

**UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO**

FACULTAD DE CIENCIAS

**“EL JUEGO CIRCULAR DEL CAOS EN EL
DNA Y COMPRESIÓN FRACTAL DE
IMÁGENES”**

**TESIS
CONJUNTA
PARA OBTENER EL TITULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN Y
MATEMÁTICO
PRESENTAN
GUSTAVO CARREÓN VAZQUEZ
JESÚS ENRIQUE HERNÁNDEZ ZAVALA**

DIRECTOR: DR. PEDRO EDUARDO MIRAMONTES VIDAL

2004



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Why is geometry often described as 'cold' and 'dry'? One reason lies in its inability to describe the shape of a cloud, a mountain, a coastline, or a tree. Clouds are not spheres, coastlines are not circles, and bark is not smooth, nor does lightning travel in a straight line. [...] The existence of these patterns challenges us to study those forms that Euclid leaves aside as being 'formless', to investigate the morphology of the 'amorphous'.

Benoit B. Mandelbrot

Fractal geometry will make you see everything differently. There is danger in reading further. You risk the loss of your childhood vision of clouds, forests, flowers, galaxies, leaves, feathers, rocks, mountains, torrents of water, carpets, bricks, and much else besides. Never again will your interpretation of these things be quite the same.

Michael F. Barnsley

A mis padres Maricela y Gustavo, a Paty, a mis hermanos Iridián e Iván.

Gustavo

En memoria de mis abuelos Nina y Abundio, a mis padres Gloria y Jesús, a mis hermanos Ana y Arturo.

Enrique

Agradecimientos

Agradecemos al big bang y a aproximadamente 15 mil millones de años de evolución, que nos han permitido llegar a este momento en el el espacio y el tiempo junto con nuestras familias, conocer a nuestros amigos y aprender de nuestros profesores; con todos ellos hemos compartido experiencias que nos han ayudado a ser mejores en lo personal y en lo académico.

Especialmente agradecemos a nuestros padres por todas las clases de apoyo que nos han brindado y por haber construido los cimientos de nuestra formación con su amor y ejemplo. A nuestros hermanos con los que hemos convivido y compartido experiencias, a nuestros abuelos, tíos, primos y amigos por alentarnos y darnos buenos consejos para seguir adelante con nuestras metas.

A nuestros profesores les agradecemos por su amistad y por compartir sus conocimientos, por guiarnos a través de caminos inexplorados, por expandir nuestros conocimientos ayudándonos a combatir el fantasma de la ignorancia.

Agradecemos a nuestros amigos de la “unidad” que han estado con nosotros desde niños y hemos crecido juntos compartiendo vivencias y horas de estudio, también a todos los que hemos conocido a lo largo de nuestras vidas. En especial a Paty por aguantarnos y estar siempre que se le necesita; a Vicente por apoyarnos moral y académicamente durante el transcurso de nuestras carreras; a Juan (jurema) por dedicar tiempo en la elaboración de ilustraciones para este trabajo. A Iván por tener paciencia y esperar su turno para utilizar la computadora.

Agradecemos a nuestros amigos de la Facultad de Ciencias por su amistad y su apoyo desde los primeros semestres, por compartir buenos y malos momentos académicos, y por compartir alegres momentos de esparcimiento. En especial a los Computólogos y los Macacos; y en general a todos los amigos que en algún momento en nuestra estancia en Ciencias nos brindaron su amistad y su apoyo.

Agradecemos a nuestro maestro y amigo Pedro Miramontes por compartir sus ideas con nosotros ya que sin ellas este trabajo no se hubiera llevado a cabo y por iniciarnos en temas fascinantes y desconocidos por nosotros, a nuestros sinodales por el tiempo dedicado en la revisión de este trabajo y por complementar nuestro conocimiento en el área, igualmente a Faustino Sánchez por compartir su conocimiento con nosotros.

En lo personal quiero dar un gran agradecimiento a mi amigo Jesús por su amistad y sus buenos consejos, por compartir grandes momentos a lo largo de nuestras vidas, como es, la consumación de nuestras carreras, y por el esfuerzo y dedicación que puso de su parte para la realización de este proyecto.

Quiero hacer un agradecimiento especial a mi gran amigo de toda la vida con el que he compartido grandes momentos en mi carrera profesional y en mi vida personal. Juntos empezamos a conocer este maravilloso mundo de las matemáticas y es un gran honor cerrar este ciclo tan importante de mi vida con Gustavo mi gran amigo.

Índice general

1. Introducción	1
2. Conceptos sobre el DNA	5
2.1. La estructura del DNA	5
2.2. Proceso de replicación del DNA	9
2.3. Algunos hechos sobre los genes	9
2.4. Proceso de transcripción-traducción	10
2.5. El código genético	13
2.6. Mutaciones	13
3. El juego del caos	17
3.1. Reglas básicas del juego del caos	18
3.2. Fractales	19
3.3. Mapeos como generadores de sucesiones	21
3.4. El juego del caos circular	23
3.5. El <i>shift</i> de Bernoulli y el juego del caos circular	25
3.6. Jugando con nucleótidos	30
4. Sistemas de funciones iteradas	33
4.1. Espacio métrico y transformaciones	33
4.2. Transformaciones afines	36
4.3. Propiedades de espacios métricos y subconjuntos	40
4.4. Transformaciones de contracción	45
4.5. Espacio de Hausdorff	47
4.6. Mapeos de contracción en el espacio \mathcal{H}	51
4.7. Sistemas de funciones iteradas	51
4.8. Algoritmo de la máquina fotocopidora	53
4.8.1. Ejemplo de un IFS	54

4.9. IFS con probabilidades	57
4.9.1. EL algoritmo de iteraciones aleatorias	57
4.10. El teorema del collage	58
4.11. Medida invariante	60
5. Compresión fractal	63
5.1. Bases de la compresión fractal de imágenes	64
5.1.1. La mejor compresión fractal de imágenes	66
5.1.2. Auto semejanza en imágenes	67
5.2. Metodología para la compresión y descompresión fractal de imágenes	68
5.2.1. Sistemas de funciones iteradas locales	68
5.2.2. El teorema del collage para IFS locales	72
5.2.3. Transformación fractal para imágenes binarias	74
5.2.4. Un IFS local asociado a un esquema de compresión para imágenes en escala de grises	78
5.3. Algoritmo para hacer la compresión y descompresión frac- tal de imágenes en escala de grises	82
5.3.1. Algoritmo para la compresión fractal	83
5.3.2. Algoritmo para la descompresión fractal	85
6. Aplicaciones a la clasificación de organismos a partir del DNA	89
6.1. Método de clasificación de organismos	89
6.2. Dendrogramas asociados a organismos	101
7. Discusión	107

Índice de cuadros

2.1. Cuadro de asignación de dígitos binarios a los nucleótidos de secuencias de DNA de a cuerdo con sus características.	8
2.2. Código genético estándar. El primer nucleótido del triplete es indicado por la primer columna del lado izquierdo, el segundo es indicado por las cuatro columnas centrales y el último es indicado por la última columna de la derecha. Aminoácidos codificados por cada codón del ARN mensajero. ALA: Alanina, CYS: Cisteína, ASP: Ácido aspártico, GLU: Ácido glutámico, PHE: Fenilalanina, GLY: Glicina, His: Histidina, ILE: Isoleucina, LYS: Lisina, LEU: Leucina, MET: Metionina, ASN: Asparagina, PRO: Prolina, GLN: Glutamina, Arg: Arginina, SER: Serina, TMR: Treonina, VAL: Valina, TRP: Triptofano, TYR: Tirosina, y STOP denota los aminoácidos Pardo, Ámbar y Ocre que no codifican pero que sirven como signos de puntuación para el fin del mensaje.	14
4.1. Medidas de dos bolas con distintas iteraciones, se puede ver que entre más iteraciones la proporción de puntos converge a un valor.	61
5.1. Se muestra el conjunto de las 7 transformaciones para el cuadrado. Las rotaciones son en el sentido de las manecillas del reloj.	74
5.2. Código del IFS local para la imagen G	78
6.1. Organismos utilizados para su clasificación. A cada organismo se le asocia una letra que sirve de clave para los Cuadros 6.4 y 6.5.	90
6.2. Medida invariante sobre una bola con distintas iteraciones. . .	93

6.3. Para cada organismo se muestra la coordenada de la esquina inferior izquierda de la celda donde se encuentra el máximo conteo de puntos.	96
6.4. Matriz simétrica de distancias para cadenas binarias obtenidas mediante la regla <i>WS</i>	99
6.5. Matriz simétrica de distancias para cadenas binarias obtenidas mediante la regla <i>YR</i>	100

Índice de figuras

2.1. Estructura de doble hélice del DNA.	6
2.2. Las bases componen la parte variable del DNA, los nucleótidos adenina (A) y guanina (G) se conforman de dos anillos por lo que se les llama “bases grandes”, mientras que la timina (T) y citosina (C) por tener un anillo, se les llama “bases pequeñas”.	7
2.3. Enlaces de hidrógeno entre nucleótidos; un triple enlace ocurre entre G y C por lo que se les llama “bases fuertes”, mientras que un doble enlace ocurre entre A y T por lo que se les llama “bases débiles”.	8
2.4. Proceso de replicación del DNA.	9
2.5. Proceso de edición que ocurre en los genes durante el proceso de transcripción para poder formar el RNA mensajero (RNA_m).	11
2.6. Proceso de transcripción para formar el RNA mensajero.	12
2.7. Proceso de traducción el cual a partir de codones se forman los aminoácidos.	12
2.8. Fenómeno de mutación, puede ocurrir durante los procesos de replicación o de transcripción.	15
3.1. Movimiento browniano sobre el plano.	18
3.2. Triángulo de Sierpinski, generado por el juego del caos.	20
3.3. Juego del caos para el mapeo logístico.	22
3.4. (A) Juego del caos para el mapeo tienda de apache y (B) juego del caos para el mapeo diente de sierra.	23
3.5. Juego del caos para el mapeo logístico en el cuadrado.	24
3.6. Mapeos en el cuadrado. (A) Juego del caos para el mapeo tienda de apache y (B) para el mapeo diente de sierra.	24
3.7. Juego del caos con el mapeo logístico, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.	26

3.8.	Juego del caos con el mapeo tienda de apache, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.	26
3.9.	Juego del caos con el mapeo diente de sierra, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.	27
3.10.	En la figura podemos ver un polígono con 8 vértices, y se muestra una cadena binaria de entrada, cada vértice se encuentra etiquetado por secuencias binarias que corresponden a un número decimal entre 0 y 7, en sentido contrario a las manecillas del reloj, se puede ver cómo se aplica el shift de Bernoulli a la secuencia de entrada que indica en qué orden se van tomando los vértices.	29
3.11.	Juego del caos en el cuadrado utilizando una secuencia de letras generadas aleatoriamente.	31
3.12.	Juego del caos en el cuadrado utilizando la cadena de DNA del <i>Homo sapiens</i> . A cada vértice se le asigna un nucleótido, comenzado por el vértice superior izquierdo en dirección contraria a las manecillas del reloj se asignan A, G, C y T respectivamente.	31
4.1.	Ejemplos de métricas para el plano euclidiano entre los puntos p_1 y p_2	34
4.2.	Transformaciones f y f^{-1} en el plano euclidiano. Se ilustra cómo se modifica el conjunto S al aplicarle una transformación f ; así mismo, si aplicamos la transformación inversa al conjunto $f(S)$ obtenemos el conjunto original S ; también se ejemplifica con un solo punto.	35
4.3.	Composición de transformaciones.	36
4.4.	Transformaciones básicas.	38
4.5.	Imagen representando sucesivas ampliaciones sobre una sucesión de Cauchy.	41
4.6.	Ampliaciones alrededor del punto fijo.	42
4.7.	Punto límite del conjunto S	43
4.8.	S es cerrado si sus puntos límite están dentro del conjunto.	44
4.9.	Comparación entre el conjunto S y el conjunto de puntos límite de S	45
4.10.	Ejemplo de punto fijo.	46
4.11.	La transformación tiene la propiedad de contracción.	47
4.12.	La representación de elementos en el espacio de Hausdorff.	48

4.13. Se muestra la menor distancia que hay entre un punto x y el conjunto B	49
4.14. La distancia entre los conjuntos A y B se calcula tomando la distancia máxima de entre las distancias mínimas de los puntos de A al conjunto B	50
4.15. La aplicación de un conjunto de transformaciones de contracción al punto B en el espacio de Hausdorff $\mathcal{H}(\mathbb{R}^2)$, el resultado es la unión de los nuevos puntos.	52
4.16. Plantilla arbitraria para generar un atractor con el algoritmo de la fotocopidora.	54
4.17. Iteraciones generadas por el IFS.	55
4.18. Un Conjunto inicial distinto al de la Figura 4.17 convergiendo al triángulo de Sierpinski.	56
4.19. Helecho de Barnsley generado por un IFS con probabilidades utilizando como conjunto inicial el triángulo de Sierpinski. . .	59
4.20. El cuadro negro sobre el triángulo de Sierpinski generado por el juego del caos representa una bola.	61
5.1. Proceso del algoritmo de la fotocopidora para una imagen cualquiera, escalando la imagen a un medio y copiandola tres veces, como resultado de 5 iteraciones se converge al triángulo de Sierpinski.	65
5.2. Helecho de Barnsley generado por cuatro transformaciones. . .	66
5.3. Una sucesión decreciente de conjuntos convergiendo a un atractor de un IFS local.	71
5.4. La transformación w manda R_i a un dominio D_i que es de menor tamaño.	73
5.5. La imagen G es cubierta por dominios D_i de forma adecuada. . .	75
5.6. Se escoge un conjunto de coordenadas (x,y) adecuado para la imagen G	76
5.7. Se escoge un R_i y una transformación w_i para un dominio específico D_i , tal que la distancia de Hausdorff $h(w_i(R_i \cap G), D_i \cap G)$ sea mínima.	77
5.8. El código del IFS local es usado para descomprimir una imagen mediante la iteración del operador W_{local}	79
5.9. Elemento del espacio $\mathbf{X} = \square \times [a, b]$	80
5.10. Se observa el efecto que tiene la transformación de contracción v_i sobre los elementos z_1 y z_2	80

5.11. Proceso para el operador \mathbf{W}_{local}	82
5.12. En (A) se divide una imagen en 16 dominios homogéneos sin intersectarse, en (B) se muestra la misma figura con sus 9 posibles rangos y respectivas coordenadas.	84
5.13. Representación bidimensional de la conclusión del paso 3, cada D_i ha obtenido su R_i , su simetría y su escala de gris Q_i	85
5.14. Se muestra un diagrama que une los algoritmos de compresión y descompresión, se inicia con la imagen G pasando por el proceso de compresión y obteniendo su código del cual se muestra una parte en la tabla central, por último se aplica el proceso de descompresión y se obtiene una imagen G' que es parecida a G	87
6.1. Malla sobre el atractor del juego del caos circular correspondiente a la secuencia del <i>Homo sapiens</i> . La malla cubre un cuadrado cuyos lados tienen longitud 2 y está centrado en el origen.	92
6.2. Representación de un bola sobre el atractor del juego del caos circular utilizando la secuencia binaria WS del cromosoma 21 del <i>Homo sapiens</i>	93
6.3. Medida invariante sobre el atractor del juego del caos circular con la secuencia correspondiente al <i>Homo Sapiens</i>	94
6.4. Histograma para los organismos (A) <i>T. maritima</i> , (B) cromosoma 21 del <i>H. sapiens</i> , (C) <i>S. solfataricus</i> y (D) <i>B. subtilis</i> , con los ejes x y y se muestra la orientación de cada uno de ellos.	95
6.5. (A) Imagen en escala de grises asociada al histograma del organismo <i>T. maritima</i> . (B) Imagen construida a partir de su código IFS. (C) Imagen en escala de grises asociada al histograma correspondiente al cromosoma 21 del organismo <i>H. sapiens</i> . (D) Imagen construida a partir de su código IFS.	97
6.6. (A) Imagen en escala de grises asociada a la medida invariante del organismo <i>S. solfataricus</i> . (B) Imagen construida a partir de su código IFS. (C) Imagen en escala de grises asociada a la medida invariante del organismo <i>B. subtilis</i> . (D) Imagen construida a partir de su código IFS.	98
6.7. Dendrogramas generados respectivamente con las matrices de distancias WS y YR por el método de pesos ponderados.	103

- 6.8. Dendrograma para la matriz de distancias WS generado por el método de vecinos cercanos y vecinos lejanos, respectivamente. 104
- 6.9. Dendrograma para la matriz de distancias YR generado por el método de vecinos cercanos y vecinos lejanos, respectivamente. 105

Capítulo 1

Introducción

A lo largo del tiempo la ciencia y la tecnología han evolucionado de acuerdo a las necesidades del contexto social y cultural en el que se desarrollan, cada cultura genera sus propios métodos y técnicas para adaptarse a su entorno; por ejemplo, los sistemas de numeración, la fabricación de embarcaciones y diversas herramientas que facilitan la vida cotidiana. Conforme estos conocimientos crecen, la humanidad puede ser capaz de mantener un intercambio cultural con distintas sociedades dando lugar a una expansión y enriquecimiento de sus conocimientos. Esto ha implicado el surgimiento de nuevas técnicas que benefician la comunicación entre ellas.

En la actualidad las vías de comunicación han ayudado al desarrollo científico y tecnológico facilitando la propagación de información de una manera rápida y eficaz. Con el surgimiento de las computadoras, la humanidad pudo manejar grandes cantidades de información, de una manera antes inimaginable. Con la ayuda de esta herramienta se hace primordial el intercambio de datos de manera rápida y confidencial a lugares geográficamente distantes, una solución ha sido transmitir éstos por medio de cables conectados entre las computadoras. Al transcurrir los años, el número de computadoras conectadas se fue incrementando, formando pequeñas redes de computadoras creadas para los fines específicos de alguna comunidad. A su vez, estas comunidades tuvieron la necesidad de hacer un intercambio de información entre ellas por lo que se inició la conexión entre sus redes, formando así una gran red de redes mundial, ahora conocida como Internet, la vía de comunicación electrónica más importante de nuestros días.

Con este intercambio de grandes cantidades de información a grandes velocidades, desaparecen las barreras geográficas, formando un sistema unificado limitado solamente por el flujo y el almacenamiento de datos que aumenta día a día.

En la búsqueda de métodos y técnicas para dar solución a los problemas de almacenamiento y transmisión de grandes cantidades de datos, se han creado algoritmos de compresión los cuales reducen la cantidad de éstos explotando sus redundancias y conservando sus características principales para no perder información al momento de la descompresión. Se han generado estándares que regulan el funcionamiento de estos algoritmos. Al principio se reconstruía intacta la información comprimida, pero con el surgimiento de nuevos tipos de datos como imágenes, audio y video que requieren de grandes espacios para su almacenamiento, se ha considerado perder un porcentaje de esta información tomando en cuenta las limitaciones de los sentidos del ser humano; por ejemplo, el formato de audio mp3 el cual elimina las frecuencias que no son perceptibles al oído humano reduciendo el tamaño de los archivos aproximadamente en un 80% y conservando una alta fidelidad en el sonido. Con los actuales métodos de compresión se necesita adecuar al constante crecimiento de la transmisión y almacenamiento de datos, por lo que los científicos se han dado a la tarea de buscar nuevas formas de poder comprimir éstos de la manera más eficiente.

En las últimas décadas se ha descubierto un algoritmo capaz de reducir en un 99% el tamaño en bytes de imágenes; este algoritmo se basa en teorías matemáticas relativamente recientes, utiliza el concepto de autosemejanza para encontrar redundancias en diferentes partes de la imagen a diferentes escalas, esta propiedad es una característica de los *fractales*. Por tal motivo, a este método se le conoce con el nombre de *compresión fractal*. La importancia de este algoritmo radica en el código que genera la compresión de una imagen; este código, a diferencia de los generados por otros algoritmos de compresión, es manejable ya que se compone de una matriz definida por números lo que hace posible utilizar este algoritmo para diferentes propósitos; por ejemplo, hacer comparaciones entre imágenes, calculando qué tanto difieren las matrices que pertenecen a cada imagen.

Por otro lado se ha encontrado la forma de asociar una imagen al código genético de organismos, utilizando una técnica llamada *juego del caos*; ésta

nos brinda una imagen única para cada organismo. La cual, mediante la compresión fractal, se le puede obtener un código que también es único; de esta manera se puede obtener un algoritmo de clasificación robusto el cual nos permite observar diferencias estructurales en los organismos a partir de su DNA.

En el capítulo 2 se puede encontrar una descripción detallada de la molécula de DNA junto con sus procesos básicos como es el de replicación, transcripción-traducción y mutación. También se describirá la forma en que se manipularán las cadenas de DNA para este trabajo y un cuadro donde se encuentra el código genético.

En el capítulo 3 se describe la técnica llamada juego del caos, así como su generalización que se llamará juego del caos circular; se definirán algunos mapeos para ejemplificar el funcionamiento del juego del caos, particularmente nos enfocaremos al mapeo *shift* de Bernoulli ya que manipula secuencias binarias. También se encuentra una breve sección que describe a grandes rasgos el concepto de fractal.

En el capítulo 4 se darán las bases teóricas para entender la compresión fractal de imágenes. Se dará una breve introducción a los espacios métricos y transformaciones sobre ellos, llegando a la definición de un espacio métrico de Hausdorff, para finalmente llegar a la definición de un sistema de funciones iteradas y dar a conocer el teorema del *collage*; concluiremos esta sección dando una definición de la medida invariante de un sistema de funciones iteradas.

En el capítulo 5 se darán los fundamentos de la compresión fractal para imágenes, definiremos lo que es un mapeo de contracción local para llegar a la definición de un sistema de funciones iteradas local, éstas nos servirán para explicar el teorema del collage aplicado a sistemas de funciones iteradas locales. Los conceptos antes mencionados se aplicarán a imágenes en blanco y negro y posteriormente se definirá un método para compresión y descompresión de imágenes en escala de grises. Se concluirá por describir los algoritmos de compresión y descompresión fractal desarrollados por Michael Barnsley.

En el capítulo 6 se describen los resultados obtenidos al aplicar el algoritmo de compresión fractal a imágenes del juego del caos que son generadas mediante

secuencias de DNA. Se muestran dendrogramas en los cuales se aprecian las diferencias entre 10 diferentes organismos.

Para concluir en el capítulo 7 se da una discusión de la eficiencia del método de clasificación propuesto.

Este trabajo es de gran importancia por el paradigma científico que se está viviendo, es decir el de los *sistemas complejos*, ya que una de sus características es la interdisciplinaridad y el estudio de fenómenos con un enfoque no lineal. Se ha logrado conjuntar diferentes disciplinas científicas como la biología, las matemáticas y las ciencias de la computación dando lugar a un enriquecimiento entre ellas. Este tipo de prácticas científicas ayudarán a quitar las barreras entre las diferentes disciplinas permitiendo una unificación de la ciencia que logre tener un beneficio real para la sociedad en general.

Capítulo 2

Conceptos sobre el DNA

2.1. La estructura del DNA

La molécula de DNA es un polímero lineal construido con una sucesión de cuatro monómeros distintos llamados nucleótidos. Cada monómero tiene una parte constante que es una molécula de fosfato y desoxirribosa, y una parte variable que puede ser cualquiera de las siguientes cuatro bases nitrogenadas o nucleótidos: adenina (A), guanina (G), timina (T) y citosina (C). El DNA es una doble hélice formada por dos polímeros lineales antiparalelos en forma de escalera, los *pasamanos* de la escalera son moléculas de fosfato y desoxirribosa (azúcar) mientras que los *peldaños* son una de las cuatro bases anteriormente mencionadas, una de cada lado y unidas entre sí mediante puentes de hidrógeno (ver Figura 2.1).

Adicionalmente, se debe mencionar que las bases nitrogenadas pueden ser de dos tipos, la A y la G pertenecen a la familia de las purinas y son moléculas de dos anillos que llamaremos *grandes*; por otra parte la T y la C pertenecen a la familia de las pirimidinas y son moléculas de un anillo que llamaremos *pequeñas* (ver Figura 2.2).

Como cada peldaño esta formado por una base de cada lado, para mantener homogénea la anchura de la escalera es necesario que si en un lado de la doble hélice se encuentra una purina, del lado opuesto le corresponda una pirimidina. Esto, aunado al hecho de que no todas las uniones purina-pirimidina son posibles debido a la incompatibilidad de formar adecuadamente los enlaces de hidrógeno, esto determina el siguiente principio de complementaridad en-

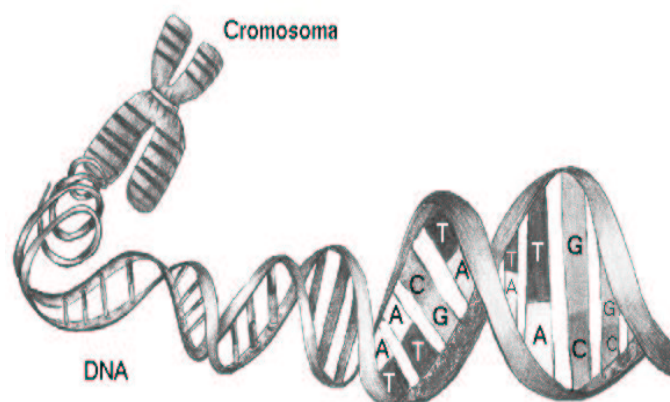


Figura 2.1: Estructura de doble hélice del DNA.

tre los lados de la escalera: cada vez que aparece una molécula de adenina (A) del otro lado existe una timina (T) con la cual esta apareada y viceversa, mientras que cada vez que aparezca una de citosina (C) se hallará apareada con una de guanina (G) y viceversa. Por ello, la información contenida en la molécula se puede deducir de una sola rama de la doble hélice (ver Figura 2.3).

La unión de los peldaños de la escalera no es homogénea; las parejas A-T están unidas mediante dos puentes de hidrógeno, mientras que las parejas C-G lo están mediante tres puentes de hidrógeno (ver Figura 2.3). Esto sugiere que las secciones de la cadena más abundantes en G o C sean, en promedio, más rígidas que aquéllas en las cuales la abundancia relativa de A-T sea mayor, en el sentido de la energía necesaria para separar las dos ramas de la molécula. Por otra parte, la diferencia de tamaño entre las purinas y pirimidinas hace que la escalera no sea geoméricamente uniforme; si de un solo lado se alternan purinas y pirimidinas, la escalera se encontrará dislocada, de manera que podemos afirmar que en este caso la molécula es en promedio rugosa. Si localmente se encuentran sólo purinas la cadena será lisa. De aquí en adelante, llamaremos a las bases G y C fuertes, A y T débiles.

La cadena de DNA contiene toda la información necesaria para la producción de cadenas de aminoácidos (polipéptidos) que, a su vez, dan lugar a las

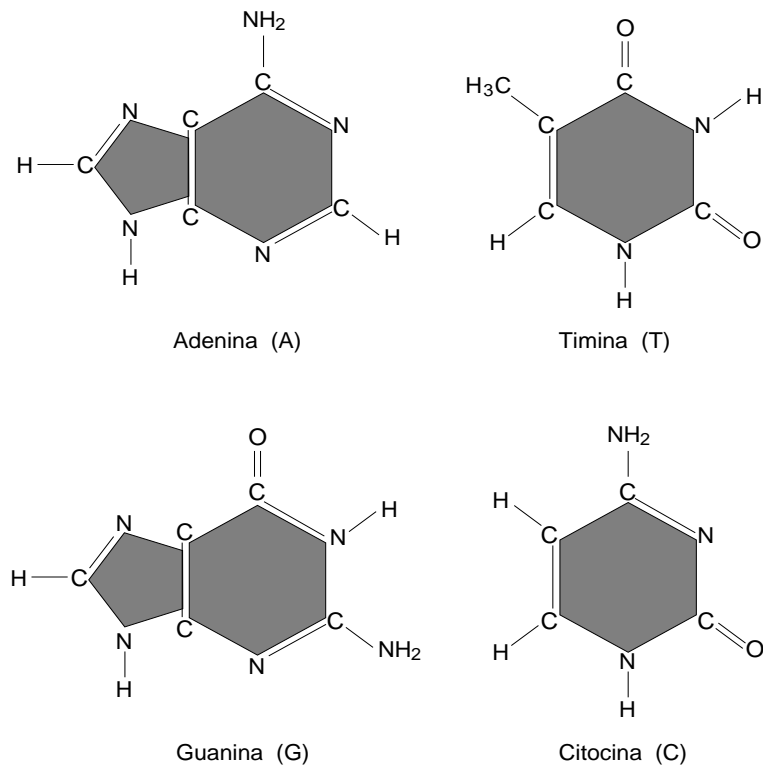


Figura 2.2: Las bases componen la parte variable del DNA, los nucleótidos adenina (A) y guanina (G) se conforman de dos anillos por lo que se les llama “bases grandes”, mientras que la timina (T) y citosina (C) por tener un anillo, se les llama “bases pequeñas”.

proteínas que son la materia prima de todo ser vivo. En el DNA se encuentra codificada la herencia morfológica y funcional que una célula transmite a sus descendientes.

Se ha hablado de las relaciones fuertes o débiles, grandes o pequeñas, así como de su naturaleza purínica o pirimídica que existen entre los cuatro nucleótidos; para propósitos de este trabajo, las cadenas de DNA se verán como secuencias de letras A,G,C y T y utilizaremos sus propiedades de relación para poder trabajar con ellas en forma binaria, es decir cadenas de ceros y unos. Se pueden tener tres formas de asignar ceros y unos a las cuatro letras, una es por la relación fuertes-débiles a lo que llamaremos la clasificación

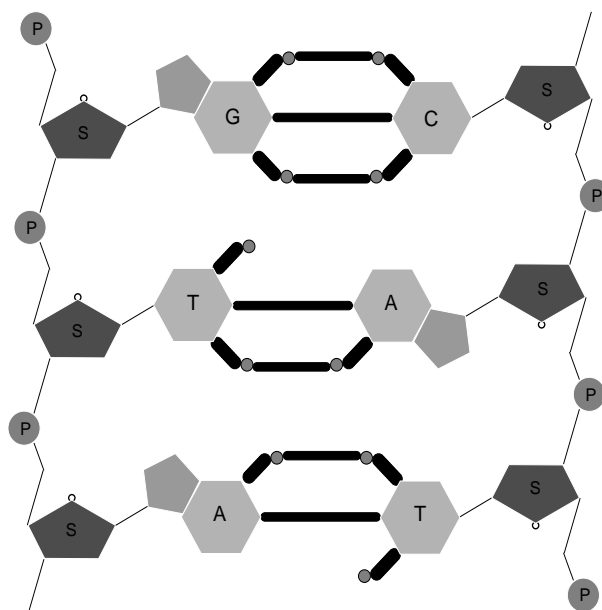


Figura 2.3: Enlaces de hidrógeno entre nucleótidos; un triple enlace ocurre entre G y C por lo que se les llama “bases fuertes”, mientras que un doble enlace ocurre entre A y T por lo que se les llama “bases débiles”.

WS ; otra es la relación grandes-pequeñas, llamada YR y la última relación que se da por su naturaleza amínada-cetónica le llamaremos MK que es la combinación faltante de las dos anteriores [18]; la asignación se muestra en el Cuadro 2.1.

$$WS = \begin{cases} \text{Fuertes} & \text{C,G} & 1 \\ \text{Débiles} & \text{A,T} & 0 \end{cases} \quad YR = \begin{cases} \text{Grandes} & \text{C,T} & 1 \\ \text{Pequeñas} & \text{A,G} & 0 \end{cases}$$

$$MK = \begin{cases} \text{Amínadas} & \text{A,C} & 1 \\ \text{Cetónicas} & \text{T,G} & 0 \end{cases}$$

Cuadro 2.1: Cuadro de asignación de dígitos binarios a los nucleótidos de secuencias de DNA de acuerdo con sus características.

