

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

**Unidad Zacatenco**

**Departamento de Computación**

**Análisis de las reglas de autómatas celulares en dos  
dimensiones vía diagramas de de Bruijn**

Tesis que presenta

**Paulina Anaid León Hernández**

para obtener el Grado de

**Maestra en Ciencias en Computación**

Director de la Tesis

**Dr. Sergio Víctor Chapa Vergara**

**Dr. Amilcar Meneses Viveros**

México, D.F.

Noviembre 2012



# Resumen

Los autómatas celulares son la representación matemática más simple de los sistemas complejos; además, han mostrado ser extremadamente útiles en la idealización de comportamientos dinámicos gracias a su simplicidad. Por si fuera poco, son motores conceptuales con los que se estudia la formación general de patrones.

Una de las herramientas que ha probado ser útil para el análisis de la dinámica global y la interacción local de los autómatas celulares son los diagramas de de Bruijn. Estos diagramas son un tipo especial de grafos dirigidos, resultado de una serie de trabajos enfocados al estudio del corrimiento de secuencias de símbolos para codificar información.

Los diagramas de de Bruijn han sido explotados en el análisis de la dinámica de los autómatas celulares unidimensionales para buscar configuraciones particulares, tales como ancestros, Jardines del Edén y autómatas reversibles; sin embargo, este mismo análisis para autómatas celulares en dos dimensiones ha sido escaso.

En esta tesis se ofrece una forma de explorar sistemáticamente configuraciones específicas por medio de los diagramas de de Bruijn, desde una configuración inicial en dos dimensiones. Dicho análisis se centra en dos reglas de interés debido a su comportamiento, por un lado, una regla de comportamiento complejo: el Juego de la vida y por otro, una regla de comportamiento caótico: la Regla de Difusión.



# Abstract

Cellular automata are the simplest mathematical representations for many classes of complex systems; also cellular automata have proven to be extremely useful in order to idealize the dynamical behaviour of many real complex systems. As result of their underlying simplicity, cellular automata are also powerful conceptual engines for studing general patterns formation.

De Bruijn diagrams have been a useful tools in order to perform a systematic analysis of cellular automata. In this way, de Bruijn diagrams are a special kind of directed graphs, known originally as shift register sequences. These diagrams arise due to a series of works focused to study shift register sequences of symbols for the encoding of information.

They have been used to compute particular kind of configurations, such as ancestors, complex patterns, cycles, Garden of Eden configurations and formal languages, specially in one dimensional cellular automata. However, there do not exist too many research for analyzing two dimensional cellular automata.

Thus, this thesis offers a way to explore systematically specific configurations by de Bruijn diagrams from initial configurations in two dimensional cellular automata. Such analysis in concentrated mainly in Game of Life and Diffusion Rule.



# Agradecimientos

Quiero agradecer a las dos instituciones que fueron parte fundamental de este logro, primero a Consejo Nacional de Ciencia y Tecnología (CONACyT) y al Centro de Investigación y Estudios Avanzados del IPN (CINVESTAV), y al departamento de Computación por darme la oportunidad de cursar mis estudios de maestría en sus instalaciones y por el apoyo económico que me fue brindado.

Le agradezco a mis directores de tesis: al Dr. Sergio Chapa Vergara por permitirme desarrollar este tema bajo su asesoría, por su tiempo para discusiones del tema para lograr y llevar a cabo con éxito esta meta. Al Dr. Amilcar Meneses Viveros, por su apoyo en el desarrollo de la tesis, por las charlas y discusiones que enriquecieron mi conocimiento, así como por su tiempo para ampliar mis dudas y sobretodo por su motivación. A mis revisores de tesis. la Dra. Sonia Mendoza Chapa y el Dr. Adriano de Luca Pennacchia por su tiempo para revisar el documento y por los comentarios oportunos que recibí del mismo.

Quiero agradecerle al Dr. Genaro Juárez Martínez, mentor y amigo, que me guió y motivó para realizar mis estudios de posgrado, por su tiempo, apoyo y enseñanza, por todas esas platicas que enriquecían mi conocimiento, tanto en lo profesional como en lo personal, por los regaños merecidos y sobretodo por su valiosa amistad.

Al Dr. Harold McIntosh quiero darle un especial agradecimiento, ya que se ha convertido en una gran motivación, agradezco sus comentarios, y también agradezco el tiempo que invirtió para transmitirme un poco de su conocimiento, le agradezco las charlas que me enriquecieron en lo profesional, pero sobretodo en lo personal.

También agradezco al M.C. Miguel Olvera Aldana, mentor y amigo, de quien siempre recibí motivación en los momentos difíciles, por sus atinados consejos que me han enriquecido en lo profesional y en lo personal, pero sobretodo por su valiosa amistad y su cariño.

Por otro lado, quiero dar un especial agradecimiento a mis padres Herminio y Ofelia, quienes han sido un gran ejemplo de vida y de lucha, sin ellos y sin su apoyo incondicional no hubiera logrado las metas que he logrado, quienes se esforzaron por hacer de mí una mujer de provecho y de quienes he recibido la mejor herencia: el

valor del trabajo. A mi hermana Teresa, quien siempre ha creído en mí, quien ha estado para darme ánimos en momentos de debilidad y para compartir conmigo mis más grandes alegrías, quien me aconsejaba y escuchaba siempre. Y a toda mi familia, quien me ha cobijado bajo su protección y cariño, y que de alguna u otra manera también son cómplices de este logro.

Quiero agradecer también a mis amigos, que me apoyaron y escucharon, que me llenaron de animos cuando ya no podía más, con quienes a lo largo de este tiempo me enriquecí como ser humano, aquellos que creyeron en mi y me apoyaron incondicionalmente en esta aventura, con quienes compartí momentos inolvidables y charlas muy interesantes especialmente a Ukranio, Rodrigo, Daniel Jiménez y Ana Helena.

A mis compañeros de maestría, por esos momentos de apoyo, de unión y esas largas noches en las que compartíamos desvelos y en ocasiones sufrimientos a causa de la maestría; por enseñarme el valor del trabajo en equipo, y por todos aquellos momentos que compartimos juntos, que han hecho de esta etapa uno de los momentos más hermosos de mi vida.

Y finalmente, y no por ello menos importante, quiero agradecerle a mi compañero de locuras, de aventuras, a mi amigo y cómplice, a Julio Cesar, por su tiempo, su apoyo incondicional, su paciencia y comprensión, por creer en mi, en mi trabajo, por enseñarme la difícil labor de darle equilibrio a mi vida, por los agradables momentos y las amenas platicas que me motivaban en los momentos difíciles.



# Índice general

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>Agradecimientos</b>	<b>VII</b>
<b>Índice general</b>	<b>IX</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>XI</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes y motivación . . . . .	1
1.2. Planteamiento del problema . . . . .	2
1.3. Objetivos . . . . .	3
1.4. Estructura de la tesis . . . . .	4
<b>2. Grafos y autómatas</b>	<b>7</b>
2.1. Conjuntos y relaciones . . . . .	7
2.2. Grafos . . . . .	11
2.2.1. Representaciones de un grafo . . . . .	13
2.2.2. Trayectorias y ciclos . . . . .	15
2.3. Teoría de autómatas . . . . .	16
2.3.1. Alfabetos, cadenas y lenguajes . . . . .	17
2.3.2. Autómatas finitos . . . . .	18
<b>3. Diagramas de de Bruijn en AC 1D</b>	<b>23</b>
3.1. Autómatas celulares . . . . .	23
3.1.1. Definición . . . . .	23
3.1.2. Introducción al estudio de los AC unidimensionales . . . . .	25
3.2. Diagramas de de Bruijn . . . . .	28
3.2.1. Antecedentes . . . . .	28
3.2.2. Definición . . . . .	29
3.3. Análisis de autómatas celulares en 1D vía de Bruijn . . . . .	32

<b>4. Diagramas de de Bruijn en 2D</b>	<b>39</b>
4.1. Definición de AC en 2 dimensiones . . . . .	39
4.2. Extensión de vecindades parciales . . . . .	43
4.3. Análisis de los AC 2D vía diagrama de de Bruijn . . . . .	46
<b>5. Experimentación</b>	<b>53</b>
5.1. Análisis . . . . .	53
5.2. Algoritmos utilizados . . . . .	54
5.3. Casos de estudio y resultados . . . . .	56
5.3.1. Regla de comportamiento complejo: El Juego de la Vida . . . . .	56
5.3.2. Regla de comportamiento caótico: Regla de Difusión . . . . .	57
<b>6. Resultados, discusión, conclusiones y trabajo a futuro</b>	<b>63</b>
6.1. Resultados . . . . .	63
6.2. Discusión . . . . .	64
6.3. Conclusiones . . . . .	66
6.4. Trabajo a futuro . . . . .	66
<b>A. Matrices de de Bruijn para Juego de la Vida</b>	<b>69</b>
A.1. Vecindades parciales de $4 \times 2$ . . . . .	69
A.2. Vecindades parciales de $4 \times 3$ . . . . .	74
<b>B. Matrices de de Bruijn la Regla Difusión</b>	<b>83</b>
B.1. Vecindades parciales de $4 \times 2$ . . . . .	83
B.2. Vecindades parciales de $4 \times 3$ . . . . .	88
<b>C. Manual técnico</b>	<b>97</b>
<b>Bibliografía</b>	<b>104</b>

# Índice de figuras

2.1. Digrafo de la relación $R$ en $A$ . . . . .	10
2.2. Ejemplos de grafos . . . . .	12
2.3. Ejemplos de grafos etiquetado . . . . .	13
2.4. Camino Euleriano . . . . .	16
2.5. Ejemplo de autómata finito . . . . .	19
2.6. Ejemplo de autómata finito determinístico . . . . .	21
3.1. Ejemplos de mallas de dimension 1 2 y 3 de AC . . . . .	25
3.2. Ejemplos de reglas de AC estudiadas . . . . .	26
3.3. Vecindad de radio $r$ en un AC 1D . . . . .	27
3.4. Secuencia de de Bruijn y sus subsecuencias de longitud 2. . . . .	30
3.5. Elementos de un diagrama de de Bruijn . . . . .	31
3.6. Diagrama de de Bruijn para $m = 1, n = 3$ . . . . .	31
3.7. Evoluciones en un tiempos $t$ con relación a la vecindad . . . . .	33
3.8. Diagrama de de Bruijn para la regla 240, de AC 1D (2,1). . . . .	35
4.1. Vecindades en dos dimensiones . . . . .	40
4.2. Posibles vecindades parciales de von Neumann . . . . .	42
4.3. Vecindades parciales de la vecindad hexagonal . . . . .	43
4.4. Vecindad de Moore extendida en horizontal . . . . .	44
4.5. Vecindad de Moore extendida en vertical . . . . .	45
4.6. Vecindad de Moore extendida en dos dimensiones . . . . .	45
4.7. Vecindad hexagonal extendida en dos dimensiones verticalmente . . . . .	46
4.8. Vecindad hexagonal extendida en dos dimensiones . . . . .	47
4.9. Conjunto $V$ para una vecindad parcial de $2 \times 3$ células y 2 estados. . . . .	49
4.10. Relaciones entre dos nodos . . . . .	49
4.11. Ejemplo de relaciones entre dos nodos . . . . .	50
4.12. Tipos de corrimientos . . . . .	50
4.13. Ejemplo de diagrama de de Bruijn . . . . .	51
5.1. Permanencia el Juego de la Vida . . . . .	58
5.2. Corrimiento en diagonal Juego de la Vida . . . . .	59
5.3. Permanencia Difusión . . . . .	60
5.4. Corrimiento en diagonal Difusión . . . . .	61

6.1. Vecindad parcial de von Neumann . . . . .	65
A.1. Vecindad parcial $4 \times 2$ . . . . .	69
A.2. Vecindad parcial de $4 \times 3$ . . . . .	74
A.3. Posibles valores que toman las células evaluadas en $t + 1$ . . . . .	75
B.1. Vecindad parcial $4 \times 2$ . . . . .	83
B.2. Vecindad parcial de $4 \times 3$ . . . . .	88
B.3. Posibles valores que toman las células evaluadas en $t + 1$ . . . . .	89

# Índice de tablas

2.1. Ejemplo de tabla de transición. . . . .	21
3.1. Clasificación de la evolución . . . . .	36
4.1. Matriz de de Bruijn para vecindades parciales $2 \times 3$ . . . . .	52
5.1. Evaluación de células en $t$ tiempo vs tamaño vecindad parcial . . . . .	53
A.1. Matriz de de Bruijn de vecindad parcial $4 \times 2$ del Juego de la Vida . . . . .	70
A.2. Matriz de de Bruijn de vecindad parcial $4 \times 3$ del Juego de la Vida . . . . .	76
B.1. Matriz de de Bruijn de vecindad parcial $4 \times 2$ para la regla de Difusión . . . . .	84
B.2. Matriz de de Bruijn de vecindad parcial $4 \times 3$ para la regla Difusión . . . . .	90



# Capítulo 1

## Introducción

### 1.1. Antecedentes y motivación

Los autómatas celulares son una clase de sistemas matemáticos determinísticos, discretos en espacio y tiempo caracterizados por la interacción local y una forma de evolución inherentemente paralela. El concepto fue introducido por John von Neumann en los años 40's como modelos biológicos de autoreproducción [von Neumann(1966)]. Sin embargo, los autómatas celulares han sido reintroducidos y usados como modelos matemáticos en una gran cantidad de contextos, dada la habilidad de generar una variedad de comportamientos complejos que producen patrones bajo un conjunto de reglas sencillas. Ejemplos de aplicación son los modelos físicos, químicos, bioquímicos, entre otros. La importancia de estudiar los autómatas celulares se debe a su utilidad como simuladores de sistemas dinámicos discretos, además son vehículos conceptuales para el estudio de formación de patrones y complejidad, así como también modelos originales de fundamentos físicos [Gutowitz(1991), Ilachinski(2001), Codd(1968)].

