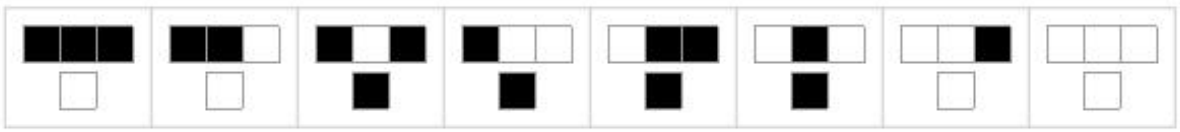


Figura 1.18: Regla 60. [8]

Podríamos preguntarnos, ¿cómo se comportan las 256 reglas posibles de autómatas celulares elementales? La forma más sencilla para responder esa pregunta es simplemente exhibiéndolos todos. Es común que, cuando se estudia el comportamiento de un autómata, se escoja una condición inicial sencilla, típicamente un único cuadro encendido (con un 1) en el centro del renglón inicial. Así, veamos cómo se comportan los autómatas celulares elementales al ser evolucionados a partir de la condición inicial más sencilla posible (ver figuras 1.19 y 1.20).

[8] Imagen tomada de [18].

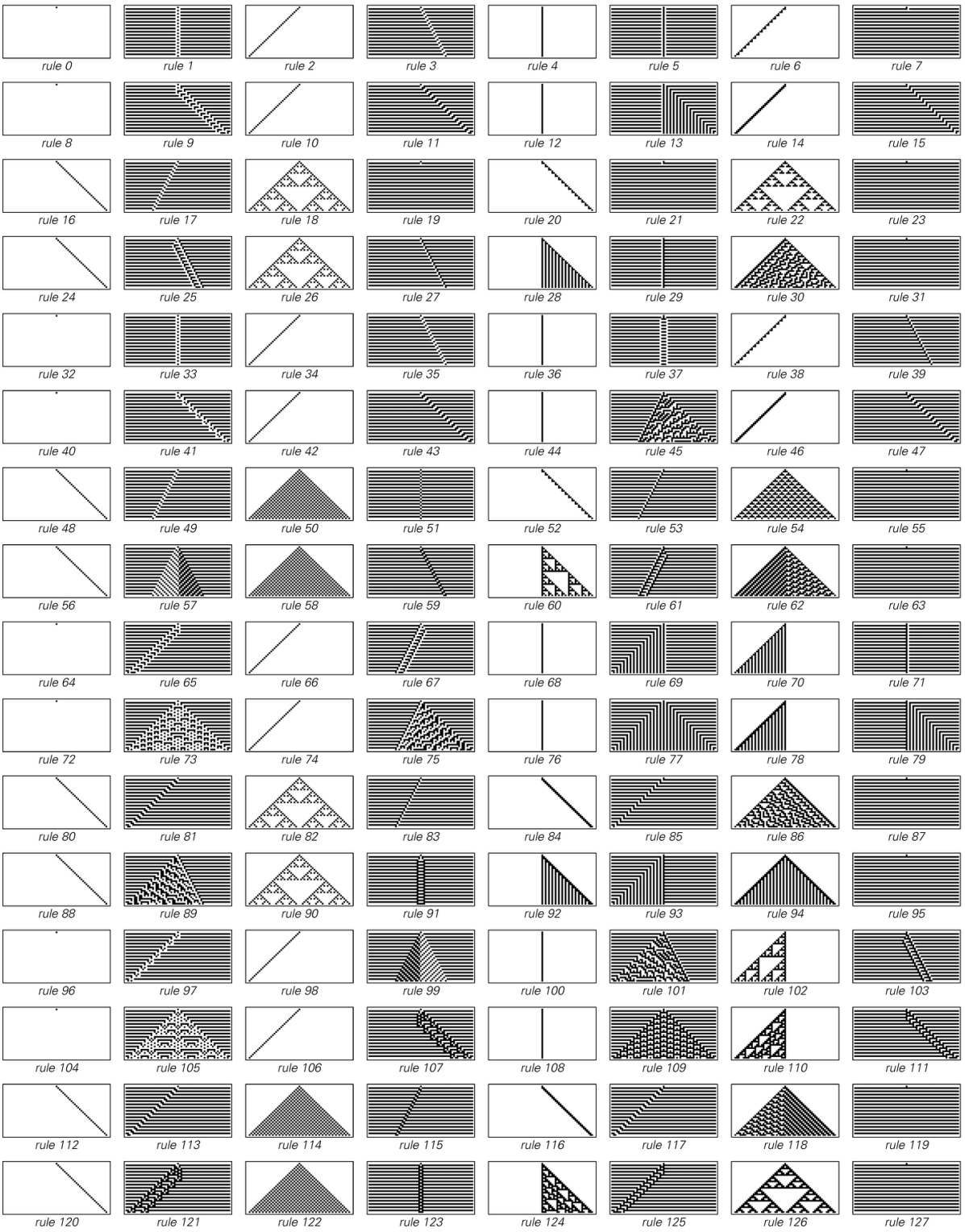


Figura 1.19: Primeros 127 autómatas celulares elementales. [9]

[9]Weisstein, Eric W. *Elementary Cellular Automaton.* From MathWorld: A Wolfram Web Resource. https://mathworld.wolfram.com/ElementaryCellularAutomaton.html

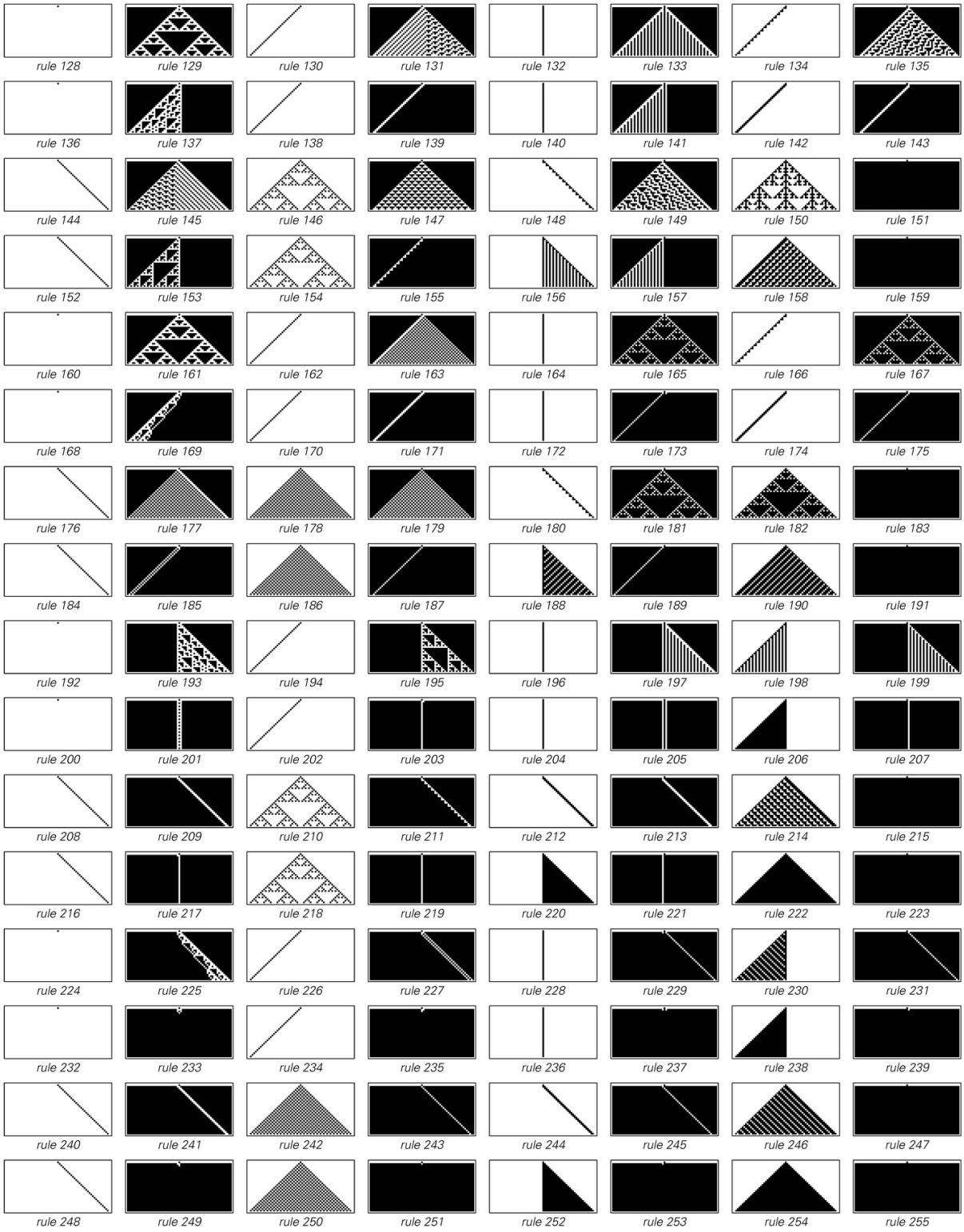


Figura 1.20: Últimos 127 autómatas celulares elementales. [10]

[10]Weisstein, Eric W. *Elementary Cellular Automaton.* From MathWorld: A Wolfram Web Resource.

En su libro *A New Kind of Science*, Wolfram estudia muchos otros tipos de autómatas celulares -y en general de modelos de cómputo-, como los de más de dos estados, autómatas celulares bidimensionales, autómatas celulares móviles, máquinas de Turing, sistemas de sustitución, sistemas de etiquetas, sistemas simbólicos, entre otros, y nota experimentalmente que ninguno de éstos presenta comportamiento esencialmente más complejo que los autómatas celulares elementales [18]. Así justifica que para estudiar complejidad emergente a partir de reglas sencillas, basta con limitarse al estudio de autómatas celulares elementales, como lo haremos en el presente trabajo.

https://mathworld.wolfram.com/ElementaryCellularAutomaton.html

Capítulo 2

Autómatas celulares aditivos

Existe un subconjunto de autómatas celulares elementales que resulta particularmente atractivo debido a la facilidad que presentan bajo el análisis algebraico. En principio, la regla de evolución $f : S^m \to S$ con $m \leq 3$ puede ser cualquier función. En esta sección consideraremos aquellos autómatas cuya regla de evolución f es del tipo

$$f(x, y, z) = \alpha_{-1}x + \alpha y + \alpha_{+1}z.$$

A este tipo de autómatas les llamaremos **aditivos**. A lo largo de esta sección procederemos como en [11], y ahora iniciamos con la siguiente definición.

Definición 2.1. *Sean $d = 1, S = \mathbb{Z}_2, N = (-1, 0, 1)$ y $f : \mathbb{Z}_2^m \to \mathbb{Z}_2$ con $m \leq 3$. Diremos que el autómata celular elemental $A = (d, S, N, f)$ es **aditivo** si $f(x, y, z) = \alpha_{-1}x + \alpha y + \alpha_{+1}z$, para $\alpha_i \in S$.*

Una pregunta relevante es, ¿cuántos de los 2^8 autómatas celulares con estados en $S = \mathbb{Z}_2$ son aditivos?

La pregunta es sencilla de responder por medio de un cálculo combinatorio. Buscamos todas las reglas de evolución de la forma

$$f(x) = \alpha_{-1}x + \alpha_0 y + \alpha_{+1}z$$

con $\alpha_i \in \mathbb{Z}_2$. Así, podemos pensar en $\alpha_i = 1$ como "escogerla" y en $\alpha_i = 0$ como "ignorarla". De esta manera, la pregunta anterior se transforma en la pregunta: ¿de cuántas formas podemos escoger elementos de un conjunto de tres elementos? Simplemente la cantidad de subconjuntos de dicho conjunto, es decir, 2^3.

Ahora bien, queremos investigar la forma de las reglas, de acuerdo con la clasificación de Wolfram de ser aditivas. Así, sea

$$a_i^{t+1} = \alpha_{-1}a_{i-1}^t + \alpha_0 a_i^t + \alpha_{+1}a_{i+1}^t$$

la regla de evolución de un autómata. Por medio de una sustitución directa, podemos encontrar la regla que le corrresponde a la vecindad con configuración $a_{i-1}^t, a_i^t, a_{i+1}^t$ de la siguiente manera:

- $a_{i-1}^t = a_i^t = a_{i+1}^t = 0$ se tiene $a_i^{t+1} = \alpha_{-1}0 + \alpha_0 0 + \alpha_{+1}0 = 0$ de forma que

$$\frac{000}{0}$$

- $a_{i-1}^t = a_i^t = 0, a_{i+1}^t = 1$ se tiene $a_i^{t+1} = \alpha_{-1}0 + \alpha_0 0 + \alpha_{+1}1 = \alpha_{+1}$ de forma que

$$\frac{001}{\alpha_{+1}}$$

- $a_{i-1}^t = 0, a_i^t = 1, a_{i+1}^t = 0$ se tiene $a_i^{t+1} = \alpha_{-1}0 + \alpha_0 1 + \alpha_{+1}0 = \alpha_0$ de forma que

$$\frac{010}{\alpha_0}$$

- $a_{i-1}^t = 0, a_i^t = 1, a_{i+1}^t = 1$ se tiene $a_i^{t+1} = \alpha_{-1}0 + \alpha_0 1 + \alpha_{+1}1 = \alpha_0 + \alpha_{+1}$ de forma que

$$\frac{011}{\alpha_0 + \alpha_{+1}}$$

- $a_{i-1}^t = 1, a_i^t = a_{i+1}^t = 0$ se tiene $a_i^{t+1} = \alpha_{-1}1 + \alpha_0 0 + \alpha_{+1}0 = \alpha_{-1}$ de forma que

$$\frac{100}{\alpha_{-1}}$$

- $a_{i-1}^t = 1, a_i^t = 0, a_{i+1}^t = 1$ se tiene $a_i^{t+1} = \alpha_{-1}1 + \alpha_0 0 + \alpha_{+1}1 = \alpha_{-1} + \alpha_{+1}$ de forma que

$$\frac{101}{\alpha_{-1} + \alpha_{+1}}$$

- $a_{i-1}^t = 1, a_i^t = 1, a_{i+1}^t = 1$ se tiene $a_i^{t+1} = \alpha_{-1}1 + \alpha_0 1 + \alpha_{+1}0 = \alpha_{-1} + \alpha_0$ de forma que

$$\frac{110}{\alpha_{-1} + \alpha_0}$$

- $a_{i-1}^t = a_i^t = a_{i+1}^t = 1$ se tiene $a_i^{t+1} = \alpha_{-1}1 + \alpha_0 1 + \alpha_{+1}1 = \alpha_{-1} + \alpha_0 + \alpha_{+1}$ de forma que

$$\frac{111}{\alpha_{-1} + \alpha_0 + \alpha_{+1}}$$

Así, las únicas reglas aditivas son de la forma

$$\frac{111}{\alpha_{-1} + \alpha_0 + \alpha_{+1}} \quad \frac{110}{\alpha_{-1} + \alpha_0} \quad \frac{101}{\alpha_{-1} + \alpha_{+1}} \quad \frac{100}{\alpha_{-1}} \quad \frac{011}{\alpha_0 + \alpha_{+1}} \quad \frac{010}{\alpha_0} \quad \frac{001}{\alpha_{+1}} \quad \frac{000}{0}$$

Sustituyendo las 2^3 posibles combinaciones, llegamos a que los índices de las reglas aditivas son

Combinación	Índice en base 2	Índice en base 10
$\alpha_{-1} = \alpha_0 = \alpha_{+1} = 0$	00000000	0
$\alpha_{-1} = \alpha_0 = 1$	00111100	60
$\alpha_{-1} = \alpha_{+1} = 1$	01011010	90
$\alpha_0 = \alpha_{+1} = 1$	01100110	102
$\alpha_{-1} = \alpha_0 = \alpha_{+1} = 1$	10010110	150
$\alpha_{+1} = 1$	10101010	170
$\alpha_0 = 1$	11001100	204
$\alpha_{-1} = 1$	11110000	240

que corresponden a los autómatas celulares siguientes (ver figura 2.1).

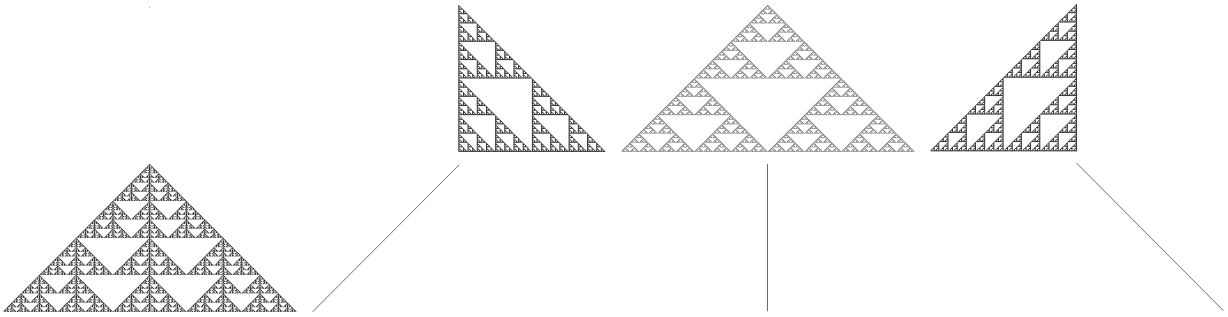


Figura 2.1: Todos los autómatas celulares elementales aditivos en orden ascendente.

Debido a que resulta imposible visualizar todo el espacio unidimensional $\mathbb{Z}$, usualmente se trabaja en una sección finita de $N \in \mathbb{Z}^+$ celdas, suponiendo que el resto de las celdas permanecen en blanco (en el estado 0_S). Así lo haremos a lo largo de este trabajo.

Dado un autómata celular en un espacio finito de N celdas cuyo estado al tiempo t contiene a las celdas en los estados $a_0^t, \ldots, a_{N-1}^t$ con $a_i^t \in S$, la configuración c_t del autómata puede ser codificada en el polinomio

$$A^t(x) = \sum_{i=0}^{N-1} a_i^t x^i$$

donde el estado de la i-ésima celda, a_i^t, será el coeficiente de x^i.

Definición 2.2. *Sea $A = (d, S, N, f)$ un autómata celular sobre un espacio finito unidimensional de N celdas. El **polinomio característico** de A al tiempo t es*

$$A^t(x) = \sum_{i=0}^{N-1} a_i^t x^i$$

donde $a_i^t \in S$ para toda $i = 0, \ldots, N-1$.

Observación 2.1. *A veces es más cómodo trabajar con polinomios generalizados que tienen potencias tanto positivas como negativas de x. A dichos polinomios generalizados les*

llamaremos ***dipolinomios****. Así, $H(x)$ es un* ***dipolinomio*** *si existe $m > 0$ tal que $x^m H(x)$ es un polinomio.*

A la regla de evolución $f(a_{i-1}, a_i, a_{i+1}) = \alpha_{-1}a_{i-1} + \alpha_0 a_i + \alpha_{+1}a_{i+1}$ la denotaremos simplemente por $a_i^t = \alpha_{-1}a_{i-1}^{(t-1)} + \alpha_0 a_i^{(t-1)} + \alpha_{+1}a_{i+1}^{(t-1)}$.

Definición 2.3. *Sea $A = (d, S, N, f)$ un autómata celular sobre un espacio unidimensional tal que $f(a_{i-1}, a_i, a_{i+1}) = \alpha_{-1}a_{i-1} + \alpha_0 a_i + \alpha_{+1}a_{i+1}$ con $\alpha_i \in S$. El* ***dipolinomio de evolución*** *de A es el dipolinomio*

$$\mathbb{T}(x) = \alpha_{-1}x + \alpha_0 + \alpha_{+1}x^{-1}$$

Observación 2.2. *Notemos lo siguiente:*

Sea $A^t(x) = \sum_{i=0}^{N-1} a_i x^i$ el polinomio característico de A al tiempo t, y sea $\mathbb{T}(x) = \alpha_{-1}x + \alpha_0 + \alpha_{+1}x^{-1}$ su dipolinomio de evolución. Notemos que

$$(\alpha_{-1}x + \alpha_0 + \alpha_{+1}x^{-1}) \sum_{i=0}^{N-1} a_i x^i = \sum_{i=0}^{N-1} (\alpha_{-1}a_{i-1} + \alpha_0 a_i + \alpha_{+1}a_{i+1})x^i = \sum_{i=0}^{N-1} f(a_{i-1}, a_i, a_{i+1})x^i$$

es decir,

$$\mathbb{T}(x)A^t(x) = A^{(t+1)}(x) \tag{2.1}$$

de forma que para obtener la configuración del autómata al tiempo $t+1$, basta con multiplicar el polinomio característico al tiempo t por el dipolinomio de evolución. Notemos que la multiplicación de polinomio se hace de la manera usual, donde las operaciones de los coeficientes son las del anillo S.

Observación 2.3. *La ecuación 2.1 nos dice que si $A = (d, S, N, f)$ es un autómata celular tal que $A^0(x) = 1$ y $\mathbb{T}(x)$ es su dipolinomio de evolución, entonces para toda $t \in \mathbb{Z}^+$, $\mathbb{T}^t(x) = A^t(x)$. Es decir, $\mathbb{T}^t(x)$ es la* ***función generadora*** *de los renglones de A. Ver sección 2.3 para una aplicación del resultado mencionado.*

Como resultado de la ecuación 2.1 los autómatas celulares aditivos presentan la propiedad de **superposición**, es decir, la configuración que se alcanza al tiempo t a partir de la condición inicial $A^0(x) + B^0(x)$ es $A^t(x) + B^t(x)$, donde $A^t(x)$ y $B^t(x)$ son las evoluciones (por separado) de $A^0(x)$ y $B^0(x)$ respectivamente. Por esta razón, a los autómatas celulares aditivos también se les conoce como **autómatas celulares lineales**. Formalizemos lo anterior en el siguiente teorema.

Teorema 2.1. *Sea $A^0(x) + B^0(x)$ el polinomio característico de un autómata celular A al tiempo 0, y sean $A^t(x)$, $B^t(x)$ los polinomios característicos al tiempo t de A al evolucionar a partir de $A^0(x)$ y $B^0(x)$ respectivamente y de forma independiente. Entonces*

$$f^t(A^0(x) + B^0(x)) = A^t(x) + B^t(x)$$

Demostración. Por la ecuación 2.1 sabemos que $f^t(A^0(x)+B^0(x)) = (A^0(x)+B^0(x))\mathbb{T}^t(x)$ donde $\mathbb{T}^t(x)$ es el dipolinomio de evolución de A. Luego,

$$(A^0(x) + B^0(x))\mathbb{T}^t(x) = A^0(x)\mathbb{T}^t(x) + B^0(x)\mathbb{T}^t(x) = A^t(x) + B^t(x)$$

lo cual completa la prueba. □

Observación 2.4. *Dado que cualquier configuración inicial puede ser expresada como suma de configuraciones iniciales "básicas", digamos $\Delta(x) = x^j$ consistentes de una única celda en el estado 1_S, el principio de superposición implica que cualquier evolución puede ser obtenida como suma de las evoluciones a partir de configuraciones básicas $\Delta(x)$. [11] Esto simplifica significativamente el cálculo de la evolución de un autómata celular aditivo.*

Dentro de las propiedades extraordinarias que presentan los autómatas celulares aditivos destaca la de alcanzar cualquier configuración posible. Esto lo formalizamos en el siguiente

Teorema 2.2. [1] *Sea A un autómata celular aditivo no trivial. Entonces A no tiene Jardínes del Edén.*

Demostración. Sean A un autómata celular aditivo no trivial, f su regla de evolución y G su función de transición global. En virtud del teorema del Jardín del Edén (1.10), basta con probar que A no tiene gemelos. Supongamos entonces que $p = (N(D), f), q = (N(D'), f)$ son dos patrones gemelos de longitud m y veamos que $p = q$. Sea $\Delta(x_i)$ la configuración $c(i) = 1, c(j) = 0$ para toda $j \neq i$. Podemos descomponer a $G^{N(D)\longrightarrow D}(q)$ como suma de evoluciones de básicos, es decir

$$G^{N(D)\longrightarrow D}(q) = \sum_{i:q(i)=1} G(\Delta(x_i))$$

y análogamente

$$G^{N(D)\longrightarrow D}(p) = \sum_{j:p(j)=1} G(\Delta(x_j)).$$

Dado que $G^{N(D)\longrightarrow D}(q) = G^{N(D)\longrightarrow D}(p)$ entonces

$$\sum_{i:q(i)=1} G(\Delta(x_i)) = \sum_{j:p(j)=1} G(\Delta(x_j))$$

de forma que, dado que A es no trivial, $\{i : q(i) = 1\} = \{j : p(j) = 1\}$ y así $p = q$. Luego, A no tiene Jardines del Edén. □

[1] Argumentos similares a los expuestos en este teorema pueden consultarse en [7], pero esta prueba es original del autor.

El teorema anterior implica que los autómatas celulares aditivos no triviales siempre son sobreyectivos, y por lo tanto la medida uniforme de Bernoulli sobre el espacio de configuraciones es invariante bajo aplicaciones de un autómata celular lineal. Esto podría ser usado para generar sucesiones pseudoaleatorias de números, simplemente iterando la función de transición global de un autómata celular a partir de una configuración inicial aleatoria (ver figura 2.2).

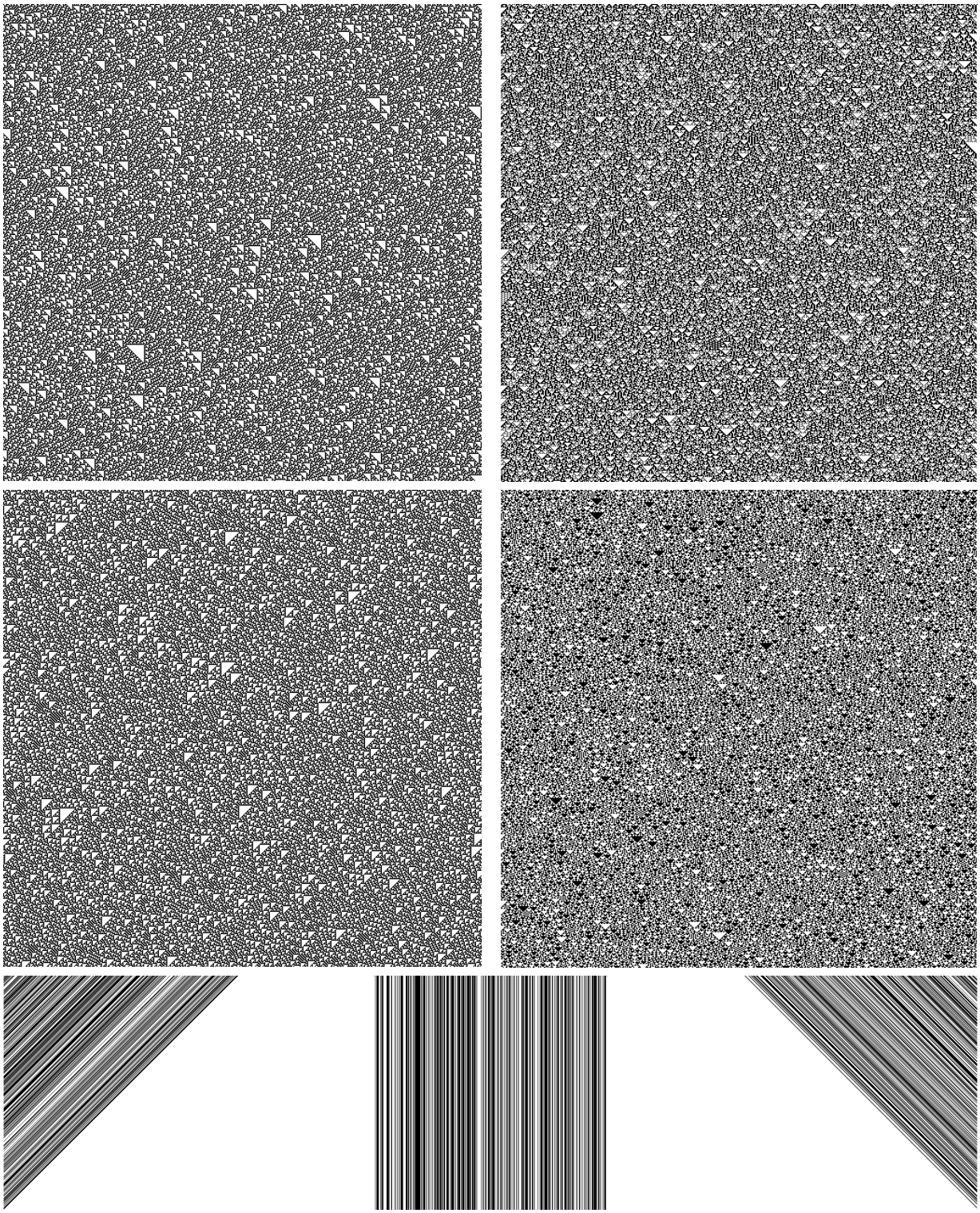


Figura 2.2: Evolución de los autómatas celulares aditivos $60, 90, 102, 150, 170, 204, 240$ (en orden lexicográfico) a partir de una configuración inicial aleatoria. La aleatoriedad se mantiene durante la evolución de cada autómata.