2.2. Proceso de replicación del DNA

La doble hélice del DNA se abre como un cierre, para dar lugar a dos procesos diferentes de la vida celular. Uno de ellos es el proceso de replicación, en el cual el DNA se duplica generando dos copias idénticas (salvo por mutaciones y errores de copiado) quedando una de estas copias en cada célula producto de la división celular (ver Figura 2.4). El otro proceso, llamado de transcripción-traducción, es aquél mediante el cual la información contenida en el DNA determina la manera en que una célula produce los polipéptidos que a la larga darán lugar a las proteínas. A un segmento de DNA que contiene la información necesaria para la síntesis de una proteína se le llama *gen estructural*.

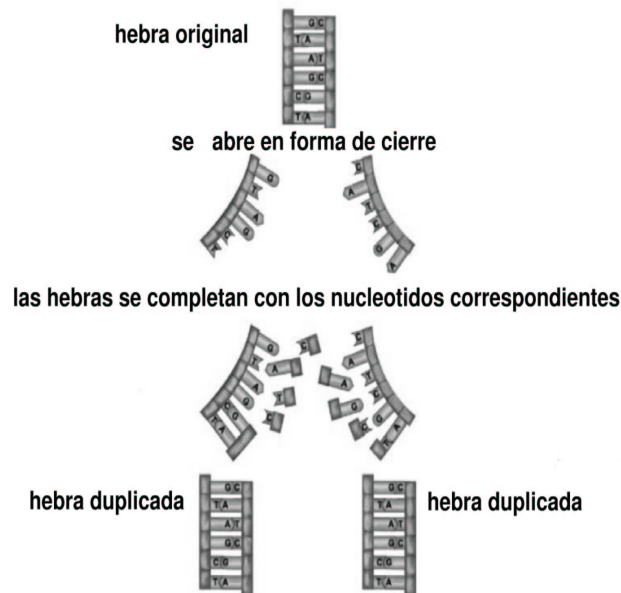


Figura 2.4: Proceso de replicación del DNA.

2.3. Algunos hechos sobre los genes

Las células se clasifican como procariontes y eucariontes; las primeras no tienen un núcleo cuya pared separe el material genético del resto del proto-

plasma, un ejemplo de ellas son las bacterias. Las células eucariontes, por otra parte, poseen un núcleo bien definido en el cual reside el DNA; todos los organismos pluricelulares están conformados por células eucariontes, en éstas, no toda la molécula de DNA está constituida por genes; de hecho, estos conforman una mínima parte de la secuencia. Los genes están separados entre sí, por largos segmentos intergénicos cuya función (si es que la tienen), no ha sido aún descubierta. Más aún, un gen está constituido por uno o varios segmentos llamados *exones* que contienen la información que codifica la producción de polipéptidos y por islas entre ellos llamadas *intrones* cuya utilidad o falta de ella tampoco se conoce a la perfección (ver Figura 2.5), esto pasa durante el proceso de transcripción en el cual se crea el RNA mensajero (RNA_m) que es el que lleva la información necesaria al ribosoma y así poder codificar los diferentes aminoácidos. Al conjunto de material genético de una célula se le llama *genoma*.

Como dato importante tenemos que un virus tiene del orden de 1000 megabases¹; los animales y plantas más comunes tienen proximadamente 2000 megabases; el humano tiene alrededor de 3000 megabases y algunos sapos y salamandras pueden llegar a tener 20000 megabases [8].

2.4. Proceso de transcripción-traducción

La manera en que se lleva a cabo el proceso de transcripción-traducción es el siguiente: en algún momento de la vida celular, el segmento del DNA correspondiente a un gen estructural (de aquí en adelante, simplemente gen) se abre como se mencionó anteriormente y en lugar de duplicarse en una molécula similar, la sucesión de bases se transcribe en una molécula de ácido ribonucleico (RNA) siguiendo el principio de complementaridad mencionado anteriormente. Se obtiene una copia de las ramas de DNA, con la única diferencia que en el RNA se reemplaza la timina (T) por uracilo (U), que también es una pirimidina con propiedades geométricas y químicas muy semejantes a las de la timina, es en esta etapa que al RNA se le conoce como RNA mensajero.

Después en las células eucariontes, con la ayuda de un sistema formado por proteínas, conocido como espliceosoma, se eliminan los segmentos intrónicos

¹Una megabase equivale a 1000 bases o nucleótidos

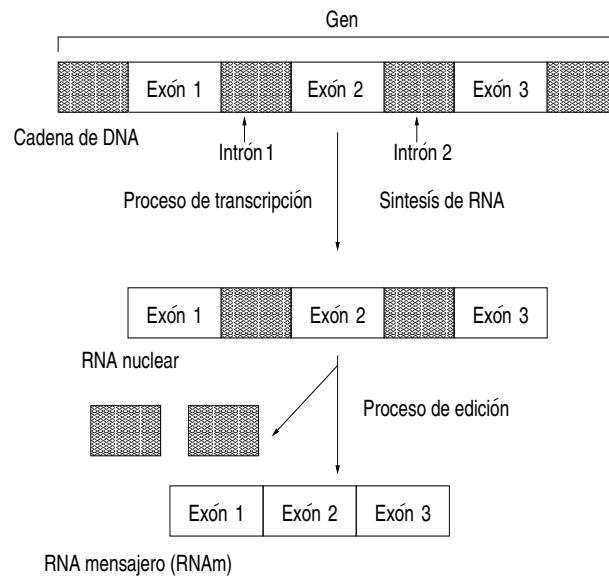


Figura 2.5: Proceso de edición que ocurre en los genes durante el proceso de transcripción para poder formar el RNA mensajero (RNA_m).

y se pegan en sucesión las partes codificadoras; este proceso se conoce como edición y es ilustrado en la Figura 2.5. Con esto termina la primera parte del proceso que es la transcripción (ver Figura 2.6).

La molécula resultante, que es el RNA mensajero, viaja a una estructura de la célula conocida como ribosoma. En ésta, se lee secuencialmente el RNA y por cada tres bases se agrega un aminoácido a una cadena lineal conocida como polipéptido que es la precursora de la proteína (ver Figura 2.7) . A cada terna de bases se le llama codón.

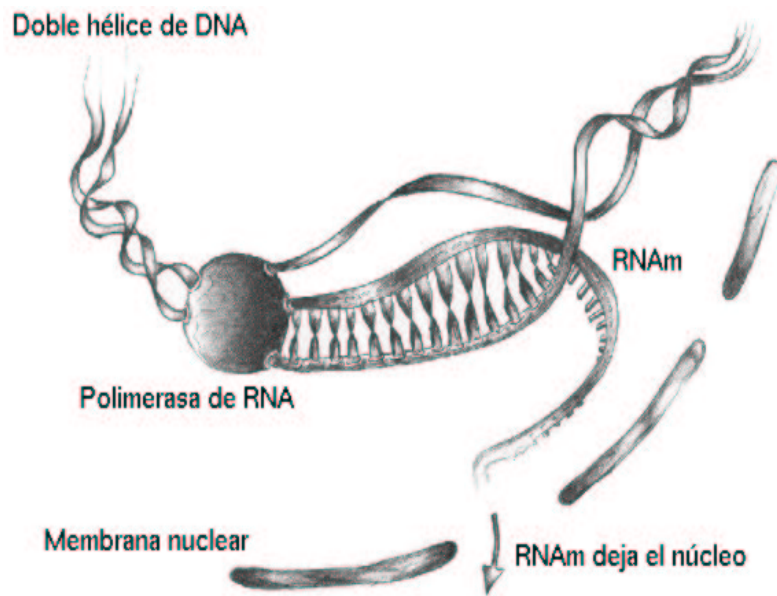


Figura 2.6: Proceso de transcripción para formar el RNA mensajero.

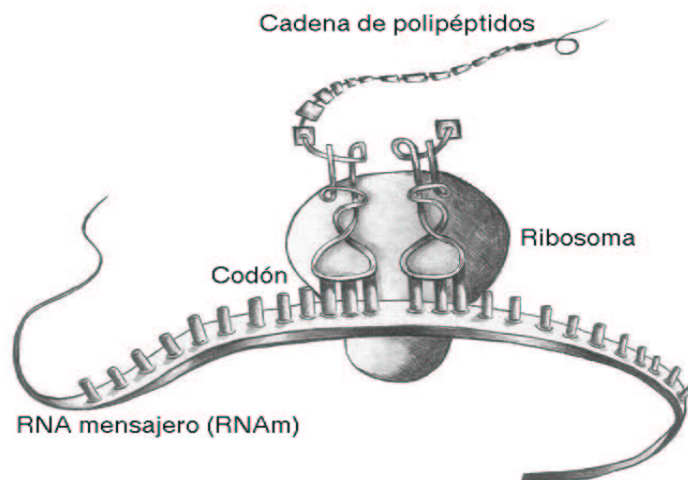


Figura 2.7: Proceso de traducción el cual a partir de codones se forman los aminoácidos.

2.5. El código genético

Dado que existen cuatro bases, pueden formarse 64 codones distintos de los cuales todos, excepto tres, dan lugar a aminoácidos. Sin embargo, en la naturaleza existen tan solo 20 aminoácidos diferentes que se emplean en la construcción de proteínas, de lo que se infiere que aunque hay codones distintos, son sinónimos para un aminoácido, en el Cuadro 2.2 se muestra la equivalencia entre secuencias de DNA que contiene las cuatro bases y los aminoácidos, a la información contenida en la tabla se le denomina *código genético* estándar, esta asociación se hace durante el proceso de traducción [7].

Una proteína tiene una configuración tridimensional que esta dada por una secuencia de aminoácidos. La función de una proteína depende de su estructura, por lo que es necesario saber qué forma tiene para predecir su funcionamiento.

2.6. Mutaciones

Durante los procesos anteriores, puede suceder que, debido a un error en la duplicación o en la transcripción, se cambie una base por otra; a este fenómeno se le llama sustitución o también mutación (ver Figura 2.8). La replicación de DNA tiene que ser extremadamente exacta para evitar la introducción de mutaciones, es decir el cambio de una base por otra, en las copias del genoma. Algunas mutaciones, pueden ocurrir como un error de replicación o debido a los efectos químicos y físicos que alteran directamente la estructura química del DNA. Existen varios tipos de mutaciones, pero el más común es el que mantiene invariante el tamaño de la molécula mutada, es decir, que mantiene su condición de purina o pirimidina en la sustitución. Estas mutaciones pueden ser corregidas por enzimas correctoras del DNA; aquellas mutaciones que escapan del proceso de corrección llegan a tener características permanentes en su descendencia. Las mutaciones son una fuente de variabilidad en los organismos que les otorga la plasticidad necesaria para que éstos, en su momento, respondan ante presiones selectivas, dando lugar a la evolución biológica.

	T	C	A	G	
T	PHE	SER	TYR	CYS	T
T	PHE	SER	TYR	CYS	C
T	LEU	SER	STOP	STOP	A
T	LEU	SER	STOP	TRP	G
C	LEU	PRO	HIS	ARG	T
C	LEU	PRO	HIS	ARG	C
C	LEU	PRO	GLN	ARG	A
C	LEU	PRO	GLN	ARG	G
A	ILE	THR	ASN	SER	T
A	ILE	THR	ASN	SER	C
A	ILE	THR	LYS	ARG	A
A	MET	THR	LYS	ARG	G
G	VAL	ALA	ASP	GLY	T
G	VAL	ALA	ASP	GLY	C
G	VAL	ALA	GLU	GLY	A
G	VAL	ALA	GLU	GLY	G

Cuadro 2.2: Código genético estándar. El primer nucleótido del triplete es indicado por la primera columna del lado izquierdo, el segundo es indicado por las cuatro columnas centrales y el último es indicado por la última columna de la derecha. Aminoácidos codificados por cada codón del ARN mensajero. ALA: Alanina, CYS: Cisteína, ASP: Ácido aspártico, GLU: Ácido glutámico, PHE: Fenilalanina, GLY: Glicina, His: Histidina, ILE: Isoleucina, LYS: Lisina, LEU: Leucina, MET: Metionina, ASN: Asparagina, PRO: Prolina, GLN: Glutamina, Arg: Arginina, SER: Serina, TMR: Treonina, VAL: Valina, TRP: Triptofano, TYR: Tirosina, y STOP denota los aminoácidos Pardo, Ámbar y Ocre que no codifican pero que sirven como signos de puntuación para el fin del mensaje.

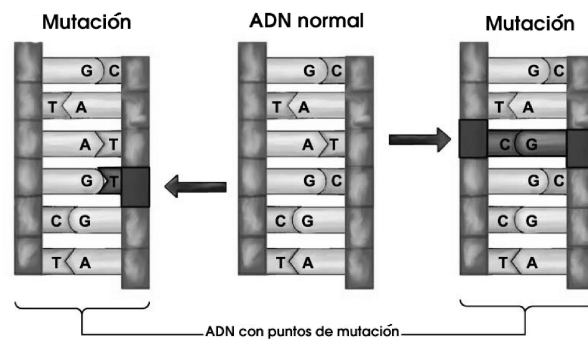


Figura 2.8: Fenómeno de mutación, puede ocurrir durante los procesos de replicación o de transcripción.

Capítulo 3

El juego del caos

Si pensamos en algo aleatorio asociado a una imagen, podríamos imaginar un lienzo de pintura lleno de manchas de colores cubriéndolo por completo y sin ningún orden o patrón específico.

También se puede sugerir el siguiente ejemplo; el movimiento browniano, descubierto por el botánico Robert Brown, aproximadamente en 1827 [19]. Brown se dió cuenta que las partículas de polen que había metido en un frasco, repentinamente cambiaban de dirección, esto debido al choque del polen con las partículas de aire. Podemos describir este choque de partículas dibujando la trayectoria de una de ellas en el plano, lo que tenemos que hacer es tomar un punto en el plano, escoger cualquier dirección al azar, caminar una cierta distancia y parar, después escoger otra dirección al azar, volver a recorrer una distancia y parar; si repetimos este procedimiento por algún tiempo, veremos que la trayectoria de nuestra partícula no se aleja mucho de los primeros trazos, por lo que llena densamente una zona del plano pero sin ningún patrón reconocible (ver Figura 3.1).

Estos ejemplos muestran que a partir de una conducta azarosa no se obtiene ningún patrón, pero no siempre es así. Se estudiará el *juego del caos* en el cual se pueden obtener patrones bien definidos de una conducta totalmente aleatoria.

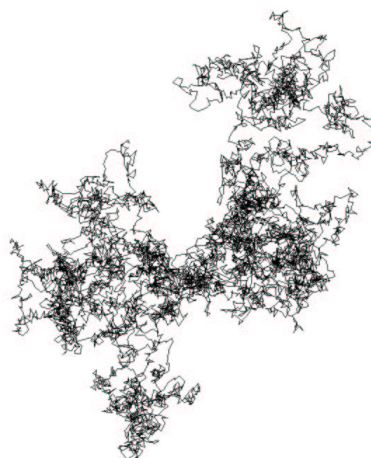


Figura 3.1: Movimiento browniano sobre el plano.

3.1. Reglas básicas del juego del caos

El juego del caos es un proceso iterativo totalmente aleatorio el cual puede producir imágenes perfectamente ordenadas y determinísticas; este juego lo inventó Michael F. Barnsley [4], el juego consiste en tener tres puntos fijos, los cuales son los vértices de un triángulo, se asocia a cada vértice una etiqueta de identificación. La probabilidad con que podemos escoger cada una de estas es $1/3$; es decir tenemos la misma posibilidad de escoger cualquiera de las tres, después de haber iniciado el juego con un punto x_0 como condición inicial, la sorpresa que obtenemos es el triángulo de Sierpinski, generado a partir de un proceso totalmente aleatorio.

Para empezar el juego se necesita un dado de seis caras, en donde dos caras están marcadas con el número 1, otras dos con el número 2 y las dos últimas con el número 3, este dado nos servirá como una máquina generadora de números aleatorios, con esto nos aseguramos que la posibilidad de que caiga cualquiera de los tres números sea la misma, es decir $1/3$.

Ahora se escogen tres puntos que pertenecen al plano, éstos servirán de vértices para formar un triángulo¹, asignaremos a cada uno de los vértices una

¹Para obtener un mejor resultado los vértices deben formar un triángulo equilátero.

etiqueta con los números 1, 2 y 3 respectivamente. Después se selecciona un punto en el plano de preferencia en el interior del triángulo, este punto nos servirá de condición inicial y lo etiquetaremos como x_0 , ahora hacemos uso de nuestro dado, supongamos que el número generado después de haber tirado el dado es 3, entonces obtenemos el punto medio entre el punto x_0 y el vértice del triángulo marcado con el número tres, este nuevo punto es etiquetado como x_1 , este es nuestro nuevo punto para seguir jugando, de nuevo usamos el dado que dará como resultado un número entre 1 y 3 para después calcular el punto medio entre el punto x_1 y el vértice elegido correspondiente al número que resultó de haber tirado el dado. Este proceso se repite durante un número muy grande de veces. De manera general, supongamos que hemos jugado k veces, entonces se ha generado una sucesión que se compone de k puntos $\{x_1, x_2, \dots, x_k\}$. Cada vez que se utiliza el dado se obtiene un número n , con $n \in \{1, 2, 3\}$, esto genera un nuevo punto x_{k+1} , que está situado en el punto medio de x_k y el vértice marcado con la etiqueta n . Después de todo este proceso totalmente hecho al azar, se esperaría que la sucesión de puntos tuvieran una distribución uniforme sobre el interior del triángulo, pero la distribución de los puntos está lejos de parecerse a este comportamiento. Si al término del juego, se dibuja la sucesión de puntos, la figura que se obtiene es el conocido triángulo de Sierpinski (ver Figura 3.2). El triángulo de Sierpinski puede ser mejor apreciado si el número de veces que jugamos es grande, debido a que esta figura es el *atractor del sistema*, es decir que para cualquier condición inicial dada, siempre tendremos el mismo comportamiento. El triángulo de Sierpinski es uno de los ejemplos más representativos de un *fractal*.

3.2. Fractales

El concepto de fractal se ha venido construyendo desde finales del siglo XIX y principios del siglo XX, pero tuvo grandes problemas por la introducción de dimensiones fraccionarias, lo que causó que muchos matemáticos no tomaran el concepto de forma seria. Este fue retomado por Benoit Mandelbrot [15], que propuso el nombre para estas figuras con dimensión fraccionaria, las llamo *fractales*, término que proviene del adjetivo latino *fractus*, que significa irregular o interrumpido. Mandelbrot afirma que las figuras geométricas convencionales pueden representar los objetos hechos por el hombre y que la geometría fractal es la forma para representar los objetos hechos por la naturaleza. En los últimos años, con tantos avances en los equipos de cómputo, nos damos cuen-

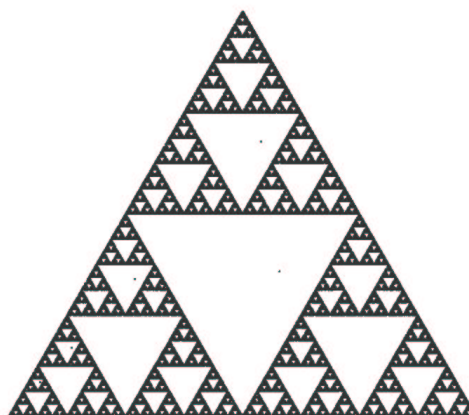


Figura 3.2: Triángulo de Sierpinski, generado por el juego del caos.

ta que la idea de Mandelbrot no es tan descabellada, ya que por medio de la computadora podemos observar imágenes de calidad artística y hermosos paisajes hechos por la composición de diferentes fractales. Más aún, podemos observar comportamientos fractales en diferentes dinámicas utilizadas para la descripción de diferentes fenómenos en la naturaleza; por ejemplo, el sistema de Edward N. Lorenz [10], que inició como un modelo para predecir el clima; o el comportamiento de la ecuación logística que es un modelo que se utiliza para describir el crecimiento de una población; éstos son sólo dos ejemplos de muchos que tratan de modelar fenómenos en la naturaleza. Por otro lado, en el juego del caos se pueden observar fractales como el triángulo de Sierpinski y veremos después que éstos son una herramienta importante para la compresión de imágenes.

Hemos dicho que una característica de las figuras fractales es su dimensión fraccionaria, pero la idea principal es la propiedad de autosemejanza a cualquier escala, es decir que si ampliamos alguna parte de la figura, podremos encontrar en esa porción el total de ella y si ampliamos alguna parte de la porción, se encontrará nuevamente el total, y así sucesivamente, lo que hace que los fractales tengan implícito un proceso de autosemejanza infinito.

Muchos fractales se construyen por medio de procesos de iteración; es decir, que a partir de una figura geométrica simple y haciendo una repetición de

ella, se consigue una figura de gran complejidad iterando un número suficientemente grande de veces.

3.3. Mapeos como generadores de sucesiones

Como vimos anteriormente, para jugar al juego del caos necesitamos de la ayuda de una máquina generadora de números aleatorios, en cualquier lenguaje de programación se tienen funciones predeterminadas que generan números pseudo-aleatorios, pero así como podemos utilizar las funciones determinadas de un lenguaje de programación podemos utilizar algún mecanismo diferente para generar tales números. Heinz-Otto Peitgen [19] utilizó los números generados por el mapeo logístico para el juego del caos, se quería obtener de nuevo el triángulo de Sierpinski, también utilizó diferentes mapeos que algunas veces generaban patrones fractales y otras veces daban lugar a distribuciones de puntos sin algún patrón discernible. La distribución de puntos que resulta de la ecuación logística, es distinta a la distribución del triángulo de Sierpinski. A continuación entraremos en detalle de como utilizar los diferentes mapeos dentro del juego del caos.

Utilizaremos el mapeo logístico en su régimen caótico total, es decir, cuando $\lambda = 4$.

$$x_{k+1} = \lambda x_k(1 - x_k) \quad (3.1)$$

Este mapeo produce números entre 0 y 1; se comienza por dividir el intervalo en 3 partes iguales, es decir 1 : $[0, 1/3)$, 2 : $[1/3, 2/3)$, 3 : $[2/3, 1]$, a cada intervalo se le asigna un vértice del triángulo en el que se aplicará el juego del caos. Se genera una sucesión de k números provenientes de la ecuación logística, a cada elemento en la sucesión se le asocia el número del intervalo al que pertenece, entonces se calcula el punto medio entre una condición inicial dada x_0 y el vértice del triángulo que corresponde según la sucesión. Se repite este proceso hasta terminar con la sucesión de números, es decir k veces. El resultado, como se dijo antes, está lejos de ser el triángulo de Sierpinski. Esto se ha realizado también para el mapeo tienda de apache (Ec.(3.2)) y el mapeo diente de sierra (Ec. (3.3)); los resultados han sido diferentes distribuciones de puntos que están lejos de formar el triángulo de Sierpinski; sin embargo, la

dinámica obliga a que estos puntos se encuentren en el atractor del sistema. Los cúmulos de puntos se agrupan de manera autosemejante. (ver Figuras 3.3 y 3.4).



Figura 3.3: Juego del caos para el mapeo logístico.

Se ha intentado obtener diferentes distribuciones de puntos incrementando el número de vértices de la figura geométrica; existe una técnica empleada por Takashi Tsuchiya, de la Universidad de Meisei en Tokio [29] que da resultados sorprendentes, utilizando los mapeos antes mencionados como generadores de números. Se encontró que para cada mapeo existe un atractor particular; esto se hace mediante la generalización del juego del caos, es decir, aumentando el número de vértices del triángulo, jugando así en un cuadrado, en un pentágono, en un hexágono y así hasta llegar a un polígono de suficientes lados que se pueda considerar como aproximación de una circunferencia; esto es, en esencia, el juego del caos circular, que se explicará con detalle en la próxima sección. En las Figuras 3.5 y 3.6 se pueden ver los atractores del juego del caos en el cuadrado utilizando los mapeos de las ecuaciones (3.1), (3.2) y (3.3).

$$x_{n+1} = \begin{cases} 2x_n & \text{si } 0 \leq x \leq \frac{1}{2} \\ 2 - 2x_n & \text{si } \frac{1}{2} < x \leq 1 \end{cases} \quad (3.2)$$

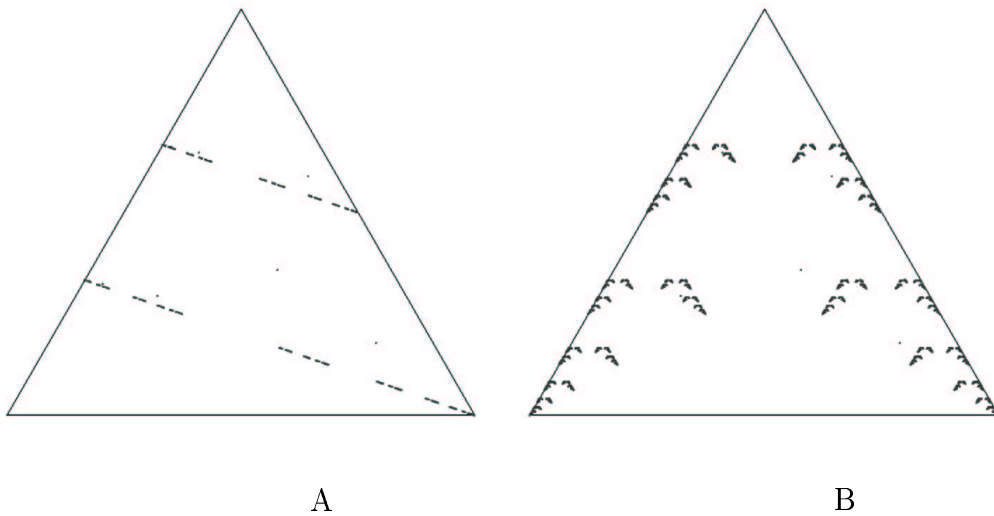


Figura 3.4: (A) Juego del caos para el mapeo tienda de apache y (B) juego del caos para el mapeo diente de sierra.

$$x_{n+1} = \begin{cases} 2x_n & \text{si } 0 \leq x \leq \frac{1}{2} \\ 2x_n - 1 & \text{si } \frac{1}{2} < x \leq 1 \end{cases} \quad (3.3)$$

3.4. El juego del caos circular

El juego del caos circular es una generalización del juego propuesto por Barnsley, esta forma de hacer el juego fue utilizada por Tsuchiya como técnica para tener una visión cualitativa del comportamiento en régimen caótico de algunos mapeos, Tsuchiya lo probó con mapeos tales como la ecuación logística, el mapeo tienda de apache y el mapeo diente de sierra. Lo que se ha probado con esta generalización, es que cada mapeo puede tener un atractor que lo puede distinguir de los otros, es decir funciona como una huella digital para cada mapeo.

La generalización del juego del caos, comienza por incrementar el número de vértices, de tres vértices se incrementa a cuatro, después a cinco, etc.; se ha visto que al utilizar el mapeo logístico (Eq. (3.1)), el mapeo tienda (Eq. (3.2)) y el mapeo diente de sierra (Eq. (3.3)) los tres vértices no son

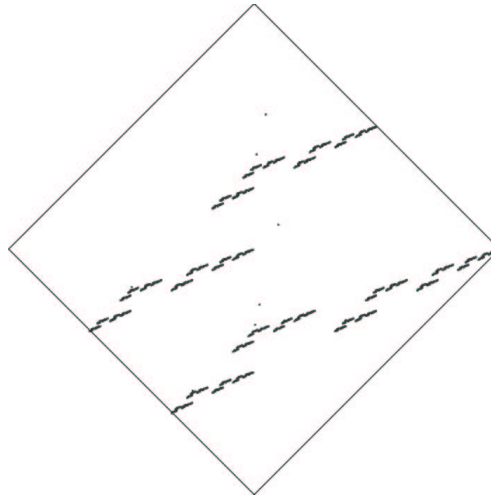


Figura 3.5: Juego del caos para el mapeo logístico en el cuadrado.

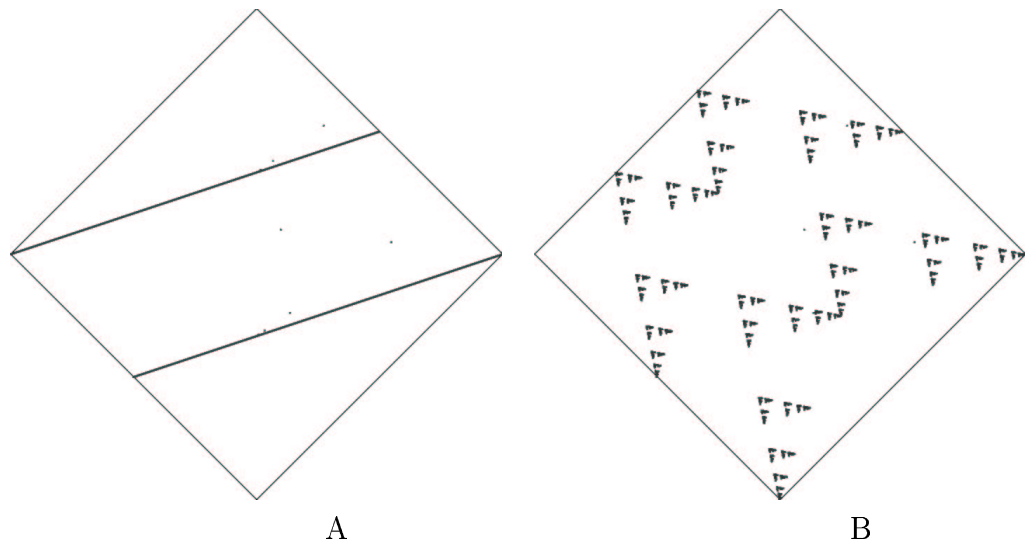


Figura 3.6: Mapeos en el cuadrado. (A) Juego del caos para el mapeo tienda de apache y (B) para el mapeo diente de sierra.

suficientes para observar el atractor del sistema; al incrementar el número de vértices, se converge al atractor del juego del caos circular correspondiente a

cada sistema; las figuras 3.5 y 3.6 muestran el juego del caos para el número de vértices igual a cuatro.

La generalización se hizo como sigue; tenemos V vértices para comenzar el juego, entonces se tiene que dividir el intervalo $[0, 1]$ en V subintervalos cada uno etiquetado con un número desde 1 hasta V respectivamente de la siguiente manera

$$1 : [0, \frac{1}{V}), \quad 2 : [\frac{1}{V}, \frac{2}{V}), \quad \dots, \quad V : [\frac{V-1}{V}, 1]$$

El paso siguiente es escoger una condición inicial x_0 y obtener una sucesión de números proveniente de algún mapeo que nos ayudará a seleccionar el vértice, dependiendo de a qué intervalo pertenezca. Como en el juego de Barnsley, se calcula la distancia entre el i -ésimo vértice seleccionado y la condición inicial x_0 , se obtiene un nuevo punto x_1 ; en la siguiente iteración se vuelve a seleccionar otro vértice y se calcula la distancia entre el nuevo vértice y el punto x_1 ; este proceso se repite un número suficientemente grande de veces. Al final tendremos el atractor del juego del caos para V vértices; si V es suficientemente grande, se tendrá el atractor del juego del caos circular, el cual es único para cada mapeo. En las Figuras 3.7 a la 3.9 se puede ver el comportamiento de las sucesiones de números generadas por diferentes mapeos. En la sección anterior se generó para $V = 4$; al incrementar el número de vértices a $V = 16$ se observa una tendencia de los puntos a agruparse sobre una figura (ver Figuras 3.7, 3.8 y 3.9) y al ir incrementando el número de vértices, el atractor correspondiente al mapeo utilizado se hace visible; de hecho, es suficiente con $V = 64$ para poder apreciar el atractor del juego del caos circular, cuando $V = 1024$ el atractor se aprecia mejor (ver Figuras 3.7, 3.8 y 3.9) .