Por varios años el comportamiento global y local en los autómatas celulares unidimensionales ha sido estudiado desde varios puntos de vista; por un lado, Wolfram realizó un análisis estadístico de la dinámica global de los autómatas celulares en una dimensión y propone su clasificación de acuerdo a su comportamiento [Wolfram(1983)]. Por otro lado, Wuensche realizó el estudio de la dinámica global de los autómatas celulares basándose en algoritmos invertibles para el cálculo de preimágenes para la comprensión de topologías de conjuntos de atracción [Wuensche(2011)]. Jen utiliza el concepto de las secuencias de de Bruijn [Ralston(1982)] y la teoría de corrimientos para estudiar los autómatas celulares unidimensionales para el cálculo de preimágenes [Jen(1986), Jen(1988)]. Voorhees estudia los autómatas celulares desde el sentido estricto de las matemáticas usando conceptos de algebra booleana y teoría de campos finitos, además de hacer uso de conceptos de teoría de grafos para el cálculo de ancestros [Voorhees(2006), Voorhees(1996)]. McIntosh retoma el trabajo de Jen y realiza el estudio vía de Bruijn para las reglas de autómatas celulares unidimensionales, en las cuales se buscaron ciclos, periodos y ancestros. Además de dicho análisis,

McIntosh utilizó una notación matricial booleana que representa la relación entre los nodos del diagrama [McIntosh(2009)]. Posteriormente se han realizado más trabajos relacionados al análisis de los autómatas celulares unidimensionales vía diagramas de de Bruijn, debido a que han demostrado ser una herramienta poderosa capaz de encontrar propiedades interesantes del comportamiento del autómata celular, así como también para la construcción de lenguajes y la manipulación de cadenas para una dimensión mediante expresiones regulares. Por ejemplo, en 2006 Gómez realizó el cálculo de preimágenes de autómatas celulares de una dimensión por medio de diagramas de de Bruijn, con la finalidad de estudiar las propiedades topológicas que se ven como esquemas útiles para la clasificación de autómatas celulares [Gómez(2008)]. En ese mismo año se utilizaron los diagramas de de Bruijn para el análisis del modelo del tránsito vehicular mediante la regla 184 de los autómatas celulares lineales, con lo cual se encontraron comportamientos cíclicos con periodo cinco [Rodríguez and Chapa(2004)]. Sutner utiliza los diagramas de de Bruijn para determinar si un autómata celular es reversible o es subyectivo [Sutner(1991)].

El estudio de los autómatas celulares en dos dimensiones ha sido enfocado a áreas distintas a las realizadas en una dimensión. Por ejemplo, los primeros estudios se desarrollaron por John von Neumann [von Neumann(1966)] donde utiliza autómatas celulares en dos dimensiones para hacer la construcción de una *máquina lógicamente universal*. Posteriormente, en los 70's, John Horton Conway retoma el trabajo de von Neumann y simplifica el autómata celular al cual se le dio el nombre de *El juego de la vida*. En cambio el estudio analítico de los autómatas celulares en dos dimensiones ha sido poco explorado, por ejemplo, en los 90's McIntosh inicio el estudio de Juego de la Vida donde ha encontrado estructuras complejas de tamaño  $n$  [McIntosh(2010b)] y Eppstein ha realizado el estudio para pequeños espacios de evolución mediante diagramas de de Bruijn utilizando características combinatorias y dependencias iterativas [Sutner(1991)]. Sin embargo, estos estudios no han sido suficientes para encontrar propiedades del comportamiento global de los autómatas celulares como las que se han estudiado en los autómatas celulares de una dimensión.

## 1.2. Planteamiento del problema

El estudio de los autómatas celulares en dos dimensiones se ha visto limitado debido a la gran cantidad de información que se requiere procesar y analizar. Sin embargo, los estudios que se han realizado a autómatas celulares de dos dimensiones, se ha observado que el análisis de la interacción local y global de un espacio de evolución presenta una cierta analogía a la realizada en autómatas celulares de una dimensión. Quizá una de las formas con la cual se encuentre una analogía de manera más sencilla es utilizando los diagramas de de Bruijn, debido a que utiliza conceptos básicos, tales como, conjuntos, grafos y relaciones, aplicables a estructuras abstractas.

Por tal razón la hipótesis de la cual se parte se describe como: si los diagramas de



de Bruijn han sido utilizados para analizar el comportamiento local y global de los autómatas celulares unidimensionales, además de ser útiles para conocer las cadenas formadas por dichos autómatas, se espera que serán útiles de igual forma para el análisis y formación de cadenas en autómatas celulares de dos dimensiones.

Los problemas que se derivan de lo anterior abarcan diversas áreas. El primer problema a tratar, consiste en el análisis de las estructuras matemáticas análogas en ambas dimensiones, las cuales pueden llevar a conocer propiedades similares entre los autómatas celulares de dimensión uno y dos.

El siguiente problema, consiste en la manipulación de elementos que funcionan como nodos en los diagramas de de Bruijn, debido a que el número de elementos crece exponencialmente dependiendo del número de estados que tienen el autómata celular y del número de células que forman la *vecindad* necesaria para evaluar al autómata celular. Estos problemas de manipulación son referentes a los siguientes puntos:

- Búsqueda de relaciones entre los elementos, llamados *vecindades parciales*, que funcionen como nodos del diagrama.
- Manipulación de las matrices de incidencias que genera el diagrama, para la búsqueda de caminos.
- Búsqueda de las matrices de incidencias para generar matrices de de Bruijn.

Por otra parte, existe una amplia variedad de reglas de autómatas celulares en dos dimensiones, cada una de las cuales tienen diferentes usos, por ejemplo, pueden ser para modelación de sistemas, o para el estudio de la teoría de la computación. Por lo cual, se requirió evaluar las reglas de autómata celular en dos dimensiones, para seleccionar las más adecuadas para ejemplificar el análisis sistemático. Así, se selecciona la regla del Juego de la Vida, debido a su complejidad y a la gran variedad de estructuras que emergen como resultado de la evaluación de la función de transición. La segunda regla seleccionada es la Regla de Difusión, la cual, a pesar de considerarse como una regla caótica, en su comportamiento local se encuentra una gran cantidad de estructuras que emergen como resultado de la evaluación de la función de transición, las cuales en conjunto forman estructuras más complejas. En ambas reglas, se ve una área de oportunidad para encontrar estructuras nuevas que emergen como resultado de la evolución.

### 1.3. Objetivos

La presente tesis pretende dar una introducción al análisis de los autómatas celulares en dos dimensiones por medio de los diagramas de de Bruijn, tomando como base los estudios realizados en una dimensión.

Por tal motivo, el objetivo general de esta tesis es buscar posibles propiedades similares entre los autómatas celulares de una dimensión y de dos dimensiones mediante los diagramas de de Bruijn, ejemplificando su uso con las reglas de *el Juego de la Vida* y *Regla de Difusión*.

Lo anterior se desglosa en los siguientes objetivos particulares:

- Formalizar los componentes de los diagrama de de Bruijn para el caso particular de dos dimensiones.
- Proponer un método que permita realizar el cálculo de los diagramas de de Bruijn para autómatas celulares en dos dimensiones.
- Implementar algoritmos para el cálculo de los diagramas de de Bruijn para autómatas celulares en dos dimensiones.
- Realizar experimentos que validen el uso de los algoritmos y la eficiencia de la metodología propuesta en dos reglas de autómatata celular: Juego de la Vida y Regla de Difusión.
- Analizar los diagramas de de Bruijn para autómatas celulares de dos dimensiones y buscar propiedades similares en una dimensión.
- Encontrar las propiedades similares que se pueden observar por medio de los diagramas de de Bruijn para autómatas celulares en dos dimensiones.

### 1.4. Estructura de la tesis

La tesis se encuentra dividida en 6 capítulos, en la cual se desarrolla de manera formal y experimental el trabajo realizado para el análisis de los autómatas celulares en dos dimensiones vía diagramas de de Bruijn, y la cual queda dividida como se describe a continuación.

En el segundo capítulo se discutirá sobre los conceptos básicos de conjuntos y grafos, los cuales son herramientas fundamentales para poder comprender la teoría de autómatas. Esta teoría ha sido de gran utilidad en las ciencias de la computación, los cuales han dado pauta a los lenguajes formales.

El tercer capítulo da un panorama general de los autómatas celulares, los cuales se definen como una tupla de cinco elementos, donde las variables a considerar para esta tesis son la dimensión y la vecindad. Se define un tipo de vecindad, a la cual se le nombra como *vecindad parcial* que funge como nodos en los diagramas de de Bruijn. Los diagramas de de Bruijn son un tipo especial de grafos dirigidos, que ayudan como herramientas para el análisis de ciertas características, y propiedades derivadas de

algoritmos de la teoría de grafos.

El cuarto capítulo realiza la discusión sobre los autómatas celulares en dos dimensiones y se describen las definiciones propuestas de los componentes de los diagramas de de Bruijn; además realiza la descripción de la metodología propuesta para analizar los autómatas celulares en dos dimensiones.

El quinto capítulo describe de manera general los algoritmos propuestos para atacar el problema computacional, derivado de las definiciones propuestas en el capítulo 3. Además de mostrar los experimentos que validan la efectividad de la metodología propuesta, aplicado a dos diferentes reglas, el Juego de la Vida y la Regla de Difusión.

Finalmente, en el sexto capítulo se discutirán los resultados, las definiciones propuestas, así como las limitantes que tiene, las conclusiones y el trabajo a futuro.



# Capítulo 2

## Grafos y autómatas

En este capítulo se discuten las definiciones básicas de las matemáticas discretas que serán de utilidad para el desarrollo de la tesis. En la sección 2.1 se describen las definiciones y características de conjuntos, relaciones, operaciones que se realizan entre relaciones y sus representaciones. Posteriormente, en la sección 2.2 se da una introducción a la teoría de grafos, en la cual se describen cinco tipos de representaciones de los grafos y una discusión sobre los conceptos de trayectorias, caminos y circuitos. Finalmente, en la sección 2.3 se da una introducción de los lenguajes formales y las definiciones básicas de autómatas finitos.

### 2.1. Conjuntos y relaciones

Uno de los conceptos quizá más importante de las matemáticas y sencillo de entender son los *conjuntos*. Los cuales surgen como propuesta del matemático ruso George Cantor al intentar unificar los distintos campos de las matemáticas [Abellanas(1991), Ullman and Aho(1995), Marshall(1971)]. De esta manera, esta sección presenta una breve discusión acerca de las propiedades y operaciones básicas de los conjuntos, que darán pauta al desarrollo de las siguientes secciones. la primeras definiciones estarán relacionadas a los conjuntos y sus operaciones básicas.

**Definición 1** *Un **conjunto** es cualquier colección de objetos bien definidos por medio de alguna o algunas propiedades en común.*

Por ejemplo, la colección de todas las sillas de madera, la colección de todos los pájaros negros de una parvada, o la colección de los números reales comprendidos entre cero y uno, son cada uno, un conjunto.

**Definición 2** *Se llama **elemento** o **miembro** a cada uno de los objetos que componen un conjunto.*

Una forma de describir un conjunto con un número finito de objetos, es hacer una lista de elementos del conjunto y encerrarla entre llaves. Así, el conjunto de todos los

enteros positivos menores a 4, se escribe como  $\{1, 2, 3\}$ . Se emplean letras mayúsculas, como  $A, B, C$  para designar los conjuntos y las letras minúsculas, como  $a, b, c, x, y, z$  para designar los elementos de los conjuntos. Para indicar que  $x$  es un elemento del conjunto  $A$ , se escribe  $x \in A$ , por el contrario, para indicar que  $x$  no es un elemento de  $A$ , se escribe  $x \notin A$ .

Por ejemplo: Sea  $A = \{1, 3, 5, 7\}$ . Entonces  $1 \in A, 3 \in A$  pero  $2 \notin A$ .

En ocasiones no es conveniente describir un conjunto por medio de una lista de todos sus elementos. Una manera útil de definir un conjunto, es especificando una propiedad que los elementos del conjunto tengan en común. Se utiliza la notación  $P(x)$  para denotar una oración o enunciado  $P$  relativo al objeto variable  $x$ . El objeto definido por  $P(x)$  escrito en la forma  $x|P(x)$ , es siempre la colección de elementos  $x$  para los cuales  $P$  es cierto.

Por ejemplo:  $Z^+ = \{x|x \text{ es un entero positivo}\}$ .

Las operaciones básicas que se definen sobre los conjuntos son las siguientes:

**Definición 3** La operación  $A \cup B$ , la unión de  $A$  con  $B$  se define como el conjunto resultante de elementos que están en  $A$  o están en  $B$ .

**Definición 4** La operación  $A \cap B$ , la intersección de  $A$  con  $B$  se define como el conjunto resultante de elementos que están en  $A$  y están en  $B$ .

**Definición 5** La operación  $A - B$ , la diferencia de  $A$  y  $B$  se define como el conjunto resultante de elementos que están en  $A$  y no están en  $B$ .

**Definición 6** El conjunto potencia de  $A$ , es el conjunto formado por todos los subconjuntos de  $A$ .

En el estudio de cualquier clase de objetos es frecuente realizar el análisis de los objetos en sí mismos, observando de qué partes consta, o cuales son sus elementos y relaciones que existen, tanto entre las partes que componen dicho objeto, como entre distintos objetos de la misma clase. De esta manera, el concepto de relación es utilizado como una herramienta básica que permite analizar y estudiar conjuntos de diferentes tipos. Sin embargo, antes de definir lo que es una relación, se hace la definición de otros conceptos, como elementos bases de una relación, estos elementos son los siguientes:

**Definición 7** Un **par ordenado**  $(a, b)$  es un listado de los objetos  $a$  y  $b$  en un orden prescrito, donde  $a$  en primer término y  $b$  en segundo.