2.1. Reducibilidad computacional

Una pregunta fundamental para el estudio de los autómatas celulares es: ¿existirá una forma más eficiente de calcular el renglón t-ésimo de un autómata celular, sin tener que simular todo el autómata? O bien, ¿existirá una forma de calcular G^t sin tener que calcular $G^1, G^2, \ldots, G^{t-1}$? ¿Existirá algún otro proceso de cómputo que, a pesar de ser recursivo, sea más eficiente que el simulador de autómatas? Amén de que habría que especificar a lo que nos referimos con "eficiente", podemos pensar, en principio, en la idea intuitiva de que un algoritmo o proceso de cómputo es más eficiente que otro siempre que implique una cantidad menor de "esfuerzo computacional" -es decir de pasos- para obtener el mismo resultado.

Esta pregunta -o más bien preguntas, como veremos más adelante- es central para el estudio de sistemas computacionales, ya que no sólo ataca un problema de optimización, sino principalmente de **reducibilidad computacional**. En [18], Stephen Wolfram plantea que una fórmula matemática -y, de hecho, cualquier proceso cognitivo- puede ser pensada como una computación -o bien un cálculo-, y podemos medir la cantidad de operaciones que nos lleva evaluar una fórmula para obtener un resultado. De la misma manera, podemos analizar un algoritmo para saber en cuántas operaciones logramos obtener la salida deseada. Entonces, si existiera una forma de calcular el renglón t-ésimo de un autómata celular sin tener que calcular sucesivamente los renglones anteriores, ya sea mediante alguna fórmula o algún algoritmo más eficiente, ésta sería **computacionalmente reducible**, en tanto que podemos llevar a cabo un proceso de cómputo más eficiente para obtener el mismo resultado. En concreto, Wolfram [18] plantea que un autómata celular es computacionalmente irreducible si, para conocer el estado t-ésimo del sistema, no existe un proceso computacional alterno a evolucionar el sistema según la función de evolución f. Luego, un autómata celular es reducible si no es irreducible.

Tenemos ante nosotros dos preguntas que en principio parecen ser la misma pero que vale la pena distinguir [3]:

1. ¿Existe algún sistema computacional que compute el t-ésimo renglón de un autómata celular dado más rápido que simplemente simular el autómata?

2. ¿Existe algún sistema computacional que compute el t-ésimo renglón de un autómata celular dado sin necesidad de computar los $t-1$ renglones anteriores?

Para poder abordarlas podemos comparar el orden de crecimiento de distintas funciones. Así, tenemos la siguiente definición.

Definición 2.4.

Dada una función $g(n)$, denotaremos por

$$\Theta(g(n)) := \{f(n) : \exists\, c_1, c_2 \in \mathbb{R}^+, n_0 \in Dom(g) = Dom(f) : \forall n \geq n_0, \\ 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)\}$$

$$O(g(n)) := \{f(n) : \exists\, c \in \mathbb{R}^+, n_0 \in Dom(g) = Dom(f) : \forall n \geq n_0, \\ 0 \leq f(n) \leq cg(n)\}$$

$$o(g(n)) := \{f(n) : \forall c \in \mathbb{R}^+, \exists\, n_0 \in Dom(g) = Dom(f) \cap \mathbb{R}^+ : \forall n \geq n_0, \\ 0 \leq f(n) < cg(n)\}$$

$$\Omega(g(n)) := \{f(n) : \exists\, c \in \mathbb{R}^+, n_0 \in Dom(g) = Dom(f) : \forall n \geq n_0, \\ 0 \leq cg(n) \leq f(n)\}$$

$$\omega(g(n)) := \{f(n) : \forall c \in \mathbb{R}^+, \exists\, n_0 \in Dom(g) = Dom(f) \cap \mathbb{R}^+ : \forall n \geq n_0, \\ 0 \leq cg(n) < f(n)\}$$

Observación 2.5.

- *Dado que los conceptos anteriores son conjuntos, formalmente tendríamos que escribir, por ejemplo,* $f \in O(g(n))$*. Sin embargo, optaremos por la notación* $f = O(g(n))$ *ya que presenta algunas ventajas.*
- *Si* $f(n) = o(g(n))$ *entonces* $lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$*. Es decir, o denota una cota superior "relajada" o "distante" de* g*.*
- *Análogamente, si* $f(n) = \omega(g(n))$ *entonces* $lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$*. Es decir,* ω *denota una cota inferior "relajada" o "distante" de* g*.*
- $f(n) = \omega(g(n))$ *si y sólo si* $g(n) = o(f(n))$*.*

Veamos entonces, como una primera aproximación a las preguntas que hemos planteado, al número de operaciones necesarias para calcular el t-ésimo renglón de un autómata celular elemental por medio de su simulación.

Supongamos que c_0 es una cadena de longitud l. Para calcular $G^1(c_0)$, necesitaremos aplicarle la regla de evolución f a l celdas, y obtendremos $l + 2$ nuevas celdas. Análogamente, para calcular $G^2(c_0)$ a partir de $G^1(c_0)$, necesitaremos $l + 2$ pasos y obtendremos $l + 4$ celdas, y así sucesivamente. Para calcular recursivamente $G^t(c_0)$ son necesarios $\sum_{i=1}^{t-1}(l + 2i) = t^2 + tl - l - n$ pasos, donde $t^2 + tl - l - n = O(t^2)$ [3].

Notemos que esto no significa que para cualquier autómata celular elemental se necesitan $O(t^2)$ pasos para computar el t-ésimo renglón, sino que ese es el número máximo de pasos que se requieren para simular sus primeros t renglones. No es difícil convencerse de que podemos obtener el t-ésimo renglón del autómata elemental 0 (y de varios otros autómatas elementales con comportamiento trivial) en tiempo $O(1)$.

Cabe recalcar otra sutil diferencia: una cosa es *simular* un autómata celular por t pasos y otra es calcular su t-ésimo renglón. Simular un autómata celular es exactamente lo mismo que calcular su t-ésimo renglón aplicando G de forma recursiva, mientras que calcular el t-ésimo renglón de un autómata no necesariamente exige un método recursivo. Por ejemplo, simular el autómata celular elemental con Regla 0 por t pasos requeriría $O(t)$ pasos,

mientras que obtener su t-ésimo renglón únicamente requiere $O(1)$ pasos, pues basta con algo del estilo "print(0)".

Para poder abordar formalmente las preguntas planteadas, necesitamos hacer evidente el modelo de cómputo en el que vamos a trabajar. Es decir, tenemos que hacer evidente qué es lo que nuestra computadora -teórica- puede o no hacer y en cuánto tiempo lo logra. En este caso, serán las *Máquinas de Acceso Aleatorio* (RAM) [17].

Definición 2.5. *(Máquina de Acceso Aleatorio)*
*Una **Máquina de Acceso Aleatorio** (RAM, por sus siglas en inglés) es un modelo de cómputo en el que la computadora consta de una memoria y una unidad de procesamiento tales que:*
Memoria:

- *Es una sucesión infinita de celdas numeradas en las que podemos guardar una cantidad arbitraria (pero finita) de bits.*

Unidad de procesamiento:

- *Tiene un número finito de registros (celdas) en las que podemos realizar las siguientes operaciones en los tiempos indicados:*
  - *Escribir un número finito de bits en algún registro o copiar la información de un registro a otro en tiempo $O(1)$.*
  - *Tomar los enteros a, b escritos en dos registros y realizar $a+b, a-b, a*b$ y a/b en tiempo $O(1)$.*
  - *Tomar los enteros a, b escritos en dos registros, compararlos con $a < b, a > b$ o $a = b$ y saber cuál de ellas es verdadera en tiempo $O(1)$.*
  - *Tomar una dirección D contenida en algún registro, leer el contenido de la celda D de la memoria y escribirlo en otro registro, o bien, escribir el contenido de otro registro en la celda con índice D de la memoria, todo lo anterior en tiempo $O(1)$.*

Al registro en la posición i-ésima lo denotaremos por r_i.

Dicho lo anterior, el algoritmo para simular un autómata celular elemental por medio de iteraciones sucesivas de $G^t(c_0)$ en el modelo RAM sería el siguiente:

Algoritmo 1: Calcular $G^t(c_0)$

Datos: $f : \{0,1\}^m \to \{0,1\}$
Entrada: $c_0 := 1$
Salida: $G^t(c_0)$

$r_0 \leftarrow c_0$

for $i \in \{0, \ldots, t-1\}$ **do**

1. Leer r_i
2. Computar $G(r_i)$ *i.e.* $f(x)$ para cada bit de $x \in r_i$
3. $r_{i+1} \leftarrow G(r_i)$

end
return r_t

Dado que estamos usando el modelo RAM, si la longitud de c_0 es $l < t$, entonces el algoritmo anterior tiene complejidad $O(1) + O(tl + \frac{t(t+1)}{2}) + O(1) = O(t^2)$, como lo insinuado anteriormente (esto es, pues la longitud de l se va incrementando por dos unidades a cada iteración).

Consideremos la siguiente definición.

Definición 2.6. *(Máquina de Acceso Aleatorio que representa a un autómata celular elemental)*

Sea E_n la n-ésima configuración (el n-ésimo renglón) de un autómata celular elemental A. Una Máquina de Acceso Aleatorio RAM_A será llamada ***Máquina de Acceso Aleatorio que representa a*** *A (RAM ACE) si:*

1. *Para toda $n \geq 0$, RAM_A con entrada n computa E_n.*
2. *Los registros de RAM_A contienen sucesivamente y en orden creciente de $i = 0, \ldots, n-1$ las configuraciones E_i.*
3. *Para toda $n \in \mathbb{N}$, E_{n+1} se obtuvo a partir de E_n aplicando f a todas las celdas activas.*

En lo sucesivo abreviaremos las condiciones anteriores por "RAM simula a A".

Observación 2.6. *Una RAM ACE es simplemente un programa que simula el comportamiento de un autómata celular elemental, como los que hemos estado mostrando a lo largo de este trabajo.*

Podemos reformular las preguntas anteriores en términos de Máquina de Acceso Aleatorio de la siguiente manera:

1. ¿Existe alguna Máquina de Acceso Aleatorio que, con entrada n, compute E_n más rápido que RAM_A?

2. ¿Existe alguna Máquina de Acceso Aleatorio que con entrada n compute E_n sin computar E_i para $i = 1, \ldots, n-1$? Es decir, ¿que no sea la Máquina de Acceso Aleatorio que representa a A?

Ambas preguntas guardan una estrecha relación con el concepto que queremos precisar pues, de existir una Máquina de Acceso Aleatorio que compute E_n más rápido que RAM_A, entonces podremos decir que A es computacionalmente reducibile. Por otro lado, si existe alguna Máquina de Acceso Aleatorio que con entrada n compute E_n sin computar E_i para $i = 1, \ldots, n-1$ y que además lo haga más rápido que cualquier proceso recursivo de cómputo de E_n, entonces también podremos decir que A es computacionalmente reducibile. Ambas preguntas son una guía para el concepto que perseguimos y nos permitirán acercarnos a la definición de reducibilidad computacional.

Teorema 2.3. *Sea A un autómata celular elemental tal que el número de celdas que cambian de una configuración a otra está acotado, entonces existe una RAM ACE RAM_A tal que computa E_n en tiempo $T(RAM_A) = O(n)$.*

Es decir, sea $L(t) : \mathbb{N} \to \mathbb{N}$ la función que al t-esimo renglón le asocia el número de celdas que cambiaron con respecto al renglón anterior. Si existe $m \in \mathbb{N}$ tal que para toda $t \in \mathbb{N}$ $L(t) < m$ entonces existe una RAM ACE RAM_A tal que computa E_n en tiempo $T(RAM_A) = O(n)$.

Demostración. Sea A un autómata celular elemental en el que existe $m \in \mathbb{N}$ tal que para toda $t \in \mathbb{N}$, $L(t) < m$. Entonces la RAM ACE RAM_A que lo representa tardaría $m * n$ pasos en computar cada E_n para toda $n \in \mathbb{N}$. Luego $T(RAM_A) = O(n)$. □

Corolario 2.1. *Para los autómatas celulares elementales con reglas* $0, 2, 4, 6, 8, 10, 12, 14, 16, 20, 24, 32, 34$ *(entre otras) existen Máquina de Acceso Aleatorio que los representan tales que computan E_n en tiempo $O(n)$.*

Demostración. Para todas ellas $L(t) < 3$ para toda $t \in \mathbb{N}$. □

Es interesante observar que hay autómatas celulares triviales tales que existen Máquinas de Acceso Aleatorio que computan E_n en $O(n)$, y por lo visto anteriormente para muchos otros existen Máquinas de Acceso Aleatorio que computan E_n en $O(n^2)$. Sin embargo, para ninguno de los 256 autómatas celulares elementales existe una Máquina de Acceso Aleatorio M que compute E_n en $O(n^2) > O(T(M)) > O(n)$. [3]

Ahora bien, para precisar el concepto de reducibilidad computacional en términos de Máquinas de Turing que representan autómatas, vale la pena recordar cómo lo plantea Wolfram

> *Indeed, whenever computational irreducibility exists in a system it means that in effect there can be no way to predict how the system will behave except by going through almost as many steps of computation as the evolution of the system itself. [18]*

En términos coloquiales podríamos traducir lo anterior como: cualquier proceso de cómputo (Máquina de Acceso Aleatorio) que computara el n-ésimo estado de un autómata (E_n) tendría que pasar por una cantidad de pasos mayor o igual a la del simulador del autómata (RAM ECA). Recordemos que para un autómata celular elemental no trivial una RAM ECA RAM_A simula a A en tiempo $O(n^2)$. Esto da pie a la siguiente definición.

Definición 2.7. *(Irreducibilidad computacional)*
*Sea A un autómata celular elemental. Diremos que A es **computacionalmente irreducible** (CIR) si y sólo si cualquier Máquina de Acceso Aleatorio M que compute E_n en tiempo $T(M)$ es tal que $T(M) = \Omega(n^2)$.*

Es decir, no existe una forma más eficiente de calcular E_n que simplemente simular el autómata.

Observación 2.7. *Notemos que la definición de irreducibilidad computacional no exige que la Máquina de Acceso Aleatorio M compute $E_1, E_2, \ldots, E_{n-1}$ antes de computar E_n, sino que, sea cual sea la forma en la que M compute E_n, lo haga en una cantidad mayor de pasos que el simulador de A. Es decir, que la forma más eficiente de calcular E_n sea simplemente simulando el autómata.*

Observación 2.8. *La definición anterior nos permite responder a las preguntas planteadas inicialmente. Si un autómata celular A es computacionalmente irreducible y RAM_A es la Máquina de Acceso Aleatorio que lo simula, entonces:*

1. *No existe una Máquina de Acceso Aleatorio que, con entrada n, compute E_n más rápido que RAM_A.*
2. *No existe una Máquina de Acceso Aleatorio que, con entrada n, compute E_n sin computar $E_1, E_2, \ldots, E_{n-1}$ **más rápido** que lo que tarda RAM_A en simular a A.*

Definición 2.8. *(Reducibilidad computacional) Sea A un autómata celular elemental. Diremos que A es **computacionalmente reducible** (CR) si y sólo si no es computacionalmente irreducible. En particular, si A es computacionalmente reducible, entonces existe una Máquina de Acceso Aleatorio M que compute E_n en tiempo $T(M)$ tal que $T(M) = O(n^2)$.*

Ejemplo 2.1. *Los autómas celulares con reglas $0, 2, 4, 6, 8, 10, 12, 14, 16, 2024, 32, 34$ (entre otras) son computacionalmente reducibles ya que es posible calcular E_n en tiempo $O(1)$, con algo del estilo*

```
if i mód 2 == 0
    print(x)
else
    print(y)
```

Las secciones siguientes están dedicadas a demostrar, mediante un análisis matemático basado en la teoría de números, que ciertos autómatas celulares elementales relevantes son, en efecto, computacionalente reducibles.

2.2. Pseudocódigos

A continuación se encuentran, en forma de pseudocódigo, los códigos que utilicé tanto para generar las imágenes de este capítulo como para realizar las exploraciones computacionales estudiadas.

Como el lector se dará cuenta, las funciones involucradas son una calca de las funciones presentadas en la introducción de este capítulo.

Nuestra primera función toma como entrada una matriz de $n \times n$, $M \in M_{n,n}(\{0,1\})$ que representará el espacio de configuraciones del autómata, y un entero positivo $k \in \mathbb{Z}^+$ que determinará hasta qué configuracion c_k queremos evolucionar el autómata. También requiere de una función $f : \{0,1\}^m \to \{0,1\}$ tal que a cada celda le asigne un estado en función de los estados de sus vecinos en la configuración anterior (para un ejemplo ver el algoritmo 6). Para autómatas celulares elementales $m \leq 3$. En el primer renglón estará codificada, con ceros y unos, la configuración inicial.

Ésta función es $G : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$.

Algoritmo 2: Función $G : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$

Datos: $f : \{0,1\}^m \to \{0,1\}$
Entrada: $M \in M_{n,n}(\{0,1\}), k \in Z^+$
Salida: M modificada.

$n \leftarrow$ número de columnas de la matriz M

Función *G(M,k)*
    for $i \in \{2, \ldots, k\}$ **do**
        for $j \in \{1, \ldots, n\}$ **do**
            $M_{i,j} = f(M_{i-1,j}, M_{i,j-1}, M_{i,j})$
        end
    end
    return M
end

Ahora necesitamos una función para colorear las entradas la matriz $M \in M_{n,n}(\{0,1\})$ que representa el espacio de configuraciones del autómata. Una forma sencilla de hacerlo es modificando otra matriz $M' \in M_{n,n}(\{0,1\})$ de tipo RGB con base en las entradas de la matriz M. Así, las entradas de la matriz M' serán elementos del conjunto $\{(x,y,z) : x,y,z \in \mathbb{R} \cap [0,1]\}$, donde x representa el porcentaje de rojo, y el porcentaje de verde y z el porcentaje de azul del pixel correspondiente. Es necesaria esta otra matriz M' pues comúnmente no se puede transformar de una matriz con entradas en $\mathbb{Z}$ a una matriz con entradas en $\mathbb{R}^3$ (que representan colores). La matriz M' inicialmente es una matriz cuyas entradas son todas blancas, es decir, $M'_{i,j} = (1,1,1)$ para toda $0 \leq i,j \leq n$.

Algoritmo 3: Función $ColorearMatriz : M_{i,j} \to \mathbb{R}^3$

Entrada: $M \in M_{n,n}(\{0,1\}), M' \in M_{n,n}(\mathbb{R}^3)$
Salida: M' modificada.

$n \leftarrow$ número de columnas de la matriz M

Función *ColorearMatriz(M,M')*
  for $i \in \{1,\ldots,n\}$ **do**
    for $j \in \{1,\ldots,n\}$ **do**
      if $M_{i,j} = 1$ **then**
        $M'_{i,j} = (0,0,0)$
      else
        $M'_{i,j} = (1,1,1)$
      end
    end
  end
  return M'
end

Un ejemplo de cómo generar y colorear los primeros k renglones de un autómata celular elemental con regla de evolución f y configuración inicial $c_0(0) = 1$ y $c_0(i) = 0$ para toda $i \neq 0$ sería

```
AC = zeros(Int8, (2k,2k))
AC' = ones(RGB{Float64}, 2k, 2k)
AC[1,k] = 1
ColorearMatriz(G(AC,K),AC')
```

2.3. La Regla 60 y el Triángulo de Pascal

> Le cœur a ses raisons que la raison ne connaît point [...] Nous connaissons la vérité non seulement par la raison, mais encore par le cœur.
>
> *Blaise Pascal, Pensées*

2.3.1. Números de Fermat

El matemático francés Pierre de Fermat conjeturó, a mediados del siglo XVII, que todos los números de la forma

$$F_n = 2^{2^n} + 1 \qquad n = 0, 1, 2, \ldots \tag{2.2}$$

eran primos. En su honor, los números de esta forma son conocidos como *números de Fermat*. En efecto, los primeros números de Fermat

$$F_0 = 3, \quad F_1 = 5, \quad F_2 = 17, \quad F_3 = 257, \quad F_4 = 65537,$$

son primos. En 1732 el gran Leonhard Euler demostró que $F_5 = 651 \cdot 6700417$ no es primo, refutando la conjetura de Fermat. Hasta la fecha no se sabe si existe una cantidad infinita de primos de esta forma (primos de Fermat).

Definición 2.9. *Un **número de Fermat** es un número natural de la forma*

$$n = 2^{2^m} + 1, \qquad m \in \mathbb{N}.$$

Denotamos por F_n al n-ésimo número de Fermat.

Definición 2.10. *Un **primo de Fermat** es un número de Fermat que además es primo.*

Hasta finales del siglo XVIII, los números de Fermat no eran más que una curiosidad matemática. El panorama cambió cuando Carl Friedrich Gauss $(1777 - 1855)$ demostró que un polígono regular puede ser construído con regla y compás si su número de lados es

$$n = 3, 4, 5, 6, 8, 10, 12, 15, 16, 17, \ldots$$

Precisamente, demostró que un polígono regular con n lados es construible con regla y compás si

$$n = 2^i F_{m_1} F_{m_2} \cdots F_{m_j}$$

donde $n \geq 3, i \geq 0, j \geq 0$, y $F_{m_1}, F_{m_2}, \ldots, F_{m_j}$ son primos de Fermat distintos.

En 1837 el matemático francés Pierre Wantzel demostró que esta condición no sólo es suficiente, sino también necesaria. Por ende, hasta la fecha no sabemos cuántos polígonos regulares son construibles con regla y compás.

Teorema 2.4. *(Gauss-Wantzel) Existe una construcción Euclideana de un polígono regular (*i.e. *con regla y compás) si y sólo si su número de lados es $n = 2^i p_1 p_2 \cdots p_j$, donde $i \geq 0, j \geq 0, n \geq 3$ son enteros y $p_1, p_2, \ldots, p_j$ son primos de Fermat distintos.*

Demostración. La demostración, basada en teoría de Galois, se puede consultar en el capítulo 16 de [10]. □

Observación 2.9. *Hasta la fecha únicamente conocemos cinco primos de Fermat (F_i con $i = 0, 1, \ldots, 4$). Luego, por el teorema 2.4 sabemos que existen* 31 *polígonos regulares con un número impar de lados que son construibles con regla y compás, pues*

$$\sum_{j=1}^{5} \binom{5}{j} = 2^5 - 1 = 31.$$

Los lados de dichos polígonos son

$$n = 3, 5, 15, 17, 51, 85, 255, 257, \ldots$$

Los únicos polígonos construibles con regla y compás conocidos con un número primo de lados son tales que

$$n = F_i \qquad p.a. \quad i = 0, 1, \ldots, 4$$

i.e.

$$n = 3, 5, 17, 257, 65537$$

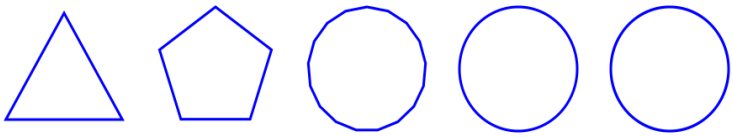


Figura 2.3: Únicos polígonos regulares construibles conocidos con un número primo de lados.

2.3.2. Primeras consideraciones

Los números de Fermat no solamente son interesantes por el teorema de Gauss-Wantzel, sino también por su conexión con otros objetos matemáticos. En particular, resulta que estos números -junto con sus productos- pueden ser extraídos de uno de los objetos combinatorios más fascinantes y estudiados en la historia de las matemáticas: el Triángulo de Pascal. Más aún -y como veremos a continuación -, dado que podemos caracterizar al Triángulo de Pascal como un autómata celular, los números de Fermat establecen un puente natural entre los polígonos construibles, la combinatoria oculta en el Triángulo de Pascal, y el mundo de los programas simples.

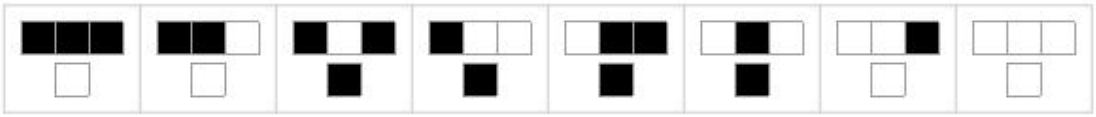


Figura 2.4: La Regla 60, indexada por su representación en base 2 : $00111100_2 = 60_{10}$. [2]

[2]Imagen tomada de Weisstein, Eric W. *Rule 60.* From MathWorld: A Wolfram Web Resource. https://mathworld.wolfram.com/Rule60.html

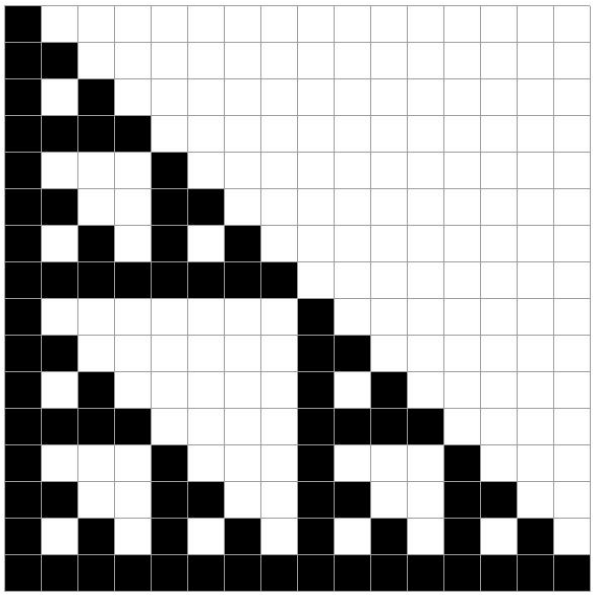


Figura 2.5: Primeros 15 pasos en la evolución de la Regla 60. [3]

Notemos que $f(x, y, z) = x + y \pmod 2$ de forma que su dipolinomio de evolución es $\mathbb{T}(x) = x + 1$. Así, por la ecuación 2.1, la evolución al tiempo t de la Regla 60 está dada por

$$\mathbb{T}(x)^t 1 = (x+1)^t = \sum_{i=0}^{t} \Big[\binom{t}{i} \pmod 2 \Big] x^i \tag{2.3}$$

Dicha expresión revela que la regla 60 es precisamente una forma rectangular del triángulo de Pascal reducido módulo 2.