3.5. El *shift* de Bernoulli y el juego del caos circular

Para propósitos de este trabajo se ha utilizado el mapeo shift o corrimiento de Bernoulli [19] que es la versión binaria del mapeo diente de sierra (Ec. (3.3)). El mapeo shift provee un atractor que es único, independientemente de la condición inicial utilizada; la densidad de puntos sobre el atractor cambia

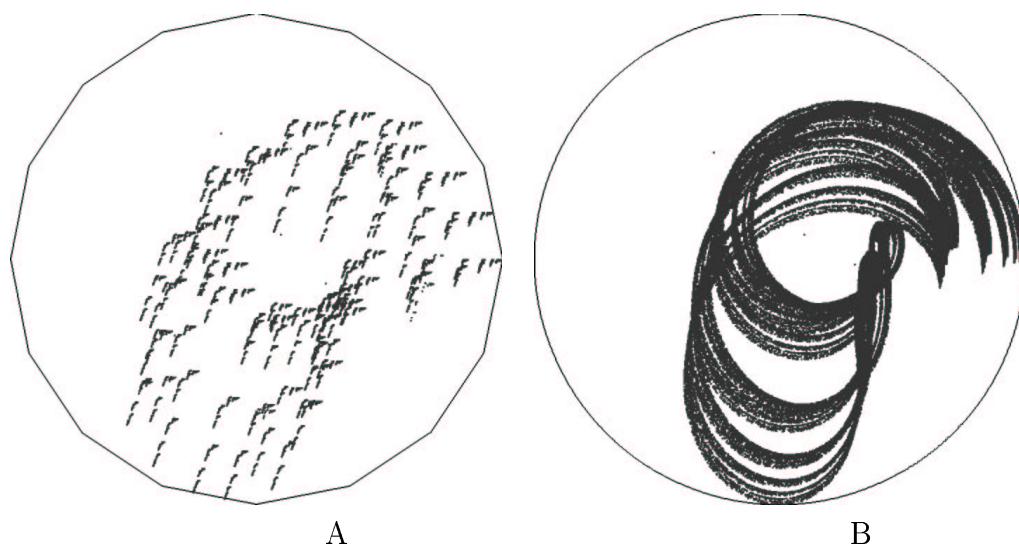


Figura 3.7: Juego del caos con el mapeo logístico, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.

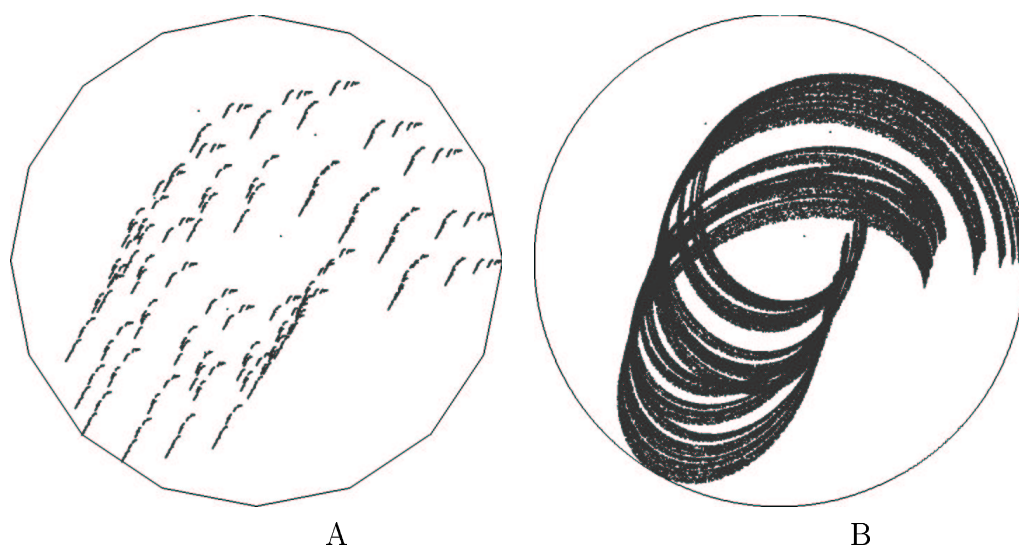


Figura 3.8: Juego del caos con el mapeo tienda de apache, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.

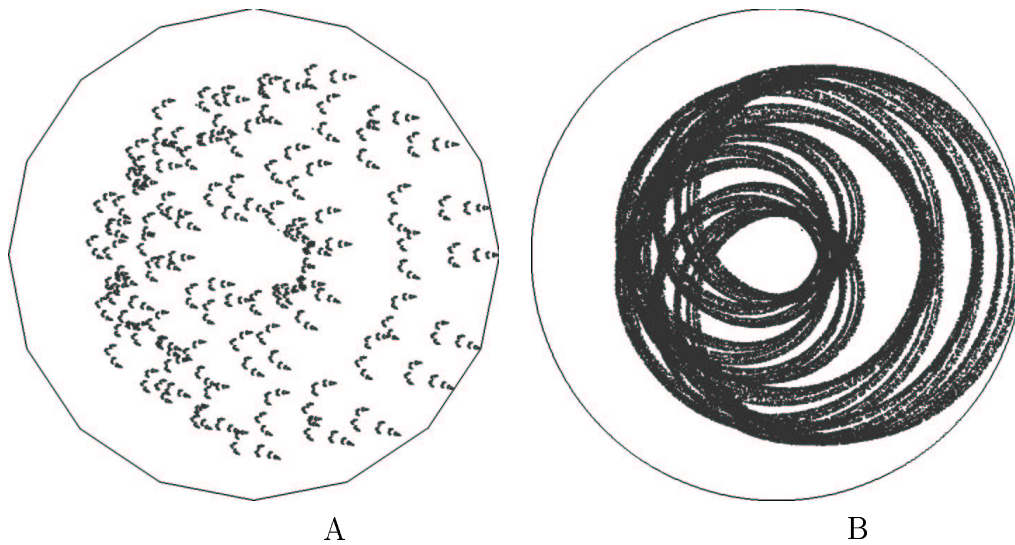


Figura 3.9: Juego del caos con el mapeo diente de sierra, (A) con un polígono de 16 lados y (B) con un polígono de 1024 lados.

dependiendo de la secuencia de números sobre la que se utiliza el mapeo. Se han hecho pruebas sobre secuencias de números totalmente aleatorias. El objetivo de este trabajo es obtener información de las secuencias de DNA en forma binaria² utilizando el juego del caos circular junto con el mapeo shift.

A continuación se explicará cómo obtener la representación binaria del mapeo diente de sierra y, de esta forma, poder utilizarlo en el juego del caos.

Para comenzar, se utiliza la función parte fraccionaria de un número $Frac(x)$ que se define como sigue:

$$Frac(x) = x - k \text{ si } k \leq x < k + 1, k \text{ entero} \quad (3.4)$$

esta ecuación elimina la parte entera de un número dejando sólo su parte fraccionaria, por ejemplo $Frac(5.98345) = 0.98345$. Llamaremos $S(x)$ al mapeo (3.3) que se puede escribir en términos de la función $Frac(x)$ como sigue:

$$S(x) = Frac(2x) \text{ para } 0 \leq x < 1. \quad (3.5)$$

²La forma binaria del DNA se obtiene a partir de las relaciones vistas en el capítulo 2.

Este mapeo toma valores entre $[0, 1)$, esto es posible ya que la primera parte del mapeo diente de sierra (3.3) cubre el intervalo $0 \leq x < 1/2$, entonces $0 \leq S(x) = 2x < 1$ y $Frac(2x)$ dan el mismo resultado, por otro lado si $1/2 \leq x < 1$ se tiene que $0 \leq S(x) = 2x - 1 < 1$ por lo que $2x - 1$ se comporta de igual forma que $Frac(2x)$ entonces $Frac(2x) = 2x - 1$, con esto se cubre el intervalo $[0, 1)$ obteniendo una equivalencia entre $S(x)$ y $Frac(2x)$.

Los valores que toma la ecuación 3.5 se encuentran en el intervalo $[0, 1)$ por lo que son de la forma $0.a_0a_1a_2\dots$

La forma de convertir un número binario $b_0b_1b_2\dots b_k$ donde $b_i \in \{0, 1\}$ a su representación decimal fraccionaria es multiplicando cada uno de sus términos de la siguiente manera

$$x = b_0\frac{1}{2} + b_1\frac{1}{2^2} + b_2\frac{1}{2^3} + \dots = 0.a_0a_1a_2\dots, \quad (3.6)$$

multiplicando la ecuación (3.6) por 2 se hace un corrimiento de un dígito hacia la izquierda

$$2x = b_0 + b_1\frac{1}{2} + b_2\frac{1}{2^2} + \dots = a_0.a_1a_2\dots, \quad (3.7)$$

entonces si se aplica la función $S(x)$ a estos números se recorre un dígito hacia la izquierda del punto decimal y se desecha la parte entera como sigue:

$$x = b_0\frac{1}{2} + b_1\frac{1}{2^2} + b_2\frac{1}{2^3} + \dots \Rightarrow S(x) = b_1\frac{1}{2^1} + b_2\frac{1}{2^2} + \dots \quad (3.8)$$

Esto se ha utilizado para poder introducir secuencias en el juego del caos, de la siguiente manera; se toman las secuencias de DNA en forma binaria, para empezar a jugar con las secuencias dependemos del número de lados del polígono como sigue: si tenemos un polígono de 8 vértices entonces tenemos que tomar cadenas de tres en tres dígitos, cada una de estas pequeñas cadenas forman a su vez un número en base decimal en donde el más pequeño que podemos representar es el número cero $0 = 000$ y el mayor es $7 = 111$ cada uno de estas secuencias indican el vértice hacia donde se dirigirá, entonces la relación entre el número de vértices y tamaño de la cadena que se toma es $V = 2^3$ en donde 3 es la longitud de la cadena y $V = 8$ es el número de vértices del polígono; entonces, cuando tomamos cadenas de 4 dígitos, el polígono tiene $V = 2^4 = 16$ vértices; en general, tenemos que el número de vértices de un polígono está dado por $V = 2^n$ en donde n es la longitud de

la cadena que se toma. El procedimiento que se acaba de describir utiliza el mapeo shift de la siguiente manera, una vez leída la cadena que indica el vértice se calcula el punto medio entre la condición inicial y el vértice correspondiente, el corrimiento desecha el primer dígito a la izquierda y se vuelve a construir el siguiente número en forma decimal; por ejemplo, si la secuencia inicial es $10111\dots$ y tenemos un polígono de 8 lados se toman los tres primeros dígitos que construyen el número decimal $101 = 5$ después se aplica el shift y se desecha el primer dígito de la izquierda que en este caso es 1 y la secuencia restante es $0111\dots$; de nuevo, se vuelven a tomar los tres primeros elementos y se calcula su número decimal correspondiente $011 = 6$ y se desecha el 0 y así, la cadena de nuevo se recorre hacia la izquierda quedando $111\dots$, este proceso se repite hasta el final de la cadena (ver Figura 3.10).

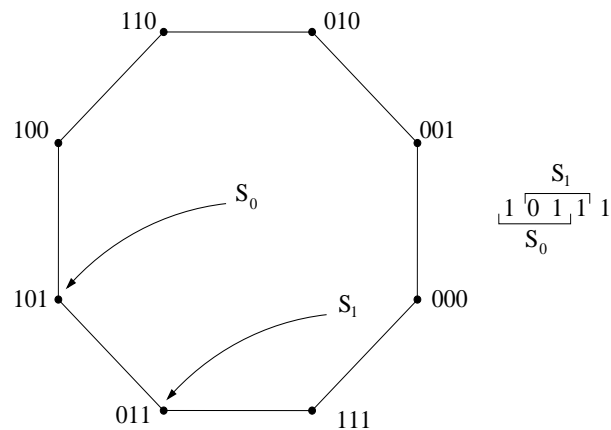


Figura 3.10: En la figura podemos ver un polígono con 8 vértices, y se muestra una cadena binaria de entrada, cada vértice se encuentra etiquetado por secuencias binarias que corresponden a un número decimal entre 0 y 7, en sentido contrario a las manecillas del reloj, se puede ver cómo se aplica el shift de Bernoulli a la secuencia de entrada que indica en qué orden se van tomando los vértices.

3.6. Jugando con nucleótidos

Considerando las secuencias de DNA como sucesiones de letras que representan a los cuatro nucleótidos —adenina (A), guanina (G), citocina (C) y timina (T)— se pueden generar imágenes de gran complejidad utilizando el juego del caos en un cuadrado. Se comienza por etiquetar cada vértice del cuadrado con una letra del alfabeto $\{A, G, C, T\}$, el orden en que se acomoden repercutirá en la distribución de los puntos. Las reglas vistas anteriormente se conservan; lo que ahora decidirá qué vértice debemos elegir, será la letra que indica la secuencia de DNA escogida, es decir si tenemos la secuencia AAGTC tomaremos un punto inicial x_0 dentro del cuadrado y como la primera letra de la secuencia es A entonces el segmento de recta que tomamos es el que va del punto inicial x_0 al vértice correspondiente a la letra A, se calcula el punto medio y se etiqueta como x_1 , se tomará el punto medio entre x_1 y la siguiente letra que indica la secuencia, esto se repite hasta que la secuencia se termine, el resultado es una imagen en la cual podemos ver repetidas veces la formación de cuadrados dentro de otros cuadrados, esto sugiere la relación de la imagen con un fractal (ver Figura 3.12).

En la Figura 3.11 se muestra el ejemplo del juego del caos utilizando una secuencia de letras generada aleatoriamente por una computadora, la imagen muestra una distribución uniforme de los puntos y no se distingue ningún patrón ordenado. Barnsley [4], hace un análisis al juego del caos relacionado con los *sistemas de funciones iteradas con probabilidades*, en estos sistemas, a cada vértice del cuadrado se le asigna una probabilidad diferente de ocurrencia, las probabilidades escogidas correctamente generan figuras con distribuciones parecidas a la de la Figura 3.12. Podemos darnos cuenta de los patrones maravillosos que las secuencias de DNA pueden generar en el juego del caos, podemos decir que la probabilidad con que aparece una letra en una cadena tuvo que ser escogida bajo cierto criterio, con esto nos damos cuenta que podemos aprovechar la información que nos brinda el juego del caos utilizando secuencias de DNA.

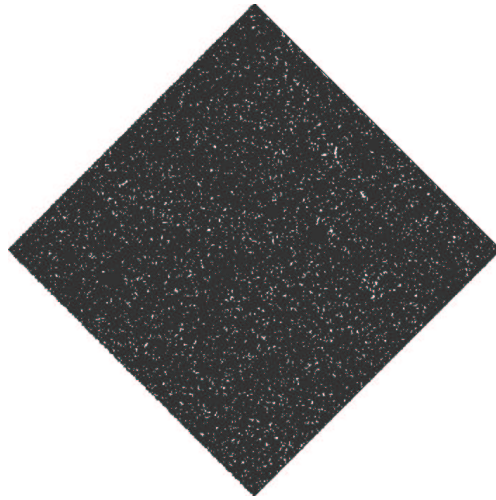


Figura 3.11: Juego del caos en el cuadrado utilizando una secuencia de letras generadas aleatoriamente.

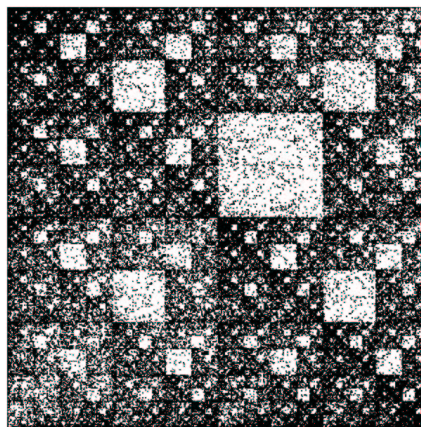


Figura 3.12: Juego del caos en el cuadrado utilizando la cadena de DNA del *Homo sapiens*. A cada vértice se le asigna un nucleótido, comenzado por el vértice superior izquierdo en dirección contraria a las manecillas del reloj se asignan A, G, C y T respectivamente.

Capítulo 4

Sistemas de funciones iteradas

En la sección anterior se generaron estructuras específicas que pertenecen a una gran familia de fractales con la simple regla de tomar el punto medio entre el vértice correspondiente del polígono y el punto que se determinó en el paso anterior. Ahora se generarán aplicando un conjunto de *transformaciones afines contractivas* a un conjunto arbitrario de \mathbb{R}^2 .

Los sistemas de funciones iteradas se componen de un conjunto de transformaciones afines que trabajan sobre un espacio métrico el cual provee información de la naturaleza y la forma del conjunto al que le estamos aplicando las transformaciones.

En los siguientes apartados se explicará el concepto geométrico de una transformación afín y de un espacio métrico ya que son necesarios para entender la teoría de los sistemas de funciones iteradas.

4.1. Espacio métrico y transformaciones

Los sistemas de funciones iteradas se desarrollan en un entorno que para los propósitos de este trabajo, será el *plano euclidiano* conocido también como \mathbb{R}^2 ; se definirá una *métrica o una medida de distancia* en este plano; puede ser, por ejemplo, la distancia euclidiana o cualquier otra que cumpla con las propiedades que se definirán. Así, con una medida de distancia y un entorno, se tiene un *espacio métrico*.

Denotemos al espacio métrico sobre el plano euclidiano junto con una función de distancia como (\mathbb{R}^2, d) donde la función $d : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ mide la distancia entre pares de puntos x, y pertenecientes al plano euclidiano. La función d debe cumplir con las siguientes propiedades y es aplicable a todo par de puntos sobre el plano.

1. La distancia que hay entre el punto x al punto y es la misma que de y a x , $d(x, y) = d(y, x)$.
2. La distancia entre cualquier par de puntos diferentes debe ser mayor que cero, $0 < d(x, y) < \infty$.
3. La distancia entre cualquier par de puntos iguales es cero, $d(x, x) = 0$.
4. La desigualdad del triángulo $d(x, y) \leq d(x, z) + d(z, y)$.

Entonces decimos que la función d es una métrica sobre el espacio \mathbb{R}^2 .

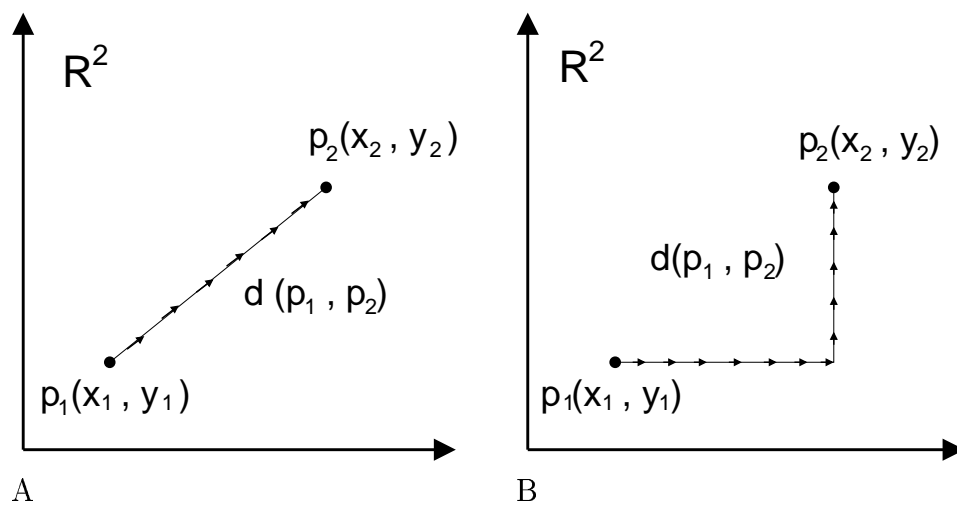


Figura 4.1: Ejemplos de métricas para el plano euclidiano entre los puntos p_1 y p_2 .

En la Figura 4.1 (A) se muestra el espacio métrico

$$(\mathbb{R}^2, d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2})$$

en donde d es la distancia euclidiana, en (B) se muestra el mismo espacio utilizando la métrica de Manhattan

$$(\mathbb{R}^2, d(p_1, p_2) = |x_1 - x_2| + |y_1 - y_2|).$$

Definición 1 a) Una transformación en \mathbb{R}^2 es una función $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$.
 b) Si S es un subconjunto de \mathbb{R}^2 , entonces, $f(S) = \{f(x) : x \in S\}$ es la imagen de S bajo f .
 c) La función f es uno-a-uno $f(x) = f(y)$ implica $x = y$.
 d) La función f es sobre si $f(\mathbb{R}^2) = \mathbb{R}^2$.
 e) La función f es llamada invertible si es uno-a-uno y sobre; en este caso es posible definir una transformación $f^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, llamada inversa de f , tal que $f^{-1}(y) = x$ donde x es el único punto tal que $f(x) = y$.

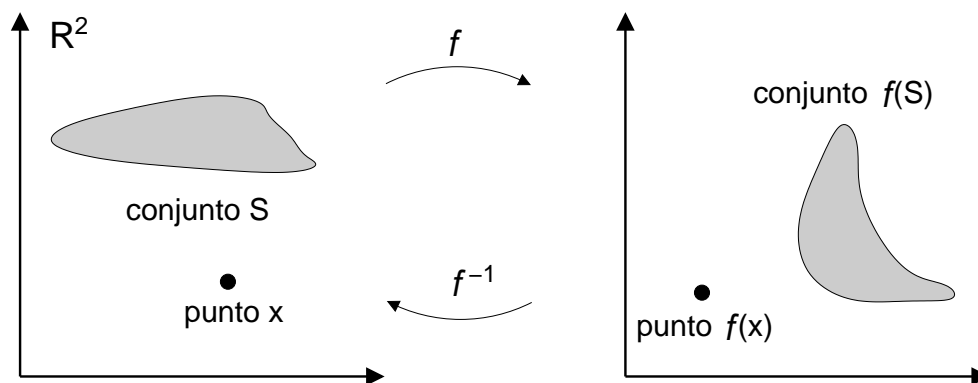


Figura 4.2: Transformaciones f y f^{-1} en el plano euclidiano. Se ilustra cómo se modifica el conjunto S al aplicarle una transformación f ; así mismo, si aplicamos la transformación inversa al conjunto $f(S)$ obtenemos el conjunto original S ; también se ejemplifica con un solo punto.

Ahora se definirán un par de conceptos más que son las iteraciones *delantera* y *trasera* de las transformaciones; esto nos servirá más adelante para explicar los sistemas de funciones iteradas.

Definición 2 Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ una transformación sobre \mathbb{R}^2 . a) La iteración delantera de f son las transformaciones $f^n : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, para

$n = 0, 1, 2, \dots$ b) Si f es invertible, entonces la iteración trasera de f son las transformaciones $f^{\circ(-m)} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ para $m = 1, 2, 3, \dots$ (note que no se incluye el cero).

Cuando se hace la *composición* de estas transformaciones quiere decir que se aplican a un conjunto de puntos en el plano (o solo un punto) tantas veces como se indique, si se tiene n igual a cero, $f^{\circ 0}$ es la identidad y $f^{\circ 0}(S) = S$, se mantiene el mismo conjunto, es decir, la transformación no se aplica, si se tiene n igual a 1, la transformación se aplica una vez al conjunto de puntos, y así sucesivamente. Al aplicar la transformación inversa se regresa iteración tras iteración al conjunto original.

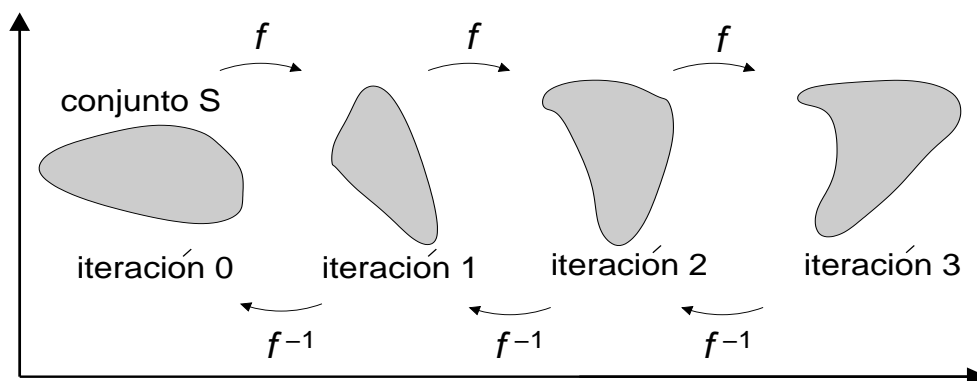


Figura 4.3: Composición de transformaciones.

En la Figura 4.3 se muestran tres iteraciones para un conjunto de puntos arbitrario, cuando se aplican todas las transformaciones delanteras se llega al conjunto marcado como tercera iteración $f^{\circ 3}(S)$; bajo cada iteración las imágenes $f(S)$ son distintas del conjunto original S sobre el cual se aplican. Ahora, si se aplican las iteraciones traseras, se regresa al conjunto inicial S .

En la siguiente subsección se estudiará un tipo de transformaciones con una estructura particular que denominaremos *transformaciones afines*.

4.2. Transformaciones afines

Las *transformaciones afines o lineales*, como se verá, son las funciones que, al ser aplicadas sobre un conjunto de puntos, generan fractales. Las transfor-

maciones son la parte más importante de los sistemas de funciones iteradas, ya que con las transformaciones correctas, podemos generar patrones complicados de gran semejanza con los que se encuentran en la naturaleza como nubes, montañas, bosques o flores.

Los cuatro tipos de transformaciones afines básicas son: traslación, escalamiento, reflexión y rotación; y son transformaciones lineales. Son importantes por su simplicidad matemática, pueden expresarse mediante la multiplicación de una matriz por un vector (para hacer esto se eleva la dimensionalidad de la matriz a \mathbb{R}^3) o mediante la multiplicación de una matriz por un vector, más otro vector. Otro rasgo importante es, como vimos anteriormente, que podemos componer varias transformaciones; por ejemplo, una rotación con una traslación, para formar una sola transformación que hace exactamente lo mismo que las transformaciones originales pero en un solo paso.

Formalmente, una transformación afín es una función $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ de la forma

$$w(x, y) = (ax + by + e, cx + dy + f), \quad (4.1)$$

donde $a, b, c, d, e,$ y f son números reales.

Con la notación equivalente de matrices y vectores, tenemos lo siguiente:

$$w(\mathbf{x}) = w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \mathbf{A}\mathbf{x} + \mathbf{T}.$$

Aquí $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ es una matriz de 2×2 y \mathbf{T} es un vector columna $\begin{pmatrix} e \\ f \end{pmatrix}$.

El vector columna con las componentes x y y representa un punto en el plano euclidiano. Al aplicar las transformaciones se debe pensar en conjuntos de puntos sobre el plano como se explicó anteriormente, ya que la finalidad es manipular los conjuntos para obtener diferentes figuras.

La *traslación* se aplica cuando se desea mover cada punto a una nueva posición que difiere de la posición original en un vector constante, es decir, sumándole a cada vector original el vector de los incrementos. En la Figura 4.4 se representa un cuadrado en el plano que se traslada a una nueva posición. Matemáticamente se describe de la siguiente manera

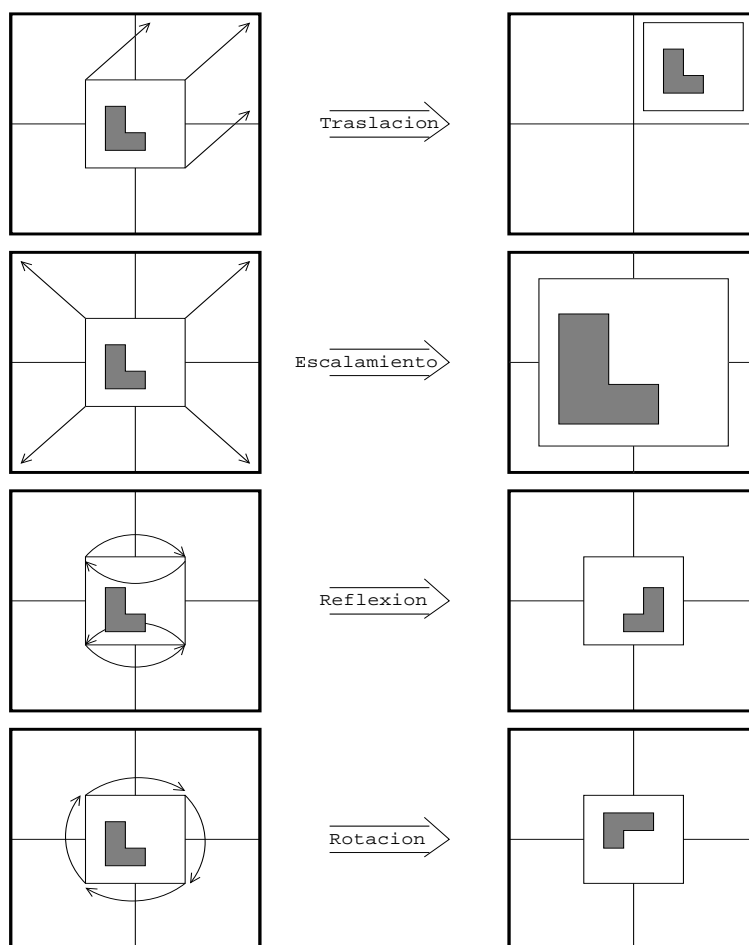


Figura 4.4: Transformaciones básicas.

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} x + e \\ y + f \end{pmatrix}$$

en donde e y f representan los incrementos sobre las componentes x y y .

El *escalamiento* de un conjunto de puntos se refiere a la contracción o expansión del conjunto. Se requiere de un factor de escalamiento que denotaremos como s y actúa sobre las componentes x y y de la siguiente manera

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix}.$$

Si el factor s es mayor que 1 entonces el conjunto se expande. Si s es menor que 1 entonces el conjunto se contrae. Ahora, si deseamos escalar un conjunto de tal forma que la escala horizontal y la vertical difieran, se debe escoger valores distintos para las componentes x y y

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} s_h & 0 \\ 0 & s_v \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} s_h x \\ s_v y \end{pmatrix}.$$

El factor s_h modifica la escala horizontal del conjunto mientras que s_v modifica la escala vertical. En la Figura 4.4 se representa el escalamiento de un cuadrado en donde $s_h = s_v$ por lo que la escala horizontal y vertical permanecen en la misma proporción. Cabe señalar que en los sistemas de funciones iteradas que nos interesan, el factor de escalamiento s deberá ser menor que 1.

Para la operación de *reflexión* se deben cambiar los signos de los valores de la diagonal de la matriz; si se quiere hacer una reflexión tomando como referencia el eje y , la primera entrada de la matriz debe ser negativa

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}.$$

En la Figura 4.4 las líneas indican el movimiento de los puntos y la flecha representa la nueva posición del punto original. Si se requiere que la reflexión sea sobre el eje x , se debe hacer negativa la componente y . Por otro lado, si se requiere reflejar simultáneamente respecto al eje x y el eje y , se tiene lo siguiente

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -x \\ -y \end{pmatrix}.$$

La *rotación* de un conjunto de puntos consiste en girar el conjunto alrededor de un punto. Por ejemplo, la expresión utilizada para una rotación de 90 grados, es la siguiente

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} y \\ -x \end{pmatrix}.$$

En la Figura 4.4 se representa la rotación de la imagen en pasos de 90 grados.