Haciendo uso de esta definición podemos tener que:

**Definición 8** Si  $A$  y  $B$  son dos conjuntos no vacíos, se define el **conjunto producto** o **producto cartesiano**  $A \times B$  como el conjunto de todos los pares ordenados  $(a, b)$  con  $a \in A$  y  $b \in B$ .

Expresado matemáticamente lo anterior:

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

Por ejemplo: Sea  $A = \{1, 2, 3\}$  y  $B = \{r, s\}$  entonces se tendrá que:

$$A \times B = \{(1, r)(2, r)(3, r)(1, s)(2, s)(3, s)\}.$$

Es posible construir nociones matemáticas complejas utilizando herramientas sencillas de la teoría de conjuntos. Una de ellas es la *relación*; las cuales son sumamente importantes en las matemáticas y sus aplicaciones.

**Definición 9** Sea  $A$  y  $B$  conjuntos no vacíos. Una **relación**  $R$  es un conjunto de pares ordenados, donde  $R$  de  $A$  a  $B$  es un subconjunto de  $A \times B$ .

Si  $R \subseteq A \times B$  y  $(a, b) \in R$ , se dice que  $a$  está relacionado con  $b$  por  $R$  y se escribe  $a R b$ . Algunas veces  $A = B$  y en este caso se dice que  $R \subseteq A \times A$  es una *relación sobre*  $A$ .

De manera general, en una relación  $R$ , un elemento  $x \in X$  puede estar relacionado con varios elementos de  $Y$ , o con ninguno. En este sentido muchos símbolos bien conocidos son interpretados como relaciones, por ejemplo, la igualdad ( $=$ ), la desigualdad ( $<$ ,  $>$ ), entre otros.

Por ejemplo, sea  $A = \{1, 2, 3, 4, 5\}$ , se define la relación  $R$  (menor que) en  $A$ :

$$a R b \iff a < b.$$

Por lo que:

$$R = \{(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)\}.$$

Si  $R \subseteq A \times B$  es una relación de  $A$  en  $B$ , existen dos conjuntos asociados a  $R$ :

1. El dominio de  $R$  que se describe como  $Dom(R)$  es el subconjunto de  $A$  siguiente:

$$Dom(R) = \{a \in A / \exists b \in B \text{ con } (a, b) \in R\}$$

2. La imagen de  $R$  que se describe  $Im(R)$  es el subconjunto de  $B$  siguiente:

$$Im(R) = \{b \in B / \exists a \in A \text{ con } (a, b) \in R\}$$

Existen varias formas de representar una función. A continuación mencionaremos dos: la primera es la matriz de relación, válida para relaciones entre dos conjuntos finitos y la segunda, el digrafo de una relación, que sólo sirve para representar relaciones de un conjunto en sí mismo.

**Definición 10** Si  $A$  y  $B$  son dos conjuntos finitos, no vacíos, con  $m$  y  $n$  elementos respectivamente  $A = \{a_1, a_2 \dots a_m\}$  y  $B = \{b_1, b_2 \dots b_n\}$  y  $R$  una relación de  $A$  en  $B$ , llamaremos matriz de la relación  $R$  y se representa por  $M_R$  a la matriz de orden  $m \times n$  siguiente:

$$M_R = (m_{ij}) \text{ donde } m_{ij} = \begin{cases} 1 & \text{si } (a_i, b_j) \in R, \\ 0 & \text{si } (a_i, b_j) \notin R. \end{cases} \quad (2.1)$$

Por ejemplo, si se tiene  $A = \{a, b, c\}$ ,  $B = \{1, 5\}$  y  $R = \{(a, 1), (b, 1), (b, 5)\}$ , entonces  $R$  es una relación de  $A$  en  $B$  y la matriz de la relación  $R$  es:

$$M_R = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 0 \end{pmatrix}$$

Lo anterior permite ver que la relación y la matriz son características, es decir, que al conocer la relación se conoce la matriz y viceversa. De esta manera, la matriz de la relación  $R$ , de  $A$  en  $B$ ,  $M_R$  representa en cada fila a un elemento de  $A$  y en cada columna a uno de  $B$ .

Cuando la relación se da entre elementos del conjunto  $A$ , una forma de representar dicha relación es mediante un método gráfico, el cual consiste en representar a cada elemento del conjunto  $A$  mediante un punto del plano y al par  $(a, b) \in R$  unirlos mediante un arco de curva dirigido de  $a$  hacia  $b$ .

Por ejemplo, si  $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$  y  $R = \{(2, 2)(2, 4)(2, 6)(2, 8)(3, 3)(3, 9)(5, 5)(5, 7)\}$  una relación en  $A$ , el grafo de  $R$  se muestra en la figura 2.1.

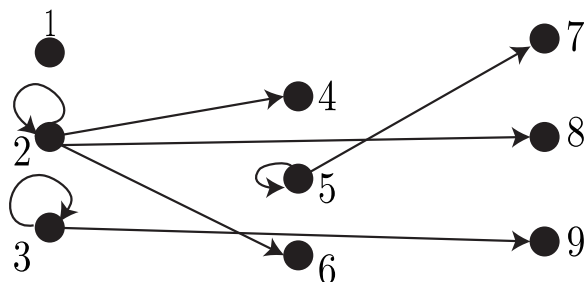


Figura 2.1: Digrafo de la relación  $R$  en  $A$



**Definición 11** *La representación gráfica se llama digrafo de la relación  $R$  y se escribe como  $G_R$ .*

De igual forma que con la matriz de relación el digrafo también caracteriza a la relación.

Algunas de las propiedades de una relación  $R$  en un conjunto  $A$  son las siguientes:

- $R$  es reflexiva si y sólo si, todo elemento de  $A$  está relacionado con él mismo.
- $R$  es irreflexiva si y sólo si, ningún elemento de  $A$  está relacionado con el mismo.
- $R$  es simétrica si, y sólo si,  $a$  relacionado con  $b$  implica  $b$  relacionado con  $a$ .
- $R$  es asimétrica si y sólo si,  $a$  relacionado con  $b$  implica  $b$  no relacionado con  $a$ .
- $R$  es antisimétrica, si y sólo si,  $a$  relacionado con  $b$  y  $b$  relacionado con  $a$  implica  $a$  igual a  $b$ .
- $R$  es transitiva si y sólo si,  $a$  relacionado con  $b$  y  $b$  relacionado con  $c$  implica  $a$  relacionado con  $c$ .

Lo anterior, es la base para estructuras más complejas de las matemáticas, por ejemplo, las relaciones de equivalencia, las de orden y las funciones.

## 2.2. Grafos

El origen de la aplicación de la teoría de grafos se atribuye a Leonhard Euler, matemático suizo, en 1736 quien resolvió el problema de los puentes de Königsberg [Calced(2010)]. Posteriormente, en 1847 Gustav Kirchhoff empleó la teoría de grafos para el análisis de las redes eléctricas [Ullman and Aho(1995)]. Y, en 1840 Francis Guthrie llega a la conjetura acerca del problema de los cuatro colores [Marshall(1971)]. Los grafos son una estructura de datos no lineal, la cual se utilizapara modelar diversas aplicaciones; además se utilizan en otras áreas, por ejemplo la teoría combinatoria .

Existen, en las matemáticas aplicadas, tanto como en las ciencias computacionales, diversas situaciones que pueden representarse mediante un esquema formado por dos cosas:

- Un conjunto (finito) de puntos.
- Un conjunto (finito) de líneas que unen algunos pares de estos puntos.

Los puntos se llaman *vértices* o *nodos* y sus uniones *aristas*; de esta manera:

**Definición 12** Un **grafo**  $G$  es un par ordenado  $(N, L)$ , donde  $N$  es un conjunto finito de vértices o nodos y  $L$  es un conjunto de pares denotados por  $\{x, y\}$ , que se denominan lados o aristas, donde  $x, y \in N$ .

Una representación gráfica de lo anterior se forma a partir del hecho de colocar puntos etiquetados por los elementos del conjunto  $N$ ; las aristas son representadas por líneas que unen los pares  $(x, y) \in L$ . Los elementos de los pares de  $L$  pueden ser ordenados o no, es decir, para representar la relación puede ser importante o no la dirección que lleve; es decir:

**Definición 13** Si  $x$  y  $y$  pertenecen a  $N$  y existe una relación  $R$ , tal que se forma el par ordenado  $(x, y)$  y este pertenece a  $L$  entonces se dice que es un **grafo dirigido**.

La figura 2.2 muestra tres ejemplos de grafos, donde únicamente el grafo de la derecha es un grafo dirigido, mientras que los otros dos son no dirigidos.

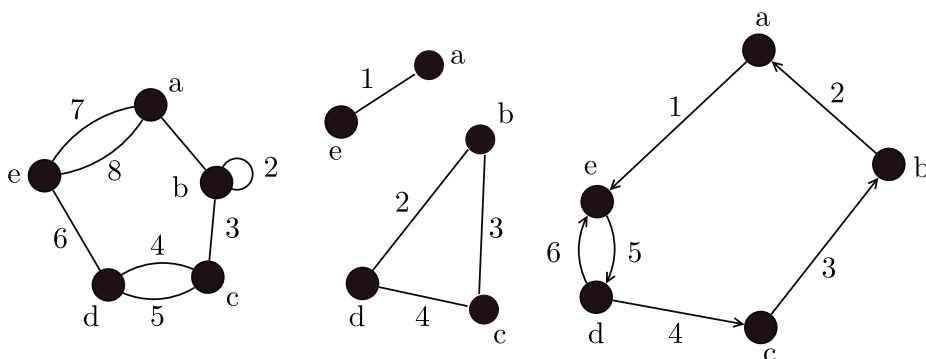


Figura 2.2: Ejemplos de grafos

Existe una variedad de aplicaciones de la teoría de grafos, la más común es utilizarla para representar los objetos y las aristas que se relacionan entre sí; por ejemplo, los mapas con sus rutas de transporte entre ciudades; diagramas de flujo de datos o procesos, o simplemente las relaciones entre personas, como las utilizadas en las redes sociales.

Una característica que poseen los grafos es el valor, peso, nombre o algún símbolo, con el cual se etiquetan las aristas. Así:

**Definición 14** Se le llama un **grafo etiquetado** cuando a las aristas del grafo se le asigna algún valor.

Los grafos etiquetados pueden ser dirigidos o no dirigidos, en la figura 2.3 se muestra un ejemplo de un grafo etiquetado no dirigido. Al hablar de nodos inevitablemente se debe hablar de sus características y propiedades, de tal manera que:

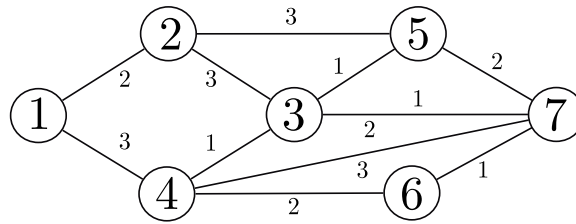


Figura 2.3: Ejemplos de grafos etiquetado

**Definición 15** El número de nodos de un grafo se llama **orden del grafo**. Y al número de aristas se le llama **talla de un grafo**. La representación matemática de la definición anterior es:

$G^n$  Grafo de orden  $n$

$|A(g)| = A$  talla de  $G$

$G(n, m)$  El grafo de orden  $n$  y talla  $m$

De tal forque que el grafo de la figura 2.3 es un grafo  $G^7$  y  $|A(g)| = 11$ ,  $G(7, 11)$  El grafo de orden 7 y talla 11

**Definición 16** El **grado de un vértice**  $x$  es el número de vértices adyacentes a él, esto es, el número de arcos incidentes a  $x$  y se denota como  $grad(x)$ .

Tomando nuevamente la figura 2.3 tenemos que los grados de sus vértices son:  $grad(1)=2$ ,  $grad(2)=2$ ,  $grad(3)=3$ ,  $grad(4)=3$ ,  $grad(5)=3$ ,  $grad(6)=2$ ,  $grad(7)=4$ .

### 2.2.1. Representaciones de un grafo

Las formas en las que se representa un grafo, depende de las características y propiedades que se deseen conocer. En la presente sección se discutirá brevemente algunas de estas representaciones que serán de utilidad en los capítulos posteriores: la representación por medio de matrices, ecuaciones simbólicas y expresiones regulares.

La representación matricial, es quizá, la forma más utilizada, en ella se representan las relaciones del conjunto  $L$ , y se distingue fácilmente de un grafo dirigido y no dirigido. Existen dos tipos de matrices, cada una con sus respectivas propiedades.

La primer matriz a definir es llamada *matriz de incidencias*, la cual está dada por  $I_g$ , donde el grafo es  $G = (N, C)$ . Los elementos de  $I_g$  son denotados por  $g_{ij}$  y son definidos como:

$$g_{ij} = \begin{cases} 1 & \text{Si la arista } j \text{ incide en el vértice } i. \\ 0 & \text{En otro caso.} \end{cases}$$

Una arista *incide* en un vértice si éste pertenece a uno de sus puntos finales. Así cada fila de  $I_g$  corresponde a los diferentes vertices de  $G$ , cada columna corresponde a las distintas aristas de  $G$  de tal forma que si  $N$  tiene  $k$  elementos y  $C$   $m$  elementos,  $I_g$  es una matriz de  $k \times m$ . Para el caso de los grafos dirigidos, la matriz de incidencia está dada por los siguientes valores:

$$g_{ij} = \begin{cases} 1 & \text{Si la arista } j \text{ es dirigida afuera del vertice } i \text{ y no es un ciclo.} \\ -1 & \text{Si la arista } j \text{ es dirigida hacia del vertice } i \text{ y no es un ciclo.} \\ 0 & \text{En todos los otros casos.} \end{cases}$$

De esta manera, las columnas de los los grafos dirigidos suman cero y la suma de las filas puede ser positiva, negativa o cero.