Observación 2.10. *Si interpretamos los primeros renglones de la regla 60 como números en base 2, notamos que todos son números de Fermat o productos de números Fermat.*

Renglón en base 2	*Renglón en base 10*
1	*1*
11	*3*
101	*5*
1111	*15*
10001	*17*
110011	*51*
1010101	*85*

Nuestro objetivo ahora es demostrar que dicho resultado se cumple para todos los renglones del autómata. Procediendo como en [4], recurriremos al siguiente teorema.

[3] Imagen tomada de Weisstein, Eric W. *Rule 60.* From MathWorld: A Wolfram Web Resource. https://mathworld.wolfram.com/Rule60.html

Lema 2.1. *Sea p un primo y sea $X, i \in \mathbb{Z}^+$. Entonces*

$$(1+X)^{p^i} \equiv 1 + X^{p^i} (mód\ p)$$

Demostración. Por inducción sobre i.

Caso base: $i = 1$

Dado que $p|\binom{p}{k} = \frac{p!}{k!(p-k)!}$, entonces $(1+X)^p = \sum_{k=0}^{p} \binom{p}{k} X^k \equiv 1 + X^p (\text{mód } p)$.

Hipótesis de inducción: Supongamos que $(1+X)^{p^i} \equiv 1 + X^{p^i} (\text{mód } p)$ con $i \geq 1$.

Paso inductivo: Veamos que del caso base y la hipótesis de inducción deducimos que

$$(1+X)^{p^{i+1}} = (1+X)^{p^i p} = [(1+X)^{p^i}]^p \equiv (1+X^{p^i})^p \equiv (1+X^{p^i p}) \equiv 1 + X^{p^{i+1}} (\text{mód } p)$$

□

Indexemos los renglones del triángulo de Pascal con $n = 0, 1, \ldots$ y las columnas con $j = 0, 1, \ldots, n$, de forma que las entradas sean los coeficientes binomiales

$$\binom{n}{j} \qquad \text{para todo} \quad 0 \leq j \leq n.$$

Ahora bien, sea

$$c(n, j) \equiv \binom{n}{j} \pmod 2$$

de tal manera que el renglón n-ésimo de la regla 60 interpretado como un número en base 2 sea

$$a(n) = \sum_{j=0}^{n} c(i, j) 2^j, \qquad n \in \mathbb{N}.$$

Ahora, si escribimos el número n en base 2 como

$$n = 2^t \alpha_t + \cdots + 2\alpha_1 + \alpha_0 \quad \text{donde} \quad \alpha_i \in \{0, 1\}, \alpha_t \neq 0$$

entonces tenemos el siguiente teorema.

Teorema 2.5. *Para toda $n \in \mathbb{N}$ se satisface que*

$$a(n) = F_t^{\alpha_t} \cdots F_1^{\alpha_1} F_0^{\alpha_0}$$

donde F_i es el i-ésimo número de Fermat.

Demostración. Observemos primero que

$$(X+1)^n = \sum_{i=0}^{n} \binom{n}{i} X^i$$

de forma que

$$(X+1)^n \equiv \sum_{i=0}^{n} c(n,i) X^i \pmod 2.$$

Además, por el Lema 2.1

$$(X+1)^n = (X+1)^{2^t\alpha_t + \cdots + 2\alpha_1 + \alpha_0} = \prod_{j=0}^{t} (X+1)^{2^j\alpha_j} \equiv \prod_{j=0}^{t} (X^{2^j}+1)^{\alpha_j} \pmod 2$$

de forma que, por la unicidad de la representación en base 2,

$$\prod_{j=0}^{t} (X^{2^j}+1)^{\alpha_j} = \sum_{i=0}^{n} c(n,i) X^i$$

Sustituyendo $X = 2$ en la ecuación anterior obtenemos

$$\prod_{j=0}^{t} (2^{2^j}+1)^{\alpha_j} = \sum_{i=0}^{n} c(n,i) 2^i$$

i.e.

$$\prod_{j=0}^{t} F_j^{\alpha_j} = a(n)$$

□

Dado que sabemos que F_i es primo para $i = 0, 1, \ldots, 4$ (y de hecho son los únicos que conocemos), el teorema 2.5 y la observación 2.9 nos dice que en los primeros 32 renglones (contando el renglón 0) de la regla 60 aparecen exactamente, y en orden, todos los polígonos construibles con una cantidad impar de lados.

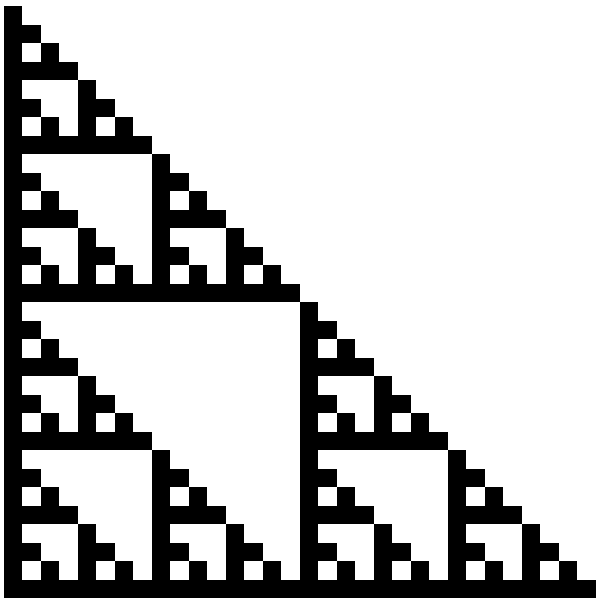


Figura 2.6: Los primeros 32 renglones de evolución de la regla 60

Como una primera aproximación al estudio de la Regla 60, queremos estimar el número de 1's que aparecen en un renglón dado del autómata en cuestión, para así saber en qué renglones aparecen los números de Fermat.

El resultado que buscamos es un corolario del célebre

Teorema 2.6. *(de Lucas)*
Sea p un primo y sean $m, n \in \mathbb{Z}^+$ tales que

$$m = m_k p^k + \cdots + m_1 p + m_0 \qquad 0 \leq m_r < p$$
$$n = n_k p^k + \cdots + n_1 p + n_0 \qquad 0 \leq n_r < p$$

Entonces

$$\binom{m}{n} \equiv \binom{m_0}{n_0}\binom{m_1}{n_1}\binom{m_2}{n_2}\cdots\binom{m_k}{n_k} (\textit{mód } p)$$

Demostración. Veamos que

$$\sum_{n=0}^{m} \binom{m}{j} X^j = (1+X)^m = \prod_{i=0}^{k} [(1+X)^{p^i}]^{m_i}$$

y por el lema 2.1 sabemos que

$$\prod_{i=0}^{k} [(1+X)^{p^i}]^{m_i} \equiv \prod_{i=0}^{k} [(1+X^{p^i})]^{m_i} \equiv \prod_{i=0}^{k} \left(\sum_{n_i=0}^{m_i} \binom{m_i}{n_i} X^{p^i n_i} \right) \equiv \prod_{i=0}^{k} \left(\sum_{n_i=0}^{p-1} \binom{m_i}{n_i} X^{p^i n_i} \right) (\text{mód } p)$$

y dado que $\sum_{n_i=0}^{p-1} X^{p^i n_i} = X^n$ se tiene que

$$\prod_{i=0}^{k}\left(\sum_{n_i=0}^{p-1}\binom{m_i}{n_i}X^{p^i n_i}\right) \equiv \sum_{n=0}^{m}\left(\prod_{i=0}^{k}\binom{m_i}{n_i}\right)X^n (\text{mód } p)$$

de forma que

$$\sum_{n=0}^{m}\binom{m}{n}X^n \equiv \sum_{n=0}^{m}\left(\prod_{i=0}^{k}\binom{m_i}{n_i}\right)X^n (\text{mód } p)$$

lo cual finaliza la prueba. □

Corolario 2.2. $\binom{m}{n}$ *es impar si y sólo si los dígitos de* n *en base* 2 *son un subconjunto de los dígitos de* m *en base* 2.

Demostración. Por el teorema 2.6 sabemos que $\binom{m}{n}$ es impar si y sólo si $\binom{m_i}{n_i} \equiv 1 (\text{mód } 2)$ para cada $i \in \{1, \ldots, k\}$. Luego, si $m_i = 0$ entonces $n_i = 0$ y si $m_i = 1$ entonces $n_i \in \{0, 1\}$, de forma que los dígitos de n en base 2 son un subconjunto de los dígitos de m en base 2. □

Ahora sí, podemos calcular el número de 1's que aparecen en el renglón n-ésimo de la Regla 60.

Teorema 2.7. *Sean* $N(n) : \mathbb{N} \to \mathbb{N}$ *la función que nos da el número de unos que aparecen en el* n*-ésimo renglón de la Regla* 60*, y* $B(n) : \mathbb{N} \to \mathbb{N}$ *la función que nos da el número de ocurrencias del dígito* 1 *en la representación binaria de* n*. Entonces* $N(n) = 2^{B(n)} \quad \forall n \in \mathbb{Z}^+$.

Demostración. Sea $n \in \mathbb{Z}^+$. Sabemos que el número que aparece en el n-ésimo renglón de la Regla 60 es $\sum_{k=0}^{n}\binom{n}{k}2^k$, de forma que el número de 1's que aparecen en dicho renglón es el número de coeficientes binomiales $\binom{n}{k}$ impares. Por el corolario 2.2 esto sucede si y sólo si los dígitos de k en base 2 son un subconjunto de los dígitos de n en base 2. Dado que hay $2^{B(n)}$ de dichos subconjuntos, se tiene que $N(n) = 2^{B(n)}$. □

Observación 2.11. *Dado que los números de Fermat tienen únicamente dos ocurrencias del dígito* 1 *en su representación binaria, pues son de la forma* $F_n = 2^{2^n} + 1$*, entonces el teorema 2.7 nos dice que éstos solamente pueden aparecer en los renglones que son potencias de dos, pues requerimos que* $2 = N(n) = 2^{B(n)}$, i.e. $B(n) = 1$.

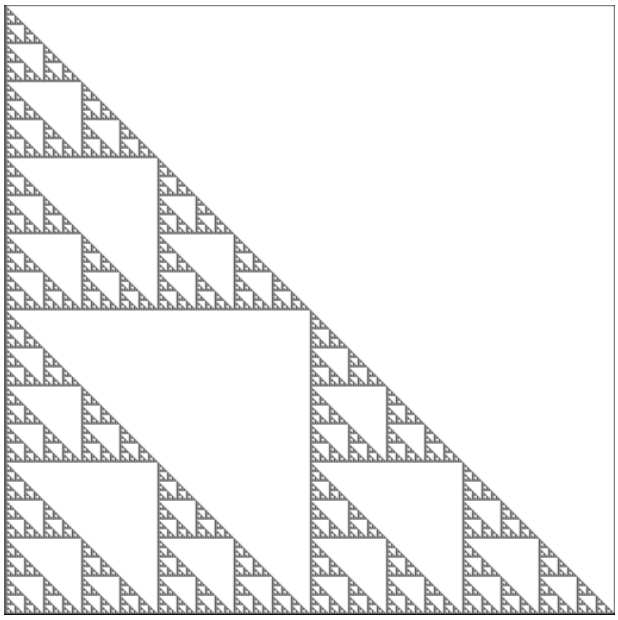


Figura 2.7: 2^{10} pasos de evolución de la regla 60.

Ahora vale la pena estimar la cantidad de números de Fermat F_n necesarios para alcanzar el renglón m-ésimo de la regla 60, para darnos una idea del crecimiento del autómata en función del crecimiento de los números de Fermat.

Lema 2.2.

$$\sum_{i=0}^{n} log_2(2^{2^i}+1) = 2^{n+1} - 1 + \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) \qquad \forall n \in \mathbb{Z}^+.$$

Demostración. Sea $n \in Z^+$. Veamos que

$$\sum_{i=0}^{n} log_2(2^{2^i}+1) = \sum_{i=0}^{n} log_2(2^{2^i}(1+\frac{1}{2^{2^i}})) = \sum_{i=0}^{n} log_2(2^{2^i}) + \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}})$$

$$= \sum_{i=0}^{n} 2^i + \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) = 2^{n+1} - 1 + \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}})$$

□

Lema 2.3.

$$0 \leq \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) \leq 1 \qquad \forall n \in Z^+.$$

Demostración. Sea $n \in \mathbb{Z}^+$. Claramente $0 \leq \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}})$ pues $log_2(1+\frac{1}{2^{2^i}}) > log_2(1) = 0$ para toda $i \in \{0,\ldots,n\}$. Así, basta con demostrar que $\sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) \leq 2$.

Por propiedades de logaritmo sabemos que $\sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) = log_2(\prod_{i=0}^{n}(1+\frac{1}{2^{2^i}}))$ y, como $(1+\frac{1}{2^{2^i}}) > 1$ para toda $i \in \mathbb{N}$, se tiene que $\prod_{i=0}^{n}(1+\frac{1}{2^{2^i}}) < \prod_{i\geq 0}(1+\frac{1}{2^{2^i}})$, de forma que $\sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) < log_2(\prod_{i\geq 0}(1+\frac{1}{2^{2^i}}))$.

Veamos ahora que $\prod_{i\geq 0}(1+\frac{1}{2^{2^i}}) = \sum_{i\geq 0}\frac{1}{2^i} = 2$. Basta con que todo factor no nulo de $\prod_{i\geq 0}(1+\frac{1}{2^{2^i}})$ sea un único sumando de $\sum_{i\geq 0}\frac{1}{2^i}$. Si dicho sumando es 1, entonces claramente es un único sumando de $\sum_{i\geq 0}\frac{1}{2^i}$. En otro caso, es un número de la forma

$$2^{-(2^{i_0}+2^{i_1}+\cdots+2^{i_k})}$$

para alguna $k \in \mathbb{N}$. Luego, $2^{i_0}+2^{i_1}+\cdots+2^{i_k}$ es la expresión en base 2 de algún natural n, por lo que es única. Así,

$$2^{-(2^{i_0}+2^{i_1}+\cdots+2^{i_k})} = 2^{-n}$$

para algún único $n \in \mathbb{N}$. Luego entonces es un único sumando de $\sum_{i\geq 0}\frac{1}{2^i}$, de forma que $\prod_{i\geq 0}(1+\frac{1}{2^{2^i}}) = \sum_{i\geq 0}\frac{1}{2^i} = 2$.

Por lo tanto

$$\sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}}) < log_2(\prod_{i\geq 0}(1+\frac{1}{2^{2^i}})) = log_2(\sum_{i\geq 0}\frac{1}{2^i}) = log_2(2) = 1$$

□

Teorema 2.8. [4]*Sea F_i el i-ésimo número de Fermat. Entonces*

$$2^{n+1} \leq \lfloor log_2(\prod_{i=0}^{n} F_i)\rfloor + 1 \leq 2^{n+1}+1 \qquad \forall n \in \mathbb{Z}^+.$$

Demostración. Sea $n \in \mathbb{Z}^+$. Por el lema 2.2 sabemos que

$$\lfloor log_2(\prod_{i=0}^{n} F_i)\rfloor + 1 = \lfloor \sum_{i=0}^{n} log_2(F_i)\rfloor + 1 = \lfloor \sum_{i=0}^{n} log_2(2^{2^i}+1)\rfloor + 1 = \lfloor 2^{n+1} - 1 + \sum_{i=0}^{n} log_2(1+\frac{1}{2^{2^i}})\rfloor + 1$$

y por el lema 2.3 tenemos que

$$2^{n+1} \leq \lfloor log_2(\prod_{i=0}^{n} F_i)\rfloor + 1 \leq 2^{n+1}+1.$$

□

Dado que el número de dígitos (en base 2) de un número $k \in \mathbb{Z}^+$ está dado por $\lfloor log_2(k)\rfloor + 1$, el teorema anterior nos dice que el número de dígitos de $\prod_{i=0}^{n} F_n$ está entre 2^{n+1} y $2^{n+1}+1$. Ahora bien, dado que la Regla 60 es un autómata tal que a cada iteración únicamente crece una celda (hacia la derecha) y el renglón 0-ésimo contiene una celda, entonces el renglón r de la Regla 60 contiene $r+1$ celdas activas. Luego, la fórmula anterior

[4]Este teorema y su demostración (junto con los lemas que lo preceden) fueron los primeros resultados del autor referentes al presente trabajo.

nos dice que el número $\prod_{i=0}^{n} F_n$ se encuentra codificado en el $2^{n+1}-1$-ésimo renglón o en 2^{n+1}-ésimo renglón. Es decir, para alcanzar el rengón $2^{n+1}-1$, es necesario multiplicar los primeros $n+1$ números de Fermat ($\prod_{i=0}^{n} F_i$).

Más aún, por la observación 2.11, sabemos que en las potencias de 2 se encuentran los números de Fermat, y dado que $\prod_{i=0}^{n} F_i$ no es un número de Fermat, entonces sabemos que se encuentra exactamente en el renglón número $2^{n+1}-1$, por lo que su número de dígitos en binario es exactamente 2^{n+1}.

Corolario 2.3. *Sea F_i el i-ésimo número de Fermat. Entonces*

$$\lfloor log_2(\prod_{i=0}^{n} F_i)\rfloor + 1 = 2^{n+1} \qquad \forall n \in \mathbb{Z}^+.$$

Corolario 2.4. *Sea F_i el i-ésimo número de Fermat. Entonces el producto $\prod_{i=0}^{n} F_i$ se encuentra codificado en el renglón número $2^{n+1}-1$ (comenzando a contar desde el renglón 0-ésimo).*

Demostración. Las demostraciones de ambos corolarios se encuentran en la exposición anterior. □

Más adelante veremos que estos resultados parciales no son más que corolarios del resultado general, que nos indica exactamente la descomposición de los renglones de la Regla 60 en productos de números de Fermat (ver ecuación 2.6).

Continuamos con nuestra investigación acerca de los números que aparecen en los renglones de la Regla 60, procederemos como en [12], definiendo la siguiente expresión:

Sea

$$t_{p,n} := \sum_{i=0}^{n} \big[\binom{n}{i} (\text{mód } p)\big] p^i$$

Notemos que en el caso $p=2$, $t_{p,n}$ simplemente es la expresión binaria del n-ésimo renglón de la Regla 60. Denotaremos por $rep_p(n)$ a la representación en base p de n, es decir, $rep_p(\sum_{j=0}^{k} n_j p^j) = n_k \dots n_1 n_0$ y, en particular, $rep_p(t_{p,n}) = \binom{n}{n} \dots \binom{n}{1}\binom{n}{0}$ (donde cada coeficiente binomial está reducido módulo p). Así, tenemos el siguiente teorema.

Teorema 2.9. *Sean p un primo, $n \in \mathbb{N}, n \geq 1, n = \sum_{j=0}^{k} n_j p^j$ con $n_k \neq 0$ y $s := n \ (mód\ p^k)$, entonces*

$$t_{p,n} = \sum_{m=0}^{n_k} \big[\sum_{j=0}^{s} \big(\binom{n_k}{m}\binom{s}{j} \ (mód\ p)\big) p^j\big] p^{mp^k} \tag{2.4}$$

Demostración. La demostración escapa las intenciones de este trabajo, y puede consultarse en [12]. □

Observación 2.12. *Lo que aparece adentro de los paréntesis cuadrados en la ecuación 2.4 es cada entrada de la expresión en base p de $t_{p,s}$ (es decir, cada letra de la palabra $rep_p(t_{p,s})$) multiplicada por $\binom{n_k}{m}$ y reducida módulo p. Así, la ecuación 2.4 puede ser vista como una fórmula recursiva para computar $t_{p,n}$ a partir de $t_{p,s}$.*

Corolario 2.5. *Sean $n \in \mathbb{N}, n \geq 1, F_k$ el k-ésimo número de Fermat. Entonces*

$$t_{2,n} = t_{2,s}F_k \tag{2.5}$$

Demostración. Notemos que para $p = 2$, tenemos que $n_k = 1$ (pues $0 < n_k < 2$ y $n_k \neq 0$), de forma que 2.4 se convierte en

$$t_{2,n} = \sum_{m=0}^{1} \big[\sum_{j=0}^{s} \big(\binom{1}{m} \binom{s}{j} \pmod 2 \big) 2^j \big] 2^{m2^k}.$$

Y más aún, dado que $\binom{1}{0} = \binom{1}{1} = 1$

$$t_{2,n} = \sum_{m=0}^{1} \big[\sum_{j=0}^{s} \big(\binom{s}{j} \pmod 2 \big) 2^j \big] 2^{m2^k}$$

i.e.

$$t_{2,n} = \sum_{m=0}^{1} \big[t_{2,s} \big] 2^{m2^k} = t_{2,s}(2^0 + 2^{2^k})$$

y finalmente

$$t_{2,n} = t_{2,s}F_k.$$

□

A partir de la ecuación anterior podemos confirmar la observación 2.11, pues para que en un renglón de la Regla 60 aparezca codificado un número de Fermat necesitamos que $t_{2,s} = 1$, es decir, $s = 0$. Así, para toda $k \in \mathbb{N}$ se tiene que

$$t_{2,2^k} = F_k$$

tal como lo afirma dicha observación.

Corolario 2.6. *Sean $n \in \mathbb{N}, n \geq 1, F_j$ el j-ésimo número de Fermat. Entonces*

$$t_{2,n} = \prod_{j:n_j=1} F_j. \tag{2.6}$$

Demostración. Por inducción sobre el número de 1's en la expansión binaria de n. Sea $B(n)$ la función que contabiliza el número de 1's en la expansión binaria de n.

Caso base: $B(n) = 2$. En dado caso $n = 2^k + 2^j$ p.a. $j, k \in \mathbb{N}, j < k$. Luego, iterando la ecuación 2.5 tenemos que

$$t_{2,2^k+2^j} = F_k t_{2^j} = F_k F_j$$

Hipótesis de inducción: Supongamos que si $B(n) = k$, entonces $t_{2,n} = \prod_{j:n_j=1} F_j$.

Paso inductivo: $B(n) = k + 1$. Supongamos sin pérdida de generalidad que $n = \sum_{j\in I} n_j 2^j$ con I un conjunto de índices de cardinalidad $k + 1$, $n_j = 1$ si y sólo si $j \in I$, y sea l el mayor natural tal que $n_l 2^l \neq 0$ en la expansión binaria de n. Sea $m = n - 2^l$. Notemos que, por construcción, $B(m) = k$. Así, iterando la ecuación 2.5 y por hipótesis de inducción tenemos que

$$t_{2,n} = F_l t_{2,m} = F_l \prod_{j:m_j=1} F_j = \prod_{j:n_j=1} F_j.$$

□

El teorema anterior nos dice que $t_{2,n}$ es el producto de todos los números de Fermat cuyos índices aparecen como exponentes de cifras significativas en la expansión binaria de n.

Ejemplo 2.2. *Sea $n = 5$, $n = 2^2 + 2^0$, de forma que $t_{2,5} = F_2 F_0 = 17 * 3 = 51$.*

Sea $n = 5346$, $n = 2^{12} + 2^{10} + 2^7 + 2^6 + 2^5 + 2^1$, de forma que $t_{2,5346} = F_{12}F_{10}F_7F_6F_5F_1$.

El lector cuidadoso podría pensar en generalizar el teorema 2.9 para $p \neq 2$. Por ejemplo, fijándonos en el triángulo de Pascal módulo p y definiendo $f_p(n) : \mathbb{N} \to \mathbb{N}$ como la función que convierte a base p el número en el renglón n y $F_n^p := p^{p^n} + 1$ como los números de Fermat generalizados (notemos que F_n^2 es simplemente el n-ésimo número de Fermat), pensar en la relación que existe entre $f_p(n)$ y F_n^p. A pesar de que, en general, no es cierto que

$$t_{p,n} = \prod_{j:n_j=1} F_n^p$$

para $p \neq 2$, sí existe una relación entre ambas funciones. Sin embargo, para expresar dicha relación es necesario introducir terminología y conceptos que escapan las intenciones del presente texto, por lo que referimos al lector a [12].

Las siguientes imágenes muestran la configuración del autómata Regla 60 después de $2^9 + 1$ pasos tomados módulo p para distintos primos.

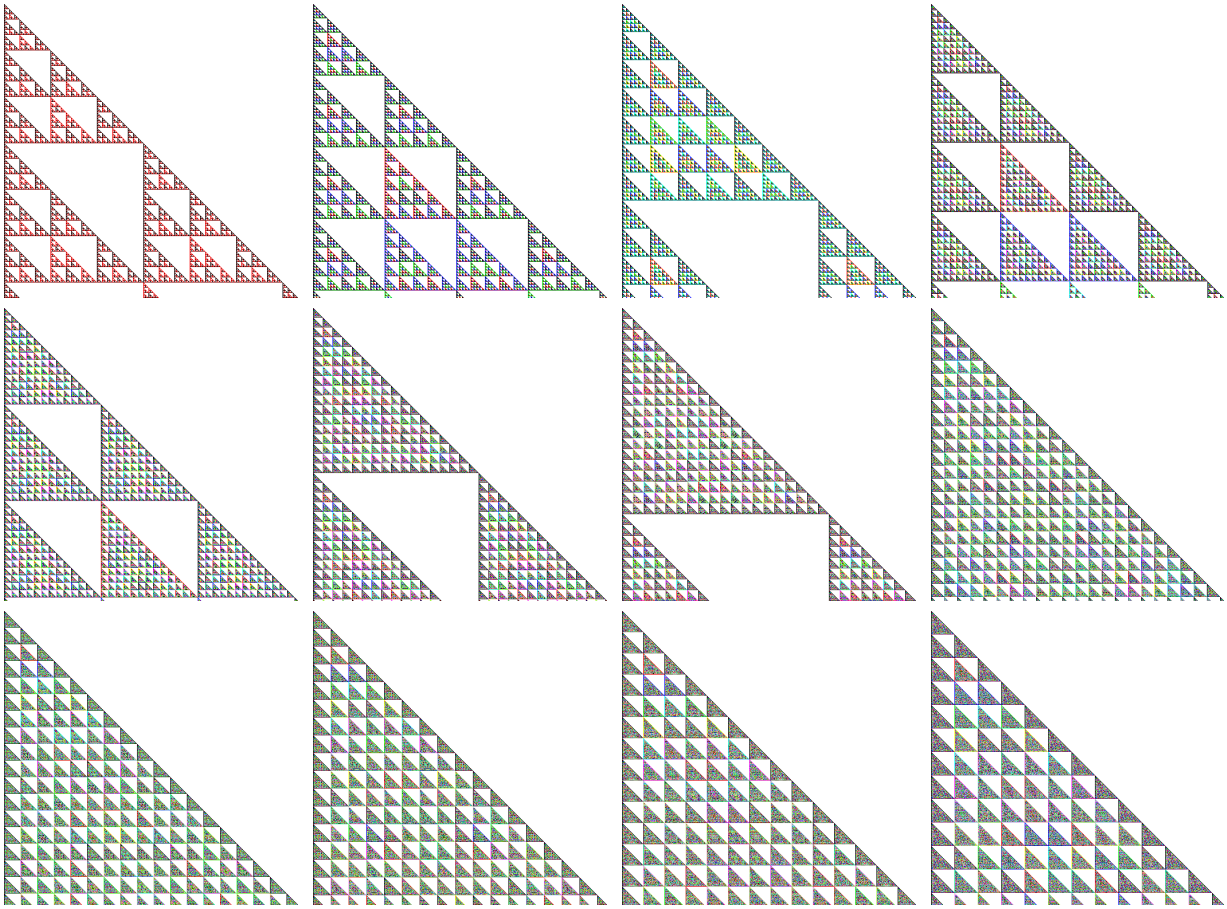


Figura 2.8: Triángulo de Pascal módulo p para $p = 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41$ (en orden lexicográfico). Distintos colores representan distintas clases residuales.

Resulta interesante observar los patrones resultantes al considerar el triángulo de Pascal módulo m, con m compuesto.