La traslación, reflexión, escalamiento y rotación son transformaciones básicas que se pueden combinar para manipular el conjunto de puntos de una manera específica. Existen otro tipo de transformaciones afines que nos permite rotar con mayor libertad y que incluyen las transformaciones básicas. A estas transformaciones se les llama *transformaciones de semejanza* y se expresan de la siguiente manera

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta & -r \operatorname{sen} \theta \\ r \operatorname{sen} \theta & r \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r \cos \theta & r \operatorname{sen} \theta \\ r \operatorname{sen} \theta & -r \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

en donde θ es llamado el ángulo de rotación y toma valores en $0 \leq \theta < 2\pi$ y r es llamado el factor de escalamiento. Estas son una forma generalizada de las transformaciones básicas.

4.3. Propiedades de espacios métricos y subconjuntos

Describiremos ciertas propiedades topológicas que nos serán de utilidad para entender el comportamiento de los conjuntos generados por los sistemas de funciones iteradas que actúan sobre el espacio métrico \mathbb{R}^2 , estas propiedades se pueden generalizar a n dimensiones pero nos enfocaremos en el plano euclidiano.

Para entender las propiedades topológicas debemos enunciar como primer lugar lo que es una sucesión de puntos. Una sucesión es un conjunto de elementos que denotaremos de la forma $\{x_n\}_{n=1}^{\infty}$ los cuales son puntos en \mathbb{R}^2 . Su ordenamiento esta dado por el índice que se denota como n y tiene la propiedad de preservar el orden.

En nuestra búsqueda y estudio de fractales generados por los sistemas de funciones iteradas se debe trabajar sobre un espacio que cumpla con ciertas propiedades los cuales llamaremos *espacios métricos completos*, anteriormente se definió a \mathbb{R}^2 como un espacio métrico, ahora se explicará lo que es

una sucesión de Cauchy para posteriormente definir a \mathbb{R}^2 como un espacio métrico completo.

Definición 3 Una sucesión $\{x_n\}_{n=1}^{\infty}$ de puntos en el espacio métrico (\mathbb{R}^2, d) es llamada una sucesión de Cauchy si para cualquier número dado $\epsilon > 0$, hay un entero $N > 0$ tal que

$$d(x_n, x_m) < \epsilon \quad \text{para toda } n, m > N.$$

Esto quiere decir que, dada cualquier distancia positiva ϵ , es posible encontrar un índice N (que depende de ϵ) tal que para cualquier par de índices n y m , que superan a N , la distancia entre los puntos correspondientes de la sucesión es menor que ϵ .

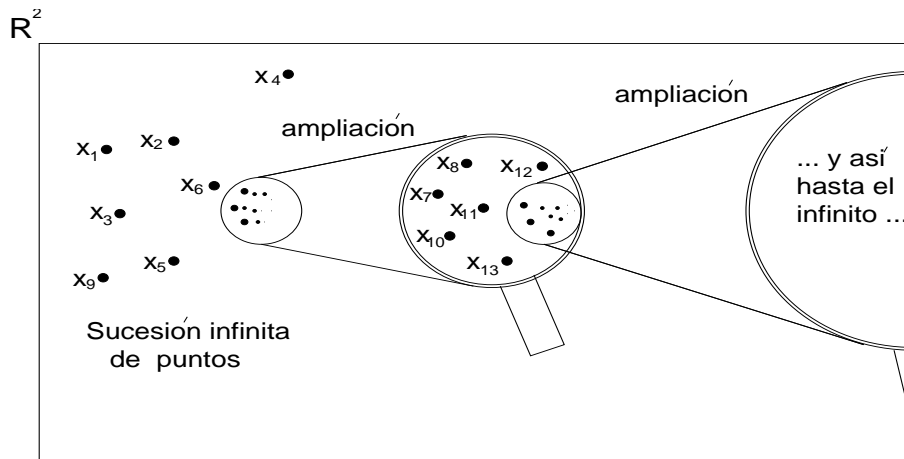


Figura 4.5: Imagen representando sucesivas ampliaciones sobre una sucesión de Cauchy.

En la Figura 4.5 se observa la representación de una sucesión de puntos en \mathbb{R}^2 , conforme la serie va avanzando los puntos se juntan cada vez más (se representa haciendo ampliaciones sobre subconjuntos de puntos), pero esto no quiere decir que converjan a un punto específico. Para saber si converge a un punto o no introduciremos la siguiente definición.

Definición 4 Una sucesión $\{x_n\}_{n=1}^{\infty}$ de puntos en el espacio métrico (\mathbb{R}^2, d) se dice que converge al punto $\mathbf{x} \in \mathbb{R}^2$, si para cualquier número dado $\epsilon > 0$, hay un entero $N > 0$ tal que

$$d(x_n, \mathbf{x}) < \epsilon \quad \text{para toda } n > N.$$

El punto $\mathbf{x} \in \mathbb{R}^2$ es donde converge la sucesión y es llamado el límite de la sucesión, lo denotaremos como

$$\lim_{n \rightarrow \infty} x_n = \mathbf{x}$$

Gráficamente lo podemos ver como un círculo de radio ϵ que satisface que dentro de él estarán todos los puntos de la sucesión a partir del índice N , donde N típicamente crece conforme ϵ es cada vez más pequeña.

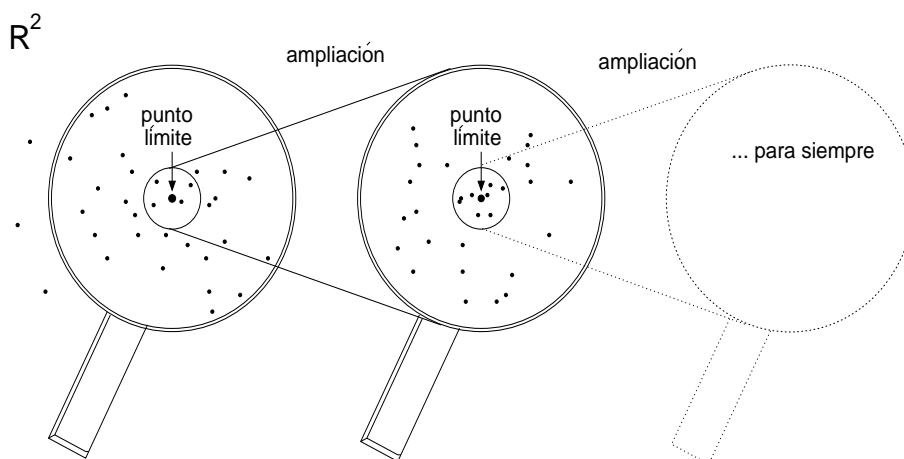


Figura 4.6: Ampliaciones alrededor del punto fijo.

Como se observa en la Figura 4.6 cada vez que hacemos una ampliación alrededor del punto fijo, son visibles más y más puntos, estas lupas son una buena analogía para los círculos mencionados anteriormente, ya que las lupas pueden desempeñar esta función, cuando hacemos un zoom es como si redujéramos el radio ϵ del círculo, y los puntos que están dentro de la lente amplificadora son los mayores a un índice N . En el infinito la serie convergerá al punto señalado como *punto límite*.

La relación entre una sucesión de Cauchy y una relación convergente es ésta: en cualquier espacio métrico, toda sucesión convergente es una sucesión de

Cauchy pero sólo en ciertos espacios métricos es posible asegurar la afirmación inversa, que toda sucesión de Cauchy sea convergente. De hecho, cuando en el espacio métrico de que se trate se cumple que las sucesiones de Cauchy convergen, se dice que el espacio es *completo*.

En los cursos de análisis matemático se prueba que los espacios (\mathbb{R}^n, d) son completos y es posible establecer, en ellos, las siguientes definiciones.

Definición 5 Sea S un subconjunto de puntos del espacio métrico (\mathbb{R}^2, d) . Un punto $\mathbf{x} \in \mathbb{R}^2$ es llamado un punto límite de S si hay una sucesión de puntos $\{x_n\}_{n=1}^{\infty}$ de $S \setminus \{\mathbf{x}\}$ tal que $\lim_{n \rightarrow \infty} x_n = \mathbf{x}$

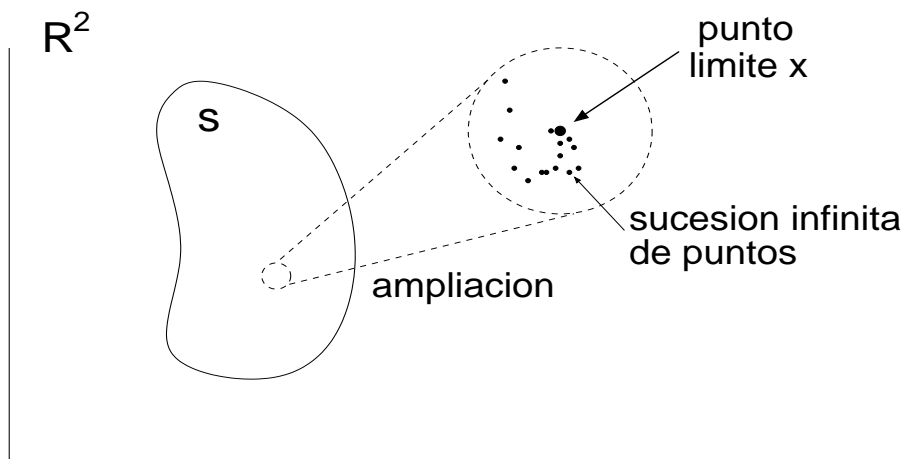


Figura 4.7: Punto límite del conjunto S .

En la Figura 4.7 se muestra el punto límite \mathbf{x} que no pertenece a la sucesión de puntos. La sucesión converge al punto pero no lo incluye. Un punto límite \mathbf{x} puede estar contenido en el conjunto S o no, y la siguiente definición explica esto.

Definición 6 Sea S contenido en \mathbb{R}^2 un subconjunto de puntos del espacio métrico (\mathbb{R}^2, d) . La cerradura (closure) de S , denotada como \bar{S} , se define como $\bar{S} = S \cup \{\text{Puntos Límite de } S\}$.

Definición 7 S es cerrado si contiene todos sus puntos límite, esto es si $S = \bar{S}$

Cuando S es un conjunto cerrado, sus puntos límite x están siempre dentro del conjunto, nunca en el exterior de S . Hay regiones en las que no hay puntos límite, como se muestra en la Figura 4.8, las ampliaciones nos revelan donde hay puntos límite, en la ampliación de la izquierda no hay puntos límite, mientras que en la otra tenemos dos. Si para cada punto x de S pudiéramos encontrar una sucesión de puntos de S que convergiera a dicho punto, tendríamos un conjunto de puntos límite con la misma cardinalidad que el conjunto S . La siguiente definición enuncia esta propiedad.

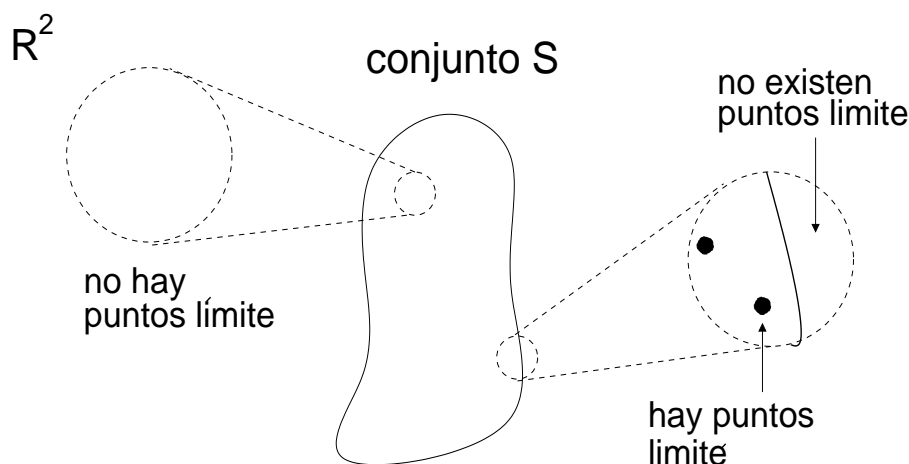


Figura 4.8: S es cerrado si sus puntos límite están dentro del conjunto.

Definición 8 *El conjunto S es perfecto si tiene el mismo número de elementos que el conjunto de todos sus puntos límite.*

En la Figura 4.9 se aprecia la comparación del conjunto S y el conjunto de sus puntos límite.

Se concluirá esta sección con la definición de conjunto compacto, este concepto es de gran importancia ya que los sistemas de funciones iteradas generan conjuntos compactos que se estudiarán para entender el comportamiento de los fractales. Con todas estas herramientas y conceptos, se podrá abordar la compresión fractal de imágenes.

Definición 9 *Sea S contenido en \mathbb{R}^2 un subconjunto del espacio métrico (\mathbb{R}^2, d) . S es compacto si para toda sucesión infinita de puntos $\{x_n\}_{n=1}^{\infty}$ en S existe una subsucesión que tiende al límite en S .*

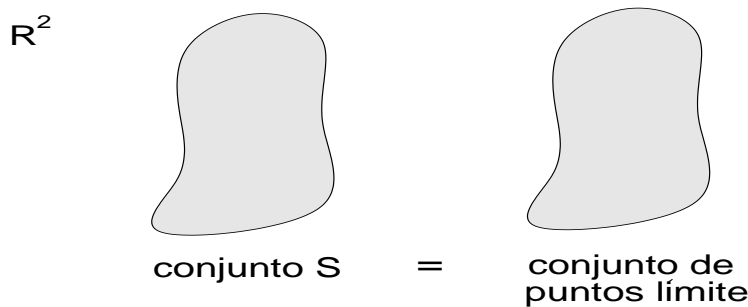


Figura 4.9: Comparación entre el conjunto S y el conjunto de puntos límite de S .

Podemos imaginarnos; por ejemplo, una sucesión infinita de puntos que esté conmutando entre dos puntos p_1 y p_2 en S , si tomamos una subsucesión con las posiciones pares tendremos un conjunto que consta sólo de puntos p_2 , y dicha subsucesión converge al punto p_2 , análogamente si tomamos las posiciones impares. De esta forma se construyen las subsucesiones para que tengan su límite en S . Otra forma de ver al conjunto compacto es que sea cerrado y acotado. Para una discusión profunda a cerca de estas definiciones véase [9], [6] y [20].

4.4. Transformaciones de contracción

Esta propiedad de las transformaciones es la clave para generar los objetos fractales, los conceptos desarrollados en la sección 4.2 son la base para entender esta propiedad, se empezará por definir qué es un punto fijo.

Definición 10 Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ una transformación en el plano euclidiano. Un punto x_f que pertenece al plano tal que al aplicar la transformación permanece en el mismo lugar, $f(x_f) = x_f$, es llamado punto fijo de la transformación.

En la figura 4.10 se ilustra cómo el conjunto que se dibujó en forma de árbol se le aplica la transformación f y se va reduciendo iteración tras iteración hasta que el árbol se ha transformado en un pequeño conjunto, al llegar a esta etapa el conjunto es ahora un punto que no cambia y es entonces cuando surge el punto fijo de la transformación f .

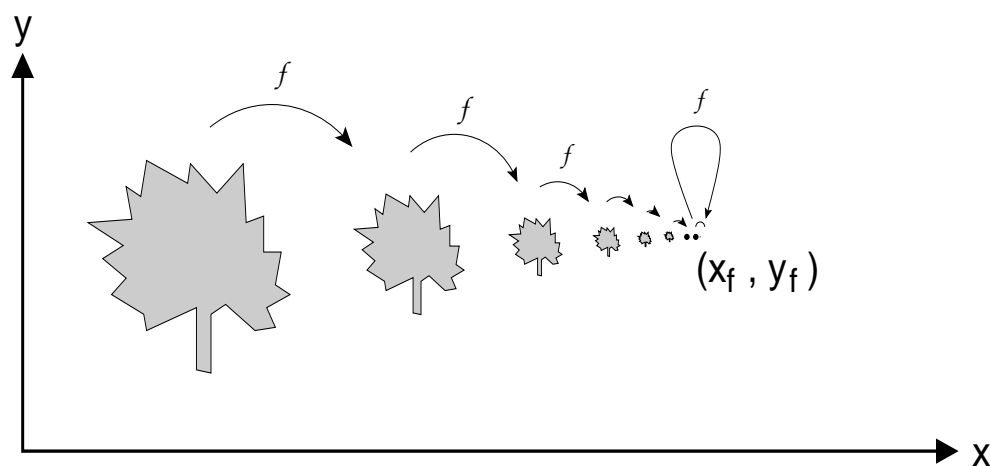


Figura 4.10: Ejemplo de punto fijo.

Se definirá lo que es una *transformación de contracción*.

Definición 11 Una función $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ sobre el espacio métrico (\mathbb{R}^2, d) es llamada transformación de contracción si hay una constante entre $0 \leq s < 1$ tal que se cumpla la siguiente desigualdad para todo x y y en el plano:

$$d(f(x), f(y)) \leq s \cdot d(x, y)$$

A la constante s se le llama *factor de contracción* para la transformación f .

En la Figura 4.11 se aplica la transformación f a dos puntos cualesquiera en el plano, los nuevos puntos transformados tienen una distancia menor que los originales; es decir, están más cerca: se puede saber que tanto se acercaron con el factor de contracción; por ejemplo, si s fuera 0.5 quiere decir que los nuevos puntos transformados distan la mitad de su distancia original. Si s toma el valor de cero los puntos transformados quedan en una sola posición, es decir quedan encimados, por lo tanto su distancia es cero, el caso contrario es cuando s esta muy cerca del valor uno, la distancia entre los puntos transformados es casi igual que la distancia de los puntos originales.

Ahora se enunciará el *teorema del mapeo de contracción* el cual nos garantiza que al aplicar una transformación de contracción repetidamente a algún conjunto S que pertenece a \mathbb{R}^2 , este converge a algún punto fijo.

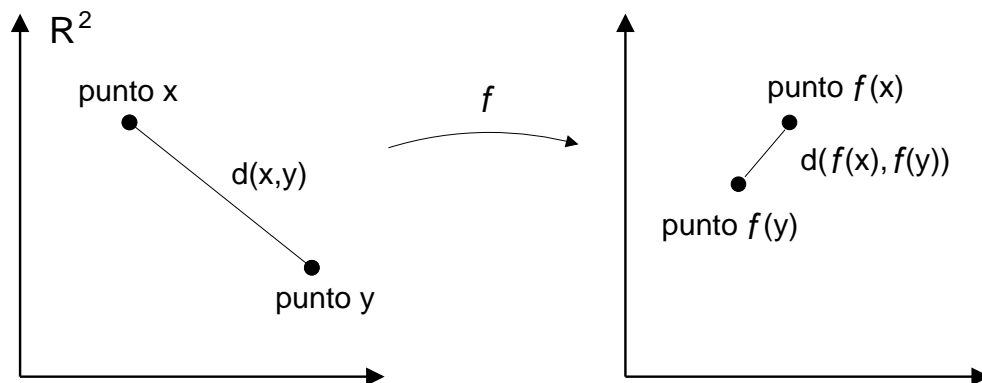


Figura 4.11: La transformación tiene la propiedad de contracción.

Teorema 1 *El teorema del mapeo de contracción.*

Sea $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ un transformación de contracción sobre el espacio métrico completo (\mathbb{R}^2, d) . Entonces f posee exactamente un punto fijo $x_f \in \mathbb{R}^2$, y más aún para cualquier punto \mathbf{x} que pertenece a \mathbb{R}^2 , la sucesión de puntos generados por las iteraciones delanteras $\{f^{\circ n}(\mathbf{x}) : n = 0, 1, 2, \dots\}$ converge al punto fijo x_f ; esto es,

$$\lim_{n \rightarrow \infty} f^{\circ n}(x) = x_f, \quad \text{para cada } x \in \mathbb{R}^2$$

Para la demostración del teorema véase [4].

Se han dado las herramientas necesarias para llegar a comprender el comportamiento de un sistema de funciones iteradas. En la siguiente sección se describirá un espacio en donde se pueden estudiar los conjuntos generados por los sistemas de funciones iteradas, de tal forma que podamos definir sencillas operaciones sobre estos conjuntos. Seremos capaces de entender estas dinámicas que dan como resultado los tan sencillos y complicados fractales.

4.5. Espacio de Hausdorff

Para estudiar los fractales que se generan con los sistemas de funciones iteradas es conveniente trabajar en el espacio de Hausdorff \mathcal{H} , ya que en este espacio los conjuntos pueden ser tratados como dibujos e imágenes binarias

pues están constituidos por puntos negros sobre un fondo blanco, esto nos será de gran utilidad cuando nos adentremos en el tema de compresión fractal.

En el espacio de Hausdorff \mathcal{H} podemos estudiar importantes subconjuntos de espacios métricos. Nos enfocaremos en el espacio métrico completo \mathbb{R}^2 con la distancia euclidiana, ya que es el espacio donde hemos definido todos nuestros conceptos. Empezaremos por definir formalmente este espacio.



Figura 4.12: La representación de elementos en el espacio de Hausdorff.

Definición 12 Sea (\mathbb{R}^2, d) un espacio métrico completo. Entonces $\mathcal{H}(\mathbb{R}^2)$ denota el espacio cuyos puntos son subconjuntos compactos no vacíos de \mathbb{R}^2 .

En la sección anterior se definió lo que es un conjunto compacto, en términos sencillos es un conjunto que “está delimitado”; por ejemplo, una imagen. Estos conjuntos en el espacio de Hausdorff son puntos como se menciona en la definición, pero sobre el plano euclidiano son conjuntos.

A continuación se definirá una métrica que opere sobre el espacio de Hausdorff para posteriormente definirlo como un espacio métrico completo.

Definición 13 Sea (\mathbb{R}^2, d) , sea x un punto en \mathbb{R}^2 y B un elemento en el espacio $\mathcal{H}(\mathbb{R}^2)$, entonces definiremos

$$d(x, B) = \min\{d(x, y) : y \in B\}$$

donde $d(x, B)$ es llamada la distancia entre el punto x y el conjunto B .

Para obtener esta distancia debemos calcular; por ejemplo, la distancia euclidiana entre todos los puntos del conjunto B y el punto x , esto genera un conjunto de números y el menor de ellos es la distancia que representa la definición anterior (ver Figura 4.13).

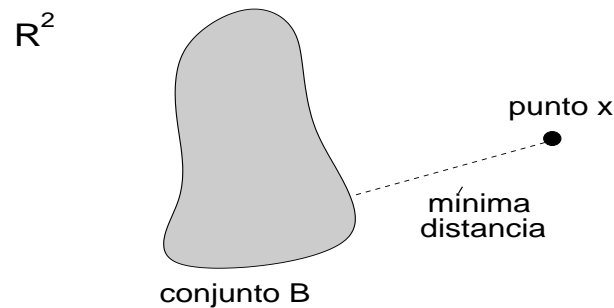


Figura 4.13: Se muestra la menor distancia que hay entre un punto x y el conjunto B .

Definición 14 Sea (\mathbb{R}^2, d) . Sean A y B pertenecientes a $\mathcal{H}(\mathbb{R}^2)$. Definimos

$$d(A, B) = \max\{d(x, B) : x \in A\}$$

donde $d(A, B)$ es llamada la distancia entre el conjunto $A \in \mathcal{H}(\mathbb{R}^2)$ y el conjunto $B \in \mathcal{H}(\mathbb{R}^2)$.

Para obtener esta distancia se aplica la definición 13 para cada punto en A y el conjunto B , de esta forma se obtiene un conjunto de distancias mínimas; posteriormente se encuentra el elemento máximo para tener la distancia entre el conjunto A y el B ; notemos que no es lo mismo calcular la distancia $d(A, B)$ y $d(B, A)$, esto servirá para la definición siguiente (ver Figura 4.14).

Definición 15 Sea (\mathbb{R}^2, d) . Entonces la distancia de Hausdorff entre los puntos A y B en $\mathcal{H}(\mathbb{R}^2)$ esta definida por

$$h(A, B) = d(A, B) \vee d(B, A)$$

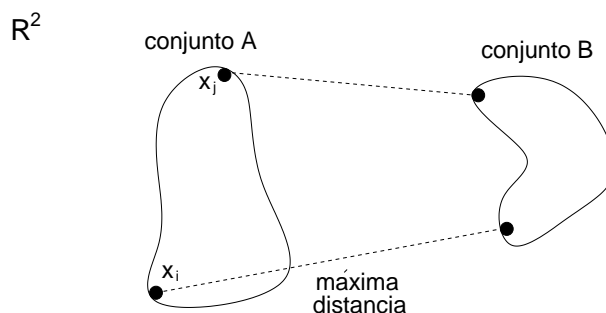


Figura 4.14: La distancia entre los conjuntos A y B se calcula tomando la distancia máxima de entre las distancias mínimas de los puntos de A al conjunto B .

donde $x \vee y$ significa el máximo de x y y . Llamaremos a h la métrica de Hausdorff sobre \mathcal{H} .

Ya que se ha definido una métrica, se puede definir el espacio de Hausdorff como un espacio métrico completo, recordando que toda sucesión de Cauchy sobre este tipo de espacios converge.

Teorema 2 Sea (\mathbb{R}^2, d) . Entonces $\mathcal{H}(\mathbb{R}^2, h)$ es un espacio métrico completo. Más aún, si $\{A_n \in \mathcal{H}(\mathbb{R}^2) : n = 1, 2, \dots\}$ es una sucesión de Cauchy, entonces

$$A = \lim_{n \rightarrow \infty} A_n \in \mathcal{H}(\mathbb{R}^2)$$

puede ser caracterizado como

$$A = \{x \in \mathbb{R}^2 \text{ tal que existe una sucesión de Cauchy } \{x_n \in A_n\} \text{ convergiendo a } x\}.$$

Para la demostración del teorema véase [4].

Sea (\mathbb{R}^2, d) un espacio métrico y sea $(\mathcal{H}(\mathbb{R}^2), h(d))$ la notación para representar su espacio de Hausdorff asociado, con la métrica $h(d)$ descrita anteriormente y donde d es la distancia euclidiana como la hemos manejado siempre. La notación $h(d)$ muestra que d es la métrica que está por debajo o subyace a la métrica de Hausdorff h .

4.6. Mapeos de contracción en el espacio \mathcal{H}

Anteriormente se definieron los mapeos de contracción para \mathbb{R}^2 y se explicaron detalladamente sus propiedades, ahora llevaremos estos conceptos al espacio de Hausdorff.

Lema 1 *Sea $w : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ un transformación de contracción sobre el espacio métrico (\mathbb{R}^2, d) con factor de contracción s . Entonces $w : \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$ definida por*

$$w(B) = \{w(x) : x \in B\}, \text{ para todo } B \in \mathcal{H}(\mathbb{R}^2)$$

es una transformación de contracción sobre $(\mathcal{H}(\mathbb{R}^2), h(d))$ con factor de contracción s .

Los sistemas de funciones iteradas como su nombre lo dice constan de un conjunto de funciones o transformaciones que serán aplicadas a un conjunto, la forma de aplicarlas se explicará en el siguiente lema, y por lo tanto llegaremos al objetivo primordial de este capítulo que es dar una definición formal de los sistemas de funciones iteradas sustentada en todos los conceptos que se han desarrollado.

Lema 2 *Sea $\{w_n : n = 1, 2, \dots, N\}$ un conjunto de transformaciones de contracción sobre $(\mathcal{H}(\mathbb{R}^2), h)$. Sea s_n el factor de contracción para la transformación w_n . Definiremos $W : \mathcal{H}(\mathbb{R}^2) \rightarrow \mathcal{H}(\mathbb{R}^2)$ como*

$$\begin{aligned} W(B) &= w_1(B) \cup w_2(B) \cup \dots \cup w_N(B) \\ &= \bigcup_{i=1}^N w_i(B), \text{ para cada } B \in \mathcal{H}(\mathbb{R}^2) \end{aligned}$$

Entonces W es una transformación de contracción con factor de contracción $s = \max\{s_n : n = 1, 2, \dots, N\}$.

4.7. Sistemas de funciones iteradas

El nombre de *sistemas de funciones iteradas* (en inglés *iterated function system*) fue inventado o usado por primera vez por Barnsley [22]. Hay otras referencias relevantes como [16] y [25]. En esta sección se usará la definición dada por Barnsley [5] modificada ligeramente para el espacio \mathbb{R}^2 .

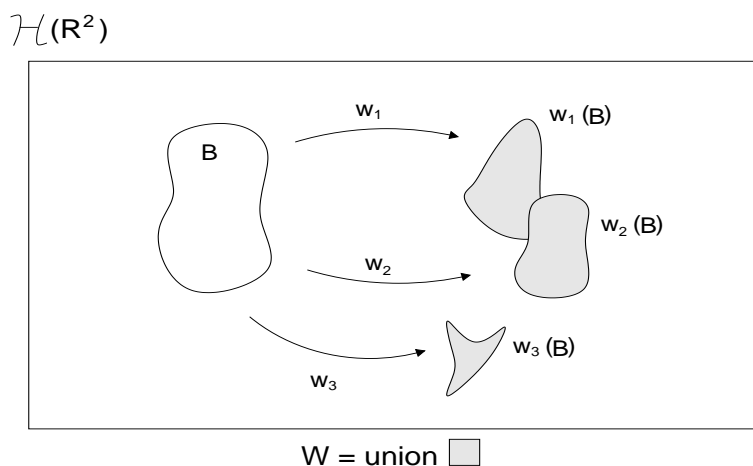


Figura 4.15: La aplicación de un conjunto de transformaciones de contracción al punto B en el espacio de Hausdorff $\mathcal{H}(\mathbb{R}^2)$, el resultado es la unión de los nuevos puntos.

Definición 16 *Un sistema de funciones iteradas consiste del espacio métrico completo (\mathbb{R}^2, d) junto con un conjunto finito de transformaciones de contracción $w_n : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ con su respectivo factor de contracción s_n , para $n = 1, 2, \dots, N$. Usaremos la abreviación “IFS” para sistemas de funciones iteradas. La notación para los IFS es $\{\mathbb{R}^2; w_n, n = 1, 2, \dots, N\}$ y su factor de contracción es $s = \max\{s_n : n = 1, 2, \dots, N\}$.*

Usaremos esta nomenclatura para denotar simplemente un conjunto de transformaciones finito actuando sobre el espacio métrico en cuestión, con la única condición de que sean transformaciones afines como las que se definieron anteriormente. El siguiente teorema es la parte central para describir la dinámica de un IFS.

Teorema 3 El teorema de los IFS

Sea $\{\mathbb{R}^2; w_n, n = 1, 2, \dots, N\}$ un Sistema de Funciones Iteradas con factor de contracción s . Entonces la transformación $W : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ definida por

$$W(B) = \bigcup_{n=1}^N w_n(B),$$

para todo $B \in \mathcal{H}(\mathbb{R}^2)$ es un mapeo de contracción sobre el espacio métrico completo $(\mathcal{H}(\mathbb{R}^2), h(d))$ con factor de contracción s ; esto es

$$h(W(B), W(C)) \leq sh(B, C)$$

para todo $B, C \in \mathcal{H}(\mathbb{R}^2)$, se tiene un único punto fijo, $A \in \mathcal{H}(\mathbb{R}^2)$, el cual obedece

$$A = W(A) = \bigcup_{n=1}^N w_n(A),$$

y viene dado por $A = \lim_{n \rightarrow \infty} W^{\circ n}(B)$ para cualquier $B \in \mathcal{H}(\mathbb{R}^2)$.