La segunda representación matricial para un grafo es la *matriz de adyacencia*  $A$ , cuyos elementos  $a_{ij}$  son iguales al número de aristas que van del vertice  $i$  al vertice  $j$ . Así  $A$  es una matriz cuadrada de  $k \times k$ , cuando  $G$  es un grafo de  $k$  nodos. Para el caso de los grafos dirigidos, la dirección de  $a_{ij}$  es interpretada exactamente como el estado de la dirección indicada por la palabras “de” y “a”. Para los grafos no dirigidos la especificación de las aristas es tomada en el sentido de las ligas que existen entre dos vertices. Para la matriz de adyacencia no es necesario nombrar las aristas del grafos.

Una derivación es la *matriz topológica*, la cual está definida como:

$$M_{ij} = \begin{cases} 1 & (i, j) \in L \\ 0 & \text{En otro caso} \end{cases}$$

La más importante aplicación de esta matriz consiste en el hecho que sus potencias describen las rutas entre los nodos que contienen multiples ligas de acuerdo a la potencia involucrada. Esto surge de la definición de la multiplicación de matrices donde:

$$\begin{aligned} [M]_{ij}^2 &= \sum_k M_{ik} M_{kj} \\ &= \sum_k \text{link}(ik) \text{link}(kj) \\ &= \sum_k \text{link}(ikj) \end{aligned}$$

Los elementos de la diagonal de esta matriz cuenta los ciclos que tiene cada nodo [McIntosh(2009)].

Otra representación de los grafos es la descrita mediante *ecuaciones simbólicas*, para lo cual se requiere que los nodos sean representados por variables, como letras mayúsculas y las aristas por constantes, representadas por letras minúsculas. Donde las ecuaciones de entrada para el grafo se escriben de la siguiente manera:

$$X = Ya + Zb + \lambda$$

Lo cual significa que, existe una ruta para llegar a  $X$ , formada por los nodos  $Y$  y  $Z$ ; el nodo  $Y$  se relaciona con el nodo  $Z$  por medio de la arista  $a$  y el nodo  $Z$  se relaciona con el nodo  $X$  por medio de la arista  $b$ . Si el nodo  $X$  es el nodo final,  $\lambda$  se puede omitir de la ecuación.

En el caso contrario, donde se tenga una ecuación de salida, es decir, que se parta del nodo  $X$ , se escribe de la siguiente forma:

$$X = aY + bZ + \lambda$$

Donde  $\lambda$  representa los nodos iniciales. En cualquiera de los casos, las entradas del diagrama son descritas por un sistema de ecuaciones los cuales pueden ser escritos en la forma matricial simbólica apropiada. Finalmente, la solución de las ecuaciones simbólicas son las expresiones regulares, los cuales representan conjuntos. En otras palabras, las expresiones regulares proporcionan una descripción concisa de las rutas a través del diagrama. [McIntosh(2009)].

### 2.2.2. Trayectorias y ciclos

Las diversas aplicaciones de la teoría de grafos se debe al uso de trayectorias que recorren el diagrama, en esta sección se dará una breve introducción de las trayectorias y los ciclos de un grafo.

La primer definición a discutir es la *trayectoria* o *camino* por medio de los cuales se recorre el grafo. Así:

**Definición 17** Una **trayectoria**  $T$  en un grafo es una secuencia de nodos  $v_1, v_2, v_3 \dots v_n$  tal que  $(v_1, v_2), (v_2, v_3), \dots (v_{n-1}, v_n)$  son arcos.

Cuando los vértices de una trayectoria son todos distintos, excepto posiblemente el primero y el último la trayectoria se define como una *trayectoria simple*.

**Definición 18** Un **ciclo** o **circuito** es una trayectoria que comienza y termina en el mismo nodo.

Un ejemplo de trayectoria que se puede formar con el grafo 2.3 es:  $T = \{1, 2, 3, 5, 7\}$ ; y un circuito, sería el formado por los vertices  $T_c = \{2, 3, 5, 2\}$ .

Existen algunas familias de grafos que se derivan del concepto de grafos conexos. Este es el caso de los *grafos de Euler* y los *grafos de Hamilton*; los cuales han hecho posible resolver el problema de los puentes de Königsberg [Giudici and Bris(1997), Calced(2010)].

**Definición 19** Se le llama **circuito Euleriano** a un circuito que recorre todos los nodos y todos los arcos pasando al menos, una vez por cada arco.

Es decir, es aquel circuito que realiza un recorrido cerrado que cubra todas las líneas de un grafo, comenzando y terminando en un mismo vertice, sin repetición y en forma continua todas las líneas de un grafo  $G$  cualquiera; o en otras palabras, es una red no dirigida conexa que puede diseñar un circuito que recorra todos sus nodos, partiendo de un nodo específico y regresando a él, pero sin pasar más de una vez por cada arco.

**Definición 20** Se le llama **trayectoria euleriana** a una trayectoria que incluye cada arco exactamente una sola vez.

La figura 2.4 se muestra un grafo, el cual al seguir la trayectoria forma un camino Euleriano, pasando por todos sus vertices y aristas una sola vez. El camino a seguir es el siguiente:  $e, c, a, b, d$ .

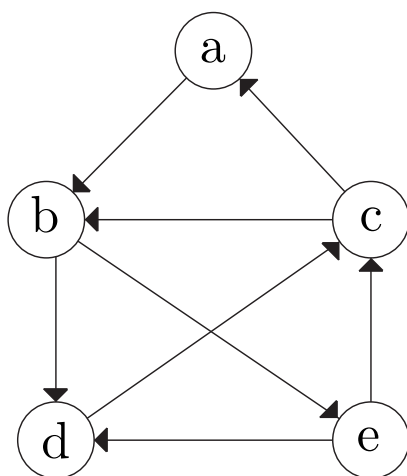


Figura 2.4: Camino Euleriano

**Definición 21** En un grafo  $G = (N, A)$ , una trayectoria (o ciclo) que contenga exactamente  $n - 1$  arcos y recorra todos los nodos, es llamado **hamiltoniano**.

No hay una regla simple para determinar la existencia de un circuito hamiltoniano como si la hay para circuitos eulerianos. Las trayectorias de un grafo son la base para la solución de problemas computacionales, además de que el análisis de dichos caminos, lleva a clasificar información, como por ejemplo, los árboles.

## 2.3. Teoría de autómatas

En la presente sección se describe brevemente una de las herramientas más utilizadas en el área de las ciencias computacionales, la cual ha sido la base para resolver

problemas de diferentes índoles, la más conocida es el reconocimiento de patrones en una máquina: la teoría de autómatas. Además, se describe brevemente como se utiliza para analizar lenguajes formales.

El autómata finito es un modelo matemático de un sistema con entradas y salidas discretas. El sistema puede estar en cualquiera de un número finito de configuraciones o “estados”. El estado del sistema resume la información concerniente a entradas anteriores y que es necesaria para determinar el comportamiento del sistema de entradas posteriores. En las ciencias de la computación encontramos muchos ejemplos de sistemas de estados finitos, y la teoría de autómatas finitos es una útil herramienta para el diseño de tales sistemas.

### 2.3.1. Alfabetos, cadenas y lenguajes

Los elementos básicos que se deben considerar para comenzar a estudiar los autómatas finitos, son los alfabetos y lenguajes que se pueden formar con ellos. Por lo tal, se inicia con el elemento básico, denominado como símbolo. Un símbolo es una entidad abstracta que se define con un propocito, por ejemplo, las letras y los números en la vida cotidiana, los cuales se utilizan para determinar el lenguaje español, o para nombrar el número de años de una persona. Los símbolos, en ocasiones, forman parte de un conjunto particular llamado *alfabeto*, el cual se define como:

**Definición 22** *Un alfabeto se define como un conjunto finito, no vacío de símbolos.*

Usualmente se utiliza el símbolo  $\Sigma$  para un alfabeto. Por ejemplo, el alfabeto binario se escribe:  $\Sigma = \{0, 1\}$ .

**Definición 23** *Una cadena o palabra es una secuencia finita de símbolos yuxtapuestos elegidos de algún alfabeto  $\Sigma$ .*

Por ejemplo, 01101 es una cadena que se forma con el alfabeto binario  $\Sigma = \{0, 1\}$ .

**Definición 24** *Una cadena vacía es la cadena con ningún símbolo. Se denota como  $\epsilon$ .*

**Definición 25** *La longitud de una cadena  $w$ , que se denota como  $|w|$ , es el número de posiciones de símbolos que componen la cadena.*

Por ejemplo 0101 tiene una longitud 4. En el caso de la cadena vacía  $|\epsilon| = 0$ .

Si  $\Sigma$  es un alfabeto, el conjunto de todas las cadenas de una cierta longitud de dicho alfabeto se expresa utilizando una notación exponencial, es decir:

**Definición 26** *Se define  $\Sigma^k$  al conjunto de cadenas de longitud  $k$ , cuyos símbolos pertenecen a  $\Sigma$ .*

Por ejemplo,  $\Sigma^0 = \{\epsilon\}$ , o si se tiene  $\Sigma = \{0, 1\}$ , entonces  $\Sigma^2 = \{00, 01, 10, 11\}$ .

Un caso particular de lo anterior, es llamado *clausura de Kleene*, la cual representa el conjunto de todas las cadenas del alfabeto  $\Sigma$  y se denota como  $\Sigma^*$ ; por ejemplo, para  $\{0, 1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$ . Visto de otra manera  $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$ . En algunas ocasiones, es posible excluir la cadena vacía del conjunto de cadenas; este conjunto se representa como  $\Sigma^+$ . Así, dos equivalencias apropiadas son:

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \dots$$

$$\Sigma^* = \Sigma^+ \cup \{\epsilon\}.$$

La *concatenación* de dos cadenas es la cadena que se forma al escribir la primera seguida de la segunda, sin que haya espacio entre ellas. Más precisamente, si  $x$  es una cadena compuesta de  $i$  símbolos,  $x = a_1a_2\dots a_i$ , y  $y$  es la cadena compuesta de  $j$  símbolos,  $y = b_1b_2\dots b_j$ , entonces  $xy$  es la cadena de longitud  $i + j$ :  $xy = a_1a_2\dots a_ib_1b_2\dots b_j$ . Por ejemplo, la concatenación de 0011010 y 0000 es 0011010000. La yuxtaposición se utiliza como el operador de concatenación.

**Definición 27** *Un lenguaje es un conjunto de cadenas de símbolos de algún  $\Sigma^*$ , donde  $\Sigma$  es un alfabeto particular.*

Si  $\Sigma$  es un alfabeto, y  $L \subseteq \Sigma^*$ , entonces  $L$  es un *lenguaje a través de  $\Sigma$* . Se debe considerar que un lenguaje no incluye todas las cadenas, únicamente las establecidas por el lenguaje  $L$ .

El conjunto vacío  $\phi$  y el conjunto formado por la cadena vacía  $\epsilon$  son lenguajes. Por ejemplo, si  $\Sigma = \{a\}$ , entonces  $\Sigma^* = \{\epsilon, a, aa, aaa, aaaa, \dots\}$ .

### 2.3.2. Autómatas finitos

Un autómata finito (AF) consiste en un conjunto finito de estados y un conjunto de transiciones de estado a estado, que se dan sobre símbolos de entrada tomados de un alfabeto  $\Sigma$ . Para cada símbolo de entrada existe exactamente una transición a partir de cada estado. Un estado inicial, por lo general denotado como  $q_0$ , es el estado donde el autómata comienza. Existen estados designados como finales o de aceptación.

La definición formal de un AF es la siguiente:

**Definición 28** *Un AF se define como la quintupla  $(Q, \Sigma, \delta, q_0, F)$  donde  $Q$  es un conjunto finito de estados,  $\Sigma$  un alfabeto de entrada finito,  $q_0 \in Q$  el estado inicial,  $F \subseteq Q$  el conjunto de estados finales y  $\delta$  la función de transición que transforma  $Q \times \Sigma \rightarrow Q$ .*



A un autómata finito (AF) se le asocia un grafo dirigido, conocido como *diagrama de transiciones* de la manera siguiente: los vértices del grafo corresponden a los *estados* del AF; si existe una transición del estado  $q$  al  $p$  sobre la entrada  $a$ , entonces existe un arco con etiqueta  $a$  que va del estado  $q$  al  $p$  en el diagrama de transiciones. Un AF acepta una cadena  $X$ , si la secuencia de transiciones correspondientes a los símbolos de  $X$  del estado inicial a un estado de aceptación.

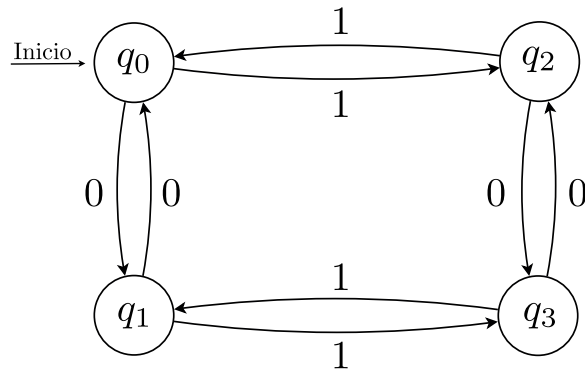


Figura 2.5: Ejemplo de autómata finito

La figura 2.5 ilustra el diagrama de transiciones de un AF. El estado inicial es  $q_0$ . Existe solo un estado final, también  $q_0$  marcado por un doble círculo. El AF acepta todas las cadenas de ceros y unos, en las que el número de ceros y unos son pares.