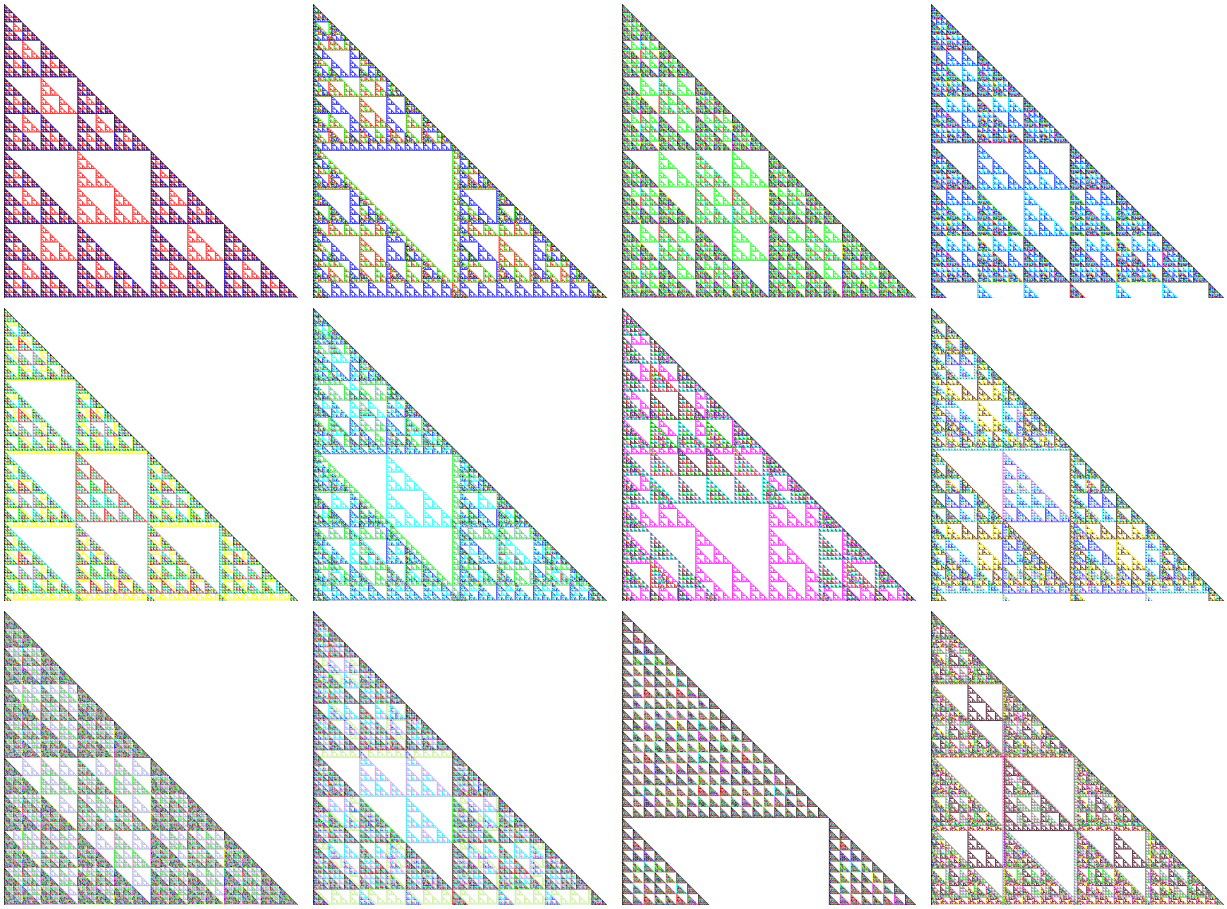


Figura 2.9: Triángulo de Pascal módulo m para $m = 4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 19, 20$ (en orden lexicográfico). Distintos colores representan distintas clases residuales.

2.3.3. Generalizaciones del Triángulo de Pascal

La ecuacion 2.1 nos sugiere la construcción de un autómata celular en cuyos renglones podamos encontrar ciertos coeficientes multinomiales

$$\binom{n}{k_1, k_2, \ldots, k_m}$$

Sea $A = (d, S, N, f)$ un autómata celular unidimensional ($d = 1$) con conjunto de estados $S = \mathbb{Z}$, radio de vecindad $N = (1, 2, \ldots, n)$ y $f(a_{i-n}, a_{i-n+1}, \ldots, a_i) = \sum_{j=-n}^{0} a_{i+j}$. Dicho autómata tendrá como dipolinomio de evolución a

$$\mathbb{T}(x) = 1 + x + x^2 + \cdots + x^n.$$

La ecuación 2.1 y el teorema multinomial nos aseguran que, si $A^0(x) = 1$, entonces el polinomio característico del renglón t-ésimo del autómata propuesto será

$$(1+x+x^2+\cdots+x^n)^t 1^t = \sum_{k_0+k_1+\cdots+k_N=t;\ \ k_i\geq 0} \binom{t}{k_0,\ldots,k_n} \prod_{1\leq m\leq t} x^{mk_m}$$

donde

$$\binom{t}{k_0,\ldots,k_n} = \frac{t!}{k_0!\cdots k_n!}.$$

Observemos que los coeficientes del polinomio característico son precisamente los coeficientes multinomiales que suman n, y las variables x_j simplemente nos indican el lugar en el que aparecen dichos coeficientes. Para obtener dichos coeficientes módulo m, basta con tomar $S = \mathbb{Z}_m$ y realizar las operaciones de los coeficientes en el anillo correspondiente.

2.3.3.1. Ejemplos

Consideremos el autómata celular $A = (1, \mathbb{Z}_2, N, f)$ donde $f(a_{i-2}, a_{i-1}, a_i) = \sum_{j=-2}^{0} a_{i+j}$. Dicho autómata es la Regla 150 desplazada presentado en la figura 2.1. Observemos que en su renglón t-ésimo se encuentran codificados los coeficientes del polinomio

$$(1+x+x^2)^t$$

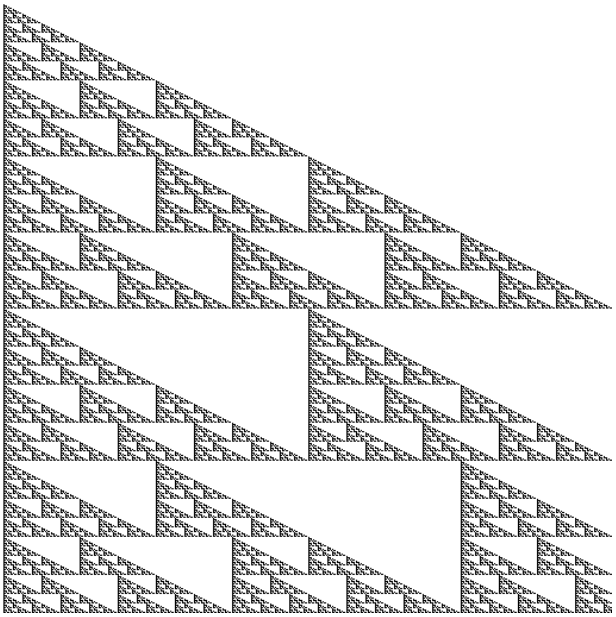


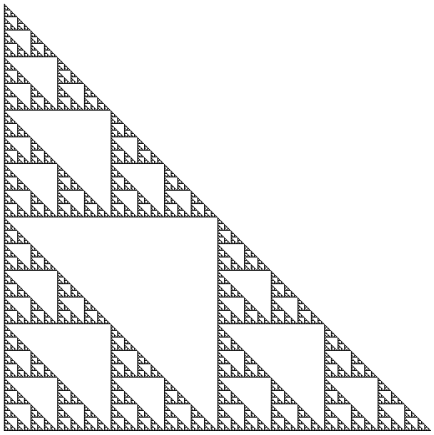
Figura 2.10: Autómata con dipolinomio de evolución $\mathbb{T}(x) = 1 + x + x^2$ y conjunto de estados $S = \mathbb{Z}_2$.

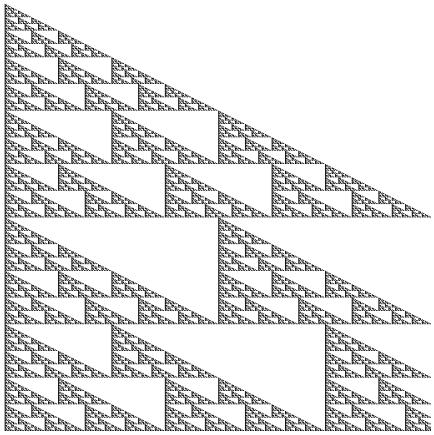
Más aún, para construir un autómata en cuyos renglones aparezcan los coeficientes de cualquier polinomio

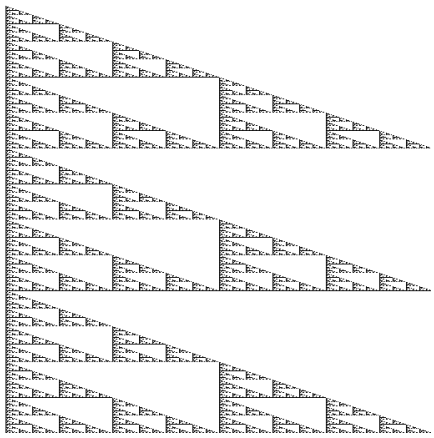
$$\mathbb{T}(x) = (\alpha_0 + \alpha_1 x^1 + \alpha_2 x^2 + \cdots + \alpha_n x^n)$$

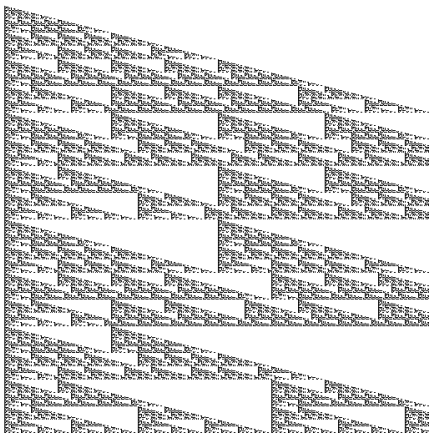
basta con considerar al autómata unidimensional $A = (d, S, N, f)$ $(d = 1)$ con conjunto de estados $S = \mathbb{Z}$, radio de vecindad $N = (-n, \ldots, -2, -1, 0)$ y $f(a_{i-n}, a_{i-n+1}, \ldots, a_i) = \sum_{j=-n}^{0} \alpha_{-j} a_{i+j}$.

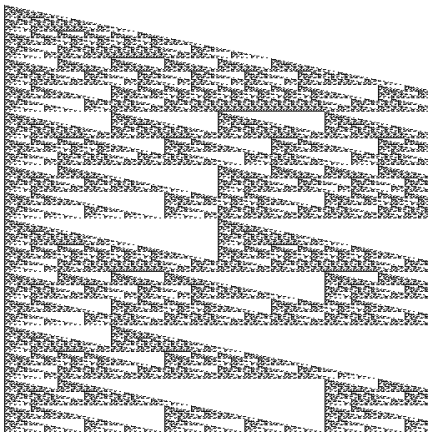
Las siguientes imágenes corresponden a autómatas unidimensionales de la forma $A = (1, \mathbb{Z}_2, N, f)$ con $N = (-n, \ldots, -2, -1, 0)$ y $f(a_{i-n}, a_{i-n+1}, \ldots, a_i) = \sum_{j=-n}^{0} \alpha_{-j} a_{i+j}$ para distintos valores de n. En cada caso, el dipolinomio de evolución de cada uno de ellos está dado por $\mathbb{T}(x) = 1 + x + \cdots + x^n$.

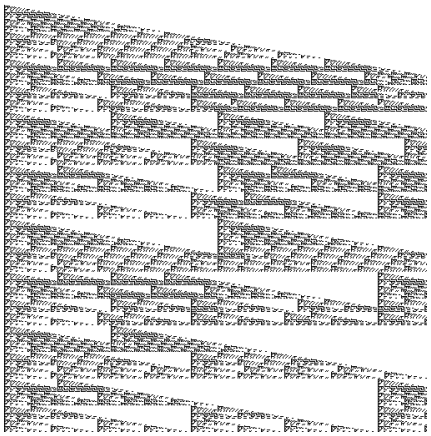


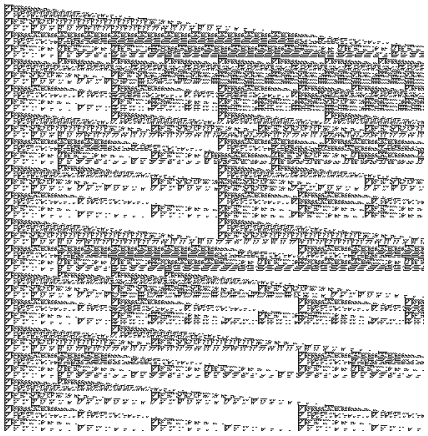


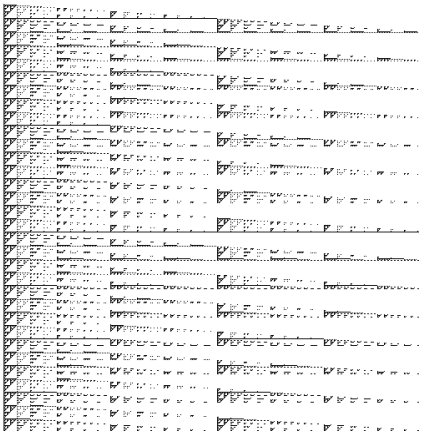


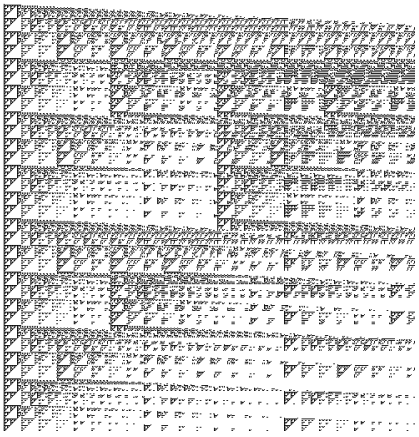


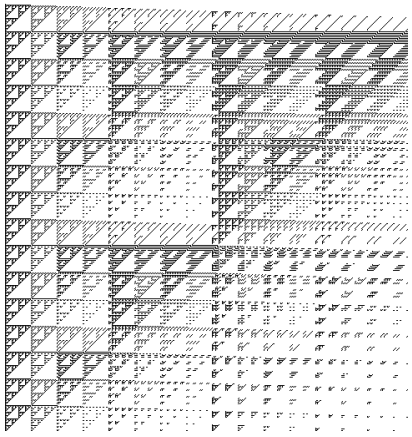












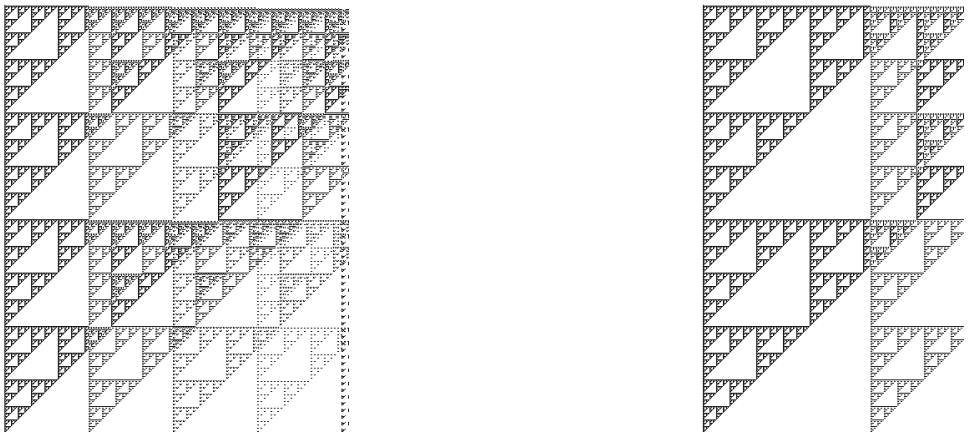


Figura 2.11: Autómatas con dipolinomios de evolución dados por $\mathbb{T}(x) = 1+x+\cdots+x^n$ con $n = 1, 2, 3, 4, 5, 6, 10, 15, 20, 30, 100, 200$ en orden lexicográfico. Todas las imágenes están alineadas.

2.3.3.2. Particiones de enteros

Una interpretación interesante de los renglones de los autómatas presentados en los ejemplos anteriores es la siguiente. Consideremos dos dados, en cuyas caras se encuentran los números $1, 2, 3, 4, 5, 6$. Al tirar los dados, podemos preguntarnos: ¿de cuántas formas puedo obtener el número k? Por ejemplo,

$$\begin{aligned} 7 &= 1+6 \\ &= 6+1 \\ &= 2+5 \\ &= 5+2 \\ &= 3+4 \\ &= 4+3 \end{aligned} \qquad \begin{aligned} 6 &= 1+5 \\ &= 5+1 \\ &= 2+4 \\ &= 4+2 \\ &= 3+3 \end{aligned}$$

Cabe mencionar que $1+6$ y $6+1$ las contaremos como formas distintas de sumar 7 pues estamos pensando en dos dados distinguibles. Ambas sumas representan escenarios distintos de los dados.

Una forma ingeniosa de abordar la pregunta anterior es considerando funciones generadoras.

Definición 2.11. *Una **función generadora** es una serie formal de potencias cuyos coeficientes codifican información sobre una sucesión dada* (a_n).

Pensemos en

$$p(x) = x^1 + x^2 + \cdots + x^6$$

y observemos que

$$\begin{aligned}p(x)^2 =& (x^1+x^2+\cdots+x^6)(x^1+x^2+\cdots+x^6)\\ =& x^{1+1}+x^{2+1}+x^{1+2}+x^{2+2}+x^{1+3}+x^{3+1}+x^{1+4}+x^{4+1}+\\ & x^{2+3}+x^{3+2}+x^{1+5}+x^{5+1}+x^{2+4}+x^{4+2}+x^{3+3}+\cdots+x^{6+6}\\ =& x^2+2x^3+3x^4+4x^5+5x^6+6x^7+5x^8+4x^9+3x^{10}+2x^{11}+x^{12}.\end{aligned}$$

Denotemos por $[x^n]p(x)$ al coeficiente de x^n en $p(x)$. Notemos que $[x^n]p(x)^2$ es exactamente el número de formas de obtener n con dos dados de 6 caras numeradas con $1,2,3,4,5,6$ ya que, por definición,

$$[x^n]p(x)^2 = \sum_{j+k=n} 1$$

con $j,k \in \{1,2,3,4,5,6\}$. Es decir, $[x^n]p(x)^2$ son las particiones de enteros positivos de n con exactamente dos factores en el conjunto $\{1,2,3,4,5,6\}$.

Definición 2.12. *Sea $n,k \in \mathbb{Z}^+$. Una **partición** de n en k factores es una secuencia de enteros positivos $(\alpha_1,\ldots,\alpha_k)$ tales que $\alpha_1+\cdots+\alpha_k = n$. Diremos que las particiones de n con k factores son m si $|\{(\alpha_1,\ldots,\alpha_k) : \alpha_1+\cdots+\alpha_k = n\}| = m$.*

Definición 2.13. *Sea $n,k \in \mathbb{Z}^+$ y (a_m) una sucesión finita de enteros. Una **partición de n en k factores de** (a_m) es una partición de n en k factores tales que $\alpha_i = a_l$ para alguna $1 \le l \le m$ para toda $i = 1,2,\ldots,k$. Diremos que las particiones de n con k factores en S son m si $|\{(\alpha_1,\ldots,\alpha_k) : \alpha_i = a_l, \alpha_1+\cdots+\alpha_k = n\}| = m$.*

Observación 2.13. *Las particiones de n en k factores de $a_1,\ldots,a_m$ las podemos pensar como el número de formas que existen de obtener el número n lanzando k dados en cuyas caras aparecen $a_1,\ldots,a_m$. Así, tenemos el siguiente teorema.*

Teorema 2.10. *Sea $n,k \in \mathbb{Z}^+$ y (a_m) una sucesión finita de enteros. La función generadora del número de particiones de n en k factores de (a_m) es $p(x)^k$ donde*

$$p(x) = \sum_{1\le i\le m} x^{a_i}$$

Demostración. Por definición

$$[x^n]p(x)^k = \sum_{a_{i_1}+\cdots+a_{i_k}=n} 1$$

para toda $n \in \mathbb{Z}^+$. □

Dado que podemos construir un autómata elemental cuyo dipolinomio de evolución sea cualquier polinomio, en particular podemos considerar autómatas cuyo dipolinomio de evolución sea de la forma $\mathbb{T}(x) = 1+x+\cdots+x^m$. En este caso tendremos que la función generadora del t-ésimo renglón será $\mathbb{T}(x)^t$, misma que codificará las particiones del entero

n con t factores en $(0, 1, \ldots, m)$ (número de formas de obtener n con t dados en cuyas caras figuran los números $(0, 1, \ldots, m)$). A continuación presentamos algunos ejemplos junto con los autómatas correspondientes.

Ejemplo 2.3. *Sea $k = 2, (a_m) = (1, 2, 3, 4, 5, 6)$. Estamos pensando en el número de formas de obtener cualquier $n \in \mathbb{Z}^+$ usando 2 dados en cuyas caras aparecen los números $(1, 2, 3, 4, 5, 6)$. Éstas están dadas por $[x^n]p(x)^2$, donde $p(x) = x + x^2 + x^3 + x^4 + x^5 + x^6$* i.e.

$$p(x)^2 = x^2 + 2x^3 + 3x^4 + 4x^5 + 5x^6 + 6x^7 + 5x^8 + 4x^9 + 3x^{10} + 2x^{11} + x^{12}.$$

En efecto,

$$\begin{array}{llllllll}
2 = 1+1 & 3 = 2+1 & 4 = 3+1 & 5 = 4+1 & 6 = 5+1 & 7 = 6+1 & 8 = 6+2 & 9 = 6+3 \\
 & = 1+2 & = 1+3 & = 1+4 & = 1+5 & = 1+6 & = 2+6 & = 3+6 \\
 & & = 2+2 & = 3+2 & = 4+2 & = 5+2 & = 5+3 & = 5+4 \\
 & & & = 2+3 & = 2+4 & = 2+5 & = 3+5 & = 4+5 \\
 & & & & = 3+3 & = 4+3 & = 4+4 & \\
 & & & & & = 3+4 & &
\end{array}$$

$$\begin{array}{lll}
10 = 6+4 & 11 = 6+5 & 12 = 6+6 \\
= 4+6 & = 5+6 & \\
= 5+5 & &
\end{array}$$

Consideremos el autómata elemental cuyo dipolinomio de evolución es $\mathbb{T}(x) = p(x) = \sum_{i=1}^{6} x^i$, $S = \mathbb{Z}_2$. Esto implica que $a_i^{t+1} = \sum_{n=-6}^{1} a_{i+n}^t$.

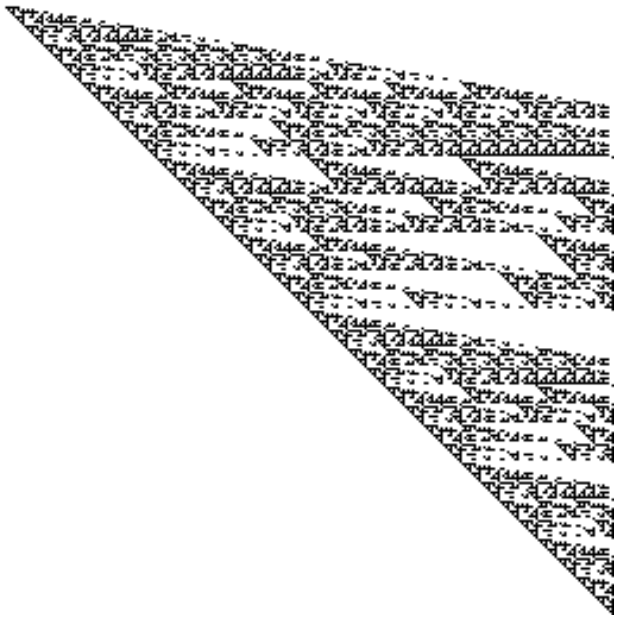


Figura 2.12: Autómata con dipolinomio de evolución $\mathbb{T}(x) = x + x^2 + x^3 + x^4 + x^5 + x^6$.

Ejemplo 2.4. *Sea* $k = 3, (a_m) = (1, 2)$. *Estamos pensando en el número de formas de obtener cualquier* $n \in \mathbb{Z}^+$ *usando* 3 *dados en cuyas caras aparecen los números* $(1, 2)$. *Éstas están dadas por* $[x^n]p(x)^3$, *donde* $p(x) = x + x^2$ i.e.

$$p(x)^3 = x^3 + 3x^4 + 3x^5 + x^6.$$

En efecto,

$$\begin{array}{llll} 3 = 1+1+1 & 4 = 2+1+1 & 5 = 1+2+2 & 6 = 2+2+2 \\ & \quad = 1+2+1 & \quad = 2+2+1 & \\ & \quad = 1+1+2 & \quad = 2+1+2 & \end{array}$$

Consideremos el autómata elemental cuyo dipolinomio de evolución es $\mathbb{T}(x) = p(x) = x + x^2$, $S = \mathbb{Z}_2$. *Esto implica que* $a_i^{t+1} = a_{i-1}^t + a_{i-2}^t$.

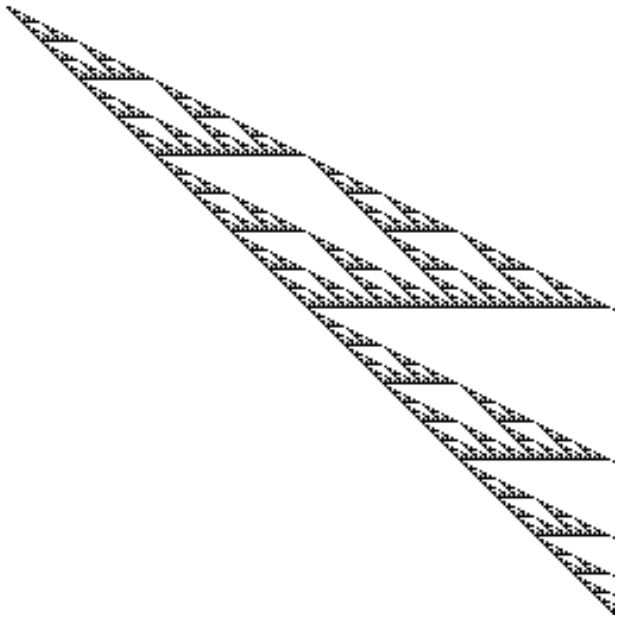


Figura 2.13: Autómata con dipolinomio de evolución $\mathbb{T}(x) = x + x^2$.

Ejemplo 2.5. *Sea* $k = 2, (a_m) = (0, 1, 8)$. *Estamos pensando en el número de formas de obtener cualquier* $n \in \mathbb{Z}^+$ *usando* 2 *dados en cuyas caras aparecen los números* $(0, 1, 8)$. *Éstas están dadas por* $[x^n]p(x)^2$, *donde* $p(x) = x^0 + x + x^8 = 1 + x + x^8$ i.e.

$$p(x)^2 = 1 + 2x + x^2 + 2x^8 + 2x^9 + x^{16}$$

En efecto,

$$\begin{array}{lllllll} 0 = 0+0 & 1 = 0+1 & 2 = 1+1 & 8 = 0+8 & 9 = 8+1 & 16 = 8+8 \\ & \quad = 1+0 & & \quad = 8+0 & \quad = 1+8 & \end{array}$$

Consideremos el autómata elemental cuyo dipolinomio de evolución es $\mathbb{T}(x) = p(x) = 1 + x + x^8$, $S = \mathbb{Z}_2$. *Esto implica que* $a_i^{t+1} = a_i^t + a_{i-1}^t + a_{i-8}^t$.