Definición 17 El punto fijo $A \in \mathcal{H}(\mathbb{R}^2)$ descrito en el teorema de los IFS se llama el atractor del IFS.

4.8. Algoritmo de la máquina fotocopidora

Podemos calcular el atractor del IFS a través de algoritmos adecuados. Existe un algoritmo llamado *El algoritmo de la máquina fotocopidora*, el cual sugiere los pasos para poder calcular el atractor del IFS. Como primer punto debemos definir el IFS, sea $\{\mathbb{R}^2; w_1, w_2, \dots, w_N\}$. Se escoge un conjunto compacto A_0 que pertenece a \mathbb{R}^2 . Se calculan sucesivamente conjuntos $A_n = W^{\circ n}(A)$ con la regla siguiente:

$$A_{n+1} = \bigcup_{j=1}^N w_j(A_n) \text{ para } n = 1, 2, \dots;$$

la cual construye una sucesión $\{A_n : n = 0, 1, 2, 3, \dots\} \in \mathcal{H}(\mathbb{R}^2)$ que converge al atractor del IFS en el espacio de Hausdorff. Para ser más claros, las transformaciones del IFS se aplican al conjunto inicial A_0 como en la Figura 4.15; este conjunto inicial puede ser cualquiera no importa la forma o el tamaño. Al nuevo conjunto generado le llamaremos A_1 y se le aplicará nuevamente el conjunto de transformaciones para obtener A_2 y así sucesivamente para obtener una sucesión de conjuntos que a la larga convergerá al atractor del IFS. Si cada A_i lo graficamos en una computadora a través del *software* adecuado, se observará la sucesión de estados convergiendo a una aproximación del atractor (es aproximado ya que el verdadero atractor surge cuando hacemos un número infinito de iteraciones).

Se le llama “algoritmo de la máquina fotopiadora” porque el proceso de fotocopiar es similar al proceso anterior. Por ejemplo, tomamos una hoja de papel

en blanco, y la reducimos de tamaño digamos en un 40 % entonces tenemos la imagen de un rectángulo en la nueva hoja, recortamos este rectángulo y lo pegamos en una hoja en blanco en cualquier posición . Creamos tantos rectángulos como queramos, este paso se representa en figura 4.16. Generamos 8 copias de esta nueva hoja ya que serán nuestras *plantillas*. Ahora tomamos una hoja con contenido (texto, imagen, etc.), y la reducimos en un factor de 40 %, este factor debe ser el mismo en todas las fotocopias, ya reducida sacamos tres copias, recortamos las reducciones y las pegamos en la primer plantilla (ver Figura 4.16), a esta plantilla le llamaremos A_1 . La plantilla A_1 la reducimos y le sacamos tantas copias como rectángulos haya, en nuestro caso son 3 copias, se recortan y se pegan en la segunda plantilla y la llamamos A_2 , y así sucesivamente, hasta terminar todas las plantillas. Después de repetir una y otra vez todo el proceso, notaremos que hay un momento en que no se encuentran diferencias entre plantillas, entonces diremos que llegamos al atractor generado por la fotocopidora.

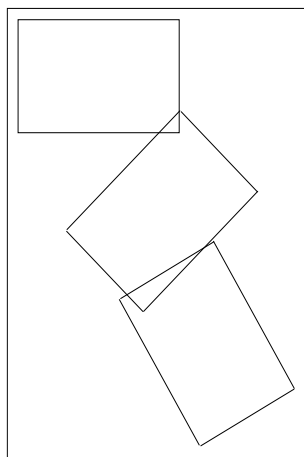


Figura 4.16: Plantilla arbitraria para generar un atractor con el algoritmo de la fotocopidora.

4.8.1. Ejemplo de un IFS

Aplicando el algoritmo anterior podemos generar el triángulo de Sierpinski. Primero definiremos el IFS adecuado. Sea $\{\mathbb{R}^2; w_1, w_2, w_3\}$ donde las w_i son transformaciones afines en \mathbb{R}^2 definidas como:

$$w_1 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

$$w_2 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 & 50 \\ 1 & 0.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 50 \\ 0 \end{pmatrix},$$

$$w_3 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0.5 & 50 \\ 50 & 0.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 50 \\ 50 \end{pmatrix}.$$

El triángulo de Sierpinski tendrá como vértices los puntos $(0,0)$, $(100,0)$, y $(100,100)$. La figura 4.17 muestra el conjunto inicial que corresponde a un cuadrado de 100 por 100 pixels, tiene estas dimensiones para mostrar de una forma simple las siguientes 8 iteraciones.

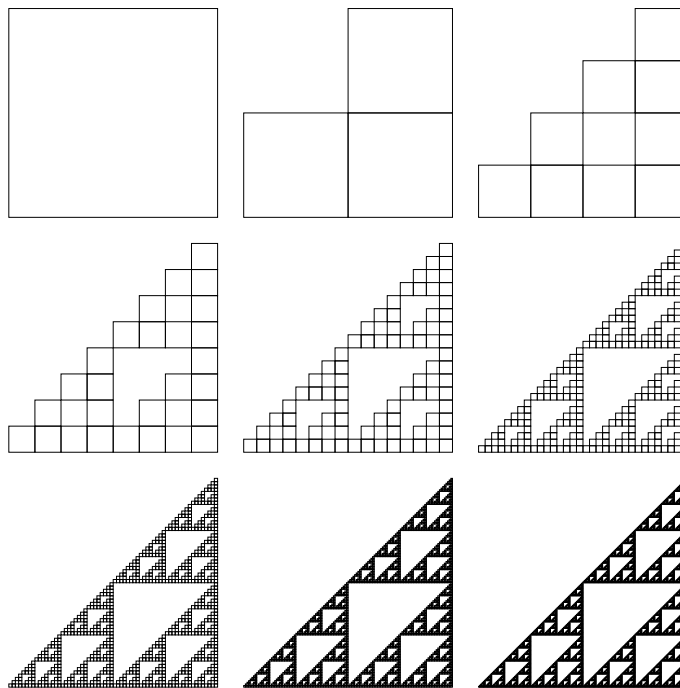


Figura 4.17: Iteraciones generadas por el IFS.

Para reforzar la idea de convergencia de un IFS en su atractor, se repite el proceso pero ahora con un conjunto inicial diferente como se muestra en

la figura 4.18. Al cabo de n iteraciones el IFS convergerá al triángulo de Sierpinski.

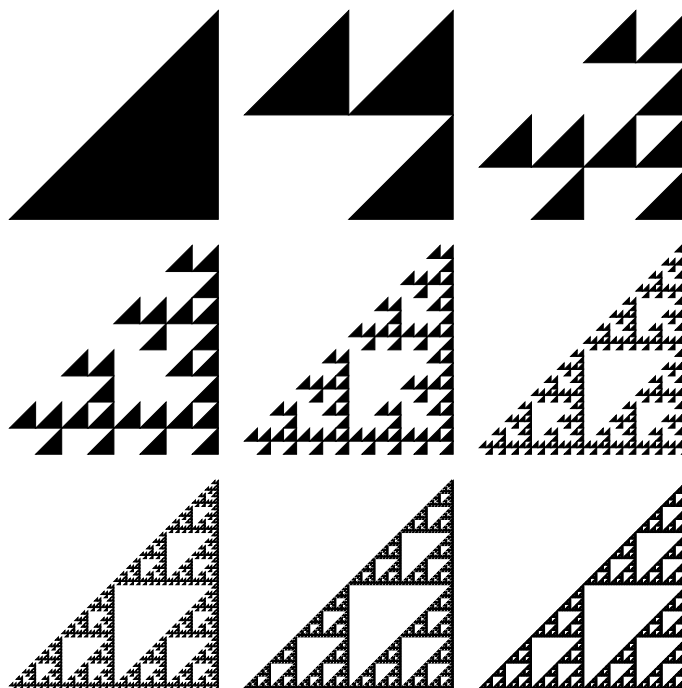


Figura 4.18: Un Conjunto inicial distinto al de la Figura 4.17 convergiendo al triángulo de Sierpinski.

Para manipular fácilmente el código podemos hacer una tabla que resuma las características del IFS anterior como la siguiente.

w	a	b	c	d	e	f	p
1	0.5	0	0	0.5	0	0	0.33
2	0.5	0	0	0.5	50	0	0.33
3	0.5	0	0	0.5	50	50	0.33

La última columna representa una probabilidad asociada a cada transformación, en el ejemplo anterior no se utilizaron. En la siguiente sección se explica la forma en que trabajan los IFS con probabilidades.

4.9. IFS con probabilidades

Se definirá un nuevo tipo de IFS que es parecido al anterior, ahora las transformaciones se aplicarán siguiendo una probabilidad, esto nos permitirá crear rápidamente una representación gráfica del atractor.

Definición 18 Un sistema de funciones iteradas con probabilidades *consiste de un IFS* $\{\mathbb{R}^2; w_1, w_2, \dots, w_N\}$ *junto con un conjunto ordenado de números* $\{p_1, p_2, \dots, p_N\}$, *tal que*

$$p_1 + p_2 + \dots + p_N = 1 \text{ y } p_i > 0 \text{ para } i = 1, 2, 3, \dots, N.$$

La probabilidad p_i esta asociada a la transformación w_i .

Las probabilidades están relacionadas con la teoría de la medida de los atractores de los IFS. Con un algoritmo adecuado podemos usar estas probabilidades para llegar a la visualización del atractor en una computadora.

El siguiente mecanismo nos permite asignar probabilidades a cada transformación y nos será de utilidad para el algoritmo que mostraremos posteriormente.

$$p_i \approx \frac{|\det A_i|}{\sum_{i=1}^N |A_i|} = \frac{|a_i d_i - b_i c_i|}{\sum_{i=1}^N |a_i d_i - b_i c_i|} \text{ para } i = 1, 2, \dots, N.$$

Se toma el símbolo \approx ya que puede surgir el caso en que $A_i = 0$ entonces asignará una probabilidad cero y estaría violando la definición, por tal motivo debemos asignar un número positivo muy pequeño o ajustarlo según convenga.

4.9.1. EL algoritmo de iteraciones aleatorias

Sea $\{\mathbb{R}^2; w_1, w_2, \dots, w_N\}$ un IFS, donde las probabilidades $p_i > 0$ han sido asignadas para cada transformación w_i con $i = 1, \dots, N$ donde $\sum_{i=1}^N p_i = 1$. Se escoge un punto cualquiera $\mathbf{x}_0 \in \mathbb{R}^2$, se escoge recursivamente y de forma independiente una transformación de acuerdo a su probabilidad. Se le aplica la transformación elegida y obtenemos un nuevo punto

$$\mathbf{x}_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \dots, w_N(x_{n-1})\} \text{ para } n = 1, 2, 3, \dots,$$

donde la probabilidad del evento $\mathbf{x}_n = w_i(x_n - 1)$ es p_i . Entonces se construye la sucesión $\{\mathbf{x}_n : n = 1, 2, 3, \dots\} \in \mathbb{R}^2$. Esta sucesión $\{\mathbf{x}_n\}_{n=1}^{\infty}$ converge al atractor del IFS. Para poder visualizar el atractor debemos graficar los últimos n puntos ya que son los que están más próximos al atractor. En la Figura 4.19 se muestra el proceso de este algoritmo calculando el *helecho* (fern) de Barnsley.

4.10. El teorema del collage

El objetivo del teorema del collage es escoger un IFS para un conjunto de puntos T tal que al iterarlo resulte un conjunto lo más parecido al conjunto T inicial, es decir se escogen las transformaciones del IFS tal que al aplicarlas sobre T la distancia entre el conjunto T inicial y el conjunto T transformado sea mínima, esto nos garantiza que el atractor del IFS estará aún más cerca del conjunto inicial T que cualquier iteración del IFS sobre T . La comprensión fractal de imágenes utiliza este teorema como fundamento teórico para garantizar su eficiencia.

Teorema 4 El teorema del collage

Sean $T \in \mathcal{H}(\mathbb{R}^2)$ y $\epsilon \geq 0$. Se escoge un IFS $\{\mathbb{R}^2; w_0, w_1, w_2, \dots, w_N\}$ con factor de contracción $0 \leq s < 1$ entonces se tiene

$$h\left(T, \bigcup_{n=1}^N w_n(T)\right) \leq \epsilon,$$

en donde $h(d)$ es la métrica de Hausdorff, entonces

$$h(T, A) \leq \frac{\epsilon}{1 - s},$$

en donde A es el atractor del IFS. Equivalentemente se tiene que

$$h(T, A) \leq (a - s)^{-1} h\left(T, \bigcup_{n=1}^N w_n(T)\right) \text{ para toda } T \in \mathcal{H}(\mathbb{R}^2).$$

Para la demostración del teorema véase [4].

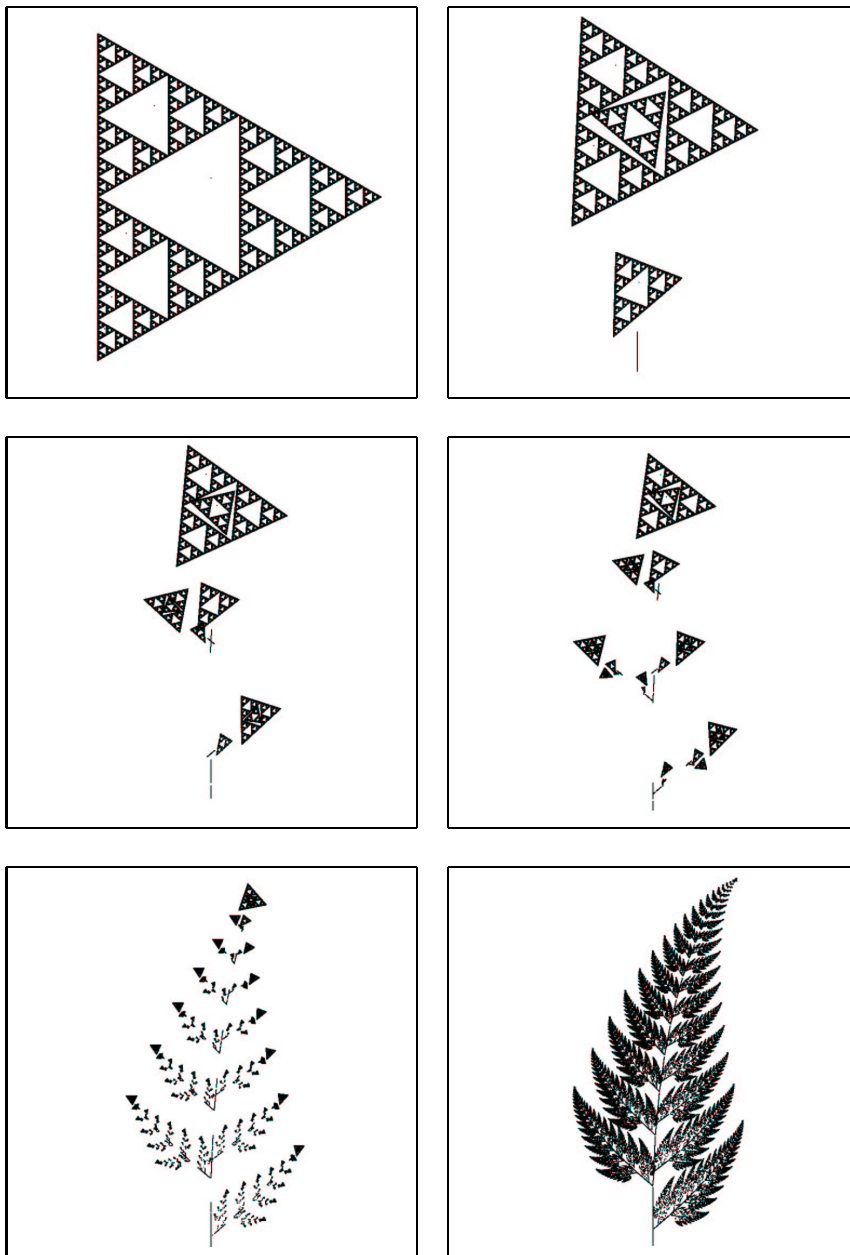


Figura 4.19: Helecho de Barnsley generado por un IFS con probabilidades utilizando como conjunto inicial el triángulo de Sierpinski.

4.11. Medida invariante

El propósito de esta sección es dar una herramienta que servirá para estudiar las distribuciones de puntos en los atractores generados por sistemas, en particular el juego del caos y los IFS. En el juego del caos utilizando el *shift* de Bernoulli encontramos distribuciones de puntos muy variadas dependiendo de la sucesión que se utilice. Otro ejemplo es el IFS que genera el helecho de Barnsley, ya que al variar las probabilidades, las distribuciones de puntos sobre el atractor cambian. Para saber cómo cambian estas distribuciones, es necesario poder medir las densidades de puntos sobre los atractores, esto se hará utilizando algo que llamaremos *medida invariante*.

A los subconjuntos de \mathbb{R}^2 que contengan puntos los llamaremos *subconjuntos de Borel* sobre \mathbb{R}^2 y los denotaremos como $\mathcal{B}(\mathbb{R}^2)$. Los subconjuntos de Borel de \mathbb{R}^2 incluyen los subconjuntos compactos no vacíos de \mathbb{R}^2 así que $\mathcal{H}(\mathbb{R}^2)$ está contenido en $\mathcal{B}(\mathbb{R}^2)$. Si \mathcal{O} es un subconjunto abierto de \mathbb{R}^2 entonces $\mathcal{O} \in \mathcal{B}(\mathbb{R}^2)$

Para medir la densidad de puntos sobre el atractor definiremos regiones B que lo cubran totalmente. B denota una bola cerrada en \mathbb{R}^2 . La densidad estará dada por un sistema que produce una sucesión de puntos $\{Z_n\}_{n=1}^{\infty}$. Definamos a

$$\mathcal{N}(B, n) = \text{número de puntos en } \{z_0, z_1, \dots, z_n\} \cap B \text{ para } n = 0, 1, 2, \dots$$

La medida invariante la denotaremos como μ , la cual es una función de $\mathcal{B}(\mathbb{R}^2) \rightarrow [0, 1] \subset \mathbb{R}$ y la definiremos como:

$$\mu(B) = \lim_{n \rightarrow \infty} \frac{\mathcal{N}(B, n)}{(n+1)}$$

Esto indica que la densidad en la bola B es la proporción de puntos producidos por el sistema que caen en la bola. Por ejemplo si A es un atractor entonces $\mu(A) = 1$ y $\mu(\emptyset) = 0$, también $\mu(\mathbb{R}^2) = 1$, lo que nos dice que todo el espacio \mathbb{R}^2 tiene la misma densidad que el atractor del sistema, por lo que el sistema no genera puntos que no estén sobre el atractor.

Para comprender estos conceptos definamos una bola $B \in \mathcal{B}(\mathbb{R}^2)$ en el triángulo de Sierpinski generado por el juego del caos, como se muestra en la

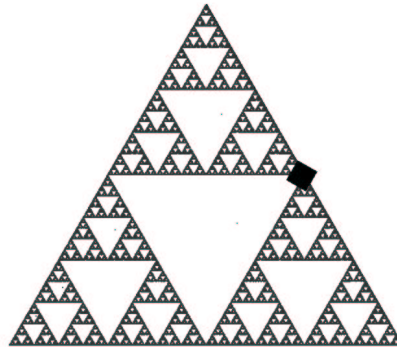


Figura 4.20: El cuadro negro sobre el triángulo de Sierpinski generado por el juego del caos representa una bola.

Figura 4.20 y aplicamos la medida μ a la bola con diferentes cantidades de iteraciones n , los resultados se muestran en el Cuadro 4.1

Si se tiene una medida sobre un conjunto de Borel entonces diremos que esta medida es de Borel, por lo que μ es una medida de Borel.

Definición 19 Sea μ una medida de Borel sobre $\square \subset \mathbb{R}^2$. Si $\mu(\square) = 1$, entonces decimos que μ esta normalizada.

Sea \mathcal{B} que denota los subconjuntos de Borel de \square . Sea $w : \square \rightarrow \square$ continua

n iteraciones	$\mathcal{N}(B_1, n)/n$	$\mathcal{N}(B_2, n)/n$
10000	0.010900	0.013900
50000	0.009500	0.014540
100000	0.009700	0.014010
500000	0.009008	0.013204
1000000	0.009145	0.013098
5000000	0.009106	0.013101

Cuadro 4.1: Medidas de dos bolas con distintas iteraciones, se puede ver que entre más iteraciones la proporción de puntos converge a un valor.

y definamos $w^{-1} : \mathcal{B} \rightarrow \mathcal{B}$. Si ν es una medida normalizada de Borel sobre \square entonces $\nu \circ w^{-1}$ también lo es. Utilizaremos estas construcciones para la siguiente definición.

Definición 20 Sea $\{\square; w_1, w_2, \dots, w_N; p_1, p_2, \dots, p_N\}$ un IFS con probabilidades. El operador de Markov asociado con este IFS se define como

$$M(\nu) = p_1\nu \circ w_1^{-1} + p_2\nu \circ w_2^{-1} + \dots + p_N\nu \circ w_N^{-1}$$

La medida de Borel μ es invariante bajo el operador de Markov $M(\nu)$. En otras palabras $M(\nu)$ define una nueva medida de Borel normalizada sobre \square . Esta nueva medida de Borel la evaluaremos sobre un subconjunto dado $B \in \square$ que es el atractor del IFS.

La iteración sobre el operador de Markov construirá las densidades sobre subconjuntos de Borel y lo podemos representar mediante histogramas. En particular existe una única medida η tal que al aplicarle el operador de Markov no varía, es decir $M(\eta) = \eta$ y cumple con

$$\lim_{n \rightarrow \infty} M^{on}(\nu) = \eta$$

A η le llamaremos el *punto fijo* del operador de Markov.

Definición 21 Sea η el punto fijo del operador de Markov. Entonces η es llamada la medida invariante del IFS con probabilidades.

Capítulo 5

Compresión fractal

Con el avance de la tecnología las cantidades de información que se manejan han crecido; por lo tanto, cada vez hay más necesidad de dispositivos para guardarla, pero no han sido suficientes para la cantidad de datos que se maneja en imágenes, audio y video. Cada vez que la calidad de resolución en imagen y video crece o, en el caso del sonido, la fidelidad, se requiere de nuevos dispositivos que logren contener tal cantidad de información; hasta el momento, los avances científicos y tecnológicos han logrado satisfacer esta necesidad, pero debemos tener en cuenta que las herramientas que tenemos a la mano están limitadas y la cantidad de datos que se manejan esta creciendo de manera exponencial; por tal motivo, los científicos se han visto en la necesidad de desarrollar técnicas que persiguen la compresión de datos.

La compresión de datos se ha buscado desde la década de los 40 como una consecuencia de la teoría de la información hecha por Claude Shannon[27]; la compresión de datos ha evolucionado desde ese tiempo y uno de los cambios más importantes es la postura de sacrificar información cada vez que se comprime, cuando la iniciativa original era que la información debía mantenerse intacta. Esta postura comenzó a aplicarse cuando se tuvo la necesidad de comprimir imágenes, audio y video. En el caso de las imágenes, las limitaciones del ojo humano juegan un factor importante en todo esto ya que, aunque es capaz de procesar grandes cantidades de información, hay detalles en la imagen codificada en píxeles que no alcanza a distinguir por lo que era aceptable perder parte de los datos originales, siempre y cuando se retuviera suficiente información para que el ojo pudiera reconocer la imagen en cuestión.

El propósito de este capítulo es tratar de explicar una técnica de compresión de imágenes basada en geometrías fractales, utilizando todo el cuerpo matemático desarrollado en el capítulo anterior y describir el algoritmo utilizado por Barnsley, ya que este algoritmo es adecuado para los fines de este trabajo. Desde que Barnsley encontró este método, se han desarrollado mejoras al algoritmo utilizando herramientas como redes de neuronas y algoritmos genéticos [28]. Por otro lado, existen artículos en los que se discute la eficiencia de diferentes algoritmos de compresión fractal utilizando sistemas de funciones iteradas [24], [23].

5.1. Bases de la compresión fractal de imágenes

La compresión fractal de imágenes matematiza las imágenes del mundo real para describir los patrones y estructuras “que viven” en ellas. Iniciemos con ideas básicas sobre este tipo de compresión.

Imaginemos una fotocopidora fuera de lo común, nuestra fotocopidora tiene propiedades especiales, cada vez que se fotocopia una imagen, ésta se reduce a la mitad y se reproduce tres veces en la copia, si la copia que acabamos de hacer la volvemos a fotocopiar y repetimos este proceso una y otra vez, lo que observamos es que todas las copias parecen converger a una misma imagen; sin importar la imagen inicial, podemos imaginarnos que la imagen final será afectada únicamente por la posición y la orientación de la imagen inicial. Llamaremos a la imagen final el *atractor* de nuestra fotocopidora (ver Figura 5.1).

La compresión fractal de imágenes fue desarrollada por Barnsley, conocido por haber comercializado este esquema, sugirió que el ver las imágenes como colecciones de transformaciones, podría llevarnos a una compresión de la imagen. Su argumento fue el siguiente: el helecho en la Figura 5.2 tiene una geometría intrincada y complicada pero es generado solamente por cuatro transformaciones afines y la probabilidad de ocurrencia de cada una de ellas. Cada transformación afín w_i es definida por seis números, $a_i, b_i, c_i, d_i, e_i, f_i$ y su probabilidad p_i , los cuales para su almacenamiento no requieren de mucha memoria en la computadora. Guardar la imagen del helecho requiere de mucha más memoria que guardar el conjunto de números antes mencionados. Si deseamos ver la imagen del helecho utilizando este conjunto de números, sólo tenemos que utilizar el método de la fotocopidora para generar

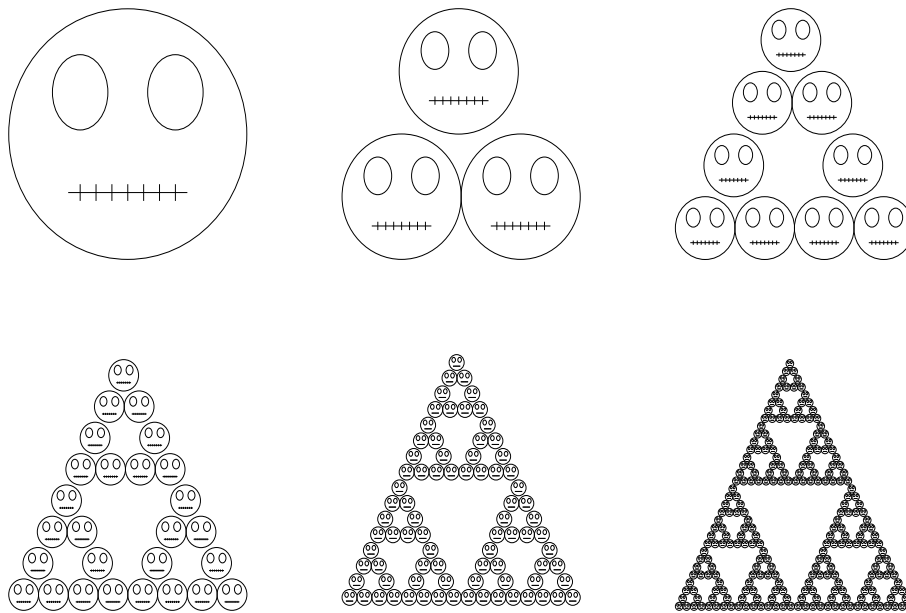


Figura 5.1: Proceso del algoritmo de la fotocopiadora para una imagen cualquiera, escalando la imagen a un medio y copiandola tres veces, como resultado de 5 iteraciones se converge al triángulo de Sierpinski.

su atractor, es decir convergerá a la imagen del helecho. Ahora supongamos que tenemos una imagen arbitraria, puede ser cualquier foto de algún álbum familiar; si existe un número finito de transformaciones que puedan generar la imagen de esa foto, entonces podemos guardarla en la memoria de la computadora de una forma compacta. El truco es encontrar las transformaciones y probabilidades adecuadas como en el helecho, esto se discutirá con detalle en las siguientes secciones.

El esquema descrito anteriormente puede ser fractal en muchos sentidos. Una imagen es guardada como una colección de transformaciones, esto tiene muchas implicaciones. Por ejemplo, el helecho de Barnsley es un conjunto que tiene detalle en cualquier escala por lo que su decodificación conserva los detalles, es decir si se hace una ampliación al helecho, digamos en un factor de 2, la imagen que resulta es dos veces más grande y sin embargo no pierde calidad en los detalles como sucede con otros formatos de imá-



Figura 5.2: Helecho de Barnsley generado por cuatro transformaciones.

genes¹, por lo tanto la imagen puede ser decodificada en cualquier tamaño. La resolución que se necesita para decodificar en tamaños muy grandes es generado automáticamente por las transformaciones. Podemos preguntarnos si esta nueva resolución es la adecuada; es decir si decodificamos una imagen de una persona en un tamaño más grande ¿podremos ver las células de la piel o tal vez los átomos? la respuesta es no. El detalle no está del todo relacionado con la resolución de la imagen cuando fue digitalizada. En ampliaciones grandes las transformaciones codificarán imágenes que están lejos de parecerse a la original. Sin embargo, en algunos casos, el detalle es realista en pequeñas ampliaciones de la original.

5.1.1. La mejor compresión fractal de imágenes

Los métodos para la compresión de imágenes pueden ser evaluados utilizando su tasa de compresión, que es la cantidad de memoria requerida para guardar una representación de la imagen en forma comprimida, dividida entre la cantidad de memoria requerida para guardar una imagen como una colección de píxeles. La tasa de compresión para el esquema fractal es difícil de medir, dado que la imagen puede ser decodificada a cualquier escala. Si decodificamos

¹JPEG, Postscript, RAW, PGM, BMP, etc

el triángulo de Sierpinski en, digamos, dos veces su tamaño, entonces podemos tener cuatro veces más la tasa de compresión, pues requerimos cuatro veces más píxeles para guardar la imagen descomprimida.

Ya se ha dicho que se han escrito artículos en los que se discute la eficiencia de diferentes algoritmos de compresión fractal, comparando la tasa de compresión de diferentes imágenes. Por ejemplo, han comparado las diferentes tasas de compresión entre los algoritmos hechos por Barnsley, Yuval Fisher y Mark Nelson [23], de donde ha resultado mejor la técnica utilizada por Fisher, que sacrifica tiempo por tasa de compresión; el algoritmo hecho por Nelson es más veloz que el de Fisher pero sacrifica tasa de compresión mientras que el algoritmo de Barnsley gasta tiempo en el proceso de compresión y su tasa es deficiente en comparación con los anteriores además de que la pérdida de información es notoria en la imagen descomprimida.

Los propósitos de este trabajo están lejos de una discusión de la eficiencia de los diferentes algoritmos de compresión fractal, por lo que utilizamos el algoritmo de Barnsley como una herramienta de clasificación que se acomoda a los objetivos primordiales de este trabajo.

5.1.2. Autosemejanza en imágenes

Para empezar a discutir la compresión de imágenes necesitamos un modelo matemático de una imagen. Generaremos una función $z = f(x, y)$ tomando la altura como la intensidad de gris que corresponde a un píxel en la posición (x, y) , con las partes más altas indicando el color blanco, y las más bajas acercándose al color negro. Este es nuestro modelo para una imagen, tomando en cuenta que la superficie es generada por la unión de las alturas de una malla finita; entonces, cuando queramos hablar de una imagen nos referiremos a la función $f(x, y)$, la cual asigna el nivel de gris para cada punto (x, y) . Cuando estemos trabajando con una imagen digitalizada y almacenada en una computadora, $f(x, y)$ será tomada de forma discreta y asignará un valor constante sobre cada píxel.