Existen dos notaciones importantes que no son autómatas, pero juegan un rol importante en el estudio de autómatas y sus aplicaciones.

La primera son las *gramáticas*, las cuales son utilizadas como modelos cuando se diseña software que procesa datos con una estructura recursiva. El ejemplo mejor conocido es el *parseo*, el componente de un compilador que procesa con las funciones recursivas de los lenguajes típicos de programación.

Las *expresiones regulares* también son denotadas como estructuras de datos, especialmente cadenas de texto. Los patrones de cadenas descritas son exactamente iguales a las que puede describir un autómata finito.

Existen diferentes tipos de autómatas, los más conocidos, por su uso y aplicación son los autómatas finitos deterministas y los no deterministas.

El término *determinista* se refiere al hecho de que cada entrada tiene solo un estado con el cual el autómata puede realizar una transición de estado. En contraste con los autómatas *no deterministas*, pueden existir varias transiciones entre estados

para uno sólo.

Un autómata finito determinista (AFD) consiste en:

- Un conjunto finito de estados,  $Q$ .
- Un conjunto finito de *símbolos de entrada*,  $\Sigma$ .
- Una *función de transición*  $\delta$ , la cual toma como argumentos un estado y un símbolo de entrada y regresa un estado.
- Un *estado inicial*  $q_0 \in Q$ .
- Un conjunto de *estados finales o aceptados*  $F$ . El conjunto  $F \subseteq Q$ .

Los AFD poseen un *lenguaje*, que es el conjunto de todas las cadenas aceptadas. Esto es, supongase que  $a_1a_2 \dots a_n$  es una secuencia de símbolos de entrada. Se inicia en el estado  $q_0$ . Posteriormente, se calcula la función  $\delta$ , donde  $\delta(q_0, a_1) = q_1$ , donde  $q_1$  es el estado que toma a un AFD después de procesar el símbolo  $a_1$ . Después se procesa el siguiente símbolo de entrada  $a_2$ , se evalúa  $\delta(q_1, a_2) = q_2$ . Así se continúa hasta evaluar todos los símbolos de entrada. Si  $q_n$  es el estado final de evaluar el último símbolo es un elemento de  $F$ , entonces la cadena de entrada  $a_1a_2 \dots a_n$  es aceptada y si no, entonces es rechazada.

Sin embargo, explicado con palabras parece complicado de entender, por ello, existen notaciones simples para comprender mejor el concepto.

La primera notación es un *diagrama de transición*, el cual es un grafo definido por lo siguiente:

- Para cada estado en  $Q$  existe un nodo
- Para cada estado  $q \in Q$  y para cada símbolo  $a$  en  $\Sigma$ , se tiene  $\delta(q, a) = p$ . Entonces el diagrama de transición tiene un arco del nodo  $q$  al nodo  $p$ , etiquetado con  $a$ .
- Existe una flecha de *inicio* en el estado  $q_0$ .
- Los nodos correspondientes a los estados aceptados son marcados por un doble círculo.

La figura 2.6 muestra el diagrama de transición para el AFD que acepta todas las cadenas con una subcadena 01. El diagrama tiene tres nodos, que corresponden a los tres estados. Existe una flecha de *Inicio* mostrando el estado inicial  $q_0$ ; el estado de aceptación  $q_1$  es representado con un doble círculo. Cada arco es etiquetado con 0 ó 1, los cuales son el resultado de la evaluación de la función  $\delta$ .

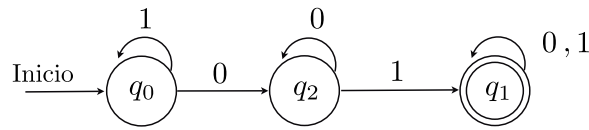


Figura 2.6: Ejemplo de autómata finito determinístico

Una tabla de transición es una representación tabular convencional de una función como  $\delta$ , que toma dos argumentos y regresa un valor. Las filas de la tabla corresponden a los estados y las columnas a los símbolos de entrada.

Por ejemplo, la tabla 2.1 corresponde a la función  $\delta$  del ejemplo mostrado anteriormente. Las características de una tabla de transición son las siguientes: el estado inicial es marcado con una flecha y los estados de aceptación son marcados con un asterisco. Las filas representan los estados y las columnas los símbolos de estados y la intersección entre ellos será el estado como resultado de la función  $\delta$ .

	0	1
$\rightarrow q_0$	$q_2$	$q_0$
$*q_1$	$q_1$	$q_1$
$q_2$	$q_2$	$q_1$

Tabla 2.1: Ejemplo de tabla de transición.

Una de las aplicaciones que se le da a los AFD es poder definir lenguajes, por lo que se define el *lenguaje* de un AFD como  $A = (Q, \Sigma, \delta, q_0, F)$ . Este lenguaje se escribe  $L(A)$  y se define como:

$$L(A) = \{w \mid \hat{\delta}(q_0, w) \in F\}$$

Lo cual significa que el lenguaje de  $A$  es el conjunto de cadenas  $w$  que inicia en el estado inicial  $q_0$  y que termina en un estado de aceptación. Si  $L$  es  $L(A)$  para un AFD  $A$ , entonces se dice que  $L$  es un *lenguaje regular*.

Las operaciones que se pueden utilizar para los lenguajes regulares son las siguientes:

- La *unión* de dos lenguajes  $L$  y  $M$ , escrito  $L \cup M$ , es el conjunto de cadenas que están en  $L$  o  $M$  o ambos. Por ejemplo,  $L = \{001, 10, 111\}$  y  $M = \{\epsilon, 001\}$ , entonces  $L \cup M = \{\epsilon, 10, 001, 111\}$ .
- La *concatenación* de lenguajes  $L$  y  $M$  es el conjunto de cadenas que pueden ser formadas al colocar una cadena de  $L$  seguida de otra cadena de  $M$ . Por ejemplo, si  $L = \{001, 10, 111\}$  y  $M = \{\epsilon, 001\}$ , entonces  $LM = \{001, 10, 111, 001001, 10001, 111001\}$ .



# Capítulo 3

## Diagramas de de Bruijn en AC 1D

En este capítulo se discutirá acerca de la definición de los autómatas celulares y un panorama general del análisis que se ha realizado a éstos. Posteriormente, se da una breve introducción a los diagramas de de Bruijn, sus antecedentes y definiciones; finalmente, se expone una discusión sobre el uso de los diagramas de de Bruijn para el análisis de los autómatas celulares de una dimensión.

### 3.1. Autómatas celulares

Los autómatas celulares (AC) son modelos prototipos para sistemas complejos y su proceso consiste de un gran número de componentes idénticos, simples y que interactúan localmente. El estudio de estos sistemas ha promovido gran interés, debido a su capacidad de generar un amplio espectro de patrones muy complejos en su comportamiento, derivados de un conjunto de reglas relativamente simples.

Existen al menos cuatro formas parcialmente intersectadas para el estudio de los autómatas celulares: como ingeniería computacional poderosa; como simuladores de sistemas dinámicos discretos, como herramientas conceptuales para la construcción de patrones y como modelos físicos.

#### 3.1.1. Definición

Los autómatas celulares son una clase de sistemas matemáticos determinísticos, discretos en espacio y tiempo caracterizados por la interacción local y una forma paralela de evolución. Los autómatas celulares pueden ser tratados como sistemas dinámicos discretos abstractos, dentro de los cuales se encuentran características interesantes de comportamiento.

Los autómatas celulares tienen las siguientes características:

- Un espacio que está formado por un conjunto finito de subespacios homogéneos, llamadas *células*, distribuidas en una rejilla regular de dimensión  $d$ .

Los autómatas pueden ser de una dimensión o también llamados lineales; dos dimensiones, tres dimensiones o hasta  $d$  dimensiones, donde su estudio es propiamente matemático.

- Cada célula puede estar en un único *estado* en determinado intervalo de tiempo  $t$ . El estado debe estar definido en un conjunto finito de estados asociados al espacio del autómata.
- Una *configuración inicial* que es el estado en el que se encuentran todas las células del espacio en el tiempo  $t = 0$ .
- El estado de las células cambia de un instante a otro, de acuerdo a un conjunto de *reglas de evolución* comunes a todas las células. Estas reglas definen cómo debe de cambiar de estado, dependiendo del estado propio y de sus vecinos.
- La *vecindad* de una célula está formado por las células adyacentes y se define igual para todas las células del autómata.

De manera formal un AC se define como la tupla  $(\Sigma, d, \rho, N, f)$ , donde los elementos se definen como:

- $\Sigma$ : el conjunto de estados finitos del autómata celular.
- $d$ : la dimensión del espacio donde se encuentran las células.
- $\rho$ : se define como  $\rho : \Sigma_1 \times \Sigma_2 \times \Sigma_3 \cdots \times \Sigma_n \rightarrow \Sigma$ , donde  $n$  especifica el número de células que se necesitan para definir la vecindad, la cual es necesaria para evaluar la función de transición local.
- $N$ : el conjunto de estados que tiene cada una de las células que forma el espacio antes de la primera evaluación de la función de transición.
- $f : \Sigma^p \rightarrow \Sigma$ : función de transición local.

La figura 3.1 muestra cómo se verían las regiones de células en una, dos y tres dimensiones.

A lo largo del estudio de los AC se han creado muchas reglas y se han analizado muchas más. Las reglas de AC más famosas en una dimensión son: las 256 reglas elementales, de dos estados con una vecindad de tres, estudiadas y analizadas con respecto a su comportamiento por Wolfram [Wolfram(2002)]. Las reglas de más de dos estados con vecindades de tres células, que han servido para encontrar reglas reversibles. En el caso de dos dimensiones, esta la máquina autoreproductora de von Neumann con 27 estados y una vecindad en forma de cruz que dió origen a la idea de la teoría de la computación y la universalidad en los AC [von Neumann(1966)]. La regla del Juego de la Vida con 2 estados y una vecindad cuadrada, estudiada y analizada por sus propiedades y simulación computacional y el comportamiento complejo que

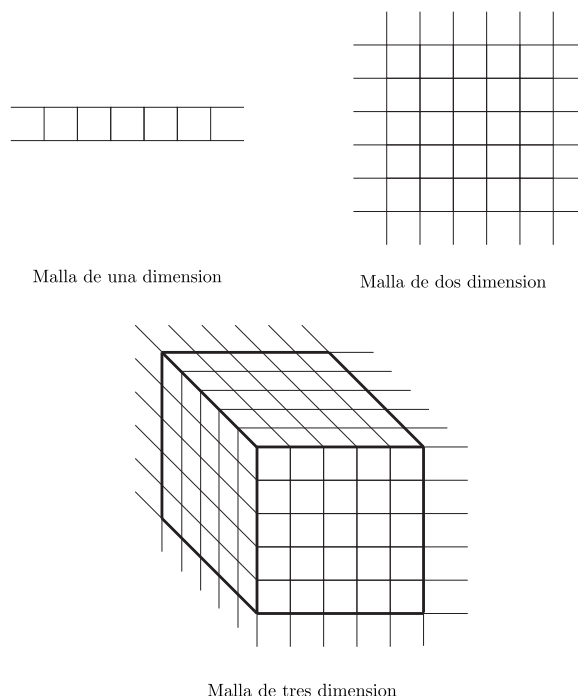


Figura 3.1: Ejemplos de mallas de dimension 1 2 y 3 de AC

presenta [Gardner(1970)]. Además de distintos modelos físicos simulados por AC tales como la regla del Juego de Billar, la regla de la hormiga, las reglas de reacción difusión, por mencionar algunas [Gutowitz(1991)]. En tres dimensiones, existen pocas reglas estudiadas, la mayoría son extensiones de reglas bidimensionales tales como Life en 3 dimensiones, o las que simulan modelos de reacción difusión en tres dimensiones [Ilachinski(2001)]. En dimensiones de orden mayor a tres se encuentran los AC estudiados por Chaté Manneville que se utilizaron para el análisis del comportamiento colectivo no trivial [Chaté and Manneville(1996)].

La figura 3.2 muestra algunos ejemplo de la evolución de la regla 75 de AC unidimensionales, el Juego de la Vida y la regla de la Hormiga, donde se aprecia que la dinámica de cada una tiene características propias, es por ellos que el estudio en cada dimensión y a cada tipo de regla puede ser variada.

### 3.1.2. Introducción al estudio de los AC unidimensionales

En el caso particular de los AC unidimensionales (AC 1D) el desarrollo del análisis se ha dado de manera significativa desde los años 80's. Después de que Wolfram realizó el estudio estadístico del comportamiento de los AC 1D (2,1), según la notación de Wolfram AC 1D, dos estados y radio 1 [Wolfram(1984)] y donde propuso la clasificación de 4 tipos de comportamientos, se dio pauta a que muchos investigadores

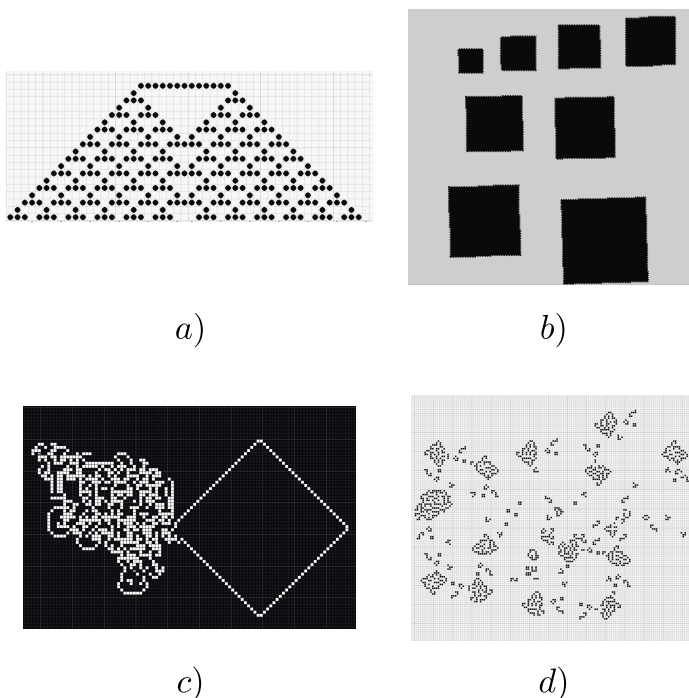


Figura 3.2: Ejemplos de reglas de AC estudiadas

siguieran esa línea y se encontraran más propiedades en los AC 1D.