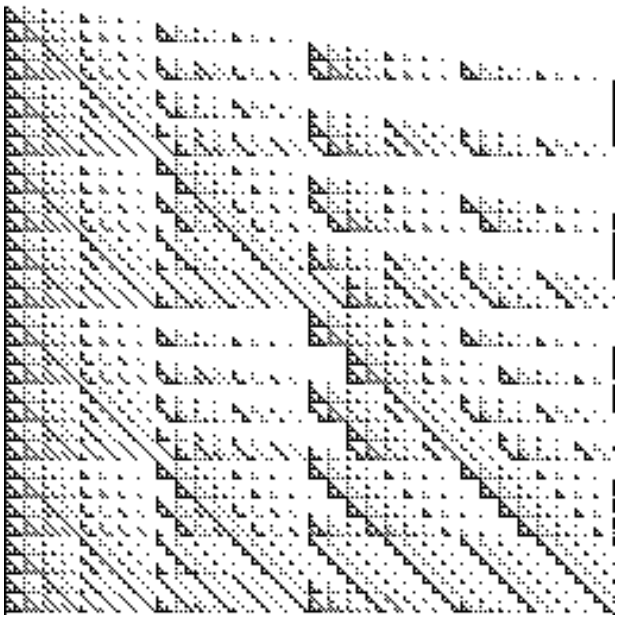


Figura 2.14: Autómata con dipolinomio de evolución $\mathbb{T}(x) = 1 + x + x^8$.

Ejemplo 2.6. *Sea* $k = 3, (a_m) = (1, 1, 2, 2, 3, 3)$. *(cada número aparece* 2 *veces, es decir, nuestro dado es de* 6 *caras). Tenemos entonces que* $p(x) = x + x + x^2 + x^2 + x^3 + x^3 = 2(x + x^2 + x^3)$ *y* $p(x)^3 = 8(x^3 + 3x^4 + 6x^5 + 7x^6 + 6x^7 + 3x^8 + x^9)$.

Si consideramos el autómata elemental cuyo dipolinomio de evolución es $\mathbb{T}(x) = p(x)$ *con* $S = \mathbb{Z}_2$, *todo el autómata va a ser el autómata trivial pues* $p(x)$ *es par para toda* $x \in \mathbb{Z}^+$. *Así, lo consideraremos con* $S = Z_3$, *de forma que* $a_i^t = 2(a_{i-1}^t + a_{i-2}^t + a_{i-3}^t)$

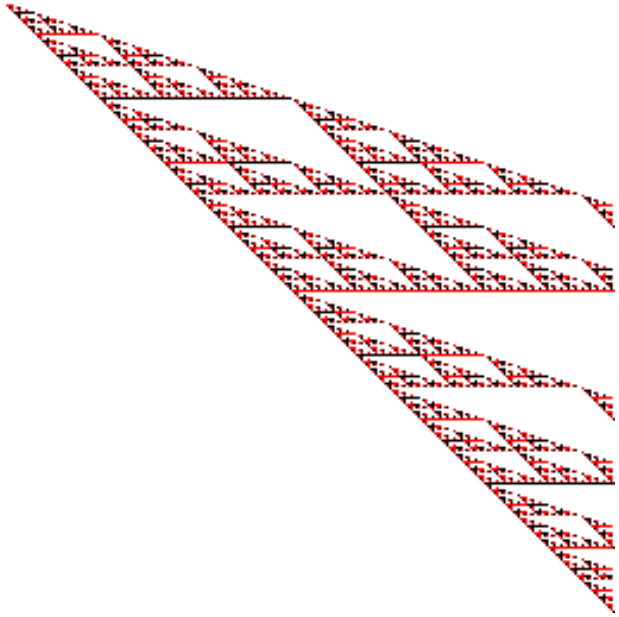


Figura 2.15: Autómata con dipolinomio de evolución $\mathbb{T}(x) = 2(x + x^2 + x^3)$ sobre $\mathbb{Z}_3$.

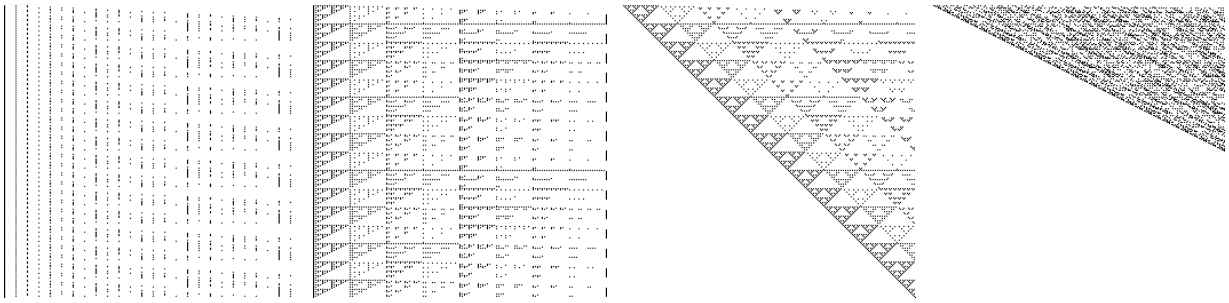


Figura 2.16: Autómatas elementales sobre Z_2 con dipolinomios de evolución $\mathbb{T}(x) = 1 + x^{10} + x^{20}$, $\mathbb{T}(x) = \sum_{n=0}^{15} x^{2n}$, $\mathbb{T}(x) = \sum_{n=0}^{15} x^{2n+1}$, $\mathbb{T}(x) = \sum_{p \text{ primo:} 1<p<100} x^p$ (en orden lexicográfico).

Más adelante veremos cómo construir un autómata que codifique las particiones de un entero $n \in \mathbb{Z}^+$ en factores de k listas finitas distintas $(a_m^1), (a_m^2), \ldots, (a_m^k)$. Esto correspondería al número de formas de obtener el número n lanzando k dados distintos tal que en las caras del primero de ellos aparecen $a_1^1, \ldots, a_m^1$, en las caras del segundo de ellos aparecen $a_1^2, \ldots, a_m^2$, etcétera. [Ver capítulo 3.1]

2.3.3.3. Autómatas que codifican probabilidades

Observemos que si $p(x)^k$ es la función generadora del número de particiones de n en k factores de (a_m), entonces $p(1)^k$ es el total de posibles configuraciones que pueden tomar los k dados al ser lanzados (pues $p(1)$ es el total de caras del dado). Así, $\frac{[x^n]p(x)^k}{p(1)^k}$ es la probabilidad de obtener el número n al lanzar k dados en cuyas caras figuran los elementos de la sucesión (a_m) (suponiendo que cada lanzamiento es independiente). Es decir, si Y_n es el evento "obtener el número n al lanzar los k dados en cuyas caras figura la lista (a_m)", entonces

$$\mathbb{P}(Y_n) = \frac{[x^n]p(x)^k}{p(1)^k}.$$

Así, tomando $S = \mathbb{R} \cap [0, 1]$ podemos interpretar los estados del autómata con dipolinomio de evolución $\mathbb{T}(x)$ como dicha probabilidad. En la n-ésima celda del renglón k-ésimo estará codificada la probabilidad de obtener n lanzando k dados en cuyas caras figura la lista (a_m).

Ejemplo 2.7. *Sea* $(a_m) = (1, 2, 3, 4, 5, 6)$ *de forma que* $p(x) = x + x^2 + x^3 + x^4 + x^5 + x^6$. *Observemos que* $p(1)^k = 6^k$ *para toda* $k \in \mathbb{Z}^k$, *de forma que*

$$\mathbb{P}(Y_n) = \frac{[x^n]p(x)^k}{6^k}$$

para toda $n \in \mathbb{Z}^+$.

Figura 2.17: Autómata en cuyo renglón k -ésimo codifica la probabilidad de obtener n al lanzar k dados de caras $(1,2,3,4,5,6)$ para $k = 1, \ldots, 5$.

Ejemplo 2.8. *Sea* $(a_m) = (1,2)$ *de forma que* $p(x) = x + x^2$. *Observemos que* $p(1)^k = 2^k$ *para toda* $k \in \mathbb{Z}^k$, *de forma que*

$$\mathbb{P}(Y_n) = \frac{[x^n]p(x)^k}{2^k}$$

para toda $n \in \mathbb{Z}^+$.

Figura 2.18: Autómata en cuyo renglón k -ésimo codifica la probabilidad de obtener n al lanzar k dados de caras $(1,2)$ para $k = 1, \ldots, 5$.

Ejemplo 2.9. *Sea* $(a_m) = (0,1,8)$ *de forma que* $p(x) = 1 + x + x^8$. *Observemos que* $p(1)^k = 3^k$ *para toda* $k \in \mathbb{Z}^k$, *de forma que*

$$\mathbb{P}(Y_n) = \frac{[x^n]p(x)^k}{3^k}$$

para toda $n \in \mathbb{Z}^+$.

Figura 2.19: Autómata en cuyo renglón k -ésimo codifica la probabilidad de obtener n al lanzar k dados de caras $(0, 1, 8)$ para $k = 1, \ldots, 5$.

2.3.4. Reducibilidad computacional

Uno de los pleanteamientos principales de Wolfram [18] es que los sistemas -en este caso autómatas celulares- con comportamiento *suficientemente sencillo* van a presentar reducibilidad computacional, es decir, que existirá otro proceso de cómputo tal que computará E_n con entrada n en una cantidad de pasos menor que la que nos llevaría simular el autómata. Específicamente escribe:

> *One can specify the number of steps t that one wants by giving the sequence of digits in t. And for systems with sufficiently simple behavior -say repetitive or nested- the pictures on page 774 indicate that one can typically determine the outcome with an amount of effort that is essentially proportional to the length of the digit sequence. [18]*

En el caso de la Regla 60, no sólo es claro "a ojo" que presenta comportamiento repetitivo y fractal, sino que en virtud de la ecuación 2.3 sabemos que el renglón t-ésimo codifica los coeficientes del polinomio $(x + 1)^t$ módulo 2, mismo que corresponde al famoso fractal conocido como Triángulo de Pascal. Entonces debería de ser el caso, según lo dicho por Wolfram, que exista una Máquina de Acceso Aleatorio M tal que, con entrada n, compute el renglón t-ésimo en tiempo $T(M) = O(n^2)$.

Teorema 2.11. *La Regla 60 es computacionalmente reducible.*

Demostración. Sea M la Máquina de Acceso Aleatorio que computa el algoritmo siguiente:

Algoritmo 4: Calcular $E_t(1)$

Entrada: t en base 2.
Salida: $E_t(1)$

$r_0 \leftarrow t$
$r_2 \leftarrow 1$
for $i \in \{1, \ldots, \lfloor log_2(t) \rfloor + 1\}$ **do**
  Leer el i-ésimo dígito de t (x_i).
  if $x_i == 1$ **then**
    $r_1 \leftarrow F_i$ en base 2.
    $r_2 \leftarrow r_2 * r_1$
  end
end
return r_2

En virtud de la ecuación 2.6 sabemos que para toda $t \in \mathbb{N}$, el algoritmo anterior en efecto computa $E_t(n)$. Ahora bien, dado que estamos trabajando en el modelo RAM, calcular F_i en base 2 es de orden $O(1)$ pues simplemente es escribir una sucesión finita de bits. Dado que el resto de las instrucciones también son de orden $O(1)$ y hay que calcular, a lo más, $\lfloor log_2(t) \rfloor + 1$ números de Fermat -pues ésa es la longitud de t escrito en base 2-, entonces el algoritmo anterior tiene complejidad $O(log_2(t))$.

□

Observación 2.14. *Quizá salte a la vista del lector el hecho de que multiplicar números de Fermat sea de orden $O(1)$, dado que son números bastante grandes. Notemos entonces que, incluso si pensamos en no usar la multiplicación de orden $O(1)$ del modelo RAM, sino en implementar una multiplicación* manual *de números de Fermat, el algoritmo anterior seguirá siendo más eficiente que simular el autómata. Esto es, multiplicar k números de Fermat $F_{a_1}, \ldots, F_{a_k}$ se reduce a calcular el conjunto potencia de $I := \{2^{a_1}, \ldots, 2^{a_k}\}$ y escribir un número con 1's en las posiciones indexadas por la suma de cada elemento de $\mathcal{P}(I)$ y 0's en el resto de las posiciones. Sea* prod *la función que multiplica k números de Fermat de forma manual, y consideremos el algoritmo siguiente:*

Algoritmo 5: Calcular $E_t(1)$

Entrada: *t en base 2.*
Salida: $E_t(1)$

$r_0 \leftarrow t$
$r_1 \leftarrow \emptyset$
$r_2 \leftarrow 1$

for $i \in \{1, \ldots, \lfloor log_2(t) \rfloor + 1\}$ **do**
  Leer el i-ésimo dígito de t (x_i).
  if $x_i == 1$ **then**
    $r_1 \leftarrow r_1 \cup \{2^i\}$
  end
  $r_2 \leftarrow prod(r_1)$
end
return r_2

Claramente hace lo mismo que el algoritmo anterior, por lo que en efecto calcula $E_t(1)$ para toda $t \in \mathbb{N}$. Ahora bien, la complejidad de $prod(I)$ es $O(2^k)$ donde $k < log_2(t) + 1$ i.e. *$O(t)$. Dado que esa función es llamada $log_2(t) + 1$ veces y el resto de las operaciones son de complejidad $O(1)$, la complejidad del algoritmo es de $O(tlog_2(t))$.*

Notemos que hemos encontrado una Máquina de Acceso Aleatorio que computa E_t con entrada t sin necesidad de computar las configuraciones anteriores y que además computa la t-ésima configuración más rápidamente que lo que lo haría el simulador de la Regla 60. En la página 744 de [18], Wolfram exhibe una Máquina de Turing -de hecho, un autómata celular- tal que con 6 estados distintos puede calcular E_t de forma recursiva -es decir, calculando previamente $E_0, E_1, \ldots, E_{t-1}$- en un tiempo menor que la simulación de la Regla 60.

2.3.5. Pseudocódigos

La siguiente función toma como entrada una matriz de $n \times n$, $M \in M_{n,n}(\{0,1\})$ que representará el espacio de configuraciones del autómata. Ésta será la función $f : S^m \to S$ que a cada celda le asocia un nuevo estado con base en el estado de las celdas de su vecindad al paso anterior.

Algoritmo 6: Función $f_{60} : S^2 \to S$

Entrada: $M \in M_{n,n}(\{0,1\}), i, j \in Z^+$
Salida: $x \in \{0,1\}$

$n \leftarrow$ número de columnas de la matriz M

Función $f_{60}(M, i, j)$
  if $j = 1$ **then**
    $M_{i,j} \equiv M_{i-1,n} + M_{i-1,j} (mód\ 2)$
  else
    $M_{i,j} \equiv M_{i-1,j-1} + M_{i-1,j} (mód\ 2)$
  end
  return $M_{i,j}$
end

En la observación 2.10 se hizo notar que al interpretar cada renglón de la Regla 60 como un número en base 10, los números resultantes eran productos de números de Fermat. Para estos fines, necesitamos:

1. Una función que, dado un renglón de la Regla 60, nos regrese su interpretación como número en base 10.

2. Una base de datos que contenga los primeros n números de Fermat.

3. Una función que factorice un número dado en números de Fermat.

Algoritmo 7: Función $PasarBase10 : \mathbb{R}^n \to \mathbb{Z}^+$

Entrada: Arreglo $\in \mathbb{R}^m$
Salida: $k \in \mathbb{Z}^+$

$n \leftarrow$ longitud de Arreglo

Función *PasarBase10(Arreglo)*
  indice = 0
  cuenta = 0
  for $i \in \{0, \ldots, n-1\}$ **do**
    if $Arreglo_{n-i} \neq 0$ **then**
      indice = $n - i$
      break
    end
  end
  for $j \in \{0, \ldots, indice - 1\}$ **do**
    cuenta = cuenta + $\text{Arreglo}_{\text{indice}-i} * 2^i$
  end
  return cuenta
end

El siguiente algoritmo se basa en la relación

$$F_{n+1} = (F_n - 1)^2 + 1 \quad , \quad \forall n \in \mathbb{N}$$

la cual se demuestra a partir de un cálculo directo.

Algoritmo 8: Crear los primeros $n+1$ números de Fermat.

Entrada: $n \in \mathbb{Z}^+$
Salida: Fermat $\in \mathbb{R}^{n+1}$

Fermat $\leftarrow$ arreglo vacío de enteros

insert 3 **in** Fermat
for $i \in \{1, \ldots, n\}$ **do**
  insert$(Fermat_i - 1)^2 + 1$ **in** Fermat
end

Para factorizar un número con números de Fermat, basta con recorrer el arreglo "Fermat" y dividir el número en cuestión entre sus divisores dentro del arreglo. Queremos que nos devuelva una lista con los índices de los números de Fermat que en efecto lo dividieron.

Algoritmo 9: Función $FactorizarFermat : \mathbb{Z}^+ \to \mathbb{R}^m$

Datos: Fermat $\in \mathbb{R}^{n+1}$
Entrada: $k \in \mathbb{Z}^+$
Salida: IndiceFactores $\in \mathrm{R}^m$.

$n \leftarrow$ longitud del arreglo Fermat
IndiceFactores $\leftarrow$ arreglo vacío de enteros

Función *FactorizarFermat(k)*
  for $i \in \{1, \ldots, n\}$ **do**
    if $k < Fermat_i$ **then**
      break
    else
      if $k \equiv 0(mód\ Fermat_i)$ **then**
        insert $i - 1$ **in** IndiceFactores
        $k = \frac{k}{\mathrm{Fermat}_i}$
      end
    end
  end
  return IndiceFactores
end

Para generar el triángulo de Pascal módulo m, no basta con generar el autómata de la regla 60, pues sí nos interesa el valor numérico de cada casilla y no sólo su estado. Así, hay que calcular el triángulo de Pascal completo mediante un algoritmo que como entrada reciba una matriz con la configuración inicial codificada en el primer renglón y un entero positivo m que será el módulo. De nuevo, dado que queremos representar un arreglo infinito con una matriz finita, identificaremos a las columnas de los extremos como adyacentes. Así, el algoritmo sería el siguiente.

Algoritmo 10: Función $CrearPascalModM : M_{i,j} \to \mathbb{R}^3$

Entrada: $M \in M_{n,n}(\mathbb{Z}^+), m \in \mathbb{Z}^+$
Salida: M modificada.

$n \leftarrow$ número de columnas de la matriz M

Función *CrearPascalModM(M,m)*
```
for i ∈ {2, ..., n} do
    for j ∈ {1, ..., n} do
        if j = 1 then
            M_{i,j} ≡ M_{i-1,n} + M_{i-1,j} (mód m)
        else
            M_{i,j} ≡ M_{i-1,j-1} + M_{i-1,j} (mód m)
        end
    end
end
return M
end
```

Para colorear la Regla 60 en blanco y negro, basta con usar la función ColorearMatriz descrita en el algoritmo 15. Pero para el caso del triángulo de Pascal módulo m necesitaremos modificar ligeramente dicha función. En este caso necesitaremos como dato un arreglo de colores y de nuevo una matriz auxiliar $M' \in M_{n\times n}(\{0,1\})$ cuyas entradas son incialmente todas blancas. Así, el algoritmo para colorear dichas imágenes sería el siguiente.

Algoritmo 11: Función $ColorearPascalModM : M_{i,j} \to \mathbb{R}^3$

Datos: ArregloColores $\in \mathbb{R}^m$
Entrada: $M \in M_{n,n}(\{0,1\}), M' \in M_{n,n}(\mathbb{R}^3), m \in \mathbb{Z}^+$
Salida: M' modificada.

$n \leftarrow$ número de columnas de la matriz M

Función *ColorearModM(M,M',m)*
```
for i ∈ {1, ..., n} do
    for j ∈ {1, ..., i} do
        for k ∈ {0, ..., m - 1} do
            if M_{i,j} ≤ k then
                M'_{i,j} = ArregloColores_{k+1}
                break
            end
        end
    end
end
return M'
end
```

Resulta útil contar con una función que, dado el color que vemos en la imagen, nos diga a qué clase residual pertenece dicho elemento del triángulo de Pascal. Para esto, dado el

mismo arreglo de colores con el que coloreamos el triángulo de Pascal previamente, basta con conocer el número en RGB del color deseado para recorrer dicho arreglo y ver a qué clase residual le fue asignado.

Algoritmo 12: Función $ChecarColor : \mathbb{R}^3 \to \mathbb{N}$

Datos: ArregloColores $\in \mathbb{R}^m$
Entrada: color $\in \mathbb{R}^3$
Salida: $k \in \mathbb{N}$

$n \leftarrow$ longitud de ArregloColores

Función *ChecarColor(color)*
  for $i \in \{1, \ldots, n\}$ **do**
    if $ArregloColor_i = color$ **then**
      print(Ese color representa a la clase $i - 1$)
    end
  end
end

2.4. La Regla 90 y el Triángulo de Pascal truncado

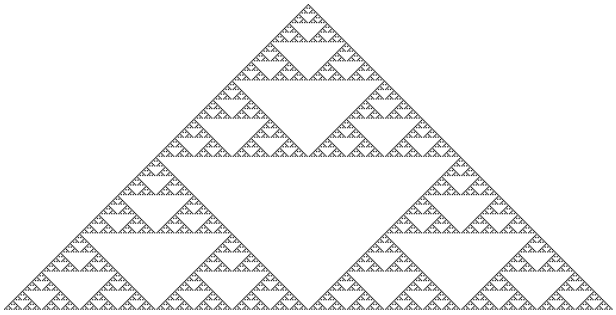


Figura 2.20: 2^8 pasos de evolución de la regla 90.

2.4.1. Relación con la Regla 60

Una propiedad que resulta no menos que asombrosa es la siguiente:

Consideremos cada renglón de la Regla 60 como una cadena de 1's y 0's, según el estado de cada celda. Interpretemos dicha cadena como una expresión en base 4, y procedamos a convertirla a base 4. Posteriormente, convirtamos el número resultante a base 2. Interpretando dicho número como una cadena de 1's y 0's, el resultado es la sucesión de estados de cada renglón de la Regla 90.

Ejemplo 2.10.

Número de renglón	*Cadena de símbolos*	*Número en base 4*	*Número en base 2*
0	*1*	*1*	*1*
1	*11*	*5*	*101*
2	*101*	*17*	*10001*
3	*1111*	*85*	*1010101*
4	*10001*	*257*	*100000001*
5	*110011*	*1285*	*10100000101*
6	*1010101*	*4369*	*1000100010001*
7	*11111111*	*21845*	*101010101010101*

Formalmente, hay que demostrar que

$$\Big(\sum_{i=0}^{n}\Big[\binom{n}{i}(\text{mód } 2)\Big]4^i\Big)_2 = c_n$$

i.e.

$$\Big(\sum_{i=0}^{n}\Big[\binom{n}{i}(\text{mód } 2)\Big]2^{2i}\Big)_2 = c_n$$

donde c_n denota la n-ésima configuración de la Regla 90.

En esta última forma podemos observar que el interpretarlo como un número en base 4 lo único que hace es añadir un 0 entre cada símbolo que aparece en la cadena de símbolos de cada renglón.

Es decir, claramente,

$$\Big(\sum_{i=0}^{n}\Big[\binom{n}{i}(\text{mód } 2)\Big]2^{2i}\Big)_2 = \sum_{i=0}^{n}\Big[\binom{n}{i}(\text{mód } 2)\Big]2^{2i}.$$

Así, basta con ver que

$$\sum_{i=0}^{n}\Big[\binom{n}{i}(\text{mód } 2)\Big]2^{2i} = c_n$$

Teorema 2.12. *Sea*

$$B : S^{\mathbb{Z}} \to \mathbb{N}$$
$$c_n \mapsto \sum_k s_k 2^k$$

la función que a cada estado c_n le asocia el natural que resulta de interpretar los estados como coeficientes de una expansión binaria. Entonces

$$B(n) = \sum_{i=0}^{n}\Big[\binom{n}{i}(\textit{mód } 2)\Big]2^{2i}, \quad \forall n \in \mathbb{N}$$

.

Demostración. Dado que $f(x, y, z) = x + z$ sobre $S = \mathbb{Z}_2$, el dipolinomio de evolución de la regla 90 es $\mathbb{T}(x) = x + x^{-1}$. Así, por la ecuación 2.1 la evolución al tiempo t está dada por

$$\mathbb{T}(x)^t 1 = (x + x^{-1})^t$$

y dado que

$$(x + x^{-1})^t x^t = (x^2 + 1)^t = \sum_{i=0}^{t} \left[\binom{t}{i} (\text{mód } 2) \right] x^{2i}$$

entonces

$$(x + x^{-1})^t = \sum_{i=0}^{t} \left[\binom{t}{i} (\text{mód } 2) \right] x^{2i-t}.$$

Luego, dado que la función $B(n)$ cuenta de derecha a izquierda las celdas con estados no nulos, tenemos que el coeficiente de x^{2t-t} en $\mathbb{T}(x)^t$ es el mismo que el coeficiente de x^0 en $B(n)$, es decir, si

$$B(n) = \sum_{i=0}^{n} b_i 2^i \quad , \quad \mathbb{T}(x) = \sum_{i=0}^{t} a_i x^{2i-t}$$

entonces

$$b_i = a_{i+t}$$

i.e. el coeficiente de x^m en $B(n)$ es el mismo que el coeficiente de x^{2m-t} en $\mathbb{T}(x)$,
de forma que

$$B(n) = \sum_{i=0}^{n} \left[\binom{t}{i} (\text{mód } 2) \right] 2^{2i}$$

□

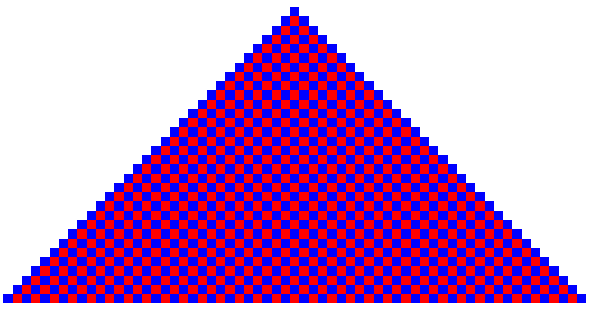


Figura 2.21: Regla 90 como dos autómatas que no interactúan entre sí.

Otra relación interesante entre ambos autómatas es que el renglón n de la Regla 90

(empezando a contar renglones desde el 0 i.e. el renglón 0 consta de un 1 en la punta del triángulo) es el renglón $2n$ de la Regla 60 para toda $n \geq 1$. Es decir,

Teorema 2.13. *Sea $n \geq 1, n \in \mathbb{N}$. Entonces*

$$\sum_{i=0}^{n} \big[\binom{n}{i} (mód\ 2)\big] 2^{2i} = \sum_{i=0}^{n} \big[\binom{2n}{i} (mód\ 2)\big] 2^{i}$$

Demostración. Por el teorema 2.6 (de Lucas) sabemos que, dado que $2n$ es par, cuando i es impar se tiene que $\binom{2n}{i} (\text{mód } 2) = 0$, de forma que

$$\sum_{i=0}^{n} \big[\binom{2n}{i} (\text{mód } 2)\big] 2^{i} = \sum_{0 \leq 2i \leq n} \big[\binom{2n}{2i} (\text{mód } 2)\big] 2^{2i}$$

Ahora bien, si

$$n = n_0 2^0 + n_1 2^1 + \cdots + n_k 2^k$$

y

$$i = i_0 2^0 + i_1 2^1 + \cdots + i_k 2^k$$

entonces

$$2n = 0 * 2^0 + n_0 2^1 + \cdots + n_{k-1} 2^k + n_k 2^{k+1}$$

y

$$i = 0 * 2^0 + i_0 2^1 + \cdots + i_{k-1} 2^k + i_k 2^{k+1}$$

de forma que por el teorema 2.6 (de Lucas) tenemos que

$$\binom{n}{i} \equiv \binom{2n}{2i} (\text{mód } 2)$$

lo cual demuestra el resultado. □

Corolario 2.7. *El 3 no aparece en la factorización en números de Fermat de ningún renglón de la Regla 90, es decir si*

$$\sum_{i=0}^{n} \big[\binom{n}{i} (mód\ 2)\big] 2^{2i} = \prod_{k=0}^{m} F_k$$

con F_k números de Fermat, entonces $F_k \neq 3$ para toda $k \in \{0, \ldots, m\}$.