Una imagen típica de una cara no contiene el tipo de auto-semejanza que podemos encontrar en el triángulo de Sierpinski. La imagen no parece contener transformaciones afines de sí misma pero, de hecho, la imagen contiene un tipo diferente de autosemejanza. En una imagen podemos encontrar regiones que son similares a diferentes escalas, tal vez idénticas en porciones;

por ejemplo, si la reflejamos en un espejo o si la rotamos 180 grados. La distinción del tipo de semejanza que vemos en la imagen de un helecho, es que la imagen está formada de copias de sí misma bajo transformaciones afines apropiadas. En una imagen del mundo real encontramos que está formada por copias de partes de ella misma propiamente transformadas. Estas partes no son copias idénticas sí mismas bajo transformaciones afines. Esto quiere decir que la imagen que codificamos como un conjunto de transformaciones no será una copia idéntica de la imagen original, pero sí una aproximación de ella.

Finalmente ¿en qué tipo de imágenes podemos esperar este tipo de autosemejanza local?, resultados experimentales sugieren que la mayoría de las imágenes que uno puede ver, pueden ser comprimidas tomando ventaja de este tipo de semejanza; por ejemplo, imágenes de árboles, casas, caras, montañas, nubes, etc. Sin embargo la existencia de esta autosemejanza local y la habilidad de un algoritmo para detectar esto, son temas que trataremos en las siguientes secciones de este capítulo.

5.2. Metodología para la compresión y descompresión fractal de imágenes

Los conceptos que se han tratado anteriormente en este capítulo serán de gran utilidad para comenzar a adentrarse en todo lo que encierra el algoritmo hecho por Barnsley. El marco teórico que se utiliza son los sistemas de funciones iteradas y, de manera central, el teorema del collage.

5.2.1. Sistemas de funciones iteradas locales

Se definirá un mecanismo con el propósito de llegar a una compresión automática de imágenes utilizando los IFS como se han definido en el capítulo anterior; se comenzará por describir el esquema para imágenes binarias con el fin de describir la metodología básica para después llegar a nuestro objetivo que son las imágenes en escala de grises.

Se han definido los mapeos de contracción y los IFS, ahora retomando un poco de estas ideas, se definirán los mapeos de contracción locales y los IFS locales.

Definición 22 Sea R un subconjunto no vacío de \mathbb{R}^2 . Sea $w : R \rightarrow \mathbb{R}^2$ una transformación y s un número real en el intervalo $0 \leq s < 1$, si

$$d(w(x), w(y)) \leq s d(x, y) \text{ para toda } x, y \text{ en } R$$

entonces w es llamado un mapeo de contracción local sobre (\mathbb{R}^2, d) . El número s es el factor de contracción para w .

Definición 23 Sea $w_i : R_i \rightarrow \mathbb{R}^2$ un mapeo de contracción local sobre (\mathbb{R}^2, d) , con factor de contracción s_i , para $i = 1, 2, \dots, N$, en donde N es un entero positivo. Entonces

$$\{w_i : R_i \rightarrow \mathbb{R}^2 : i = 1, 2, \dots, N\}$$

es llamado sistema de funciones iteradas local (IFS local). El número $s = \max\{s_i : i = 1, 2, \dots, N\}$ es llamado factor de contracción para el IFS local.

Las ideas de las definiciones anteriores no difieren en mucho de las expuestas en el capítulo anterior, lo que debe tenerse en cuenta es que ahora las transformaciones se están aplicando a una parte específica R_i y no a un conjunto arbitrario, como se había definido anteriormente; para cada una de ellas, se tiene un mapeo de contracción con un factor de contracción s_i ; entonces, para el IFS local, el factor de contracción s es el máximo número que toman las s_i .

Las definiciones 22 y 23 son de gran importancia para la compresión fractal; a continuación daremos una explicación de la forma de uso de éstas.

Sea S el conjunto de todos los subconjuntos de \mathbb{R}^2 . Ahora podemos definir el operador $W_{local} : S \rightarrow S$ de la siguiente forma:

$$W_{local}(B) = \bigcup_{i=1}^N w_i(R_i \cap B), \text{ para cada } B \in S,$$

con las condiciones adecuadas podemos tratar a W_{local} como un operador sobre el espacio de Hausdorff $\mathcal{H}(\mathbb{R}^2)$ que consiste de subconjuntos no vacíos compactos de \mathbb{R}^2 ; entonces se puede encontrar que W_{local} es un mapeo de contracción sobre ciertos subconjuntos compactos de $\mathcal{H}(\mathbb{R}^2)$, con factor de contracción s con respecto a la métrica de Hausdorff.

Diremos que un subconjunto no vacío A de \mathbb{R}^2 es un atractor o conjunto invariante del IFS local si

$$W_{local}(A) = A$$

Un IFS local puede no tener atractores o puede tener varios, esto depende de la forma del conjunto B en W_{local} ya que si tomamos un B adecuado, entonces podemos encontrar el atractor completo o parte de él; por lo tanto, en la unión de estas partes encontraremos el atractor completo del IFS local; por otra parte, si $W_{local}(B) = \emptyset$ no podremos encontrar el atractor. Cuando hablemos del atractor de W_{local} , nos referiremos a la unión de todos los atractores que pueden ser generados por W_{local} .

A continuación se dará un método que garantiza encontrar el atractor del IFS local.

Sea $\{w_i : R_i \rightarrow \mathbb{R}^2 : i = 1, 2, \dots, N\}$ un IFS local y supongamos que los conjuntos R_i son compactos. Entonces podemos definir una sucesión de subconjuntos compactos de \mathbb{R}^2 , $\{A_n : n = 0, 1, 2, 3, \dots\}$ como sigue

$$A_0 = \mathbb{R}^2$$

$$A_n = \bigcup_{i=1}^N w_i(R_i \cap A_{n-1}) \text{ para } n = 1, 2, 3, \dots$$

Es fácil observar que

$$A_0 \supset A_1 \supset A_2 \supset A_3 \supset \dots$$

Entonces $\{A_n : n = 0, 1, 2, 3, \dots\}$ es una sucesión decreciente de conjuntos compactos.

En particular, existe un conjunto compacto $A \subset \mathbb{R}^2$ tal que

$$\lim_{n \rightarrow \infty} A_n = A$$

y

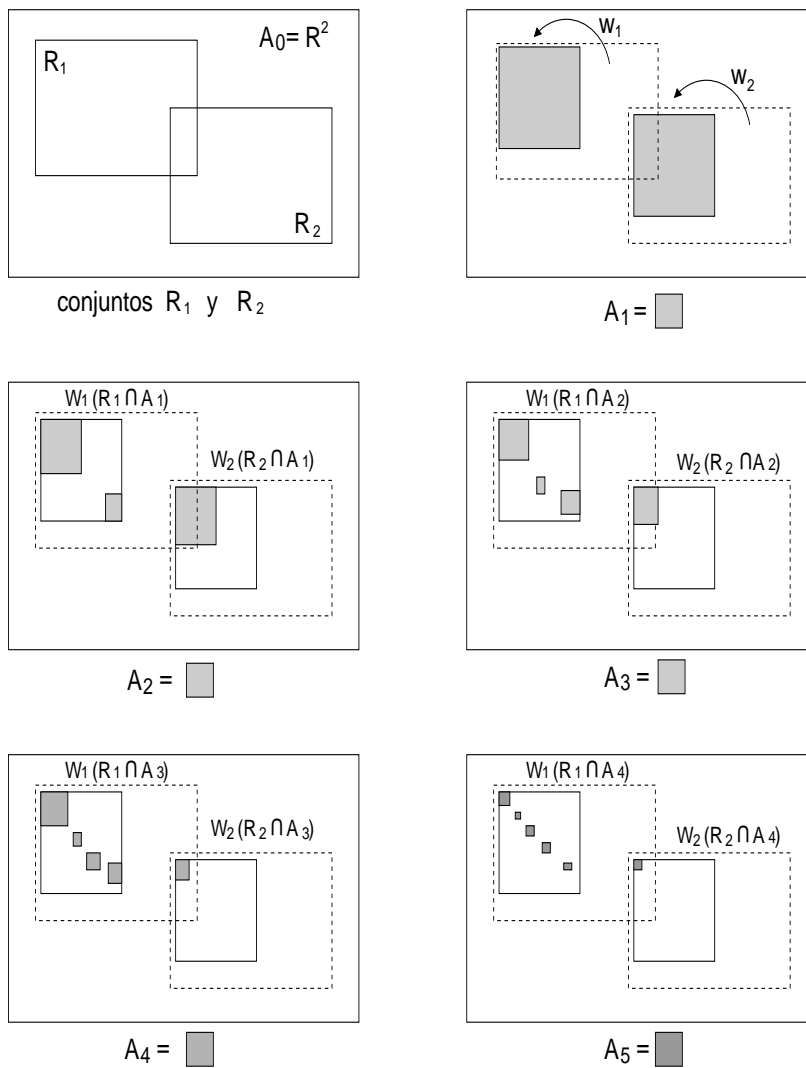


Figura 5.3: Una sucesión decreciente de conjuntos convergiendo a un atractor de un IFS local.

$$A = \bigcup_{i=1}^N w_i(R_i \cap A) = W_{local}(A),$$

tenemos que A es el atractor para el IFS local.

En la Figura 5.3 se muestran los primeros cinco elementos de la sucesión $\{A_n : n = 0, 1, 2, 3, \dots\}$ generada por el operador W_{local} , el elemento inicial A_0

es \mathbb{R}^2 ; esto garantiza la existencia del atractor siempre y cuando los elementos A_n sean diferentes del conjunto vacío y se definan dos subconjuntos R_1 y R_2 . Para contruir A_1 aplicamos el operador W_{local} a A_0 y el conjunto resultante se representa con la parte sombreada y para A_2 se vuelve a aplicar el operador y así sucesivamente para un número suficientemente grande de iteraciones para obtener el atractor. En la imagen se observa cómo en el subconjunto R_2 se converge a un punto en su parte superior izquierda, mientras que en R_1 se converge a varios puntos que se aproximan a una curva. El conjunto A_n nunca es vacío por la característica de que los conjuntos $w_i(R_i)$ siempre están contenidos en los conjuntos R_i .

5.2.2. El teorema del collage para IFS locales

Para poder hacer la compresión de una imagen se necesita encontrar un conjunto de transformaciones afines tales que su atractor sea lo más parecido a ella, esto se conseguirá utilizando el teorema del collage para IFS locales.

La idea es dividir la imagen mediante una cuadrícula; para cada cuadro en ella se debe encontrar un mapeo de contracción tal que, al aplicarlo a una cierta parte de la imagen, el resultado sea lo más parecido posible al cuadro en cuestión.

Empecemos por definir a una imagen mediante $\square = \{(x, y) \in \mathbb{R}^2 : a \leq x \leq b, c \leq y \leq d\}$, es decir un rectángulo contenido en \mathbb{R}^2 que se utilizará como el conjunto que contiene a todas las imágenes.

Sea G una imagen binaria, subconjunto de \square ($G \subset \square$); se cubre a G con pequeños cuadrados que se llamarán *dominios* (ver Figura 5.4) y se denotará mediante D_i para $i = 1, 2, \dots, N$. Entonces, para cada D_i , buscaremos una transformación afín $w_i : R_i \rightarrow \square$ con factor de contracción s , tal que

$$w_i(R_i) = D_i$$

y

$$w_i(R_i \cap G) \simeq D_i \cap G \tag{5.1}$$

Donde R_i son subconjuntos que cubren por completo a G y se llamarán *rangos* (ver Figura 5.4).

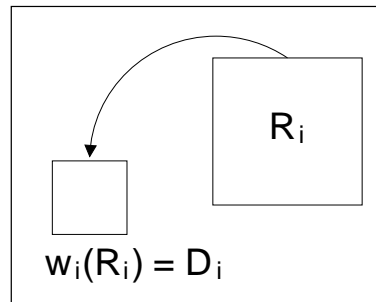


Figura 5.4: La transformación w manda R_i a un dominio D_i que es de menor tamaño.

En la ecuación (5.1) el criterio de aproximación puede estar dado por la distancia de Hausdorff entre $R_i \cap G$ y $D_i \cap G$, debemos ser capaces de minimizarla tanto como se pueda, es decir, conseguir que para $\epsilon > 0$ suficientemente pequeña se cumpla que

$$h(w_i(R_i \cap G), D_i \cap G) < \epsilon, \text{ para } i = 1, 2, \dots, N.$$

En la práctica, podemos utilizar la distancia euclidiana y las transformaciones las podemos escoger de la siguiente forma

$$w \begin{pmatrix} x \\ y \end{pmatrix} = 0.5A \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

en donde A es una simetría afín del Cuadro 5.1 y los R_i son dos veces más grandes que el tamaño de los D_i .

Cada bloque D_i es del mismo tamaño, entonces el IFS local está totalmente especificado por la coordenadas (D_x, D_y) que pertenecen a la esquina inferior izquierda de cada dominio, por las coordenadas (R_x, R_y) que corresponden a la esquina inferior izquierda de cada rango y un entero que indica la simetría afín A escogida. El resultado de este proceso le llamaremos el código del IFS local.

Simetría	Matriz	Descripción
0	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	Identidad
1	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	Reflexión en el eje y
2	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	Reflexión en el eje x
3	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	Rotación 180 grados
4	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	Reflexión en la línea $y = x$
5	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	Rotación 90 grados
6	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	Rotación 270 grados
7	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$	Reflexión en la línea $y = -x$

Cuadro 5.1: Se muestra el conjunto de las 7 transformaciones para el cuadrado. Las rotaciones son en el sentido de las manecillas del reloj.

5.2.3. Transformación fractal para imágenes binarias

El siguiente algoritmo describe los pasos principales de la compresión fractal automática de imágenes binarias, este algoritmo fue patentado por Barnsley [5].

1. Se introduce una imagen G , subconjunto de $\square \subset \mathbb{R}^2$.
2. Se cubre G con bloques dominio D_i como se ilustra en la Figura 5.5. El conjunto total de bloques dominio $\{D_i : i = 1, 2, \dots, n\}$ deben cubrir G . Los bloques D_i no se traslapan unos con otros.
3. Se introduce una colección de posibles bloques rango R_i , que están contenidos en \square , pero son tales que la intersección de ellos con G es diferente del vacío. Los rangos son cuadrados tales que sus lados son dos veces más grandes que los lados de los dominios. Las posibles coordenadas (R_x, R_y) de la esquina inferior izquierda de cada

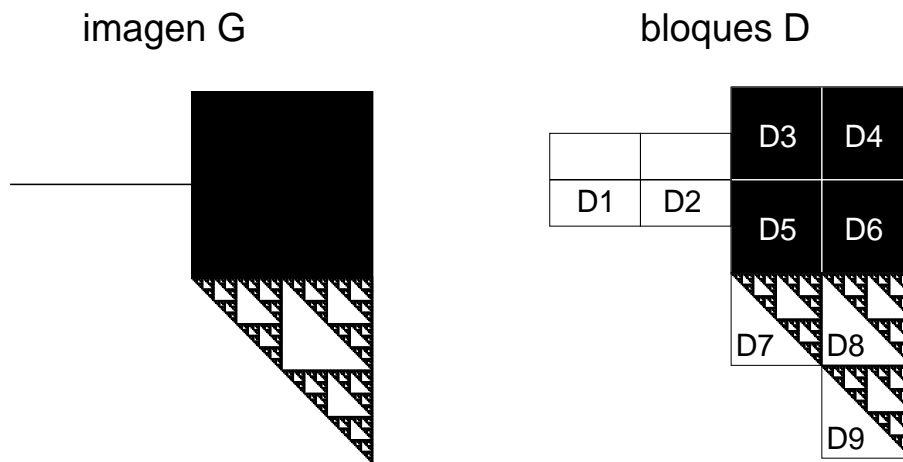


Figura 5.5: La imagen G es cubierta por dominios D_i de forma adecuada.

R_i están restringidas a un conjunto L . Correspondientemente, se define una colección \mathcal{T} de transformaciones afines contractivas locales, como mapeos que van de los bloques rango R_i a los bloques dominio D_i . Esto para $i = 1, 2, \dots, n$, es decir,

$$\mathcal{T}_i = \{w(D_i, R_x, R_y, j) : (R_x, R_y) \in L; j = 0, 1, 2, \dots, 7\}$$

en donde $w(D_i, R_x, R_y, j)$ es la transformación afín con propiedad de contracción, que va de R_i a D_i mediante la regla

$$0.5A(j) \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

en donde $A(j)$ denota la j -ésima simetría del Cuadro 5.1.

4. Para llevar a cabo el proceso de la transformación fractal, para cada i , escogemos $w_i \in \mathcal{T}$ de manera que la distancia de Hausdorff $h(w_i(R \cap G), D_i \cap G)$ sea mínima.

Para cada bloque dominio D_i se trata de escoger un bloque rango R_i y una simetría correspondiente, de manera que la parte R_i transformada

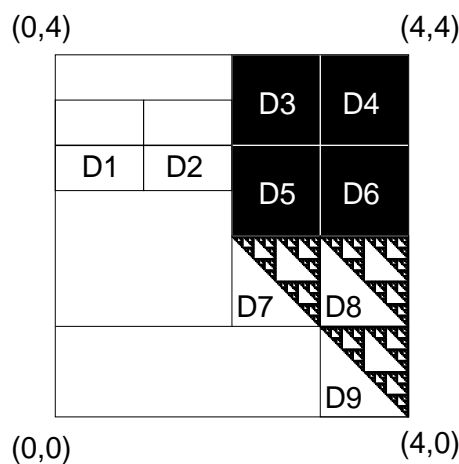


Figura 5.6: Se escoge un conjunto de coordenadas (x,y) adecuado para la imagen G .

de la imagen sea la más parecida a la parte de la imagen que está en el dominio D_i (ver Figuras 5.6 y 5.7).

El conjunto $W_{local} = \cup w_i(R \cap G)$ es el conjunto de todas las contracciones que se escogieron de acuerdo al criterio de la mínima distancia (la imagen más parecida) aplicadas a ciertos rangos; a este conjunto se le llama *el Collage de la imagen G* correspondiente al IFS local y la distancia

$$h(w_i(R \cap G), D_i \cap G)$$

es llamado el *error del collage*.

- Ahora se tienen los datos que hacen que la compresión fractal de imágenes funcione, se tienen las coordenadas (R_x, R_y) de la esquina inferior izquierda de cada bloque rango R_i y sabemos a qué bloque dominio se dirige mediante las coordenadas de la esquina inferior izquierda de cada bloque dominio (D_x, D_y) y el número de la transformación afín que se ha escogido. Todo este conjunto de datos forman la salida de la compresión y nos servirá para formar el código de un IFS local. El Cuadro

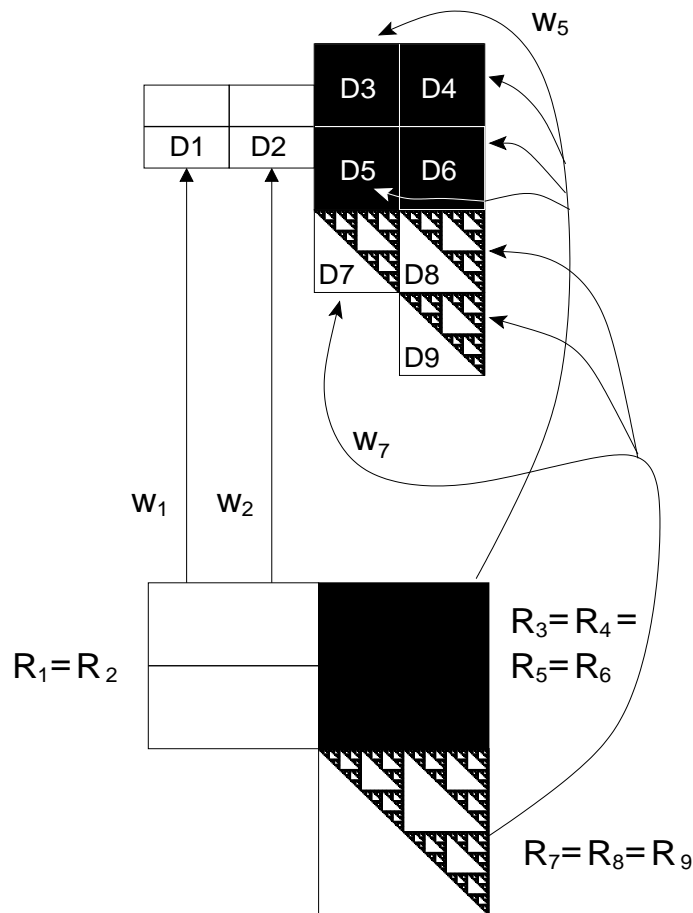


Figura 5.7: Se escoge un R_i y una transformación w_i para un dominio específico D_i , tal que la distancia de Hausdorff $h(w_i(R_i \cap G), D_i \cap G)$ sea mínima.

5.2 muestra el formato del código, en donde $\text{map}\#$ es un número i que indica la aplicación del mapeo i -ésimo (w_i).

6. Por último se aplica un algoritmo de compresión sin pérdida de información² al código del IFS local.

El siguiente paso es hacer la descompresión de la imagen comprimida, esto es de alguna forma más sencillo dado que tenemos disponible la definición de un

²Conocido como *lossless data algorithm*, tales como *tar*, *gzip*, etc.

map#	D_x	D_y	R_x	R_y	simetría
1	0	2.5	0	2	0
2	1	2.5	0	2	0
3	2	2	2	2	0
4	3	2	2	2	0
5	2	3	2	2	0
6	3	3	2	2	0
7	2	1	2	0	0
8	3	1	2	0	0
9	3	0	2	0	0

Cuadro 5.2: Código del IFS local para la imagen G .

IFS local, ya que para descomprimir una imagen, lo único que debemos hacer es aplicar el algoritmo de la fotocopidora explicado detalladamente en el capítulo anterior. De manera muy rápida, lo que se intenta hacer es reconstruir el operador W_{local} , visto en la sección anterior y calcular las aproximaciones para $\lim_{n \rightarrow \infty} W^{on}(D)$. Este proceso se puede observar en la Figura 5.8.

El algoritmo presentado anteriormente se puede utilizar sobre cualquier imagen digitalizada en forma binaria, pero es posible hacer una extensión de él y utilizarlo sobre imágenes digitalizadas en escalas de grises, de esto se hablará en la siguiente sección.

5.2.4. Un IFS local asociado a un esquema de compresión para imágenes en escala de grises

Anteriormente se describió un esquema para la compresión fractal de imágenes binarias, ahora introduciremos una estructura apropiada para hacer de forma automática la compresión fractal de imágenes en escala de grises.

Definiremos un espacio $\mathbf{X} = \square \times I$, donde $I = [a, b] \subset \mathbb{R}$, en particular este intervalo representará los posibles valores de la intensidad de grises que se encuentran en el intervalo $[0, 255]$, en donde 0 es el valor para el color negro y 255 es el valor para el color blanco. En la Figura 5.9 se observa un elemento de este espacio, el cual es una superficie que muestra diferentes intensidades

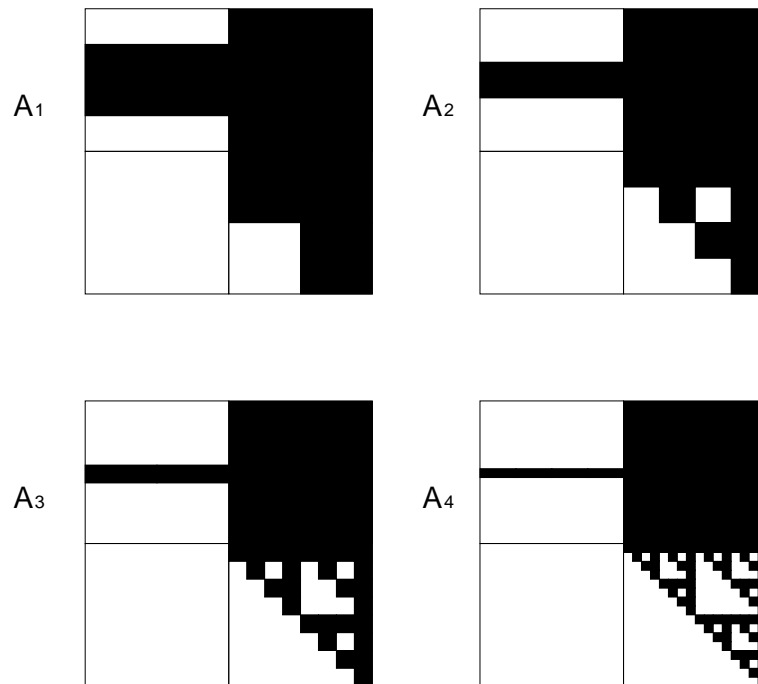


Figura 5.8: El código del IFS local es usado para descomprimir una imagen mediante la iteración del operador W_{local} .

de grises, los puntos más altos se ven en color blanco y los más bajos en color negro.

Convertiremos \mathbf{X} en un espacio métrico completo, asociándole la métrica euclidiana d , entonces tenemos (\mathbf{X}, d) . Sea $\mathcal{H}(\mathbf{X})$, el espacio de Hausdorff aplicado a \mathbf{X} con la métrica de Hausdorff h .

Consideremos a $\square \times [a, b]$ como una colección finita de conjuntos $\mathbf{D}_i, i = 1, 2, \dots, M$, tal que

$$\square \times [a, b] = \bigcup_{i=1}^M \mathbf{D}_i$$

en donde $\mathbf{D}_i \cap \mathbf{D}_j = \emptyset$ para $i \neq j$. Para cada i , sea $v_i : \mathbb{R} \rightarrow \mathbb{R}$ una transformación de contracción con factor de contracción s , con $0 \leq s < 1$, que cumple con

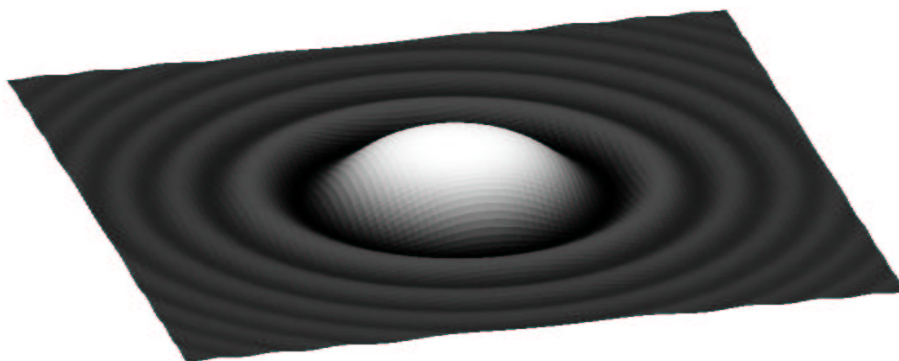


Figura 5.9: Elemento del espacio $\mathbf{X} = \square \times [a, b]$

$$|v_i(z_1) - v_i(z_2)| < s|z_1 - z_2| \text{ para todo } z_1, z_2 \in \mathbb{R}, i = 1, 2, \dots, M$$

Esto nos dice que la distancia entre dos números reales se hace menor bajo la transformación v_i en un factor s y se ilustra en la Figura 5.10.

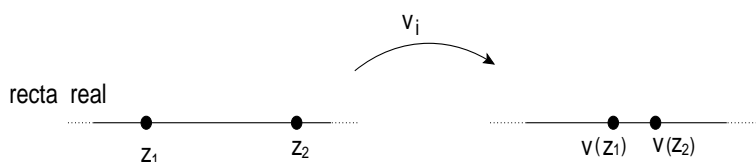


Figura 5.10: Se observa el efecto que tiene la transformación de contracción v_i sobre los elementos z_1 y z_2 .

En la sección anterior se definió el operador W_{local} que se aplica sobre imágenes binarias en el plano; ahora se introducirá una variable más haciendo referencia a la escala de grises, esto nos lleva a trabajar en un, espacio de tres dimensiones, en particular este espacio será \mathbf{X} , ya que esta acotado por

el intervalo $[0, 255]$. A continuación definiremos la estructura de las transformaciones de contracción para el IFS local en tres dimensiones.

$$\{\mathbf{w}_i : \mathbf{R}_i \rightarrow \mathbf{X} : i = 1, 2, \dots, N\}$$

en donde $\mathbf{X} = \square \times [a, b]$, $\mathbf{R}_i = R_i \times [a, b]$ y $\mathbf{w}_i : \mathbf{R}_i \rightarrow \mathbf{X}$ se define como

$$\mathbf{w}_i(x, y, z) = (w_i(x, y), v_i(z)) \text{ para todo } (x, y, z) \in \mathbf{R}_i$$

En donde w_i , R_i y D_i se definen como en el esquema para imágenes binarias.

Definamos al operador $\mathbf{W}_{local} : \mathcal{H}(\mathbf{X}) \rightarrow \mathcal{H}(\mathbf{X})$ como

$$\mathbf{W}_{local} = \bigcup \mathbf{w}_i(\mathbf{R}_i \cap B)$$

donde B es un elemento de \mathbf{X} . En la Figura 5.11 se muestra el proceso del operador \mathbf{W}_{local} , sobre una imagen B ; el cubo representa un subconjunto \mathbf{R}_i , la intersección de B con \mathbf{R}_i está dada por la superficie dentro del cubo, que nunca es vacía por que se han tomado los \mathbf{R}_i de manera adecuada. A esta superficie se le aplica la transformación de contracción \mathbf{w}_i ; el resultado de aplicar esta transformación, es la actualización del dominio \mathbf{D}_i correspondiente. Recordemos que la transformación \mathbf{w}_i se compone de w_i que actúa sobre el plano xy y la transformación v_i que actúa sobre las escalas de grises. Esto se repite para cada \mathbf{D}_i de la imagen B . Al finalizar tendremos una imagen en escala de grises distinta a la inicial.

Al igual que el \mathbf{W}_{local} para imágenes binarias podemos construir una sucesión de subconjuntos compactos de \mathbf{X} , $\{A_n : n = 0, 1, 2, 3, \dots\}$ como sigue

$$A_0 = \square \times [a, b]$$

$$A_n = \bigcup_{i=1}^N \mathbf{w}_i(\mathbf{R}_i \cap A_{n-1}) \text{ para } n = 1, 2, 3, \dots$$

El elemento A_0 es una imagen inicial arbitraria de \mathbf{X} , a la cual se le aplica el operador \mathbf{W}_{local} , conforme se avanza en la sucesión, los subconjuntos \mathbf{D}_i convergen a un conjunto invariante.

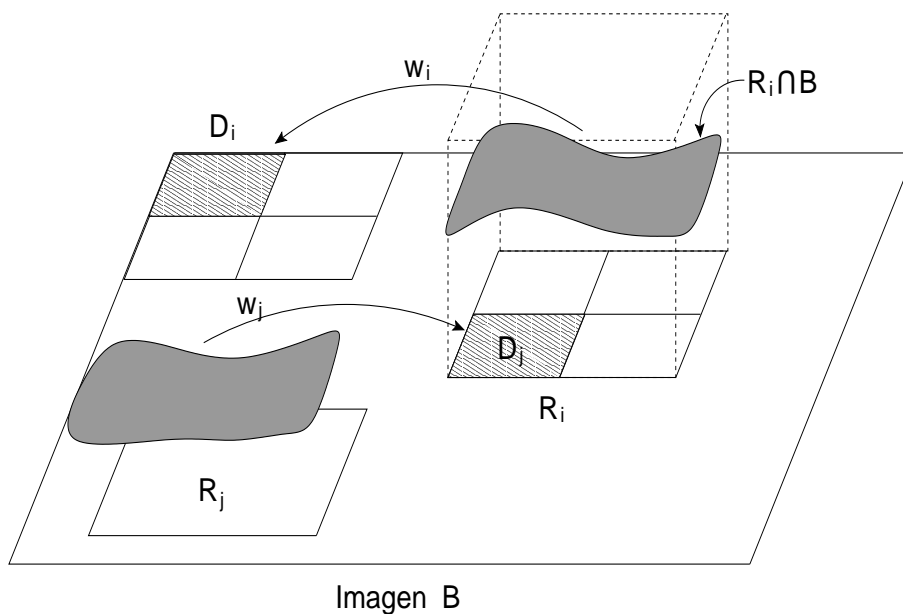


Figura 5.11: Proceso para el operador \mathbf{W}_{local}

En particular, existe un conjunto compacto $A \subset \mathbf{X}$ tal que

$$\lim_{n \rightarrow \infty} A_n = A$$

y

$$\mathbf{W}_{local}(A) = A.$$

Entonces tenemos que A es el atractor para el IFS local.