El análisis realizado a los AC 1D va desde el análisis de comportamiento y dinámica hasta el análisis riguroso de las matemáticas, para lo cual, se ha propuesto una serie de definiciones, que se mencionarán en la presente sección.

La definición de los AC 1D ha sido propuesta de diferentes formas, sin embargo para los fines de esta tesis tomaremos la siguiente:

**Definición 29** *Un autómata celular unidimensional es una secuencia infinita de símbolos que pertenecen a un alfabeto infinito  $\Sigma$ , que evoluciona en el tiempo, dependiendo de sus vecinos, en un radio  $r$  de acuerdo a una función  $f : \Sigma^{2r+1} \rightarrow \Sigma$ .*

Con la finalidad de estudiar de manera formal la dinámica y propiedades de las reglas unidimensionales, se propuso la siguiente definición de una vecindad unidimensional [Ilachinski(2001)]:

$$\rho_i(t) = \{x_{i+r}(t), \dots, x_{i+1}, x_i, x_{i-1}, \dots, x_{i-r}\} \quad (3.1)$$

Esta vecindad representa al conjunto de células vecinas de la célula  $x_i$ ; esto es, tomando una célula como elemento  $x_i$ , sus vecinos serán todas las células que están  $r$  posiciones a la izquierda y todas las células que están  $r$  posiciones a la derecha así como se muestra en la figura 3.3.



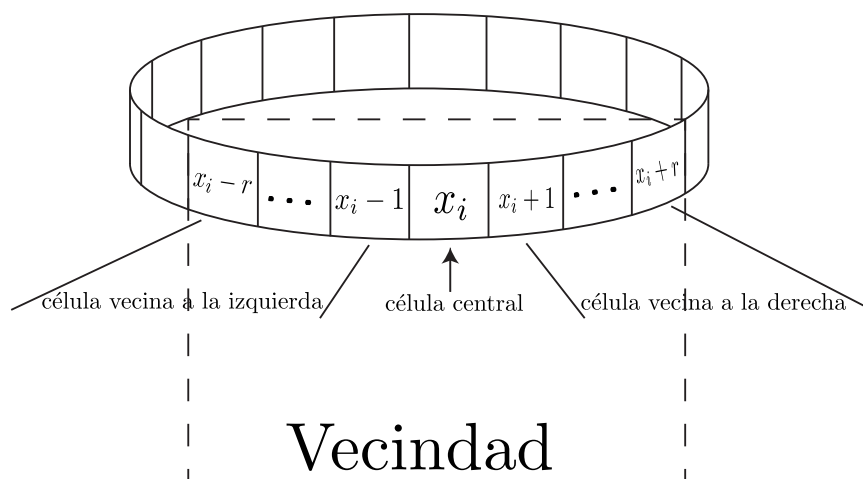


Figura 3.3: Vecindad de radio  $r$  en un AC 1D

Considerando que el resultado de la función de transición local depende de la vecindad, entonces tenemos que la función se puede definir de la siguiente manera para una vecindad de longitud  $r$ .

$$f_i(t+1) = \phi(f_{i-r}(t), \dots, f_i(t), \dots, f_{i+r}(t)); \phi : \Sigma^{2r+1} \rightarrow \Sigma \quad (3.2)$$

Recordando que los AC se definen como la quintupla  $(\Sigma, d, \rho, N, f)$  tenemos que para una AC 1D (2,1) los siguientes valores:

- $\Sigma : \{0, 1\}$ .
- $d = 1$ .
- $\rho_i(t) = \{x_{i+1}, x_i, x_{i-1}\}$ .
- $N$  es una configuración aleatoria.
- $f_i(t+1) = \phi(f_{i-1}(t), f_i(t), f_{i+1}(t)); \phi : \Sigma^{2+1} \rightarrow \Sigma$ .

Reglas de este tipo fueron estudiadas por Wolfram para su clasificación, donde el valor que cambia entre las 256 *reglas elementales* de los AC 1D es el último, la función de transición. Al iniciar la evolución de los AC desde una configuración aleatoria, Wolfram se dio cuenta que cada regla de evolución tiene su propia “personalidad”, que algunas reglas tienen similitudes y también significativas diferencias, de esta manera, después de un estudio sistemático se llegó a que los AC 1D se pueden clasificar en cuatro categorías:

- Clase I. Comportamiento uniforme.
- Case II. Comportamiento periódico.
- Clase III. Comportamiento caótico.

- Clase IV. Comportamiento complejo.

Estas clases se derivaron de una clasificación en mecánica no lineal [Wolfram(1983)].

Posterior a este trabajo, se realizó una serie de estudios relacionados, entre los que destacan los realizados por Jen, quien utiliza un cilindro y la teoría de corrimientos para encontrar la clasificación de Wolfram, por medio de un método alternativo [Jen(1986)].

La teoría de corrimientos en conjunto con la teoría de grafos, Jen y otros autores realizaron el cálculo de *Ancestros*, configuraciones que al evolucionar  $t$  tiempos llegarán a una configuración en particular, *Jardines del Eden*, configuraciones que no tienen ancestros, y *Ciclos*, configuraciones que se repiten cada  $t$  tiempos, por medio de una herramienta llamada diagramas de de Bruijn, tema que se abordará en la siguiente sección.

Se realizaron los estudios en los AC 1D cuyo comportamiento global es invertible, es decir, que cada uno de estos autómatas puede regresar a todos los estados globales que produjo anteriormente. Estos AC llamados *reversibles* y su estudio matemático resulta relevante debido a la convergencia que tiene en diversos campos de investigación [Morita(2008)].

A finales de los 80's aparecen trabajos importantes acerca de los AC 1D reversibles: Morita prueba la existencia de AC reversibles que soportan computación universal y son autorreproductivos [Morita(2008)]; Hillman implementa algoritmos para encontrar AC 1D reversibles [Hillman(1991)]; Boykett propone un esquema algebraico para construir AC reversibles [Boykett()]; por mencionar algunos ejemplos.

Uno de los trabajos más completos e importantes realizados sobre los AC 1D fue el realizado por Voorhees, al analizar con herramientas matemáticas los AC 1D. Entre las principales aportaciones que hizo se pueden mencionar, la clasificación de las reglas aditivas, propiedades para AC reversibles y el estudio mediante los diagramas de de Bruijn [Voorhees(1996)].

## 3.2. Diagramas de de Bruijn

En la presente sección se discutirá acerca de los antecedentes de los diagramas de de Bruijn; además de desarrollar las definiciones de los elementos.

### 3.2.1. Antecedentes

Los diagramas de de Bruijn fueron propuestos por Nicolaas Govert de Bruijn como solución del siguiente problema:

**Problema 1** Dado un  $m+1$  símbolos y un entero positivo  $n$ , encontrar un algoritmo para generar una secuencia de símbolos que tenga una longitud mínima y además, que cuando sea colocados en un círculo contengan como subsecuencias de símbolos consecutivos todas las secuencias de símbolos de longitud  $n$ .

El problema ha sido resuelto de diferentes formas desde 1894, cuando Rivière encontró una solución para  $m = 1$ ; posteriormente Flye Sainte-Marie y Mantel hicieron contribuciones al problema, y en los años 30's Martin demostró la existencia de las secuencias para cualquier  $m$  y  $n$  por medio de un algoritmo que creaba las secuencias, finalmente de Bruijn solucionó el problema mediante grafos para  $m = 1$  dejando ver factible la extensión de su trabajo a cualquier  $m$  [Ralston(1982)].

La esencia del problema es el siguiente: si se tiene un  $m = 2$  y un  $n = 2$ , entonces, se tienen tres símbolos  $S = \{0, 1, 2\}$ ; de los cuales se obtendrán subsecuencias de longitud 2.

**Definición 30** Una secuencia se define como una sucesión de símbolos de la forma:  $s_i s_{i+1} s_{i+2} \dots s_k$  donde  $k$  es la longitud de la secuencia.

El número de subsecuencias se calcula con  $(m+1)^n$ . Por ejemplo si  $m = 2$  y  $n = 2$ , entonces se tendrán  $(2+1)^2 = 3^2 = 9$  subsecuencias.

**Definición 31** La subsecuencia está definida como la sucesión de símbolos de una longitud  $n$ , que se escribe como:  $s_i s_{i+1} s_{i+2} \dots s_{i+(n-1)}$ , para cada uno de los símbolos.

Continuando con el ejemplo, las subsecuencias resultantes son:  $S_s = \{00, 01, 02, 10, 11, 12, 20, 21, 22\}$ .

**Definición 32** Se define como una secuencia de de Bruijn a toda secuencia de longitud  $(m+1)^n$  que permite generar  $(m+1)^n$  subsecuencias de longitud  $n$  al colocarse en una circunferencia.

El objetivo de colocar la secuencia en una circunferencia es para cerrar la secuencia, es decir, que el símbolo inicial y el final se unan para que no exista inicio ni fin a la secuencia. Por ejemplo, la secuencia  $s = 221201100$  genera todas la subsecuencias existentes de longitud 2 y por lo tanto es una secuencia de de Bruijn. La figura 3.4 muestra las subsecuencias.

### 3.2.2. Definición

En la presente sección se describe la definición de los diagramas de de Bruijn y la forma en la que se relacionaron con los autómatas celulares lineales, de tal manera que se convirtieron en una herramienta útil para su análisis. Tomando como referencia las definiciones de la sección anterior, se tiene que:

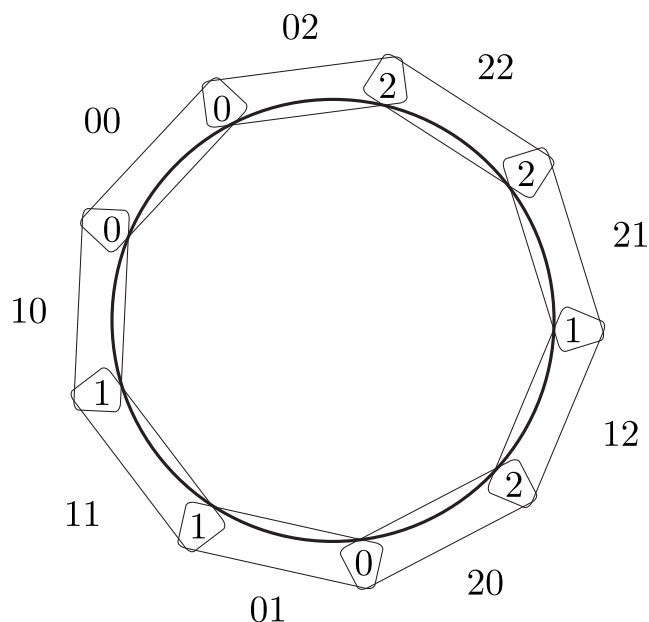


Figura 3.4: Secuencia de de Bruijn y sus subsecuencias de longitud 2.

**Definición 33** El diagrama de de Bruijn es un tipo de grafo dirigido con  $(m + 1)^n$  nodos, los cuales representan secuencias de longitud  $n$  de  $m + 1$  símbolos, donde al menos existe un símbolo que se intersecta.

**Definición 34** Un nodo es una subsecuencia de longitud  $n - 1$  de  $m + 1$  símbolos. Existen  $(m + 1)^n$  nodos en un diagrama de de Bruijn.

**Definición 35** La unión entre dos nodos a través de una intersección de símbolos entre las secuencias forma una arista, es decir, cada secuencia compartirá al menos un símbolo para poder unirse y formar una subsecuencia de longitud  $n$ .

La figura 3.5 ilustra las definiciones de secuencia, y subsecuencia, así como los nodos correspondientes.

La figura 3.6 muestra el diagrama para el caso  $n = 3$  y  $m = 1$ , donde se obtienen secuencias de longitud 2 de un alfabeto de 2 símbolos. Estas secuencias serán los nodos y las aristas representan la intersección entre los símbolos centrales.

Dada la naturaleza de los diagramas de de Bruijn se han podido aplicar a la teoría de AC. Esto se debe a que pueden representar de una forma natural y conveniente la evolución de las vecindades de un AC. Generalizando, los componentes de un diagrama de de Bruijn se pueden ver de la siguiente manera:

- El conjunto de  $m + 1$  símbolos representa los estados del AC.