Demostración. Por el teorema anterior y la ecuación 2.6, dado que $2n \equiv 0 (\text{mód } 2)$ sabemos que $3 = F_0$ no aperece en $\prod_{j:2n_j=1} F_j$. □

2.4.2. Reducibilidad computacional

Por el teorema 2.12, sabemos que podemos obtener el renglón t-ésimo de la Regla 90 obteniendo el renglón t-ésimo de la Regla 60 e intercalando ceros entre cada dígito. Así, dado que obtener el renglón t-ésimo de la Regla 60 es, a lo más, de orden $O(tlog_2 t)$ e intercalar ceros implica recorrer un número de longitud $log_2(t)$, entonces obtener el renglón

de la regla 90 puede ser calculado mediante un proceso de orden $O(tlog_2t + log_2(t)) = \mathcal{O}(tlog_2t)$, lo cual implica que la Regla 90 también es **computacionalmente reducible**.

2.4.3. Pseudocódigos

Al igual que en los algoritmos anteriores, $n \times n$, $M \in M_{n,n}(\{0,1\})$ representará el espacio de configuraciones del autómata y comenzaremos con programar la función $f : S^m \to S$ que a cada celda le asocia un nuevo estado con base en el estado de las celdas de su vecindad al paso anterior.

Algoritmo 13: Función $f_{90} : S^2 \to S$

Entrada: $M \in M_{n,n}(\{0,1\}), i, j \in Z^+$
Salida: $x \in \{0,1\}$

$n \leftarrow$ número de columnas de la matriz M

Función $f_{90}(M, i, j)$
  if $j = 1$ **then**
    $M_{i,j} \equiv M_{i-1,n} + M_{i-1,j+1}(m\acute{o}d\ 2)$
  else
    if $j = n$ **then**
      $M_{i,j} \equiv M_{i-1,j-1} + M_{i-1,1}(m\acute{o}d\ 2)$
    else
      $M_{i,j} \equiv M_{i-1,j-1} + M_{i-1,j+1}(m\acute{o}d\ 2)$
    end
  end
  return $M_{i,j}$
end

Capítulo 3

Autómatas celulares dinámicos

> Podrán morir las personas pero jamás sus ideas.
>
> *Ernesto "Ché" Guevara*

Entenderemos por autómata celular dinámico un autómata celular cuya regla de transición global G cambia en función del tiempo (G_t). Esto implica que tanto la regla local f como la vecindad del autómata también cambian en función del tiempo (f_t, N_t), de forma que únicamente la dimensión y el conjunto de estados permanece inalterado durante la evolución del autómata.

Definición 3.1. *Un **autómata celular dinámico** es una tupla $A = (d, S, N_t, f_t)$ donde N_t y f_t son funciones del tiempo* i.e. $N_t : \mathbb{Z}^+ \longrightarrow (\mathbb{Z}^d)^m$, $f_t : \mathbb{N} \longrightarrow S^{S^m}$.

Observación 3.1. *Todo autómata celular es un autómata celular dinámico con $N_t = N$, $f_t = f$ funciones constantes.*

De forma análoga a la definición de automátas celulares elementales aditivos tenemos la siguiente definición.

Definición 3.2. *Un **autómata celular dinámico aditivo** es un autómata celular dinámico tal que para toda $n \in \mathbb{Z}^+$ existen $k_1, k_2 \in \mathbb{Z}$ tales que $f_t(n) = \sum_{j=k_1}^{k_2} \alpha_j a_j^t$ para algunos $\alpha_i \in S$.*

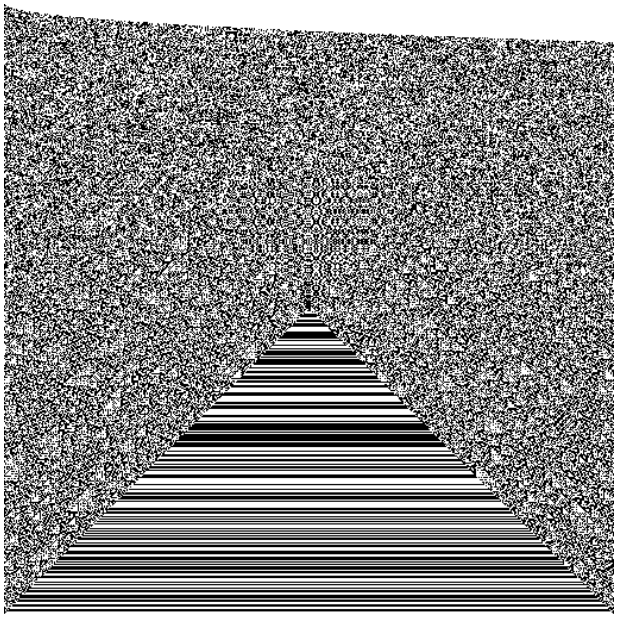
Ya en 2002, Stephen Wolfram [18] consideró distintos tipos de sistemas (autómatas celulares móviles, máquinas de Turing, sistemas de sustitución, sistemas de etiquetas, etc.) y observó que el hecho de presentar comportamiento complejo a partir de condiciones inciales sencillas y reglas de evolución sencillas no dependía de la estructura particular de los autómatas celulares.

> *For it is certainly true that cellular automata have many special features. All their elements, for example, are always arranged in a rigid array, and are always updated in parallel at each step. And one might think that features like these could be crucial in making it possible to produce complex behavior from simple underlying rules.*

Indeed, I specifically chose the sequence of systems in this chapter to see what would happen when each of the various special features of cellular automata were taken away. And the remarkable conclusion is that in the end none of these features actually matter much at all. For every single type of system in this chapter has ultimately proved capable of producing very much the same kind of complexity that we saw in cellular automata.

Es en este contexto en el que tiene sentido considerar a los autómatas celulares dinámicos.

Ejemplo 3.1. $A = (d, S, N_t, f_t)$, *con* $d = 1, S = \mathbb{Z}_2, N_t = (-t, -t+1, \ldots, 0, \ldots, t-1, t), f_t = \sum_{j=-t}^{t} a_{i+j}^t$.



Ejemplo 3.2. $A = (d, S, N_t, f_t)$ *con* $d = 1, S = \mathbb{Z}_2, N_t = (-t, 0, t), f_t = a^t_{i-t} + a^t_i + a^t_{i+t}$

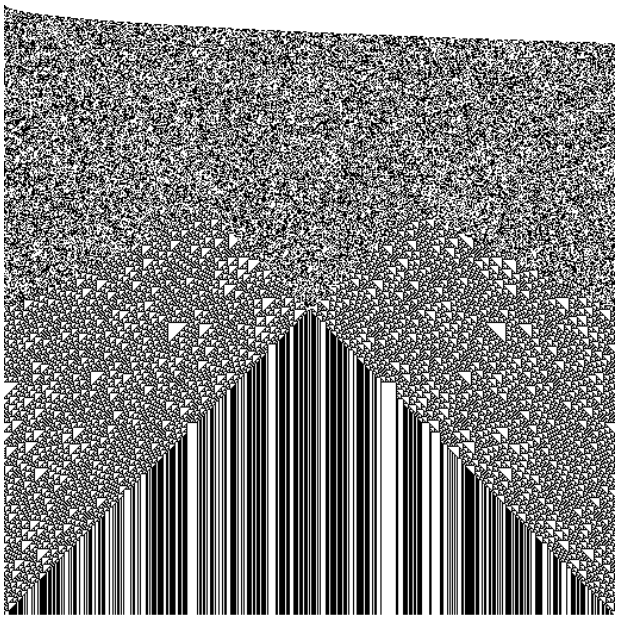


Figura 3.1: Autómata celular dinámico A. Presenta comportamiento semiregular en la frontera del triángulo formado.

Ejemplo 3.3. $A = (d, S, N_t, f_t)$ *con* $d = 1, S = \mathbb{Z}_2$

$$N_t(t) = \begin{cases} (0,1) & t = 2n \\ (-1,0) & t = 2n+1 \end{cases}$$

$$f_t(t) = \begin{cases} a^t_i + a^t_{i+1} & t = 2n \\ a^t_{i-1} + a^t_i & t = 2n+1 \end{cases}$$

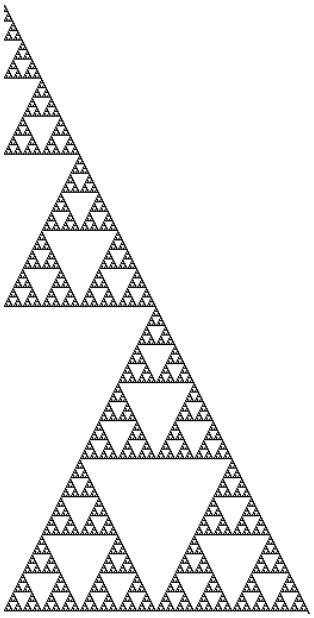
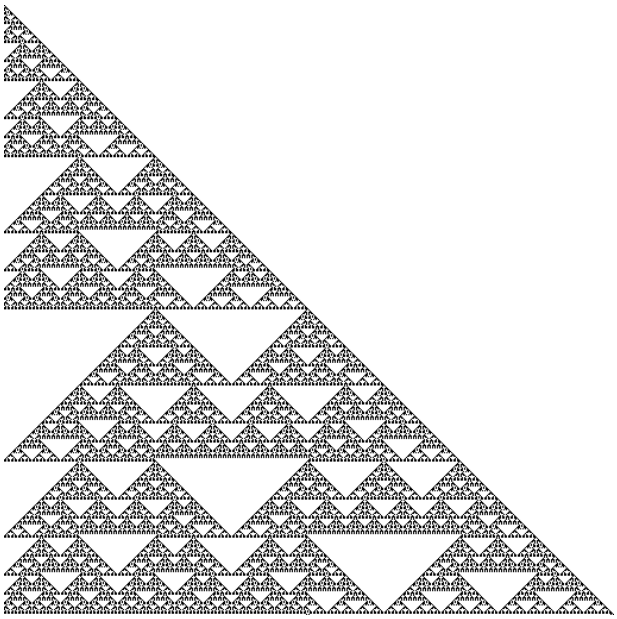


Figura 3.2: Autómata celular dinámico A. Más adelante explicaremos el patrón fractal emergente.

Ejemplo 3.4. $A = (d, S, N_t, f_t)$ *con* $d = 1, S = \mathbb{Z}_2$

$$N_t(t) = \begin{cases} (-1, 0, 1) & t = 2n \\ (-1, 1) & t = 2n + 1 \end{cases}$$

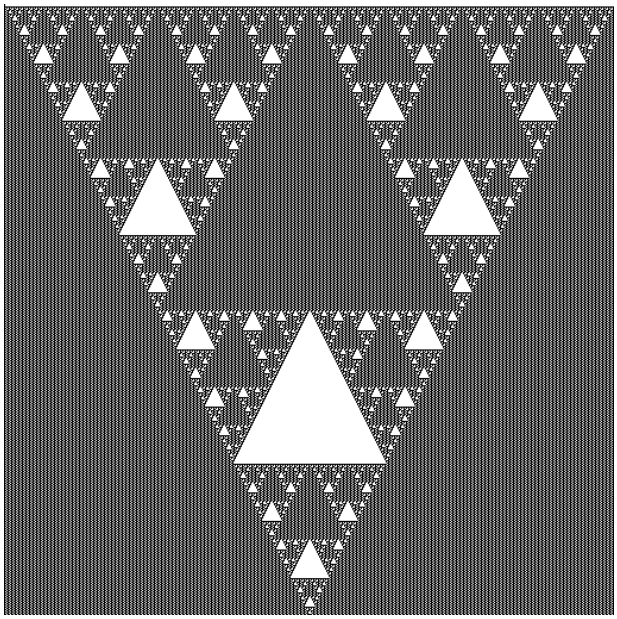
$$f_t(t) = \begin{cases} a_{i-1}^t + a_i^t + a_{i+1}^t & t = 2n \\ a_{i-1}^t + a_{i+1}^t & t = 2n + 1 \end{cases}$$



Ejemplo 3.5. $A = (d, S, N_t, f_t)$ *con* $d = 1, S = \mathbb{Z}_2$

$$N_t(t) = \begin{cases} (-2^9, \ldots, 0) & t = 2n \\ (0, \ldots, 2^9) & t = 2n+1 \end{cases}$$

$$f_t(t) = \begin{cases} \sum_{j=-2^9}^{0} a_j^t & t = 2n \\ \sum_{j=0}^{2^9} a_j^t & t = 2n+1 \end{cases}$$



La definición 3.2 sugiere analizar a los autómatas celulares dinámicos aditivos con las herramientas que usamos para analizar a los autómatas celulares elementales aditivos.

En efecto, si $f_1, f_2, \ldots$ son las reglas de evolución del autómata dinámico A_t, y $A^1(x), A^2(x), \ldots$ son los polinomios característicos de A_t a cada tiempo t y A_t es dinámico elemental, entonces los dipolinomios de evolución $\mathbb{T}_1(x), \mathbb{T}_2(x), \ldots$ satisfacerán que

$$\mathbb{T}_{i+1}(x)A^i(x) = A^{i+1}(x).$$

Así, el polinomio característico de A_t al tiempo t será

$$A^0(x)\prod_{i=1}^{t}\mathbb{T}_i.$$

En particular, si $A^0(x) = 1$, la función

$$\prod_{i=1}^{t}\mathbb{T}_i$$

será la función generadora del renglón t-ésimo del autómata dinámico elemental A_t.

Ejemplo 3.6. *En el ejemplo 3.1, dado que* $f_t = \sum_{j=-t}^{t} a_{i+j}^t$ *entonces para cada* $i \in \mathbb{Z}^+$ *se tiene que* $\mathbb{T}_i(x) = \sum_{j=-t}^{t} x^t$*, de forma que la función generadora del renglón* $t+1$*-ésimo es*

$$\prod_{i=1}^{t} \mathbb{T}_i(x) = (x^{-1} + 1 + x)(x^{-2} + x^{-1} + 1 + x + x^2) \cdots (x^{-t} + \cdots + 1 + \cdots + x^t).$$

Ejemplo 3.7. *En el ejemplo 3.3, dado que* $f_t(t) = f_t(t) = \begin{cases} a_i^t + a_{i+1}^t & t = 2n \\ a_{i-1}^t + a_i^t & t = 2n+1 \end{cases}$

Se tiene que

$$\mathbb{T}_i(x) = \begin{cases} x^{-1} + 1 & i = 2n \\ 1 + x & i = 2n+1 \end{cases}$$

de forma que la función generadora del renglón $t+1$*-ésimo es*

$$\prod_{i=1}^{t} \mathbb{T}_i(x) = (1 + x)(x^{-1} + 1) \cdots (1 + x^{-1^{t+1}}).$$

Notemos que $x\mathbb{T}(x)$ *es simplemente recorrer las celdas cuyos coeficientes son codificados por* $\mathbb{T}(x)$ *una celda hacia la derecha. Luego, dado que* $x\mathbb{T}_{2n}(x) = \mathbb{T}_{2n+1}(x)$ *para toda* $n \in \mathbb{Z}^+$ *se tiene entonces que*

$$\prod_{i=1}^{t} \mathbb{T}_i(x) = x^{-j}(1 + x)^t$$

para alguna $1 \leq 1 \leq t$*, y dado que*

$$(1 + x)^t = \sum_{i=0}^{t} \binom{t}{i} x^i$$

tomando $\binom{t}{i}$ *sobre* $S = \mathbb{Z}_2$ *explica por qué en dicho autómata dinámico aparece el triángulo de Pascal a diferentes escalas.*

3.1. Jugando al casino con otros dados

En la sección 2.3.3.2 introdujimos una forma de construir un autómata que codificara las particiones de un entero n en k factores de una lista finita (a_m). Usando autómatas celulares dinámicos aditivos podemos considerar el problema de construir un autómata que codifique las particiones de un entero $n \in \mathbb{Z}^+$ en factores de k listas finitas distintas $(a_{m_1}^1), (a_{m_2}^2), \ldots, (a_{m_k}^k)$. Podemos interpretar dicho problema como el número de formas de obtener el número n lanzando k dados distintos tal que en las caras del primero de ellos

aparecen $a^1_1, \ldots, a^1_{m_1}$, en las caras del segundo de ellos aparecen $a^2_1, \ldots, a^2_{m_2}$, etcétera.

Para cada lista finita $(a^j_{m_j})$, sea

$$p_j(x) = \sum_{i=1}^{m_j} x^{a^j_i}$$

con $j = 1, \ldots, k$. Así, en virtud del teorema 2.10, las particiones de un entero $n \in \mathbb{Z}^+$ en factores de k listas finitas distintas $(a^1_{m_1}), (a^2_{m_2}), \ldots, (a^k_{m_k})$ serían exactamente $[x^n] \prod_{j=1}^k p_j(x)$. Basta entonces con construir un autómata celular dinámico tal que

$$\begin{aligned} \mathbb{T}_i(x) &= p_i(x) \\ A^0(x) &= 1 \end{aligned}$$

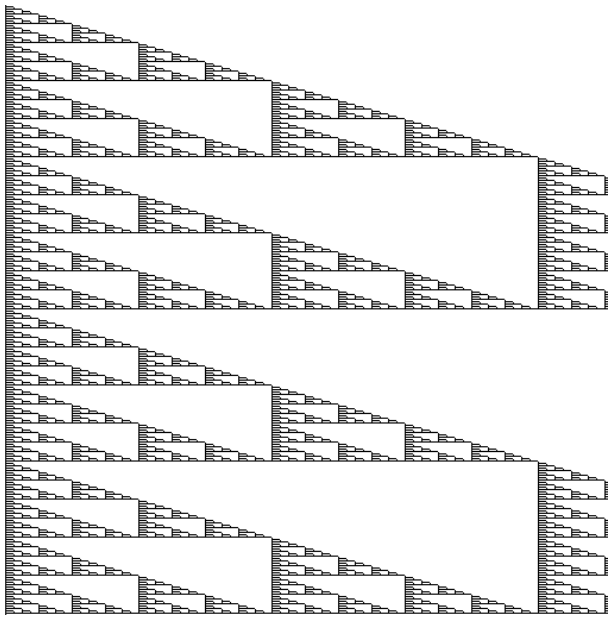
para cada $i = 1, \ldots, k$. Notemos que el polinomio característico del renglón j-ésimo, con $j = ck + r$ con $c \in \mathbb{N}, 0 \leq r < k$ sería

$$\prod_{i=1}^{k} (\mathbb{T}_i(x))^c \prod_{i=1}^{r} T_i(x)$$

La interpretación de la evolución de dicho autómata sería la siguiente: tiramos el primer dado, cuyos coeficientes estarán codificados en $T_1(x)$. Despúes tiramos el segundo dado, de forma que en el segundo renglón, cuyo polinomio característico es $T_1(x)T_2(x)$, estarán codificadas las particiones de n con factores en $(a^1_{m_1}), (a^2_{m_2})$. Tiramos el tercer dado, de forma que en el tercer renglón, cuyo polinomio característico es $T_1(x)T_2(x)T_3(x)$, estarán codificadas las particiones de n con factores en $(a^1_{m_1}), (a^2_{m_2}), (a^3_{m_3})$, y así sucesivamente. Al tirar el k-ésimo (y último) dado tendremos que en el k-ésimo renglón, cuyo polinomio característico es $\prod_{i=1}^k \mathbb{T}_i(x)$, estarán codificadas las particiones de n con factores en $(a^1_{m_1}), (a^2_{m_2}), (a^3_{m_3}), \ldots, (a^k_{m_k})$. Así, en el renglón $j = ck + r$ con $c \in \mathbb{N}, 0 \leq r < k$ estarán codificadas las formas de obtener el número n habiendo tirado c veces los k dados, y una vez más los primeros r dados. En particular, si $j = ck$, en el renglón j-ésimo estarán codificadas las formas de obtener el número n habiendo tirado c veces los k dados.

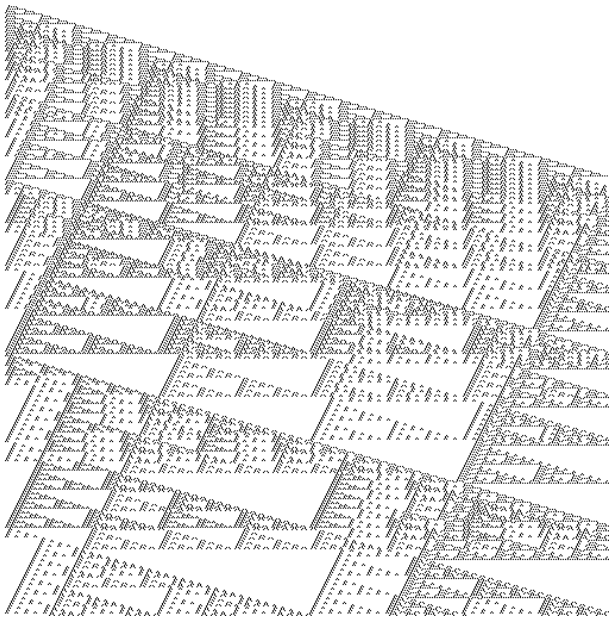
Ejemplo 3.8. *Supongamos que tenemos* 2 *dados. El primero de ellos tiene caras* $(0, 1, 2, 3, 4, 5, 6)$ *y el segundo* $(0, 1)$. *Entonces* $\mathbb{T}_1(x) = \sum_{j=0}^6 x^j$, $\mathbb{T}_2(x) = 1 + x$, *lo cual implica que*

$$a^{t+1}_i = \begin{cases} \sum_{j=0}^6 a^t_{i-j} & t = 2n \\ a^t_{i-1} + a^t_i & t = 2n+1 \end{cases}$$



Ejemplo 3.9. *Supongamos que tenemos* 2 *dados. El primero de ellos tiene caras* $(0, 2, 4, 6)$ *y el segundo* $(-1, 1)$*. Entonces* $\mathbb{T}_1(x) = 1 + x^2 + x^4 + x^6$*,* $\mathbb{T}_2(x) = \frac{1}{x} + x$*, lo cual implica que*

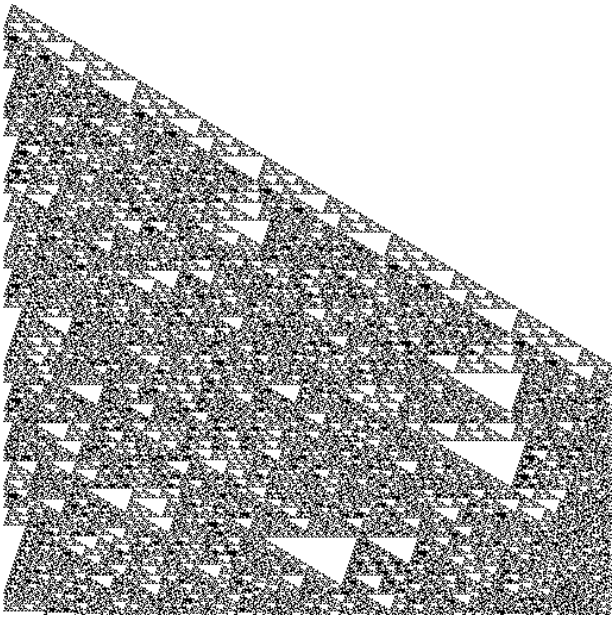
$$a_i^{t+1} = \begin{cases} a_{i-6}^t + a_{i-4}^t + a_{i-2}^t + a_i^t & t = 2n \\ a_{i-1}^t + a_{i+1}^t & t = 2n+1 \end{cases}$$



Ejemplo 3.10. *Supongamos que tenemos* 3 *dados. El primero de ellos tiene caras* $(1, 2, 3)$*, el segundo* $(-2, -1, 0)$ *y el tercero* $(0, 1, 2)$*. Entonces* $\mathbb{T}_1(x) = x^1 + x^2 + x^3$*,* $\mathbb{T}_2(x) = \frac{1}{x^2} + \frac{1}{x} + 1$

y $\mathbb{T}_3(x) = 1 + x + x^2$ *lo cual implica que*

$$a_i^{t+1} = \begin{cases} a_{i-3}^t + a_{i-2}^t + a_{i-1}^t & t = 3n \\ a_i^t + a_{i+1}^t + a_{i+2}^t & t = 3n+1 \\ a_{i-2}^t + a_{i-1}^t + a_i^t & t = 3n+2 \end{cases}$$



Vale la pena hacernos la siguiente pregunta: ¿existirán conjuntos de dados distintos tales que generen el mismo espacio de posibilidades? Es decir, ¿existirán polinomios distintos tales que sus funciones generadoras arrojen los mismos coeficientes?

No es díficil convencerse de que lo anterior es cierto, al menos en casos triviales. Por ejemplo, pensemos en un dado de seis caras con valores en $(1, 2, 3, 4, 5, 6)$, es decir, un dado convencional. Como ya sabemos, su función generadora sería $\mathbb{T}(x) = x^1 + x^2 + x^3 + x^4 + x^5 + x^6$, de forma que las particiones de n en t factores de $(1, 2, 3, 4, 5, 6)$ estarían dadas por $[x^n]\mathbb{T}(x)^t$ para toda $n \in \mathbb{N}, t \in \mathbb{Z}^+$. Ahora bien, es claro que $\mathbb{T}(x) = x(1 + x^1 + x^2 + x^3 + x^4 + x^5)$, de forma que resulta lo mismo -en términos de particiones- aventar un dado de seis caras con valores en $(1, 2, 3, 4, 5, 6)$, que dos dados: uno de ellos con una única cara con valor 1, y el otro con 6 caras con valores en $(0, 1, 2, 3, 4, 5)$.

Lo que resulta interesante es que, aunque algebraicamente es trivial, en términos de autómatas celulares no lo es tanto, pues el primer escenario corresponde a un autómata celular elemental con dipolinomio de evolución $x^1 + x^2 + x^3 + x^4 + x^5 + x^6$, mientras que el segundo corresponde a un autómata celular dinámico aditivo con dipolinomios de evolución $\mathbb{T}_1(x) = x, \mathbb{T}_2(x) = 1 + x^1 + x^2 + x^3 + x^4 + x^5$. El hecho de que $(\mathbb{T}(x))^t = (\mathbb{T}_1(x)\mathbb{T}_2(x))^t$ para toda $t \in \mathbb{N}$ nos dice que los renglones pares del autómata celular dinámico aditivo coincidirán con todos los renglones del autómata celular elemental. Los renglones impares

del autómata celular dinámico aditivo pueden ser interpretados como haber aventado un número par de veces ambos dados, y una vez más uno de ellos.