5.3. Algoritmo para hacer la compresión y descompresión fractal de imágenes en escala de grises

Para poder hacer una compresión de una imagen en escala de grises usaremos transformaciones afines de la forma

$$\mathbf{w}_i = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & P \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} R_x \\ R_y \\ Q_i \end{bmatrix}$$

para que sea más claro el manejo de esta transformación dentro del algoritmo, podemos separarla en dos transformaciones de la siguiente manera

$$w_i(x, y) = (a_i x + b_i y + R_x, c_i x + d_i y + R_y)$$

$$v_i(z) = Pz + Q_i$$

donde los coeficientes a_i, b_i, c_i y d_i son las entradas de la matriz de transformación para el plano xy de acuerdo con las simetrías del Cuadro 5.1, con un factor de contracción de 0.5. El coeficiente P es un número positivo fijo que se encuentra en $0 < P \leq 0.5$. El parámetro Q_i regula la intensidad de grises en un R_i , de manera que ésta sea lo más parecida a la de su D_i correspondiente.

El objetivo de la compresión fractal es la búsqueda de los parámetros mencionados anteriormente, éstos formarán parte del código del IFS local, que posteriormente servirá para hacer la descompresión.

5.3.1. Algoritmo para la compresión fractal

1. Se lee una imagen en formato de escala de grises, es decir cada pixel representa un número entre 0 y 255.
2. Se escoge un sistema adecuado de coordenadas para la imagen, como se muestra en la Figura 5.12 (A). Se divide la imagen en dominios cuadrados D_i del mismo tamaño tal que cubran la imagen sin intersectarse. La misma imagen es dividida en rangos R_i del doble del tamaño de los D_i ya que el factor de contracción se definió anteriormente como 0.5, éstos pueden intersectarse entre sí. Su posición está determinada por las coordenadas (R_x, R_y) que pertenecen a la esquina inferior izquierda de cada rango, como se muestra en la Figura 5.12 (B).
3. Para un dominio específico D_i se debe encontrar un rango adecuado de la siguiente forma: para cada rango R_i , se calcula el valor Q_i de la función v_i mediante la diferencia del promedio entre su escala de grises y la escala del D_i ; posteriormente se le aplica a R_i cada una de las siete

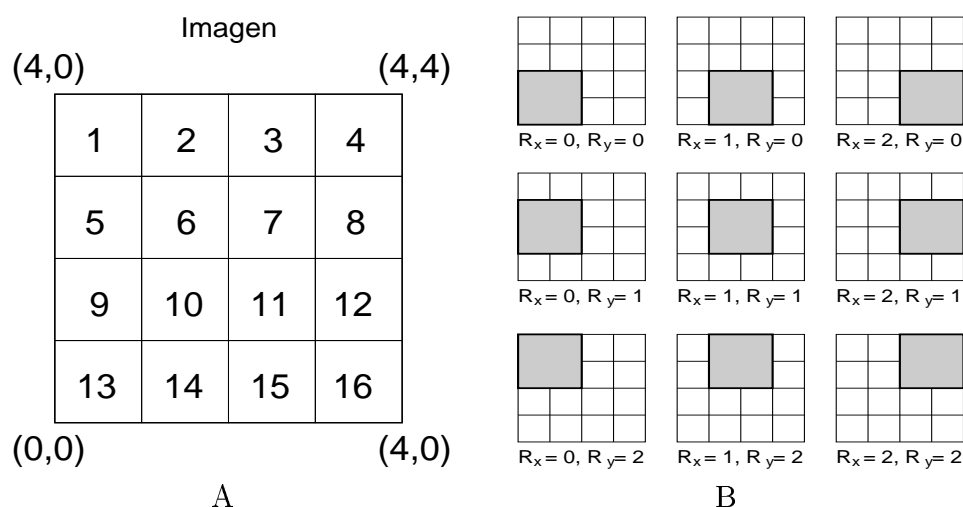


Figura 5.12: En (A) se divide una imagen en 16 dominios homogéneos sin intersectarse, en (B) se muestra la misma figura con sus 9 posibles rangos y respectivas coordenadas.

simetrías del Cuadro 5.1 para encontrar una transformación tal que la distancia entre el R_i transformado y D_i sea mínima; en particular, en el algoritmo de Barnsley se utiliza la distancia euclidiana. Cuando realicemos este proceso sobre todos los rangos de la imagen, obtendremos el mejor R_i para el dominio D_i (ver Figura 5.13).

Con esto obtenemos un código para cada D_i de la forma

$$\boxed{R_x \quad R_y \quad \text{simetría} \quad Q}$$

podemos suprimir las coordenadas de los dominios ya que la posición de ellos está determinada por el número del renglón del código, por tal motivo el tamaño de los D_i debe ser constante.

4. Se almacenan los códigos de los D_i en un archivo, su tamaño en bytes será menor que el de la imagen del paso 1.

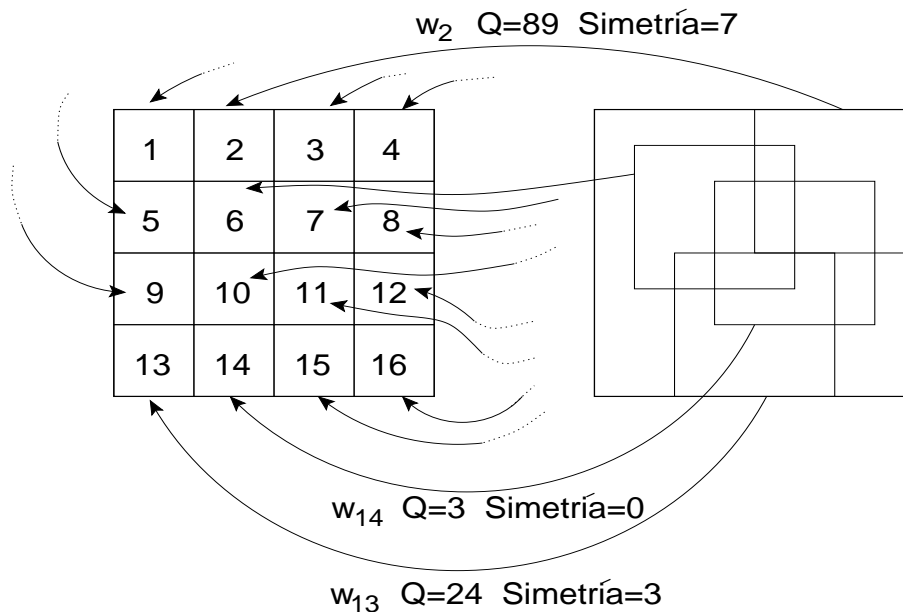


Figura 5.13: Representación bidimensional de la conclusión del paso 3, cada D_i ha obtenido su R_i , su simetría y su escala de gris Q_i .

5.3.2. Algoritmo para la descompresión fractal

1. Se lee el archivo que contiene el código de la compresión fractal y una imagen³ con las características del paso 1 del algoritmo de compresión.
2. Se define el mismo sistema coordenado que en el algoritmo de compresión; luego se divide la imagen en dominios y rangos como en el paso 2 del algoritmo de compresión.
3. El código se aplica en orden para saber a que D_i corresponde; por ejemplo, al primer código se le asocia el dominio D_1 , al dominio D_2 se le asocia el segundo código y así sucesivamente hasta el código n -ésimo que le corresponde el dominio D_n (ver Figura 5.12). Con la R_x y R_y del código encontramos el R_i al que se le aplicará la simetría y su intensidad de gris Q_i . El rango R_i transformado se sustituirá en lugar del dominio D_i (ver Figura 5.11). Cuando se termine de aplicar el código a todos los dominios, se tendrá una imagen modificada.

³La imagen puede ser la que se desee.

4. El paso anterior se repetirá para la imagen modificada y así sucesivamente, teniendo dos criterios para el término del algoritmo, cuando se cumplan las iteraciones preestablecidas o cuando la imagen ya no sea modificada por el proceso.

La imagen que resulte del algoritmo de descompresión, puede perder detalles en comparación con la imagen que generó el código en el algoritmo de compresión.

Los algoritmos presentados, son una forma simple de aplicar los conceptos teóricos desarrollados a lo largo de esta sección. Es posible hacer modificaciones que mejoren los procesos de compresión para encontrar las transformaciones adecuadas en un tiempo relativamente corto, mejorando así la calidad de la imagen descomprimida. Algunas sugerencias pueden ser, tomar diferentes factores de contracción para cada rango y extender el conjunto de transformaciones que se pueden elegir.

En la figura 5.14 se presenta un ejemplo de la compresión y descompresión de una imagen, con los algoritmos desarrollados anteriormente.

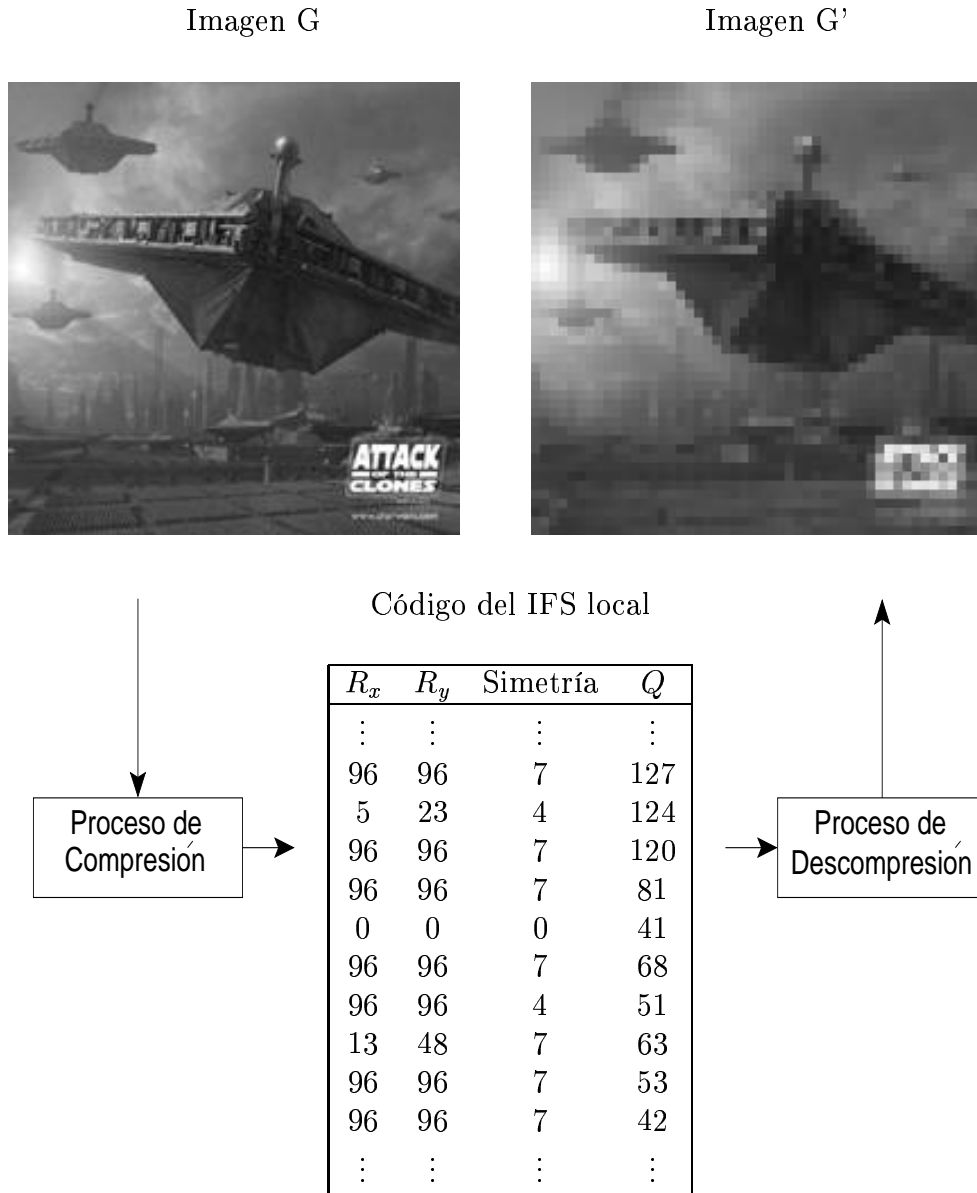


Figura 5.14: Se muestra un diagrama que uno los algoritmos de compresión y descompresión, se inicia con la imagen G pasando por el proceso de compresión y obteniendo su código del cual se muestra una parte en la tabla central, por último se aplica el proceso de descompresión y se obtiene una imagen G' que es parecida a G .

Capítulo 6

Aplicaciones a la clasificación de organismos a partir del DNA

A lo largo de los capítulos anteriores se han mostrado las bases teóricas de diferentes métodos que han sido de gran utilidad en distintos ámbitos de las matemáticas y las ciencias de la computación, nuestro propósito es conjuntar los métodos presentados para tener una herramienta de clasificación bien fundamentada, respaldada por teorías desarrolladas con bases matemáticas. El método que describiremos a continuación será para hacer una clasificación robusta de organismos a partir de su DNA.

Los organismos que se eligieron para su clasificación son los mostrados en el Cuadro 6.1, [12].

6.1. Método de clasificación de organismos

Como primer punto se deben conseguir secuencias de DNA en su formato *fasta*¹ que contienen cadenas de letras compuestas con el alfabeto A, G, C y T, que indican los cuatro nucleótidos. Convertiremos esta cadena en un formato binario con las propiedades de relación que existen entre los cuatro nucleótidos, estas son WS y YR (ver capítulo 2), la relación MK no se utilizará ya que las relaciones anteriores contienen la suficiente información para describir características principales del DNA de cada organismo. En el

¹Existen varios formatos que describen diferentes características de la secuencia de DNA.

90 Aplicaciones a la clasificación de organismos a partir del DNA

- (A) *Arabidopsis thaliana*. Pequeña planta perteneciente al grupo eucarionte, es fácilmente cultivable en cualquier región.
- (B) *Bacillus subtilis*. Organismo perteneciente al grupo de bacterias, altamente resistente a cambios físicos y se encuentra generalmente en el suelo.
- (C) *Caenorhabditis elegans*. Nemátodo de suelo que se encuentra en regiones templadas, longitud aproximada de 1mm, perteneciente al grupo de los eucariontes.
- (D) *Chlamydia trachomatis*. Parásito intracelular, relacionado con enfermedades de transmisión sexual en humanos, muy sensible a cambios de temperatura, algunas veces clasificado como bacteria y otras como perteneciente a un grupo suigeneris.
- (E) *Escherichia coli*. Bacteria que se encuentra en los intestinos de humanos y animales.
- (F) *Homo sapiens*. Especie terrestre que habita en casi todo el planeta, pertenece al grupo de los eucariontes.
- (G) *Methanococcus jannaschii*. Arqueobacteria acuática que se encuentra en chimeneas marinas de 2600 m de profundidad y con una presión de 200 atm.
- (H) *Sulfolobus solfataricus*. Arqueobacteria termofílica que se encuentra en los volcanes terrestres ricos en sulfuros y ácidos, a temperaturas entre los 70 y 90 grados C.
- (I) *Thermatoga maritima*. Bacteria que se encuentra en volcanes marinos con una temperatura de 80 grados C.
- (J) *Thermoplasma volcanium*. Arqueobacteria que crece óptimamente en temperaturas mayores a los 50 grados C.

Cuadro 6.1: Organismos utilizados para su clasificación. A cada organismo se le asocia una letra que sirve de clave para los Cuadros 6.4 y 6.5.

caso de YR nos interesa resaltar el tamaño de la molécula, mientras que en WS la cantidad de enlaces de hidrógeno con los que están unidos los nucleótidos, en ambos casos se reduce la información de la naturaleza química que distingue a cada una de las moléculas, esto nos permite manipular y estudiar cada secuencia utilizando las herramientas descritas a lo largo de este trabajo.

A continuación se presenta un ejemplo de estas conversiones:

```

Archivo fasta  ... AGTCGATCGACGATATGATAC ...
Conversión  $WS$   ... 010110011011000010001 ...
Conversión  $YR$   ... 001100110010010100101 ...

```

Se procede a utilizar el juego del caos circular sobre un polígono de 1024 lados acompañado del mapeo shift de Bernoulli como se explicó en el capítulo 3. Generaremos sucesiones de puntos siempre del mismo tamaño: es decir, cuando se utilice el juego del caos circular con secuencias binarias diferentes la cantidad de puntos sobre el atractor será la misma. Se han mostrado imágenes del atractor que se produce para el mapeo shift, el atractor para diferentes secuencias es el mismo y no es fácil discernir cómo se distribuyen los puntos. Para analizar este comportamiento recurriremos a la medida invariante. Cubriremos el atractor por completo con una malla en la cual cada celda representa una bola y se definirá un sistema de coordenadas adecuado para poder ubicar la posición de las celdas en la malla (ver Figura 6.1) .

Las bolas que contengan puntos serán subconjuntos de Borel. Sobre cada celda se calcula la cantidad de puntos que caen dentro en cada iteración, la medida invariante como se definió es un proceso infinito, pero mostraremos que al tener un número suficientemente grande de iteraciones la distribución de puntos empieza converger, ya que las diferencias entre una y otra iteración se hacen más pequeñas.

En la Figura 6.2 se muestra una bola sobre el juego del caos circular utilizando la secuencia del *Homo sapiens*, en el Cuadro 6.2 se muestra una sucesión de valores que representan una medida en la bola para diferentes cantidades de iteraciones, se observa cómo, al aumentar el número de iteraciones, la sucesión se acerca a un valor; para un número infinito de iteraciones a este valor se le conoce como la medida invariante de la bola, en el caso de las secuencias

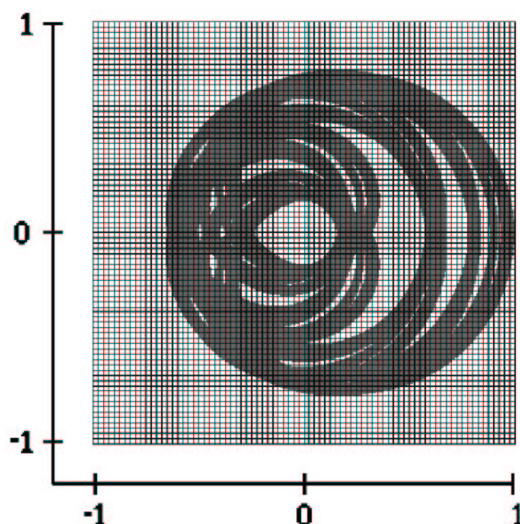


Figura 6.1: Malla sobre el atractor del juego del caos circular correspondiente a la secuencia del *Homo sapiens*. La malla cubre un cuadrado cuyos lados tienen longitud 2 y está centrado en el origen.

de DNA se puede conocer este valor ya que éstas se componen de un número finito de elementos, no es necesario utilizar la secuencia completa ya que la ocurrencia de una subcadena cualquiera esta dada por una probabilidad implícita dentro de la secuencia. Se utilizaron cadenas de tamaño 1,000,000 tomando en cuenta no exceder la longitud de la secuencia con menor número de elementos.

Sobre el conteo de puntos normalizado entre cero y uno de cada celda, se le asigna una intensidad de gris, las celdas que tienen mayor densidad de puntos están cerca del color blanco mientras que las celdas de menor densidad se acercan al color negro (ver Figura 6.3). A esta representación le llamaremos el histograma del sistema asociado a una escala de grises. Como

cada organismo tiene un histograma asociado, se le puede utilizar para hacer una clasificación midiendo qué tan distintas pueden ser sus distribuciones de puntos. Asociaremos estos histogramas con el método de la compresión fractal.

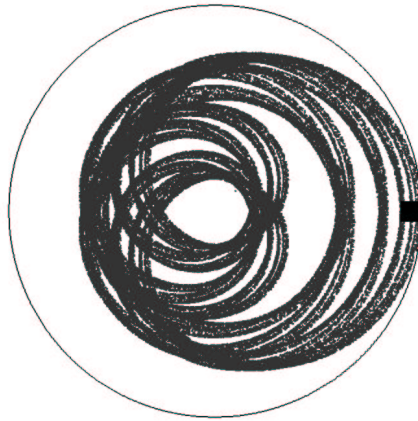


Figura 6.2: Representación de un bola sobre el atractor del juego del caos circular utilizando la secuencia binaria WS del cromosoma 21 del *Homo sapiens*.

n iteraciones	$\mathcal{N}(B_1, n)/n$
10000	0.0249
50000	0.0213
100000	0.0207
500000	0.020612
1000000	0.022051
10000000	0.024668

Cuadro 6.2: Medida invariante sobre una bola con distintas iteraciones.

Se ha descrito el método de compresión y descompresión fractal para imágenes en escalas de grises, trabajaremos con las imágenes que representan las distribuciones de puntos sobre el atractor, al hacer la compresión se genera un código (ver capítulo 5) el cual pertenece a un IFS local y éste será nuestra

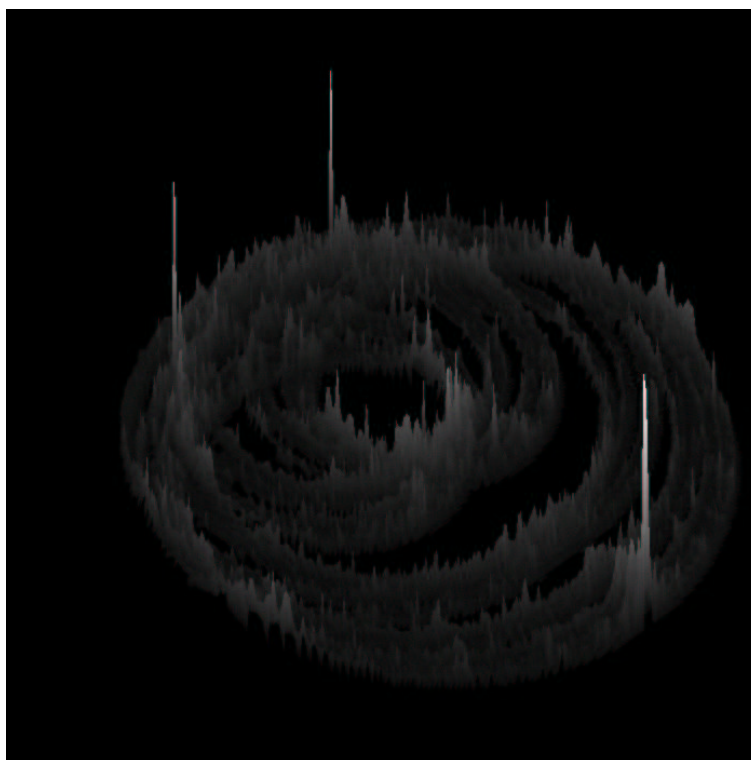


Figura 6.3: Medida invariante sobre el atractor del juego del caos circular con la secuencia correspondiente al *Homo Sapiens*.

materia prima para la clasificación de organismos. El código del IFS local nos brinda la información necesaria para reconstruir las tonalidades de grises que representan las densidades de puntos a través del proceso de descompresión. Como el IFS local fue construido apropiadamente por medio del proceso de compresión, el atractor del IFS será la imagen para la que fue construido².

En la Figura 6.4 se muestran cuatro diferentes histogramas sobre el juego del caos circular correspondientes a distintos organismos. Es notable la diferencia que se observa en las distribuciones de puntos, la máxima aglomeración de puntos se observa en las zonas blancas, cuando es muy pequeña la cantidad de puntos que caen en en la zona, ésta se ve de color negro. En la Figura 6.4 (B) se puede notar una sola región en donde la densidad de puntos es

²Ver teorema del collage en el capítulo 5.

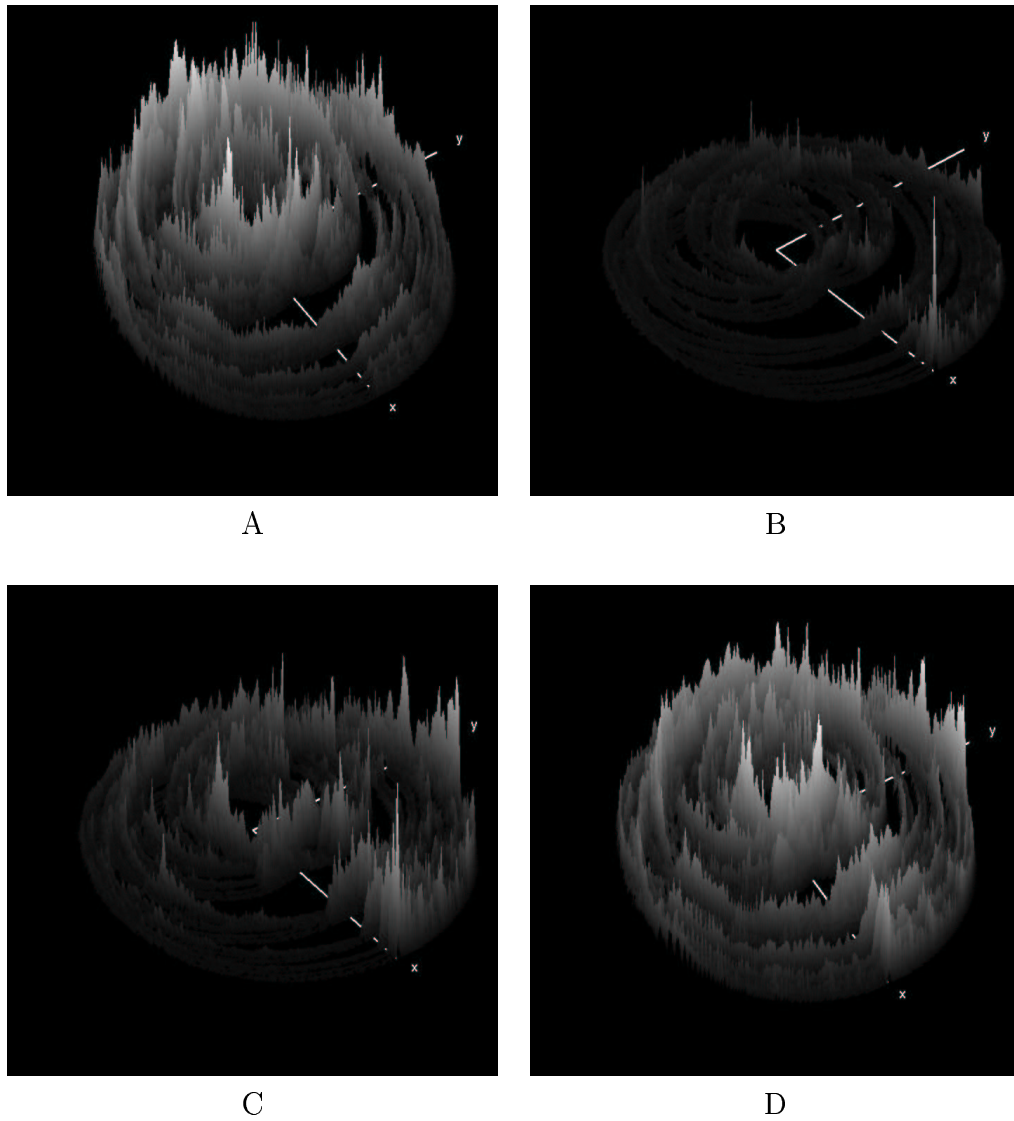


Figura 6.4: Histograma para los organismos (A) *T. maritima*, (B) cromosoma 21 del *H. sapiens*, (C) *S. solfataricus* y (D) *B. subtilis*, con los ejes x y y se muestra la orientación de cada uno de ellos.

96 Aplicaciones a la clasificación de organismos a partir del DNA

mayor; por tal motivo las demás regiones toman tonos de grises oscuros. El caso contrario se puede observar en la Figura 6.4 (D) en donde las regiones en blanco predominan, es decir, la distribución de puntos en diferentes zonas es parecida.

Para obtener una mayor información de los histogramas en el Cuadro 6.3 se muestran las coordenadas en donde la cantidad de puntos es máxima, esto nos da una referencia acerca de las distintas alturas en el histograma.

Organismo	Coordenada (x,y)	Máximo
<i>T. maritima</i>	(0.0404, -0.1717)	464
<i>H. sapiens</i>	(0.9898, 0.0)	3065
<i>S. solfataricus</i>	(0.9898, 0.0)	1300
<i>B. subtilis</i>	(0.1919, 0.101)	441

Cuadro 6.3: Para cada organismo se muestra la coordenada de la esquina inferior izquierda de la celda donde se encuentra el máximo conteo de puntos.

A cada organismo por medio de la compresión fractal se le construye un código IFS. Estos códigos serán tratados como vectores a los que calcularemos su distancia euclidiana. En las Figuras 6.5 y 6.6 se muestra la imagen original correspondiente a los histogramas de los organismos de las Figuras 6.4 con su respectiva reconstrucción a partir del código IFS generado por la compresión fractal.

En los Cuadros 6.4 y 6.5 se muestran las distancias obtenidas entre los organismos del Cuadro 6.1.