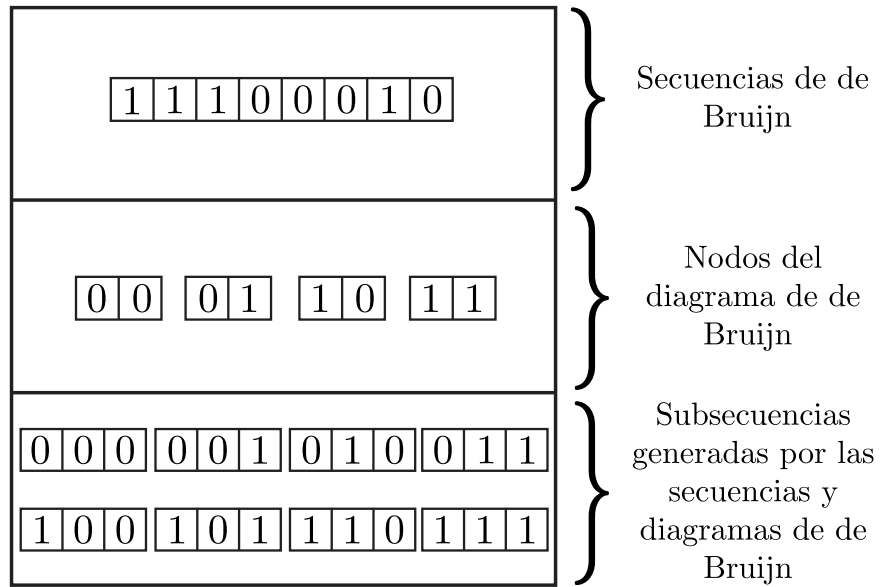


Figura 3.5: Elementos de un diagrama de de Bruijn

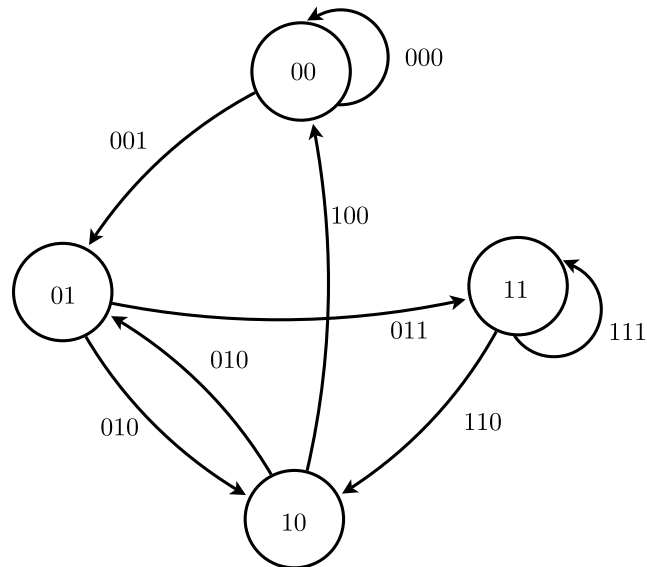


Figura 3.6: Diagrama de de Bruijn para  $m = 1$ ,  $n = 3$ .

- Las cadenas de longitud  $n - 1$ , representan una parte de la vecindad, que llamaremos *vecindades parciales*.
- Los símbolos de intersección representarán las células de AC, que serviran para unir dos *vecindades parciales* y donde al menos uno de ellos representa la célula a evaluar del AC.

Estas fueron las bases, que junto con la teoría de corrimientos de secuencias, dieron origen al desarrollo del estudio de los AC 1D y serán la base para el desarrollo del estudio de los AC en dos dimensiones, pero con sus respectivas limitantes, de las cuales se hablará a lo largo de esta tesis.

### 3.3. Análisis de autómatas celulares en 1D vía de Bruijn

Historicamente los diagramas de de Bruijn fueron creados para resolver los problemas de encontrar todas las distintas secuencias de ciertos símbolos.

El estudio de los AC 1D surge como idea de Jen al utilizar la teoría de corrimientos para encontrar configuraciones estáticas y periódicas en un cilindro de circunferencias, con lo cual ha mostrado muchas propiedades de los AC que pueden ser extraídos de diagramas que representan estos corrimientos, llamados diagramas de de Bruijn [Jen(1988)]. Jen y posteriormente Gómez realizaron la búsqueda de preimágenes utilizando los diagramas de de Bruijn [Jen(1987), Gómez(2008)]. La más simple aplicación para encontrar los ancestros es utilizando las cadenas uniformes, las cuales surgen del hecho de que cada liga representa la evolución de la célula central en una vecindad.

Los trabajos más importantes que se han realizado en el análisis de los AC 1D por medio de los diagramas de de Bruijn se desarrollarán en la presente sección.

Recordando la definición de AC 1D que se dió en la sección anterior, se puede decir que un AC 1D es un arreglo finito o infinito de células ordenadas en una malla discreta. Cada célula tendrá uno de los  $k$  valores, donde  $k$  es la cardinalidad del conjunto  $\Sigma$  y se representa como:

$$c \in \Sigma = \{0, 1, \dots, k - 1\}$$

De esta manera, podemos definir que:

**Definición 36** *La configuración de un arreglo de células es representado por una cadena  $\alpha = \dots c_{i-1}c_x c_{i+1} \dots$*

Y así, la configuración global de un AC en un tiempo  $t$  se representa por la configuración  $C^t$ .

$$C^t = \{\dots c_{i-1}c_i c_{i+1} \dots\}.$$

**Definición 37** La vecindad de un AC 1D se define como la cadena de células en un radio  $r$  alrededor de la célula a evaluar; y de que su tamaño es de  $m = 2r + 1$ .

Además, se expresa de la siguiente manera:

$$\rho_i = c_{i-r} \dots c_i \dots c_{i+r}$$

La función de transición  $f$  se evalúa en función de la vecindad  $\rho$ , así en cada paso de evolución de los AC, todas las células de la configuración  $C^t$  son evaluadas sincronizadamente para dar como resultado una configuración futura  $C^{t+1}$ .

La evolución de cada célula  $c_i^{t+1}$  dentro de su valor futuro  $c_i^{t+1}$  es definido como la salida de la función de transición local  $f$  que toma como entrada la vecindad presente  $\rho_i^t$  de la célula observada en la posición  $i$ , y se representa:

$$c_i^{t+1} = f(n^{t_i} = f(c_{i-1}^t c_i^t c_{i+1}^t))$$

La figura 3.7 muestra que, con respecto al tiempo, siempre se tomará la misma vecindad y el valor que tenga  $C^t$ .

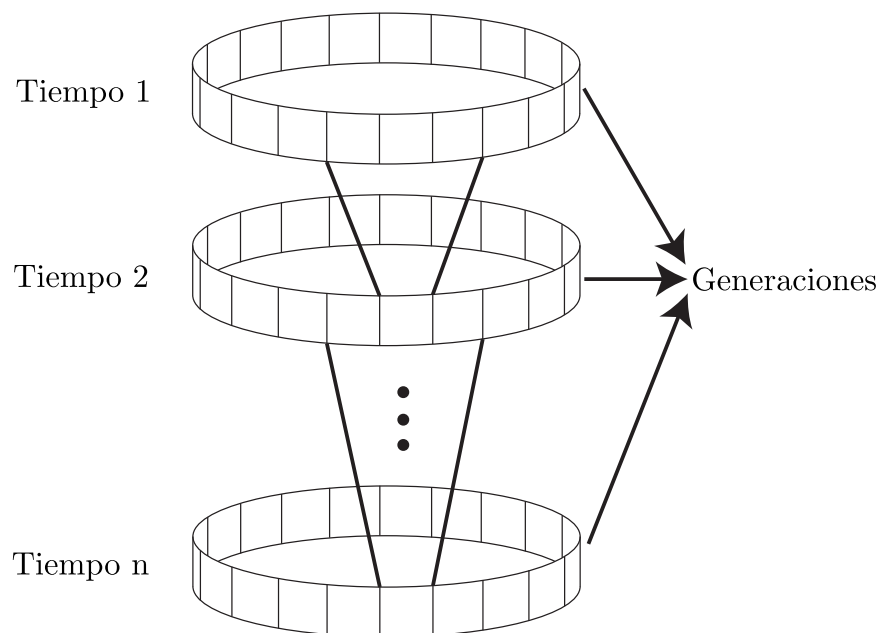


Figura 3.7: Evoluciones en un tiempos  $t$  con relación a la vecindad

Una vez que hemos definido las partes del AC para el caso específico del AC 1D definamos las partes del diagrama de de Bruijn.

La definición 34 cuando se aplica a los AC 1D queda de la siguiente manera:

**Definición 38** *Los nodos de los diagramas de de Bruijn para los AC 1D se definen como las vecindades parciales o sub cadenas de  $C$ , de longitud  $r + 1$ .*

Las ligas entre los nodos se darán mediante la condición especificada en la definición 35, donde la secuencia resultante representa una vecindad, y es de longitud  $2r + 1$  y el símbolo que se intersecta para formar la secuencia debe ser la célula a evaluar o también conocida como célula central. Las ligas se definen de la siguiente manera:

**Definición 39** *Las ligas entre nodos se dará por medio de un elemento de la sub secuencia de  $C$ , la cual representa la célula a evaluar del AC. La secuencia resultante formará una vecindad del AC 1D  $(k,r)$ .*

En el siguiente ejemplo, se desarrollarán las definiciones anteriores, y además se mencionarán algunos detalles importantes. Los elementos importantes a definir con un AC 1D  $(2,1)$ , es decir de 2 estados y radio 1, son los siguientes:

- Vecindad: la vecindad está compuesta por  $2r + 1 = 3$  elementos.
- Nodos: los nodos que componen al diagrama de de Bruijn son de una longitud  $r + 1 = 2$ .
- Ligas: las ligas forman secuencias de longitud 3.

La figura 3.8 muestra el diagrama de de Bruijn de la regla 240 de los AC 1D  $(2,1)$ , con lo cual podemos observar lo siguiente:

- El diagrama de de Bruijn está compuesto por  $k^{r+1}$ , es decir  $2^{1+1} = 4$  nodos.
- El símbolo que se intersecta es el segundo elemento de izquierda a derecha del primer nodo, y el primer símbolo de izquierda a derecha del segundo nodo.
- La etiqueta de la arista está formada por dos partes, la primera es la cadena que resulta de la intersección de dos nodos, la cual representa una vecindad y la segunda parte es el estado que tendrá la célula al ser evaluada en un tiempo  $t + 1$ .

Consecuentemente, las características de la evolución pueden ser usadas para seleccionar subgrafos, del diagrama de de Bruijn, por ejemplo, el subgrafo compuesto por las vecindades cuya célula central nunca cambie. De esta manera, las propiedades globales de los AC pueden ser leídas en términos de las cadenas que son encontradas en dichos subdiagramas; para el ejemplo anterior, las cadenas determinan todas las



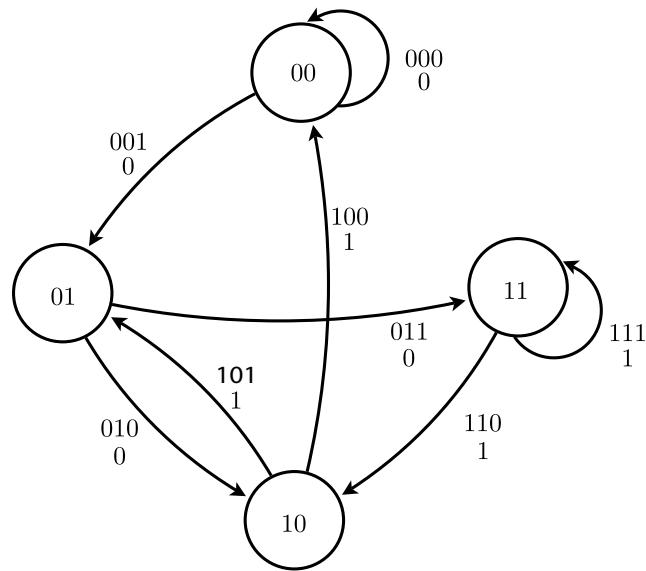


Figura 3.8: Diagrama de de Bruijn para la regla 240, de AC 1D (2,1).

configuraciones estáticas.

La ventaja de utilizar los diagramas de de Bruijn es que algunos de los problemas concernientes a los AC son transformados en problemas conocidos de seguimiento de patrones por medio de un grafo. Los grafos sin ciclos pueden ser más largos que el número total de nodos en el grafo sin repetir ningún segmento; sin embargo, deben existir otro tipo de ciclos, en el cual el segmento repetido es atravesado un número arbitrario de veces. Por ejemplo, la figura 3.8 muestra que existen una serie de ciclos que se forman al seguir la trayectoria del grafo. Si se toma el nodo 00 se forma la cadena 000 al unirse con el nodo 00, con el cual nuevamente se puede unir un número  $l$  de veces formando así cadenas del estilo 000000. Otro ejemplo, si se toma el nodo 00 y se une al 01 y después al 10 y finalmente al 00 se forma la cadena 00100, la cual se puede repetir  $l$  veces.

Los diagramas de de Bruijn dan información que se puede clasificar en dos: el análisis del comportamiento global del autómata, el cual se refiere a la formación de patrones con un comportamiento específico, tal como permanencia o desplazamiento; el análisis sistemático que sirve para obtener información del autómata celular, tales como Ancestros, Jardines del Eden y el cálculo de períodos.

### Análisis del comportamiento global

El análisis que se realiza sobre el comportamiento del AC está determinado por el estado de una célula de la vecindad y el estado de la célula central al ser evaluada y puede leerse de la regla de evolución; es decir, la vecindad en la cual la célula central no cambia se refiere a aquellas en las que el valor de la célula central en un tiempo

$t + 1$  es igual a su valor en el tiempo  $t$ . La complejidad de encontrar secuencias en las cuales no cambia el valor del estado de la célula central radica en la intersección de más de dos células sucesivas. Afortunadamente, ya existe una técnica de diagramación para tratar las secuencias en donde sus símbolos se traslapan, utilizada en la teoría de corrimientos [Golomb(1967)].