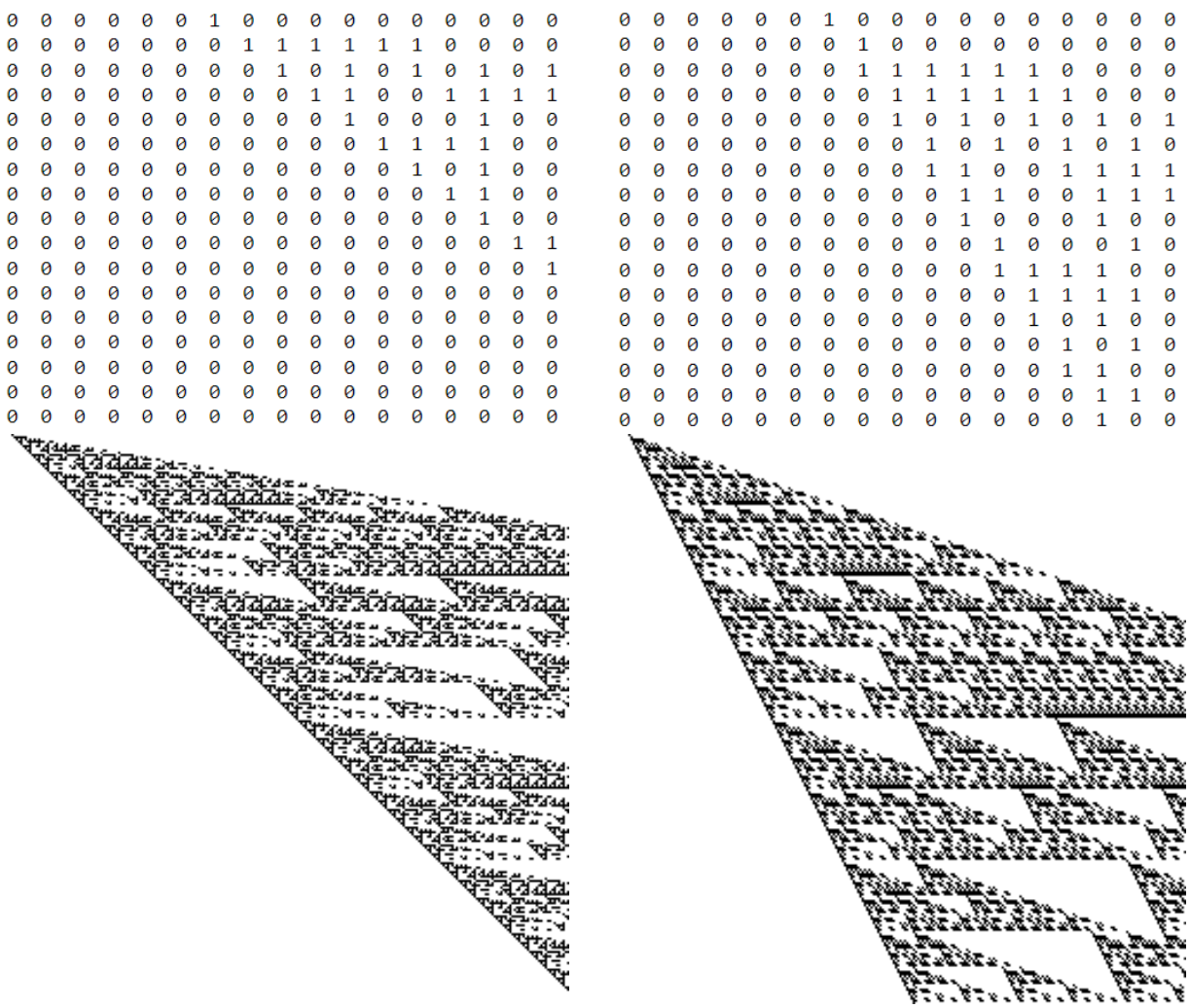


Figura 3.3: El n-ésimo renglón del autómata celular elemental (izquierda) es el $2n$-ésimo renglón del autómata celular dinámico aditivo (derecha). Las últimas dos imágenes muestran 2^8 pasos de evolución.

Esta interpretación motiva buscar factorizaciones no triviales de $x^1+x^2+x^3+x^4+x^5+x^6$ ya que, de encontrarlas, encontraríamos una relación entre autómatas celulares elementales y autómatas celulares dinámicos aditivos: precisamente que los renglones de uno (que son múltiplos de algún $k \in \mathbb{Z}^+$) coincidirán con los renglones del otro. Podemos entonces pensar a los autómatas celulares dinámicos aditivos como una representación distinta (por ejemplo, con dados distintos) del espacio generado por lo autómatas celulares elementales (con dados iguales) en virtud del siguiente

Teorema 3.1. *Sean $A = (d, S, N_t, f_t)$ un autómata celular dinámico aditivo con k dipolinomios de evolución distintos $\mathbb{T}_1, \ldots, \mathbb{T}_k$. Entonces $\prod_{i=1}^{k} \mathbb{T}_i := \mathbb{T}$ es el dipolinomio de*

evolución de un automata celular elemental $A = (d, S, N, f)$ donde $N = N_{máxt}$ donde $N_{máxt}$ es el vector de mayor longitud de $N_1, \ldots, N_k$.

Demostración. Sean $A = (d, S, N_t, f_t)$ y $\mathbb{T}_1, \ldots, \mathbb{T}_k$ como en el enunciado. Entonces para cada $j = 1, \ldots, k$, $\mathbb{T}_j = \sum_{i \in I_j} \alpha_i x^i$ con I_j un conjunto finito de índices, $\alpha_i \in S$. Así

$$[x^n] \prod_{j=1}^{k} T_j = \sum_{\substack{i \in I_1 \times \cdots \times I_k \\ i_1 + \cdots + i_k = n}} 1$$

de forma que $\mathbb{T} := \prod_{j=1}^{k} T_j$ en efecto es un dipolinomio cuya función asociada f tiene radio de vecindad $N = N_{\text{máxt}}$ donde $N_{\text{máxt}}$ es el vector de mayor longitud de $N_1, \ldots, N_k$. □

Corolario 3.1. *El renglón n-ésimo del autómata celular elemental cuyo dipolinomio de evolución es $\mathbb{T} = \prod_{j=1}^{k} \mathbb{T}_j$ es el renglón nk-ésimo del autómata celular dinámico aditivo con dipolinomios de evolución $\mathbb{T}_1, \ldots, \mathbb{T}_k$.*

Demostración.

$$\mathbb{T}^t = \Big(\prod_{j=1}^{k} \mathbb{T}_j \Big)^t = \prod_{j=1}^{k} (\mathbb{T}_j)^t$$

Es decir, aventar t veces el dado codificado por $\mathbb{T}$ es lo mismo que aventar t veces los k dados codificados, respectivamente, por $\mathbb{T}_1, \ldots, \mathbb{T}_k$. □

Ejemplo 3.11. *De acuerdo al procedimiento expuesto en [5], consideramos lo siguiente:*

$$\begin{aligned} 1 + x^1 + x^2 + x^3 + x^4 + x^5 + x^6 = \\ x(1 + x^1 + x^2 + x^3 + x^4 + x^5) = \\ x\Big(\frac{x^6 - 1}{x - 1}\Big) = x\Big(\frac{1 - x^6}{1 - x}\Big) = x\Big(\frac{(1 - x^3)(1 + x^3)}{1 - x}\Big) = \\ x\Big(\frac{1 - x^3}{1 - x}\Big)\Big(1 + x^3\Big) = x(1 + x + x^2)(1 + x^3) = \\ x(1 + x + x^2)(1 + x)(1 - x + x^2) \end{aligned}$$

de forma que aventar el dado con caras en $(1, 2, 3, 4, 5, 6)$ es equivalente a aventar los dados con caras en $(1), (0, 1, 2), (0, 1), (0, 1, 2)$ con los signos correspondientes. Y no sólo a ellos, sino a cualquier reagrupación de ellos.

No es de sorprender que los autómatas celulares dinámicos tales que el producto de sus dipolinomios de evolución resulte en el dipolinomio de evolución de un autómata celular elemental sean muy similares a éste. A continuación se muestran los autómatas celulares dinámicos aditivos cuyos dipolinomios de evolución son, en orden de aparición:

$\mathbb{T}(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6$	$\mathbb{T}_1(x) = x$ $\mathbb{T}_2(x) = 1 + x + x^2 + x^3 + x^4 + x^5$
$\mathbb{T}_1(x) = x^2$ $\mathbb{T}_2(x) = x^{-2} + 1 + x + x^2 + x^3 + x^4$	$\mathbb{T}_1(x) = x - x^2 + x^3$ $\mathbb{T}_2(x) = 1 + 2x + 2x^2 + x^3$

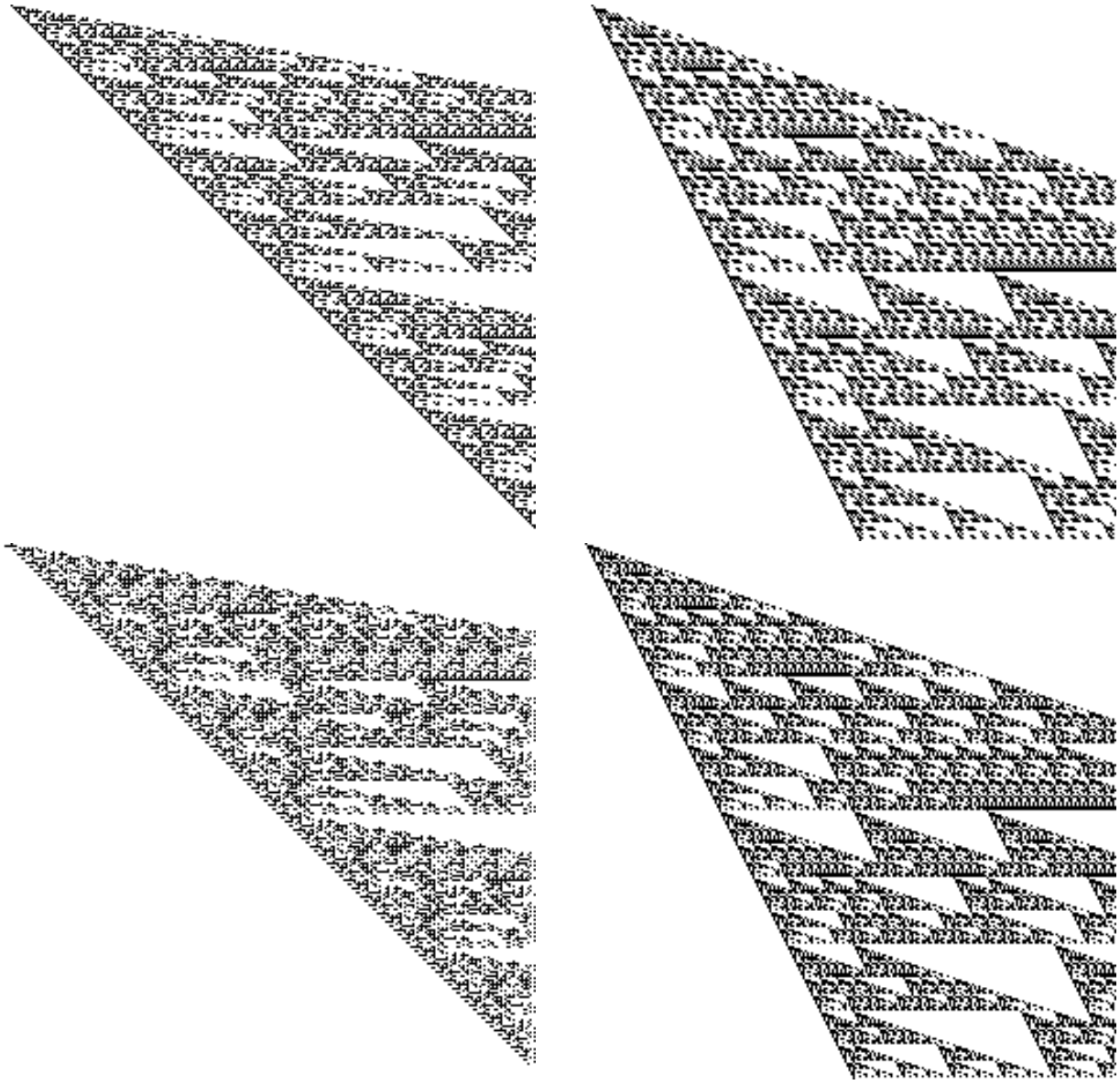


Figura 3.4: Autómatas celulares resultantes en el orden de aparición de la tabla mostrada.

Un ejemplo históricamente relevante que involucra encontrar los coeficientes de un producto de polinomios, y que por lo tanto podemos abordarlo con la teoría de autómatas celulares dinámicos, fue considerado por Euler en relación a los números pentagonales.

3.2. El Teorema de los Numéros Pentagonales de Euler

En 1775 Euler demostró el siguiente

Teorema 3.2 (Números Pentagonales de Euler)**.**

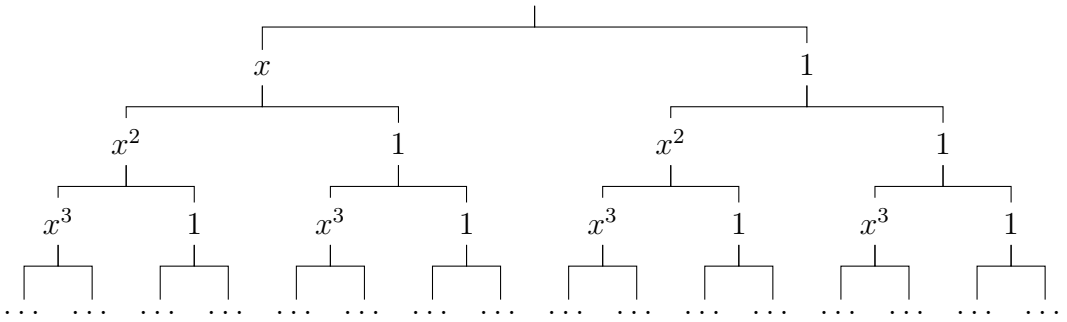
$$\prod_{n=1}^{\infty}(1-x^n)=\sum_{k=-\infty}^{\infty}(-1)^k x^{\frac{k(3k-1)}{2}}$$

es decir, que en el producto $(1-x)(1-x^2)(1-x^3)\cdots = 1-x-x^2+x^5+x^7-x^{12}-\ldots$ los exponentes con coeficientes no nulos son de la forma $\frac{k(3k-1)}{2}$ para $k=1,-1,2,-2,3,-3,\ldots$ *i.e.* números pentagonales (generalizados).

Antes de pasar a la demostración, veamos poco a poco de dónde viene este maravilloso resultado.

Pensemos primero en el coeficiente n-ésimo del producto $\prod_{j=1}^{\infty}(1+x^j)$. Podemos pensar en escoger un término de cada binomio $(1+x^j)$ dentro de $(1+x)(1+x^2)(1+x^3)(1+x^4)\ldots$ tales que al multiplicarlos nos formen el exponente n, y después sumar el número de formas en el que podemos hacer eso. Es claro que si $j>n>0$, debemos de escoger siempre al 1, pues de otra forma nunca conseguiremos que al multiplicarlos sumen n.

Por ejemplo, para $n=5$, puedo escoger, dentro de cada sumando, al 1, salvo en el binomio $(1+x^5)$, en el escogeré a x^5. O en escoger dentro de cada sumando al 1, salvo en el binomio $(1+x^1)$ y $(1+x^4)$, en los que escogeré a x^1 y x^4 sucesivamente.



No es difícil darse cuenta de que dichas "trayectorias" corresponden a las particiones de 5 con números en $\{1,2,3,4\}$ sin repetición ni distinción de orden. En efecto,

$$\begin{aligned}5&=5+0\\&=4+1\\&=3+2\end{aligned}$$

donde cada forma de representar al 5 corresponde con una única elección de términos dentro de cada binomio

$$\begin{aligned} x^5 &= x^5x^0 \\ &= x^4x^1 \\ &= x^3x^2 \end{aligned}$$

Así,

$$[x^n]\prod_{j=1}^{\infty}(1+x^j) = k$$

donde k es el número de particiones de n sin repeticiones de sumandos (no contamos particiones de la forma $a + a + \dots$) ni distinciones entre ellos ($a + b = b + a$). A este tipo de particiones las llamaremos particiones propias.

Ahora bien, fijémonos en el producto

$$\prod_{j=1}^{\infty}(1+zx^j)$$

con $z \in \mathbb{Z}$. ¿Qué sucede con $[x^n]\prod_{j=1}^{\infty}(1+zx^j)$? Lo único que va a cambiar con respecto a $[x^n]\prod_{j=1}^{\infty}(1+x^j)$ es que cada vez que multiplico los términos distintos de 1 (cada vez que los escojo dentro de mi trayectoria), aumentará en 1 la potencia de z. Es decir, la potencia de z es un contador de cuántas veces multiplico términos distintos de 1.

Por ejemplo,

$$\begin{aligned} 5 &= 5 + 0,\ z^1 \\ &= 4 + 1,\ z^2 \\ &= 3 + 2,\ z^2 \end{aligned}$$

pues

$$\begin{aligned} x^5 &= x^5x^0,\ z^1 \\ &= x^4x^1,\ z^{1+1} \\ &= x^3x^2,\ z^{1+1} \end{aligned}$$

de forma que $[x^5]\prod_{j=1}^{\infty}(1+zx^j) = z + 2z^2$, lo cual indica que 5 es representable como una suma de factores con un sumando no nulo $(5+0)$ y dos sumas de factores con 2 sumandos no nulos $(4+1, 3+2)$.

En general, $Cz^kx^n, k \in \mathbb{Z}, n \in \mathbb{Z}^+$ indica de cuántas formas es posible representar a n

con k sumandos no nulos sin repeticiones ni distinciones de sumandos.

Ya estamos en posibilidades de entender mejor el producto $\prod_{j=1}^{\infty}(1-x^j)$, pues basta con sustituir $z=-1$ en $\prod_{j=1}^{\infty}(1+zx^j)$.

Observemos que $(-1)^k = 1$ si y sólo si k es par. Luego, por lo dicho anteriormente, $C(-1)^{2m}x^n, m \in \mathbb{Z}, n \in \mathbb{Z}^+$ indica de cuántas formas es posible representar a n con $2m$ sumandos no nulos sin repeticiones ni distinciones de sumandos, mientras que $C(-1)^{2m+1}x^n, m \in \mathbb{Z}, n \in \mathbb{Z}^+$ indica de cuántas formas es posible representar a n con $2m+1$ sumandos no nulos sin repeticiones ni distinciones de sumandos. Entonces $[x^n]\prod_{j=1}^{\infty}(1-x^j)$ indica la diferencia entre el número de formas de representar a n con una cantidad par de sumandos y una cantidad impar de sumandos, ambos sin repeticiones ni distinciones.

Es decir, si $p_p(n)$ es la función que a cada $n \in \mathbb{Z}^+$ le asocia el número de particiones propias suyas con una cantidad par de sumandos y $p_i(n)$ es la función que a cada $n \in \mathbb{Z}^+$ le asocia el número de particiones propias suyas con una cantidad impar de sumandos, entonces

$$[x^n]\prod_{j=1}^{\infty}(1-x^j) = p_p(n) - p_i(n).$$

Por ejemplo,

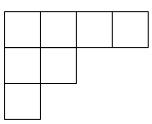
$$[x^5]\prod_{j=1}^{\infty}(1-x^j) = 1$$

pues $p_p(5) = 2$ y $p_i(5) = 1$ ya que $4+1$ y $3+2$ son las únicas particiones con un número par de sumandos de 5, mientras que $5+0$ es la única partición con un número impar de sumandos de 5 (sin repeticiones ni distinciones).

Ya tenemos entonces una muy buena idea de cómo son los coeficientes de $\prod_{j=1}^{\infty}(1-x^j)$. Cabe preguntarse, ¿qué valores pueden tomar? Para estudiar particiones de enteros, echaremos mano de una herramienta sumamente útil, que nos dará una elegantísima demostración del teorema de Euler: los diagramas de Young o de Ferrer.

3.2.1. Diagramas de Young y la demostración de Franklin

Una forma conveniente de representar particiones de enteros es mediante diagramas de Young. Un diagrama de Young es un diagrama de casillas en la que cada fila tiene una cantidad menor o igual de casillas que la fila anterior. Por ejemplo,

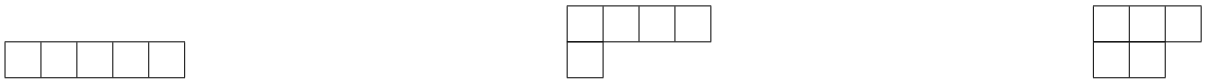


es un diagrama que denotaremos por $\lambda = (4, 2, 1)$.

Podemos entonces representar, de forma muy intuitiva, las particiones propias de un entero usando estos diagramas. Basta con representar cada número de la partición con casillas, diferenciando sumandos por renglones. Es decir, la partición $4 + 2 + 1$ estará representada por el diagrama $\lambda = (4, 2, 1)$.

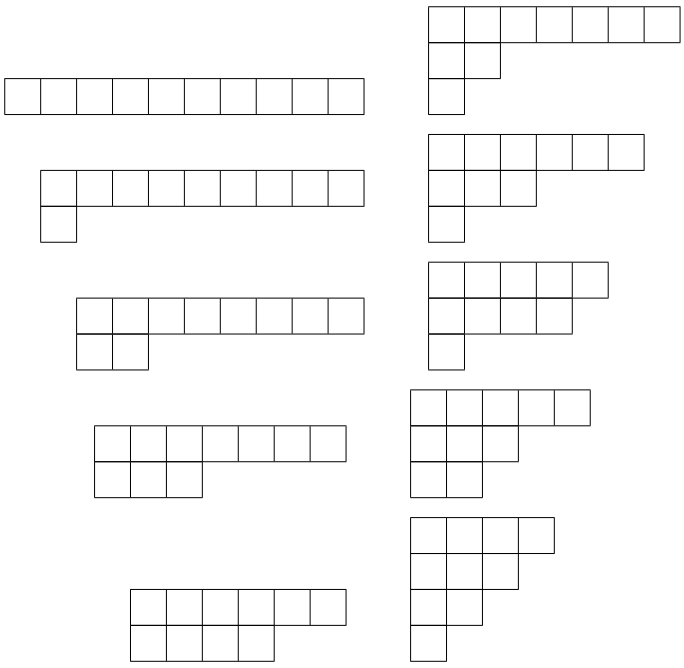
Ejemplo 3.12.

$$\begin{aligned} 5 &= 5 + 0 \\ &= 4 + 1 \\ &= 3 + 2 \end{aligned}$$



Ejemplo 3.13.

$$\begin{aligned} 10 &= 10 + 0 \\ &= 9 + 1 \\ &= 8 + 2 \\ &= 7 + 3 \\ &= 6 + 4 \end{aligned} \qquad \begin{aligned} &= 7 + 2 + 1 \\ &= 6 + 3 + 1 \\ &= 5 + 4 + 1 \\ &= 5 + 3 + 2 \\ &= 4 + 3 + 2 + 1 \end{aligned}$$



En 1881, 94 años después de la primera prueba de Euler, el matemático estadounidense Philip Franklin dió una demostración muy elegante del Teorema de los Números Pentago-

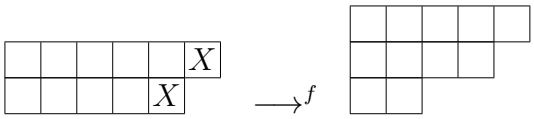
nales basada en diagramas de Young, misma que el matemático francés Hans Rademacher describió como "el primer logro importante de las matemáticas estadounidenses" [9]. La idea central de su prueba fue establecer una biyección entre los particiones con un número par de factores y las particiones con un número impar de factores, y notar que ésta falla únicamente cuando se trata de números pentagonales.

Procediendo como en [9], y para ilustrar la idea central de la prueba, veamos un ejemplo.

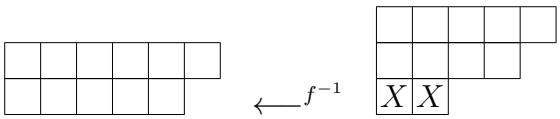
Ejemplo 3.14. *Consideremos el número* 11*, y pensemos en todas las particiones propias de* 11*. Hay exactamente* 12 *de ellas:* 6 *de ellas con un número par de sumandos, y* 6 *de ellas con un número impar de sumandos, así que las podemos aparear de la siguiente manera*

$$\begin{aligned} 10+1 &\iff 11 \\ 9+2 &\iff 8+2+1 \\ 8+3 &\iff 7+3+1 \\ 7+4 &\iff 6+4+1 \\ 6+5 &\iff 5+4+2 \\ 5+3+2+1 &\iff 6+3+2 \end{aligned}$$

¿Cómo está definida esta biyección? Pensemos en el diagrama de Young de alguna de las particiones, por ejemplo, la de $6+5$*. Sea* m *el número de casillas del último renglón, y sea* s *el número de casillas en la diagonal más larga que se puede trazar hasta la derecha (marcadas con X en e dibujo). Nos referiremos a dicha diagonal como la última diagonal del diagrama de Young. En este caso,* $m = 5$ *y* $s = 2$*. La genial idea de Franklin consiste en desplazar las casillas marcadas con X hacia abajo para formar un nuevo renglón.*



Notemos que en este caso podemos revertir el proceso, tomando los bloques del último renglón y colocándolos hasta la derecha para formar una nueva diagonal, concatenando un sólo bloque con cada uno de los primeros m *renglones.*

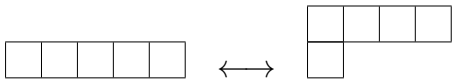


Dado que este proceso cambia la paridad del número de renglones (del número de sumandos de la partición) tenemos entonces una biyección que asocia una partición con un número par de sumandos (de renglones) con otra que tiene un número impar de ellos. Luego, su diferencia debe ser cero, por lo que $[x^{11}]\prod_{j=1}^{\infty}(1-x^j) = 0$*.*

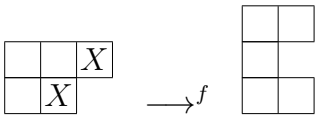
Cabe preguntarnos si esto siempre se puede llevar a cabo, en qué casos sí y en qué casos no. Veamos otro ejemplo

Ejemplo 3.15. *Consideremos las particiones propias del número* 5*. Hay exactamente* 3 *de ellas: dos con un número par de sumandos y una con un número impar de sumandos. Es claro entonces que el proceso descrito anteriormente no se puede llevar a cabo en todos los casos, por lo que en alguno de ellos no debo de poder transformar una partición en otra aplicando f seguido de f^{-1} o viceversa.*

Veamos:

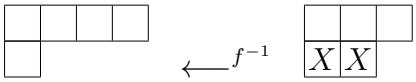


pero

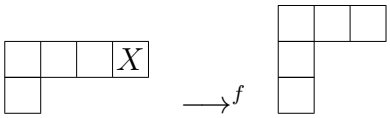


que no es una partición permitida pues un renglón del diagrama debe de tener menos elementos que su renglón anterior.

Análogamente



no cambia la paridad de los renglones y además no se cumple que f y f^{-1} sean inversas, ya que aplicar f^{-1} seguido de f no me regresa el diagrama original



Así, no podemos aparear a la partición $(3,2)$ con ninguna otra partición , por lo que $[x^5]\prod_{j=1}^{\infty}(1-x^j)=1$. Procedamos a generalizar el ejemplo anterior.

Para eso necesitaremos la siguiente definición.

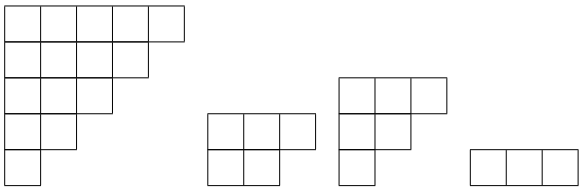
Definición 3.3. *Sean $n \in \mathbb{Z}^+$, Y_n el conjunto de diagramas de Young que representan a las particiones propias de n y $\lambda = (a_1, a_2, \ldots, a_k) \in Y_n$. Diremos que λ es un* ***diagrama de Young válido*** *si $a_i \in \mathbb{Z}^+, a_{i+1} < a_i \; \forall i \in \{1, \ldots, k\}$ y $\sum_{i=1}^{k} a_i = n$.*

Definición 3.4. *Sean $n \in \mathbb{Z}^+$, Y_n el conjunto de diagramas de Young que representan a las particiones propias de n y $Y \in Y_n$. Diremos que Y es un* ***diagrama de Young con esquina*** *si el último renglón y la última diagonal de Y comparten una casilla; en caso contrario lo llamaremos* ***diagrama de Young sin esquina****. Formalmente, $\lambda = (a_1, \ldots, a_k)$ es un diagrama de Young con esquina si $a_{i+i} - a_i = 1$ para toda $i \in \{1, \ldots, k-$*

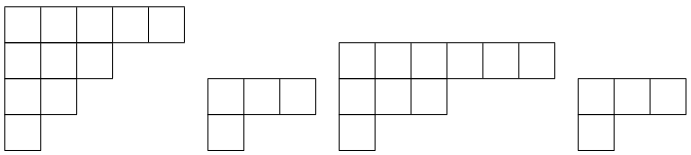
1}. *Al conjunto de diagramas de Young con esquina lo denotaremos por* Y_{ce} *y al conjunto de diagramas de Young sin esquina lo denotaremos por* Y_{se}.