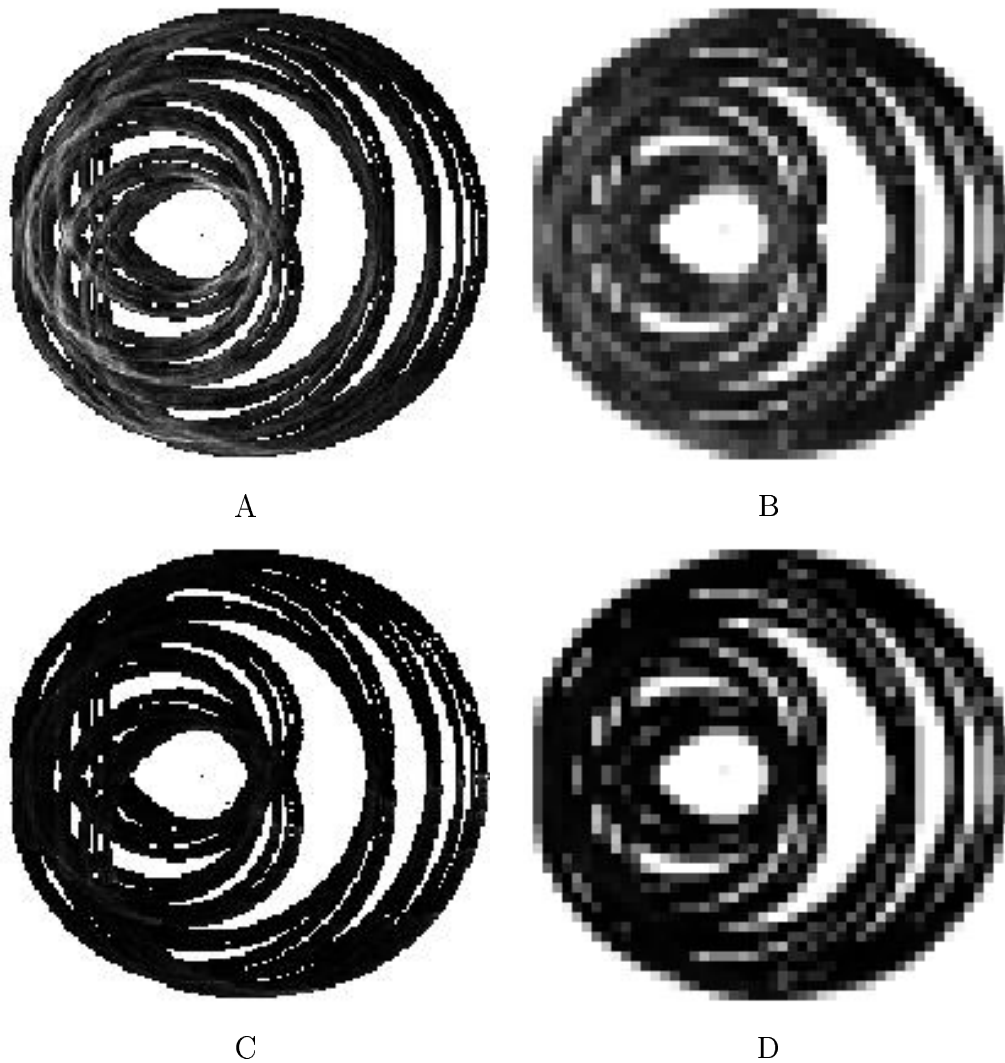


Figura 6.5: (A) Imagen en escala de grises asociada al histograma del organismo *T. maritima*. (B) Imagen construida a partir de su código IFS. (C) Imagen en escala de grises asociada al histograma correspondiente al cromosoma 21 del organismo *H. sapiens*. (D) Imagen construida a partir de su código IFS.

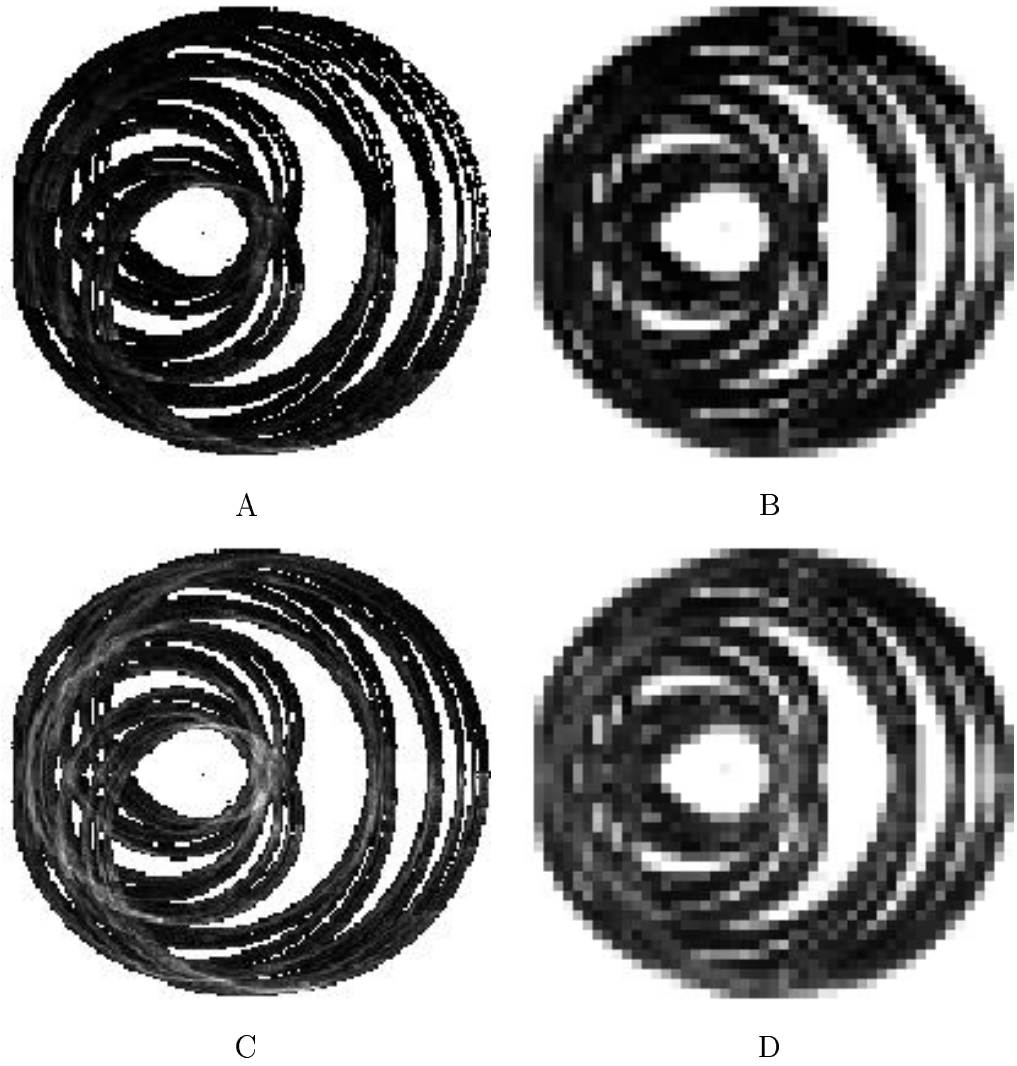


Figura 6.6: (A) Imagen en escala de grises asociada a la medida invariante del organismo *S. solfataricus*. (B) Imagen construida a partir de su código IFS. (C) Imagen en escala de grises asociada a la medida invariante del organismo *B. subtilis*. (D) Imagen construida a partir de su código IFS.

	B	C	D	E	F
B	0	1474.58	691.318	687.106	1446.67
C	1474.58	0	1522.77	1342.25	622.439
D	691.318	1522.77	0	882.022	1443.92
E	687.106	1342.25	882.022	0	1268.36
F	1446.67	622.439	1443.92	1268.36	0
G	1584.47	918.858	1512.90	1444.25	892.38
H	1258.66	944.357	1135.14	1197.50	897.104
I	668.680	1522.41	763.308	762.115	1467.59
J	1111.58	759.281	1027.62	1019.74	769.146
A	1503.33	771.774	1411.49	1359.37	746.827
	G	H	I	J	A
B	1583.47	1258.66	688.68	1111.58	1503.33
C	918.858	944.357	1522.41	759.281	771.754
D	1512.9	1135.14	763.308	1027.62	1411.49
E	1444.25	1197.5	762.115	1019.74	1359.37
F	892.38	897.104	1467.59	769.146	746.827
G	0	886.326	1615.68	914.396	823.827
H	886.326	0	1331.76	736.339	822.157
I	1615.68	1331.76	0	1150.36	1500.99
J	914.396	736.339	1150.36	0	766.8
A	823.827	822.157	1500.99	766.8	0

Cuadro 6.4: Matriz simétrica de distancias para cadenas binarias obtenidas mediante la regla WS .

100 Aplicaciones a la clasificación de organismos a partir del DNA

	B	C	D	E	F
B	0	1121.07	567.527	673.055	1259.33
C	1121.07	0	908.96	877.67	465.97
D	567.527	908.96	0	776.231	1070.19
E	673.055	877.67	776.231	0	968.982
F	1259.33	465.97	1070.19	968.982	0
G	824.788	588.551	629.213	688.55	701.25
H	547.046	889.439	461.292	663.58	1031.83
I	871.544	667.093	603.496	864.751	756.452
J	422.578	935.83	461.984	582.911	1080.25
A	1223.83	461.061	1043.5	930.834	340.031
	G	H	I	J	A
B	824.788	547.046	871.544	422.578	1223.83
C	588.551	889.439	667.093	935.83	461.061
D	629.213	461.292	603.496	461.984	1043.5
E	688.55	663.58	864.751	582.911	930.834
F	701.25	1031.83	756.452	1080.25	340.031
G	0	581.81	514.221	654.243	648.465
H	581.81	0	647.035	418.427	976.277
I	514.221	647.035	0	702.341	727.564
J	654.243	418.427	702.341	0	1045.11
A	648.465	976.277	727.564	1045.11	0

Cuadro 6.5: Matriz simétrica de distancias para cadenas binarias obtenidas mediante la regla *YR*.

6.2. Dendrogramas asociados a organismos

Con el fin de representar gráficamente la matriz de distancias recurriremos a métodos de clasificación jerárquica ascendente. Esta tomará parejas de organismos de acuerdo a su distancia en las matrices de los Cuadros 6.4 y 6.5. Se construirán cúmulos de pares de organismos y se relacionarán de acuerdo a su distancia, comenzando por los que se encuentran en el mismo cúmulo siguiendo con los que se encuentran en el cúmulo más próximo y así sucesivamente. A la representación gráfica de este proceso se le llama *dendrograma*.

Los métodos de clasificación jerárquica que se utilizaron para generar los dendrogramas son: *método de vecinos cercanos*, *método de vecinos lejanos* y *método de pesos ponderados*. El método de pesos ponderados es un promedio de los otros dos. Para una descripción detallada de estos métodos de clasificación véase [2].

Aplicamos el método de pesos ponderados a la matriz del Cuadro 6.4, para observar gráficamente que tanto difieren los organismos (ver Figura 6.7). Para saber que los resultados son confiables se utilizan los métodos extremos que son los métodos de vecinos cercanos y lejanos. Los dendrogramas generados por estos métodos se muestran en la Figuras 6.8 y 6.9.

En el dendrograma superior de la Figura 6.7 que fue hecho utilizando las secuencias WS de los organismos por el método de pesos ponderados, se agrupan en dos cúmulos notables, uno que consta de *B. subtilis*, *E. coli*, *T. maritima* y *Ch. trachomatis* que pertenecen al grupo de las bacterias, en el otro cúmulo se ven agrupados organismos que pertenecen al grupo de eucariontes y arqueobacterias. Los organismos eucariontes se encuentran más cercanos unos con otros, mientras que las arqueobacterias se alejan de los eucariontes y se mantienen cercanas entre ellas. Por el método de vecinos cercanos y lejanos se obtuvieron las mismas agrupaciones pero con variaciones en las distancias (ver Figura 6.8).

En los histogramas generados por las cadenas WS se notan claras diferencias tanto cualitativas como cuantitativas entre los organismos que pertenecen al grupo de los eucariontes y las bacterias, en los eucariontes existen zonas sobre el atractor en las que se acumula gran cantidad de puntos haciendo que otras zonas parezcan poco densas y existen algunas zonas en donde la

102 Aplicaciones a la clasificación de organismos a partir del DNA

densidad es casi nula, en el caso de las bacterias es difícil encontrar a simple vista la zona donde se aglomera la máxima cantidad de puntos, esto hace que su histograma parezca denso, las densidades que se encuentran son distintas en diferentes zonas del atractor sin llegar a una densidad nula. Las arqueobacterias a su vez comparten características de los dos grupos anteriores, es decir, podemos encontrar zonas donde la densidad de puntos es casi nula y al mismo tiempo tener zonas con distintas densidades sin poder distinguir la zona de máxima aglomeración de puntos. Este comportamiento no se puede generalizar, ya que hay organismos de distintos grupos tales que sus histogramas pueden tener características similares a histogramas de grupos a los que no pertenecen.

Con lo que respecta a los dendrogramas generados con las cadenas YR, se forma un subgrupo que se compone de eucariones mientras que las bacterias y arqueobacterias se mezclan indiferentemente ya que tenemos arqueobacterias y bacterias cerca del grupo de los eucariones y por otro lado tenemos un grupo formado por bacterias y arqueobacterias lejos de los eucariontes. Este comportamiento se observa sin importar el método utilizado. Los histogramas generados con cadenas YR son más variados y no siguen un comportamiento como el que se presenta utilizando cadenas WS.

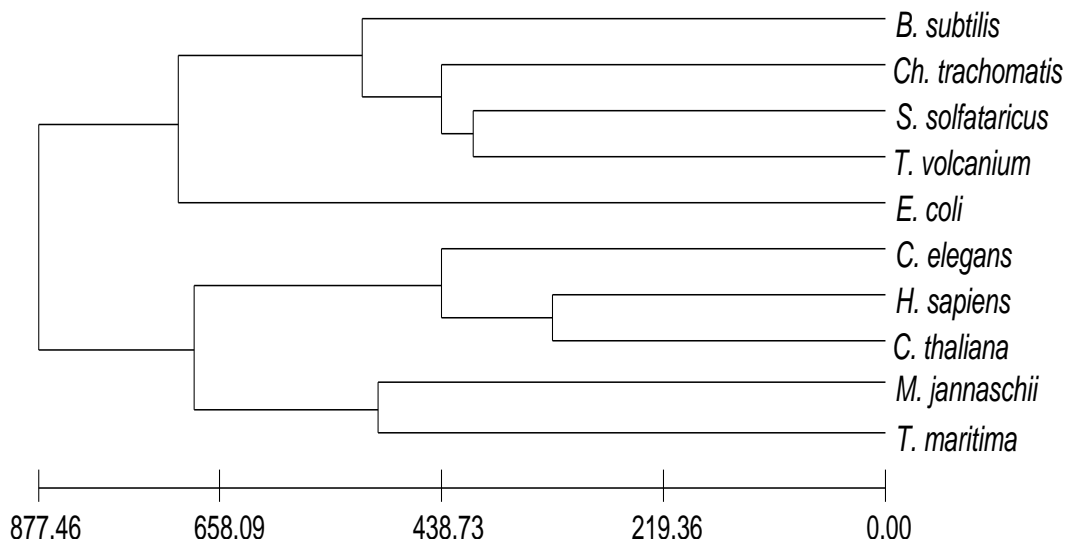
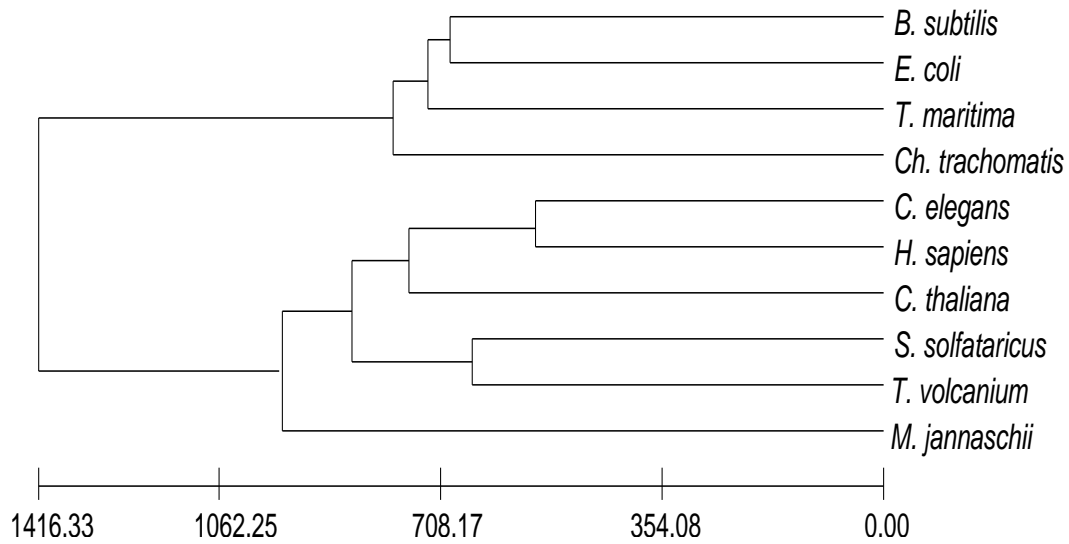


Figura 6.7: Dendrogramas generados respectivamente con las matrices de distancias WS y YR por el método de pesos ponderados.

104 Aplicaciones a la clasificación de organismos a partir del DNA

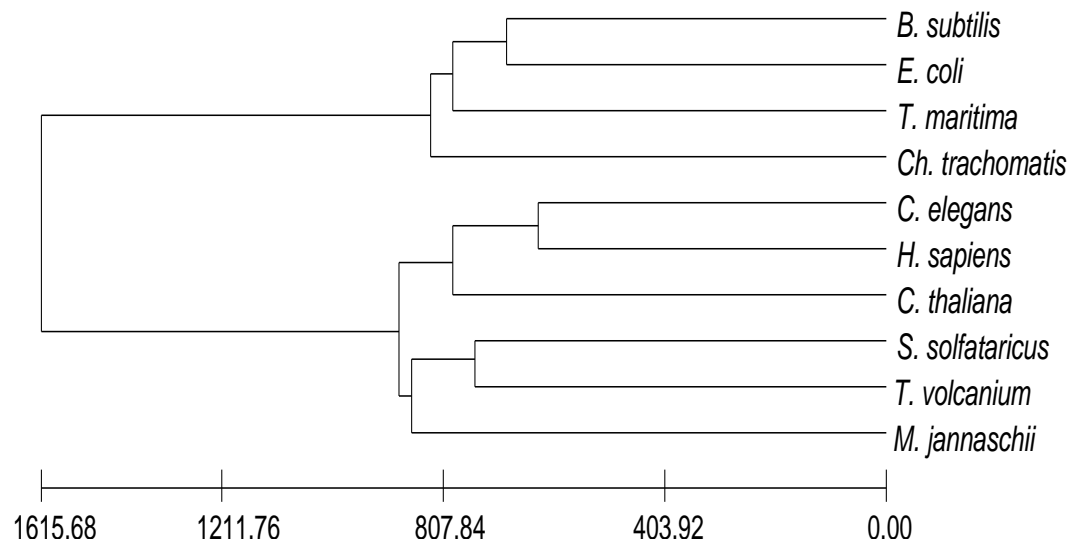
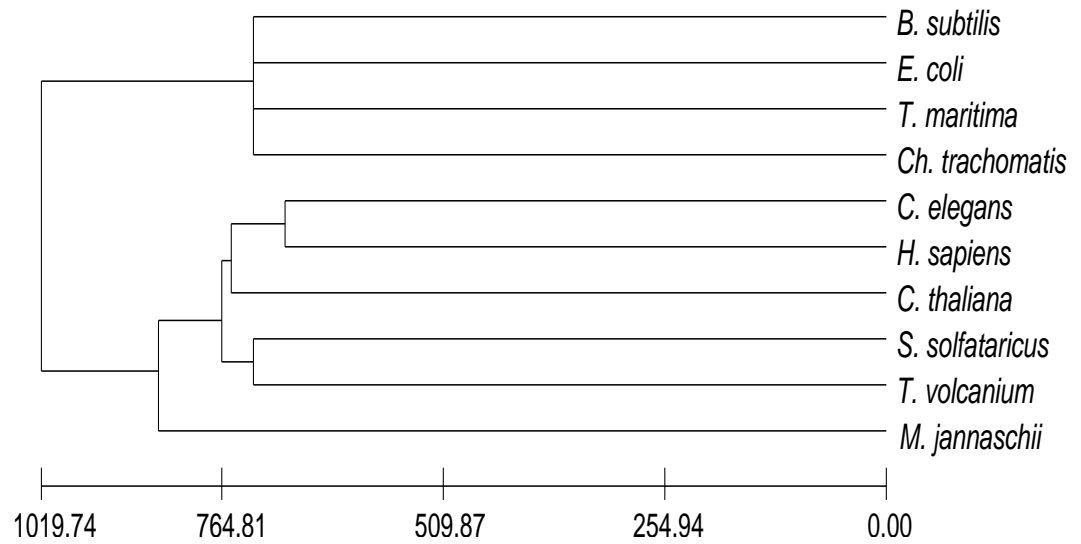


Figura 6.8: Dendrograma para la matriz de distancias WS generado por el método de vecinos cercanos y vecinos lejanos, respectivamente.

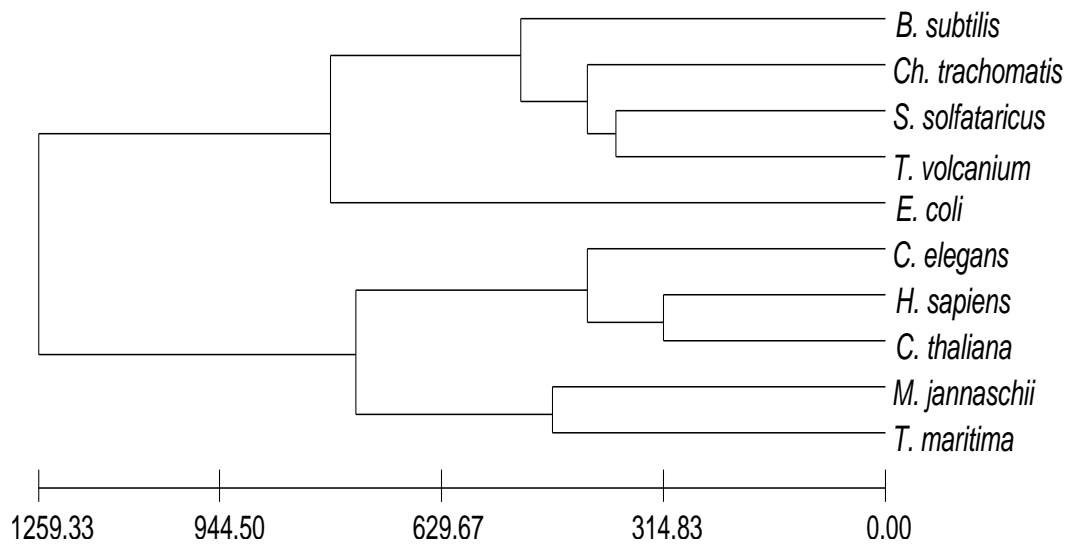
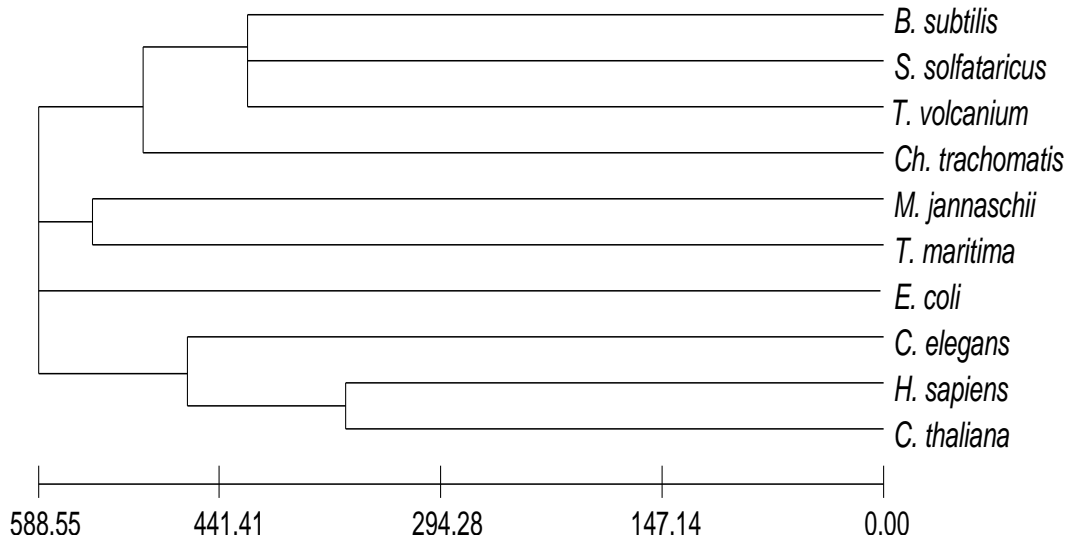


Figura 6.9: Dendrograma para la matriz de distancias YR generado por el método de vecinos cercanos y vecinos lejanos, respectivamente.

106 Aplicaciones a la clasificación de organismos a partir del DNA

Capítulo 7

Discusión

El esquema presentado es una propuesta para una clasificación de organismos utilizando cadenas de DNA en función de su naturaleza fisicoquímica y se han conjuntando métodos que han sido utilizados para el entendimiento de fenómenos no lineales. Este esquema nos proporciona una visualización cualitativa y cuantitativa de la cadena de DNA mostrando nuevas características que, en su representación de sucesión de letras, no se pueden inferir. Esto enriquece el campo de la biología molecular brindando nuevas líneas de investigación ya que, utilizando la información que esta implícita en las cadenas, se puede ayudar al mejor entendimiento del origen y evolución del DNA.

El marco conceptual y teórico que se utiliza en este trabajo, desde el punto de vista de la dinámica, es el de los sistemas complejos ya que se estudia y se analiza el DNA como un todo dejando a un lado el enfoque reduccionista que se utiliza en ciertas ramas de la biología.

Los resultados obtenidos con nuestro esquema son compatibles con la división propuesta por Carl Woese en 1990 [30] y que es aceptada hoy en día; Woese propuso los dominios eucariontes, arqueobacterias y bacterias, y el árbol filogenético se construye utilizando las distancias evolutivas entre los distintos organismos mediante el ARN mitocondrial 16s, ya que es una secuencia “perfectamente conservada”; esto significa que ha evolucionado muy lentamente, por lo que puede ser utilizada para rastrear los cambios evolutivos sucedidos a lo largo de lapsos muy largos y es fácil de manejar en laboratorios.

En este trabajo se analiza una muestra de organismos que incluye los tres grupos propuestos, permitiéndonos observar sus similitudes y diferencias con base en nuestro esquema. El siguiente paso es hacer una optimización dentro del esquema de la compresión fractal de imágenes para tener un código que permita distinguir con exactitud las diferencias entre organismos. Esto permitiría hacer un análisis detallado y exhaustivo de todas las secuencias codificadas hasta el momento, para sugerir una forma de caracterizar las fronteras que existen entre eucariontes, bacterias y arqueobacterias.

Para obtener una clasificación robusta y que pueda complementar a la propuesta original, se puede conjuntar la información de los histogramas generados por las relaciones WS y YR de cada organismo.

De este trabajo se desprende la posibilidad de usar el juego del caos circular como herramienta para obtener medidas sobre cadenas binarias; hasta donde se sabe, el juego del caos no ha sido utilizado para estos propósitos. También se pueden encontrar resultados interesantes si se cambia el orden en que se acomodan los vértices del polígono, ya que las distribuciones de puntos cambian dependiendo del ordenamiento sobre el mismo atractor; por ejemplo, un ordenamiento es dado por los códigos de Gray, los cuales tiene la propiedad de diferir en un bit entre uno y otro, fueron patentados por Frank Gray, investigador de los laboratorios BELL en 1953 [13].

En cuanto a la compresión fractal, tenemos la seguridad de que es la primera vez que se utiliza como método de clasificación de imágenes, por lo cual se podría seguir una línea de investigación que optimice los algoritmos propuestos por Barnsley cuidando que el formato del código se mantenga. Una propuesta para mejorar el algoritmo de compresión es el uso de algoritmos genéticos que obtengan una optimización al aplicar el teorema del collage.

Finalmente se hace énfasis que uno de los propósitos primordiales de este trabajo es conjuntar diversas disciplinas científicas tendiendo puentes entre ellas para enriquecer y proponer nuevos enfoques que permitan un mejor entendimiento a los problemas planteados en cada una de éstas.

Bibliografía

- [1] Angel Edward, *OpenGL A Primer*, Addison Wesley, United State of America, 2002.
- [2] Anderberg M. R., *Cluster Analysis for Application*, Academic Press, New York, 1973.
- [3] Página web del Instituto de Astrobiología de la NASA, *Carl Woose y las nuevas perspectivas en la evolución*, <http://astrobiologia.org>, 2003.
- [4] Barnsley Michael F., *Fractals Everywhere, second edition*, Academic Press, Inc., United States of America, 1993-1998.
- [5] Barnsley Michael F., *Fractal Image Compression*, Ak Peters Ltd., United States of America, 1993.
- [6] Bartle Robert G., Sherbert Donald R., *Introducción al análisis matemático de una variable*, Limusa, México, 1989.
- [7] Encyclopedia Britannica Publishers, *Hombre, ciencia y tecnología*, Ediciones Oceano, Estados Unidos, 1995.
- [8] Brown T. A. *Genomes*, Wiley-Liss, United States of America, 1999.
- [9] Chinn, W. G., Steenrod N. E., *Primeros conceptos de topología*, Alhambra, Madrid, 1975.
- [10] Devaney, R. L., *An Introduction to Chaotic Dynamical Systems, second edition*, Addison-Wesley, Redwood city, 1989.
- [11] Flake Gary William, *The computational beauty of nature: computer exploration of fractals, chaos, complex systems, and adaptation*, The MIT Press, United States of America, 1998.

- [12] Página web *GeneQuiz*, <http://jura.ebi.ac.uk:8765/ext-genequiz>, 2003.
- [13] Gray, Frank, *Pulse code communication*, 1953, U.S. Patente No. 2,632,058
- [14] Joyanes Aguilar Luis, *PROGRAMACION EN C++. Algoritmos, estructuras de datos y objetos*, Mc Graw Hill, España, 2000.
- [15] Mandelbrot Benoit B., *Fractals: From, Chance and Dimension*, W. H. Freeman and Co., San Francisco, 1977.
- [16] Mandelbrot Benoit B., *Fractal Geometry of Nature*, W.H.Freeman and Co., New York, 1982.
- [17] Marsden Jerrold E., Tromba Anthony J., *Calculo Vectorial Cuarta Edición*, Addison Wesley Longman, Estado de México, 1998.
- [18] Miramontes Vidal P., *Tesis Doctoral: Un esquema de automata celular como modelo matemático de la evolución de los ácidos nucléicos*, Universidad Nacional Autónoma de México, Facultad de Ciencias, División de Estudios de Posgrado, México, D.F., 1992.
- [19] Peitgen Heinz-Otto, Jürgens Hartmut, Saupe Dietmar, *Chaos and Fractals New Frontiers of Science*, Springer-Verlag, New York, 1992.
- [20] Rudin Walter, *Principles of Mathematical Analysis Third Edition*, Mc Graw Hill, New York, 1976.
- [21] Woo Mason, Neider Jackie, et al, *OpenGL Programming Guide Third Edition*, Addison Wesley, United States of America, 1999.

Artículos

- [22] Barnsley M., Demko S., *Iterated Functions System and the global construction of fractals*, Proceedings of the Royal Society of London, A399, 1985, 243-275.
- [23] Coello Carlos, *Compresión de imágenes usando fractales*, Soluciones avanzadas, No. 57, Mayo 1998, 53-64.

-
- [24] Fisher Yuval, *Fractal Image Compression*, SIGGRAPH '92 Course Notes.
- [25] Hutchinson J., *Fractals and Self-Similarity*, Indiana University of Journal of Mathematics, 30, 1981, 713-747.
- [26] Monarstersky Richard, *El origen de la vida en la tierra*, National Geographic, Vol 2, No. 3, 54-81, 1998.
- [27] Shannon C. E., *A mathematical theory of communication*, Bell Systems Technical Journal, Vol. 27, 1948, 379-423, 623-656.
- [28] Sun K. T., Lee S. J, Wu P. Y. *Neural network approaches to fractal image compression and descompression*, Neurocomputing, 41, 2001, 91-107.
- [29] Tsuchiya Takashi, *Circular Chaos Game Representation of 1-D Chaos and its relation to the Complex Weierstrass Function*, International Journal of Bifurcation and Chaos, Vol 9, No. 10, 1999, 2069-2080.
- [30] Woese C.R., Kandler O., Wheelis M. L., *Towards a natural system of organisms: proposal for the domains Archaea, Bacteria and Eucarya*, Proceedings of the National Academy of Sciences, Vol. 87, 4576-4579, 1990