Observación 3.2. *En el universo de los diagramas de Young válidos,* $(Y_{se})^c = Y_{ce}$.

Ejemplo 3.16. *Todos los siguientes son ejemplos de diagramas de Young con esquina:*



Ejemplo 3.17. *Todos los siguientes son ejemplos de diagramas de Young sin esquina:*



Definición 3.5. *Sean* $n \in \mathbb{Z}^+$ *y* Y_n *el conjunto de diagramas de Young que representan a las particiones propias de* n. *Definimos* $f : Y_n \to Y_n$ ***como la función que forma un nuevo renglón colocando los elementos de la última diagonal debajo del último renglón del diagrama correspondiente****. Formalmente, si* $\lambda = (a_1, a_2, \ldots, a_k) \in Y_n$ *es un diagrama de Young válido, entonces* $f(\lambda) = (a_1 - 1, a_2 - 1, \ldots, a_s - 1, a_{s+1}, \ldots, a_k, a_{k+1})$ *donde* s *es el número de elementos en la última diagonal de* λ *y* $a_{k+1} = s$.

Definición 3.6. *Sean* $n \in \mathbb{Z}^+$ *y* Y_n *el conjunto de diagramas de Young que representan a las particiones propias de* n. *Definimos* $g : Y_n \to Y_n$ ***como la función que elimina el último renglón del diagrama y coloca sus elementos, uno a uno, en los renglones anteriores (comenzando por el de mayor longitud)****. Formalmente, si* $\lambda = (a_1, a_2, \ldots, a_k) \in Y_n$ *es un diagrama de Young válido, entonces* $g(\lambda) = (a_1 + 1, a_2 + 1, \ldots, a_m + 1, a_{m+1}, \ldots, a_{k-1})$ *donde* $m = a_k$.

Observación 3.3. *Tanto* f *como* g *cambian la paridad de los renglones de* $\lambda \in Y_n$.

Requerimos que al aplicar f o g, el diagrama de Young resultante sea un diagrama válido -pues únicamente esos son los involucrados en el producto de Euler- y que además cambien la paridad de los renglones del diagrama. Esto restringe los dominios de ambas funciones ya que no siempre ocurre (por ejemplo, dada $n \in \mathbb{Z}^+$, $(n-1, 1) \notin Dom(f), (n-2, 2) \notin Dom(g)$).

De ahora en adelante denotaremos por s al número de elementos en la última diagonal de un diagrama de Young y a m al número de elementos en su último renglón.

Lema 3.1. *Para toda* $n \in \mathbb{Z}^+$

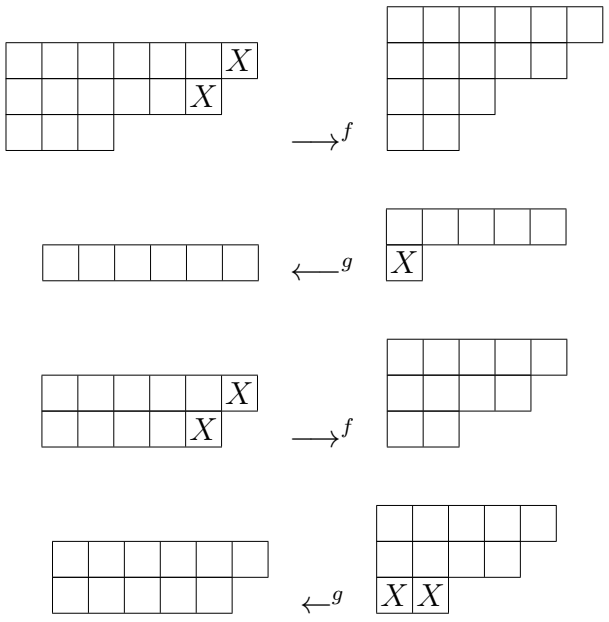
$$Dom(f) = \{Y \in Y_{se} : s < m\} \cup \{Y \in Y_{ce} : s + 1 < m\}$$

$$Dom(g) = \{Y \in Y_{se} : s + 1 > m\} \cup \{Y \in Y_{ce} : s > m\}$$

Demostración. Sea $n \in \mathbb{Z}^+$, $Y \in Dom(f)$. Si $Y \in Y_{se}$ entonces, dado que $f(Y)$ es válido, se satisface $s < m$. Análogamente, si $Y \in Y_{ce}$ entonces $s + 1 < m$.

Sea $Y \in \{Y \in Y_{se} : s < m\} \cup \{Y \in Y_{ce} : s + 1 < m\}$, entonces por definición de f, $f(Y)$ es válido, por lo que $Y \in Dom(f)$.

La prueba de $Dom(g)$ es totalmente análoga.



□

Observación 3.4. *$Y \in Dom(f)$ si y sólo si $Y \notin Dom(g)$.*

Observación 3.5. *Si $Y \in Y_{se}$ entonces $Y \in Dom(f)$ o $Y \in Dom(g)$. Si $Y \in Y_{ce}$ y $m \notin \{s, s + 1\}$ entonces $Y \in Dom(f)$ o $Y \in Dom(g)$.*

Lema 3.2. *Sean $n \in \mathbb{Z}^+$ y Y_n el conjunto de diagramas de Young que representan a las particiones propias de n. Si $Y \in Dom(f)$ entonces $f(Y) \in Dom(g)$ y $(g \circ f)(Y) = Y$. Análogamente si $Y \in Dom(g)$ entonces $g(Y) \in Dom(f)$ y $(f \circ g)(Y) = Y$.*

Demostración. Sea $\lambda = (a_1, \ldots, a_k) \in Dom(f)$. Sea s', m' el número de elementos en la última diagonal de $f(\lambda)$ y el número de elementos en el último renglón de $f(\lambda)$ respectivamente.

Si $f(\lambda) \in Y_{ce}$ entonces $s' = k + 1$, $m' = s \leq k$ en cuyo caso $s' > m'$ por lo que $f(\lambda) \in Dom(g)$. Si $f(\lambda) \in Y_{se}$ entonces $s' \geq s$ y $m' = s$ en cuyo caso $s' \geq m'$ por lo que

$f(\lambda) \in Dom(g)$. En ambos casos, por definición de f y g se cumple que $(g \circ f)(\lambda) = \lambda$.

Análogamente, sean $\lambda = (a_1, \ldots, a_k) \in Dom(g)$ y s', m' el número de elementos en la última diagonal de $g(\lambda)$ y el número de elementos en el último renglón de $g(\lambda)$ respectivamente.

Si $g(\lambda) \in Y_{ce}$ entonces dado que λ es válido, $s' = m = a_k$ y $m' = a_{k-1} + 1$ en cuyo caso $s' < a_{k-1}$ y así $s' + 1 < m'$, de forma que $g(\lambda) \in Dom(f)$. Si $g(\lambda) \in Y_{se}$ entonces $s' = m = a_k$ y $m' = a_{k-1}$ en cuyo caso $s' < m'$ y así $g(\lambda) \in Dom(f)$. En ambos casos, por definición de f y g se cumple que $(f \circ g)(\lambda) = \lambda$. □

Con ayuda de estos lemas, estamos en condiciones de demostrar el célebre teorema de Franklin.

Teorema 3.3 (Franklin, 1881)**.** *Sean $n \in \mathbb{Z}^+$, $Y \in Y_n$. $(g \circ f)(Y) \neq Y$ y $(f \circ g)(Y) \neq Y$ si y sólo si $m = s$ o $m = s + 1$. En dado caso, $n = \frac{k(3k-1)}{2}$ con $k \in \mathbb{Z}$.*

Demostración. Sean $n \in \mathbb{Z}^+$, $Y \in Y_n$. Por el lema 3.2 sabemos que $(g \circ f)(Y) \neq Y$ y $(f \circ g)(Y) \neq Y$ si y sólo si $Y \notin Dom(f)$ y $Y \notin Dom(g)$. Por la observación 3.5 $Y \notin Dom(f)$ y $Y \notin Dom(g)$ si y sólo si $Y \in Y_{ce}$ y $m \in \{s, s + 1\}$.

Si $m = s$ entonces

$$\begin{aligned} n &= m + (m + 1) + \cdots + (m + s - 1) \\ &= m + (m + 1) + \cdots + (m + m - 1) \\ &= m^2 + 1 + 2 + \cdots + m - 1 \\ &= m^2 + \frac{m(m-1)}{2} \\ &= \frac{k(3k-1)}{2} \end{aligned}$$

haciendo $k = m$, de forma que $k \in \mathbb{Z}^+$.

Si $m = s + 1$ entonces

$$\begin{aligned} n &= m + (m + 1) + \cdots + (m + s - 1) \\ &= m + (m + 1) + \cdots + (m + m - 2) \\ &= m(m - 1) + 1 + 2 + \cdots + m - 2 \\ &= m(m - 1) + \frac{(m-1)(m-2)}{2} \\ &= (m - 1)\frac{3(m-1)+1}{2} \\ &= (1 - m)\frac{3(1-m)-1}{2} \\ &= \frac{k(3k-1)}{2} \end{aligned}$$

haciendo $k = 1 - m$, de forma que $k \in \mathbb{Z}^- \cup \{0\}$.

□

Corolario 3.2. *(Teorema de los Números Pentagonales de Euler)*

$$\prod_{n=1}^{\infty}(1-x^n) = \sum_{k=-\infty}^{\infty}(-1)^k x^{\frac{k(3k-1)}{2}}$$

Demostración. Por el lema anterior sabemos que

$$[x^n]\prod_{j=1}^{\infty}(1-x^j) \neq 0 \iff n = \frac{k(3k-1)}{2} \quad \text{p.a.} \quad k \in \mathbb{Z}$$

Más aún, para cada n, de existir, hay un único diagrama de Young válido tal que $m = s$ o $m = s+1$, de forma que $[x^n]\prod_{j=1}^{\infty}(1-x^j) = (-1)^s$. Si $m = s$ entonces $(-1)^s = (-1)^m = (-1)^k$, y si $m = s+1$ entonces $(-1)^s = (-1)^{m-1} = (-1)^{-k} = (-1)^k$.

Luego,

$$[x^n]\prod_{j=1}^{\infty}(1-x^j) = \begin{cases} 0 & \text{si} \quad n \neq \frac{k(3k-1)}{2} \quad \forall k \in \mathbb{Z} \\ (-1)^k & \text{si} \quad n = \frac{k(3k-1)}{2} \quad \text{p.a.} \quad k \in \mathbb{Z} \end{cases}$$

y así

$$\prod_{n=1}^{\infty}(1-x^n) = \sum_{k=-\infty}^{\infty}(-1)^k x^{\frac{k(3k-1)}{2}}.$$

□

3.2.2. La estrella de la película

La teoría que hemos desarrollado para autómatas celulares dinámicos nos permite construir un autómata tal que en el renglón j-esimo tenga codificados los coeficientes de $\prod_{n=1}^{j}(1-x^n)$, de forma que se aproxime en cada iteración al producto infinito de Euler.

Basta con idear un autómata tal que

$$\begin{aligned} f_1 &= a_i - a_{i-1} \\ f_2 &= a_i - a_{i-2} \\ &\vdots \\ f_n &= a_i - a_{i-n} \\ &\vdots \end{aligned}$$

de forma que

$$\begin{aligned}
\mathbb{T}_1 &= 1 - x \\
\mathbb{T}_2 &= 1 - x^2 \\
&\vdots \\
\mathbb{T}_n &= 1 - x^n \\
&\vdots
\end{aligned}$$

y, si $A^0(x) = 1$, entonces

$$A^t(x) = \prod_{j=1}^{t} \mathbb{T}_j = \prod_{j=1}^{t} (1 - x^j)$$

para toda $t \in \mathbb{Z}^+$, que es precisamente el producto (parcial) del Teorema de los Números Pentagonales de Euler.

Así, el autómata celular dinámico $A = (d, S, N_t, f_t)$ con $d = 1, S = \mathbb{Z}$

$$N_t(t) = (-t, 0)$$

y

$$f_t(t) = a_i - a_{i-t}$$

con $A^0(x) = 1$ codifica en cada renglón los coeficientes del producto (parcial) del Teorema de los Números Pentagonales de Euler

$$\prod_{j=1}^{t} (1 - x^j)$$

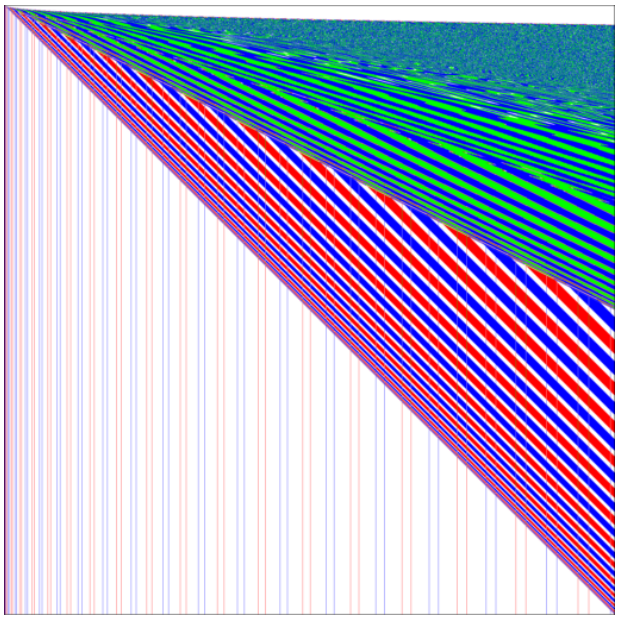


Figura 3.5: Las celdas azules representan el valor -1, las blancas el valor 0, las rojas el valor 1 y las azules cualquier número natural mayor a 2. Por el Teorema de los Números Pentagonales de Euler sabemos que eventualmente ninguna celda tendrá color verde.

3.3. Pseudocódigos

Resulta útil tener un código para programar, no sólo un autómata celular dinámico particular, sino cualquiera de ellos. Dado que los aditivos son de especies interés en este trabajo, a continuación se encuentra el pseudocódigo que utilicé para programar este tipo de autómatas celulares aditivos.

Algoritmo 14: Función $EvolucionarYVecinos : M_{i,j} \to M_{i,j}$

Datos:
$S \in M_{n\times n}(\mathbb{Z})$ Matriz que en cada entrada tiene los exponentes de los dipolinomios de evolución.
$SIG \in M_{n\times n}(\mathbb{Z})$ Matriz que en cada entrada tiene los coeficientes de los dipolinomios de evolución.
Entrada: $M \in M_{n,n}(\mathbb{Z})$
Salida: M modificada.

$contador = 0$
$n \leftarrow$ número de columas de la matriz M
$s \leftarrow$ longitud del arreglo S

Función *EvolucionarYVecinos(M)*
- **for** $i \in \{2, \ldots, n\}$ **do**
  - **for** $j \in \{1, \ldots, n\}$ **do**
    - **for** $l \in \{0, ..., s-1\}$ **do**
      - **if** $i - 1 \equiv l$ *(mód s)* **then**
        - $contador = 0$
      - **end**
      - **if** $l \neq 0$ **then**
        - **if** $j < max(S[l])$ *or* $j - min(S[l]) > n$ **then**
          - $M_{i,j} = 0$
        - **else**
          - $h = 1$ **for** $k \in S[l]$ **do**
            - $contador = contador + SIG[l][h] * M_{i-1,j-k}$
            - $h = h + 1$
          - **end**
          - $M_{i,j} = contador$ (mód 2)
        - **end**
      - **else**
        - **if** $j < max(S[l])$ *or* $j - min(S[l]) > n$ **then**
          - $M_{i,j} = 0$
        - **else**
          - $h = 1$ **for** $k \in S[s]$ **do**
            - $contador = contador + SIG[s][h] * M_{i-1,j-k}$
            - $h = h + 1$
          - **end**
          - $M_{i,j} = contador$ (mód 2)
        - **end**
      - **end**
    - **end**
  - **end**
  - **return** M
- **end**

Para colorear nuestra matriz basta con aplicar la función $ColorearMatriz$ que se muestra a continuación para obtener un autómata en blanco y negro.

Un ejemplo de la implementación del algoritmo anterior sería la siguiente:

```
S = [[1], [0,1,2], [0,1], [0,1,2]]
SIG = [[1],[1,1,1],[1,1],[1,-1,1]]
M = zeros(Int8, (2^(11),2^(11)))
M[1,3] = 1
EvolucionarYVecinos(M,S,SIG)

%Si la queremos colorerar (ver siguiente algoritmo), basta con añadir:

M' = ones(RGB{Float64}, (2^(11),2^(11)))
ColorearMatriz(M',M)
```

Que computaría el autómata cuyos dipolinomios de evolución están dados por

$$\begin{aligned}
\mathbb{T}_1(x) &= x \\
\mathbb{T}_2(x) &= 1 + x + x^2 \\
\mathbb{T}_1(x) &= 1 + x \\
\mathbb{T}_1(x) &= 1 - x + x^2
\end{aligned}$$

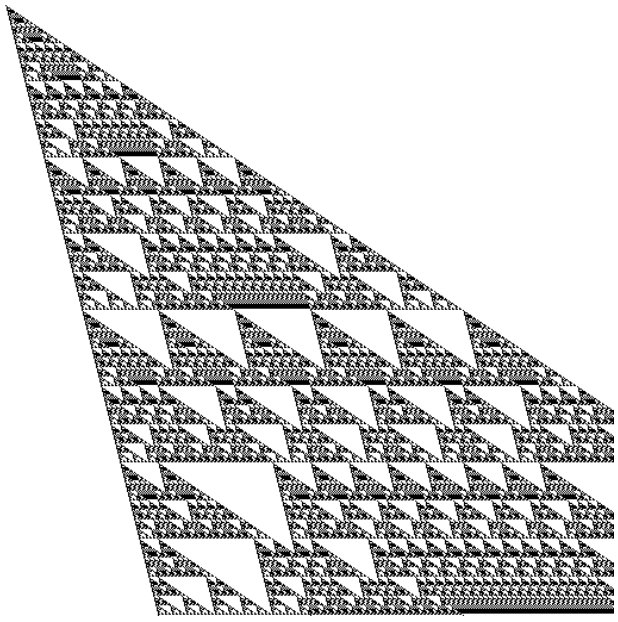


Figura 3.6: Salida del pseudocódigo en el lenguaje Julia.

Dada su relevancia, a continuación se encuentra el pseudocódigo para programar el autómata celular dinámico que codifica los coeficientes del producto involucrado en el Teorema de los Números Pentagonales de Euler. No se puede programar directamente usando el algoritmo 14 pues no tiene un número finito de dipolinomios de evolución.

Supondremos que el lector está familiarizado con el Algoritmo 2 correspondiente a la implementación de la función $G : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$. Modificaremos el Algoritmo 15 para crear una nueva función que colorea matrices según los colores elegidos (en este caso azul, blanco, rojo y verde) y presentaremos una nueva versión del algoritmo para implementar la regla de evolución.

Algoritmo 15: Función $ColorearMatriz : M_{i,j} \to \mathbb{R}^3$

Entrada: $M \in M_{n,n}(\mathbb{Z}), M' \in M_{n,n}(\mathbb{R}^3)$
Salida: M' modificada.

$n \leftarrow$ número de columnas de la matriz M

Función *ColorearMatriz(M,M')*
- **for** $i \in \{1, \ldots, n\}$ **do**
  - **for** $j \in \{1, \ldots, n\}$ **do**
    - **if** $M_{i,j} = -1$ **then**
      - $M'_{i,j} = (0, 0, 1)$
    - **else**
      - **if** $M_{i,j} = 0$ **then**
        - $M'_{i,j} = (1, 1, 1)$
      - **else**
        - **if** $M_{i,j} = 1$ **then**
          - $M'_{i,j} = (1, 0, 0)$
        - **else**
          - $M'_{i,j} = (0, 1, 0)$
        - **end**
      - **end**
    - **end**
  - **end**
- **end**
- **return** M'

end

Algoritmo 16: Función $f : \mathbb{Z}^2 \to \mathbb{Z}$

Entrada: $M \in M_{n,n}(\mathbb{Z}), i, j \in Z^+$
Salida: $x \in \{0, 1\}$

Función *f(M,i,j)*
  if $j + 1 > i$ **then**
    return $M_{i-1,j} - M_{i-1,j-i+1}$
  else
    return $M_{i-1,j}$
  end
end

Notemos que en algoritmo 16 la condición $j + 1 > i$ es porque en el lenguaje Julia el primer renglón de la matriz corresponde a $i = 1$. En lenguajes en los que el primer renglón de la matriz sea el renglón cero, bastaría con pedir $j > i$. Esta condición se verifica pues, si no se cumple, significa que el valor de la casilla que queremos restar se encuentra fuera de nuestra matriz. En teoría, nuestra matriz es infinita, por lo que suponemos que todos los valores no especificados son cero. Así, en estos casos basta con regresar $M_{i-1,j}$ pues $M_{i-1,j-i+1} = 0$.

Un ejemplo de la implementación de los algoritmos anteriores sería la siguiente:

```
AC = zeros(Int8, (2^(11),2^(11)))
AC' = ones(RGB{Float64}, (2^(11),2^(11)))
AC[1,1] = 1
u = G(M, 2^(11))
ColorearMatriz(u,AC')
```

Conclusiones

A lo largo del presente trabajo hemos establecido vínculos entre la teoría de números y la teoría de autómatas celulares. Dicha relación no se limita únicamente a abordar los mismos resultados desde perspectivas diferentes, sino en analizar distintos problemas desde alguna de las teorías para obtener respuestas que desde la otra teoría serían difíciles de obtener.

Tal es el caso de la Regla 60 y de la Regla 90, mismas que estudiamos exhaustivamente en el capítulo 2. Por un lado, desde la teoría de números, explicamos por qué los números de Fermat -y sus productos- pueden ser extraídos de un objeto tan emblemático como el Triágulo de Pascal. Por otro lado, desde la teoría de autómatas, entendemos por qué el Triángulo de Pascal se corresponde con la Regla 60 y, juntando ambos resultados, analizamos -haciendo uso de la teoría de números- los aspectos computacionales -como son la reducibilidad computacional- del Triángulo de Pascal visto como un autómata celular. Más aún, haciendo uso de la teoría de autómatas, generalizamos el Triángulo de Pascal y creamos autómatas que codifican particiones de enteros, mismas que han sido uno de los problemas más fundamentales de la teoría de números.

No deja de ser soprendente que la relación entre ambas ramas de la matemática no solamente nos permite estudiar problemas conocidos desde ópticas nuevas, sino que nos permite incluso generar nuevos objetos: éste es el caso de los autómatas celulares dinámicos, a quienes les dedicamos el tercer capítulo. Con la herramienta de las funciones generadoras extendemos la noción de autómata celular a una clase más general de objetos que nos permiten entender de una forma más completa el universo computacional. Y, por si fuera poco, generamos un autómata celular dinámico que representa a uno de los teoremas más emblemáticos de la teoría de números: el Teorema de los Números Pentagonales de Euler. Incluso desde el primer capítulo exploramos la relación entre ambas ramas de la matemática mediante algoritmos que detectan configuraciones inalcanzables por ciertos autómatas celulares o, más fundamentalmente, demostrando -usando herramienta puramente matemática- la existencia de dichas configuraciones.

Muchos de los conceptos aquí explorados -como, por ejemplo, los de irreducibilidad computacional, aleatoriedad intrínseca, universalidad computacional, etc.- son relevantes para la comprensión tanto de las matemáticas y los sistemas deductivos en general, como para el entendimiento de diversos procesos en la naturaleza y en el universo computacional. A pesar de no haber podido ahondar demasiado en ellos -tanto por las características del

presente trabajo como por las intenciones del autor- tenemos presentes que darán pie a largas y apasionadas discusiones e investigaciones, mismas que serán objeto de nuestros futuros esfuerzos, y muchas de las cuales seguramente podrán verse nutridas por los resultados aquí expuestos.

Entender el universo computacional y su relación con el universo matemático es y seguirá siendo uno de los problemas centrales de la ciencia; el presente trabajo pretende vislumbrar algunas de sus múltiples y fascinantes relaciones.

Bibliografía

[1] Amoroso, Serafino *et. al.* "Decision Procedures for Surjectivity and Injectivity of Parallel Maps for Tessellation Structures". En: *Journal of Computer and System Sciences* 6 (1972), págs. 448-464.

[2] Burks W, John Arthur *et. al.* "Theory of Self-Reproducing Automata". En: *Theory of Self-Reproducing Automata*. Urbana: University of Illinois Press, 1966, pág. 82. ISBN: 9780598377982.

[3] Delahaye, Jean-Paul *et. al.* "Unpredictability and Computational Irreducibility". En: *Irreducibility and Computational Equivalence: 10 Years After Wolfram's A New Kind of Science*. Springer Berlin Heidelberg, 2011, págs. 273-295. ISBN: 9783642354823.

[4] Fine, N. J. "Binomial Coefficients Modulo a Prime". En: *The American Mathematical Monthly* 54.10 (1947), págs. 589-592. ISSN: 00029890, 19300972.

[5] García Gutiérrez, Abel. *Particiones y funciones generadoras : del juego de dados a la teoría aditiva*. 2010. URL: `http://132.248.9.195/ptb2010/septiembre/0662036/Index.html`.

[6] Hedlund, G.A. "Endomorphisms and Automorphisms of the Shift Dynamical System". En: *Mathematical Systems Theory* 4.3 (1969), págs. 320-375. DOI: `10.1007/BF01691062`.

[7] Kari, Jarkko. *Cellular Automata*. 2013. URL: `https://www.cs.tau.ac.il/~nachumd/models/CA.pdf`.

[8] Kari, Jarkko. "Reversibility of 2D cellular automata is undecidable". En: *Physica D: Nonlinear Phenomena* 45.1 (1990), págs. 379-385. ISSN: 0167-2789. DOI: `10.1016/0167-2789(90)90195-U`.

[9] Koch, Dick. *The Pentagonal Number Theorem and All That*. 2016. URL: `https://pages.uoregon.edu/koch/PentagonalNumbers.pdf`.

[10] Křížek, Michal *et. al. 17 Lectures on Fermat Numbers: From Number Theory to Geometry*. CMS Books in Mathematics. Springer New York, NY, 2001. ISBN: 9780387953328.

[11] Martin, Olivier *et. al.* "Algebraic Properties of Cellular Automata". En: *Communications in Mathematical Physics* 93 (jun. de 1984). DOI: `10.1007/BF01223745`.

[12] Mathonet, Pierre *et. al. On digital sequences associated with Pascal's triangle*. 2022. DOI: `10.48550/ARXIV.2201.06636`. URL: `https://arxiv.org/abs/2201.06636`.

[13] Moore, E. F. "Machine models of self-reproduction". En: *Proceedings of Symposia in Applied Mathematics* 14 (1962), págs. 187-203.

[14] Munkres, James. "Topology: a first course". En: Pearson College, 2000, pág. 116. ISBN: 0131816292.

[15] Myhill, John. "Shorter Note: The Converse of Moore's Garden-of-Eden Theorem". En: *Proceedings of the American Mathematical Society* 14.4 (1963), págs. 685-686. DOI: 10.1090/S0002-9939-1963-0155764-9.

[16] Rozenberg, Grzegorz (edt) *et. al. Handbook of Natural Computing*. Springer Link, 2012. ISBN: 9783540929093.

[17] Skiena, Steven S. *The Algorithm Design Manual*. 2nd. Springer Publishing Company, Incorporated, 2008. ISBN: 1848000693.

[18] Wolfram, Stephen. *A New Kind of Science*. Wolfram Media, 2002. ISBN: 1579550088.


CIENCIAS