La tabla 3.1 muestra algunas de las características en las que se puede clasificar la información que se obtiene del diagrama de de Bruijn. Cada transición se clasifica de acuerdo al resultado de la evaluación de la función de transición de la vecindad formada de la intersección. En la tabla se desglozan seis diferentes características:

- La célula central conserva su valor original al ser evaluada la vecindad.
- La célula central al ser evaluada tiene el valor de su vecino derecho.
- La célula central al ser evaluada tiene el valor de su vecino izquierdo.
- La célula central al ser evaluada tiene el valor de cero.
- La célula central al ser evaluada tiene el valor de uno.
- La célula central al ser evaluada tiene el valor complemento,

Vecindad	000	001	010	011	100	101	110	111
Célula evolucionada	0	1	1	0	1	0	0	0
Igual	x	.	x	.	.	x	.	.
Izquierda	x	.	.	x	x	.	.	.
Derecha	x	x	.	.	.	.	x	.
Cero	x	.	.	x	.	x	x	.
Uno	.	x	x	.	x	.	.	.
Complemento	.	x	.	x	x	.	x	x

Tabla 3.1: Clasificación de la evolución

### Análisis de los ciclos en el espacio

La restricción de la periodicidad en el tiempo de un AC produce una clasificación de periodos de tiempo, dentro de los cuales la evolución debe caer eventualmente. El problema consiste en seleccionar un intervalo de tiempo, con la intención de enumerar las longitudes de todas los posibles ciclos, cuyas evoluciones se repiten después de un tiempo dado.

El curso de la evolución de cualquier autómata finito es tratado a través de un diagrama construido para este proposito o tal como una representación equivalente, como la matriz de conectividad del diagrama, donde las características a estudiar son:

- Los ciclos en los cuales todas las evoluciones terminan.
- Nodos con no ancestros, llamados Jardines del Eden.
- La longitud más larga de camino.
- El promedio y la desviación estándar de las longitudes de camino.
- La cantidad y tipo de ramificaciones del diagrama.

El análisis de los autómatas celulares se inicia al tomar las ligas para la formación de cadenas, las cuales pueden ser cadenas que cambian de una generación a otra. Una de las propiedades es cuando las cadenas tienen un bucle de longitud  $l$ , las cuales podrían de terminar la longitud de un *ciclo* o *período* de un AC.

**Definición 40** *Se define un ciclo o periodo de un AC al número de evoluciones que requiere una configuración para repetirse.*

Como consecuencia, los ciclos tienen que extraerse de fragmentos grandes del diagramas de de Bruijn, con la finalidad de obtener los ciclos de los diagramas. Otra combinación que se da es cuando un ciclo es conectado con una serie de ligas con otros ciclos, pero que no tienen una ruta de regreso. Así algunos patrones podrían ocupar el lado derecho de una cadena infinita, entonces una transición dentro de un patrón podría ocupar el lado izquierdo de la cadena. Dichas cadenas existen tanto en 1D como en 2D y son llamadas *fuses* [McIntosh(2009)]. Para obtener todos los patrones de un determinado periodo se pueden utilizar algoritmos de búsquedas de caminos, derivados de la teoría de grafos. Las células, a excepción de la célula central pueden repetirse, lo cual es tan sencillo como buscar patrones de corrimientos como patrones periódicos.



# Capítulo 4

## Diagramas de de Bruijn en 2D

En el presente capítulo, la primera sección se explicará la definición de un AC bidimensional (AC 2D), las definiciones de las vecindades principales en 2D (von Neumann, Moore y Hexagonal) con los cuales se plantea la posibilidad de que las reglas que utilicen esas vecindades puedan ser estudiadas de manera sistemática por medio de los diagramas de de Bruijn y la definición de los elementos del diagrama de de Bruijn adaptados para dos dimensiones. En la segunda sección se explicará el método que se ha propuesto para la creación de los diagramas de de Bruijn en dos dimensiones; así como la aplicación a cada una de las vecindades descritas anteriormente.

### 4.1. Definición de AC en 2 dimensiones

Se han realizado algunos trabajos iniciales con los diagramas de de Bruijn en 2D, en los 90's McIntosh planteó una forma de estudiar el Juego de la Vida por medio de los diagramas de de Bruijn [McIntosh(2010a)]. En los 2000 Eppstein utilizó los diagramas de de Bruijn para la búsqueda de patrones específicos en el Juego de la Vida, llamados *Space Ships* [Eppstein(2002)] y en 2008 Basurto y León iniciaron el estudio analítico de la regla Espiral por medio de los diagramas de de Bruijn [Basurto and León(2011)].

Se iniciará definiendo lo que es un AC 2D, donde tomando la definición formal, tenemos que la quintupla  $(\Sigma, d, \rho, N, f)$  para un AC de 2 estados y 2 dimensiones, tendrá los siguientes valores:

- $\Sigma : \{0, 1\}$ ,
- $d = 2$ ,
- $\rho_i$ : dependerá de la forma de la vecindad,
- $N$ : es una configuración inicial,
- $f_i(t + 1)$ : dependerá del valor de  $\rho$ .

A diferencia de los AC 1D donde se puede definir que existe un cadena  $C$  que representa la configuración de un arreglo de células en un tiempo  $t$  (ver definición 36), en un AC 2D se tiene un conjunto de elementos ordenados, llamado *rejilla o lattice*, la cual representará la configuración de un AC en un tiempo  $t$ . Además, las vecindades de AC 1D (definición 37) difieren de las vecindades de 2D como consecuencia de que éstas últimas dependerán totalmente de las células que se deseen incluir en la vecindad, es decir, una vecindad en 1D las células que se incluían eran exclusivamente las que se encontraban  $r$  posiciones a la izquierda y  $r$  posiciones a la derecha; sin embargo en 2D dependiendo de la forma de la rejilla (cuadrada, triangular, hexagonal), pueden tomarse  $r$  vecinos de arriba, abajo, izquierda, derecha, diagonales, por mencionar algunos. Por lo tanto, se puede definir una vecindad como:

**Definición 41** Una vecindad de un AC de dimensión mayor a 1, se define como el conjunto de células ordenadas alrededor de una célula central en una rejilla de dimensión  $d$ .

Cada célula de la vecindad tendrá un valor  $c$ ; la función de transición local  $f$  se aplicará en un tiempo  $t$  de donde se necesita conocer el valor de los estados  $c$  de cada una de las células de la vecindad  $\rho$ .

Para ejemplificar mejor las definiciones anteriores, se definirán tres vecindades AC 2D, la vecindad de von Neumann, la vecindad de Moore y la vecindad hexagonal, las cuales utilizaremos a lo largo de esta tesis y se pueden ver en la figura 4.1.

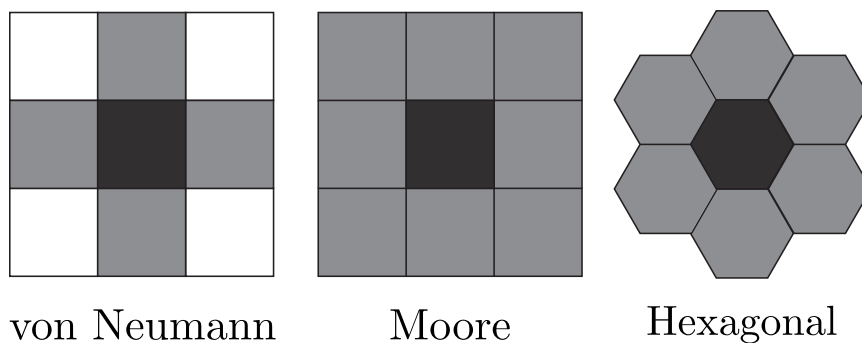


Figura 4.1: Vecindades en dos dimensiones

### Vecindad de von Neumann

La vecindad de von Neumann se define de la siguiente manera, siendo la célula  $c_{i,j}$  la célula central o la célula a evaluar.

$$\rho_{vN} = \{c_{i+1,j}, c_{i,j+1}, c_{i,j}, c_{i-1,j}, c_{i,j-1}\} \quad (4.1)$$

Por lo tanto, la función de transición  $f_i(t+1)$  se define como:

$$c_{i,j}^{t+1} = f(c_{i+1,j}, c_{i,j+1}, c_{i,j}, c_{i-1,j}, c_{i,j-1})$$

### Vecindad de Moore

La vecindad de Moore se define de la siguiente manera, siendo la célula  $c_{i,j}$  la célula central o la célula a evaluar.

$$\rho_M = \{c_{i+1,j}, c_{i+1,j-1}, c_{i+1,j+1}, c_{i,j-1}, c_{i,j+1}, c_{i,j}, c_{i-1,j-1}, c_{i-1,j+1}, c_{i-1,j}\} \quad (4.2)$$

Por lo tanto, la función de transición  $f_i(t+1)$  se define como:

$$c_{i,j}^{t+1} = f(c_{i+1,j}, c_{i+1,j-1}, c_{i+1,j+1}, c_{i,j-1}, c_{i,j+1}, c_{i,j}, c_{i-1,j-1}, c_{i-1,j+1}, c_{i-1,j})$$

### Vecindad hexagonal

La vecindad hexagonal se define de la siguiente manera siendo la célula  $c_{i,j}$  la célula central o la célula a evaluar.

$$\rho_H = \{c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j}, c_{i+1,j+1}\} \quad (4.3)$$

Por lo tanto, la función de transición  $f_i(t+1)$  se define como:

$$c_{i,j}^{t+1} = f(c_{i-1,j}, c_{i-1,j+1}, c_{i,j-1}, c_{i,j}, c_{i,j+1}, c_{i+1,j}, c_{i+1,j+1}) \quad (4.4)$$

A diferencia de los diagramas de de Bruijn para AC 1D, los diagramas de de Bruijn para AC 2D tienen una serie de restricciones que se deben considerar. La definición de cada uno de los elementos de los diagramas de de Bruijn, se describen en la presente sección para el caso específico de AC 2D.

El primer elemento importante es el conjunto de nodos, los cuales tienen a la célula central como uno de sus elementos, los cuales se definen de la siguiente forma:

**Definición 42** *Una vecindad parcial se define como un conjunto de células ordenadas en una lattice de tamaño  $n \times m$ , donde uno de sus elementos es la célula central.*

El segundo elemento de un diagrama de de Bruijn es la arista. Las aristas están determinadas por la función que relaciona a dos nodos.

**Definición 43** *Las aristas del diagrama de de Bruijn se definen como la función de intersección entre dos vecindades parciales, donde al menos uno de los elementos de cada vecindad parcial se traslapan con la finalidad de formar una vecindad completa.*

El tercer elemento, que jugará un papel importante para el análisis de los AC 2D vía diagramas de de Bruijn es la etiqueta de la arista.

**Definición 44** *Las etiquetas representan el valor de la célula central en un tiempo  $t+1$ . Este valor depende de la evaluación de la función  $f(\rho)$  de una regla de AC 2D.*

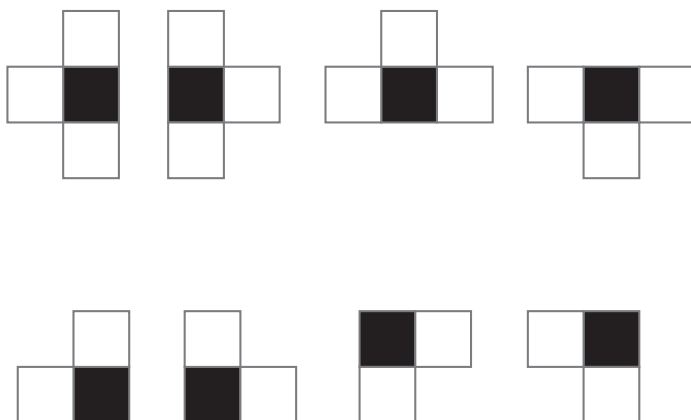


Figura 4.2: Posibles vecindades parciales de von Neumann

### Vecindad de von Neumann

Las vecindades parciales o posibles nodos de la vecindad de von Neumann que cumplen con la condición de tener a la célula central, se muestran en la figura 4.2.

Al analizar cada una de las posibles vecindades parciales, se concluye que no se cumple con la condición de ser una lattice de  $m \times n$ . Por tal motivo, el análisis para la vecindad de von Neumann con las características de la definición 42 no es posible realizarlo.

### Vecindad de Moore

Después de analizar las posibles vecindades parciales de la vecindad de Moore, se encontró que existen dos que se han apegado a la definición 42. La figura 4.3(a) muestra una lattice de  $2 \times 3$ , la célula parcial es la etiquetada en la posición  $d$ .

La relación entre dos nodos se establece cuando las células  $b, d, f$  del primer nodo se traslapan con las células  $a, c, e$  del segundo nodo. La condición para que este traslape se lleve a cabo es que el valor del estado de las células del primer nodo sea igual al valor de estado de las células del segundo nodo. En la figura 4.3(a) las células en color negro son las células que se traslapan para cumplir con la definición 43.

En el caso de la figura 4.3(b), se tiene una lattice de  $3 \times 2$ ; las células de traslape del primer nodo son  $d, e, f$  y las células de traslape del segundo nodo son  $a, b, c$  y la célula central es  $e$ .

### Vecindad hexagonal

La vecindad parcial de la vecindad hexagonal se ve en la figura 4.3. Es una lattice hexagonal de  $2 \times 2$  células. La relación que cumple con la definición 43 es cuando la célula  $c$  del primer nodo se traslapa con la célula  $b$  del segundo nodo.



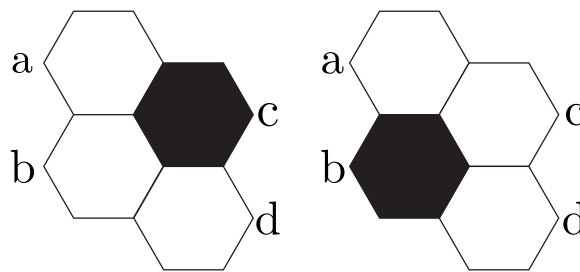
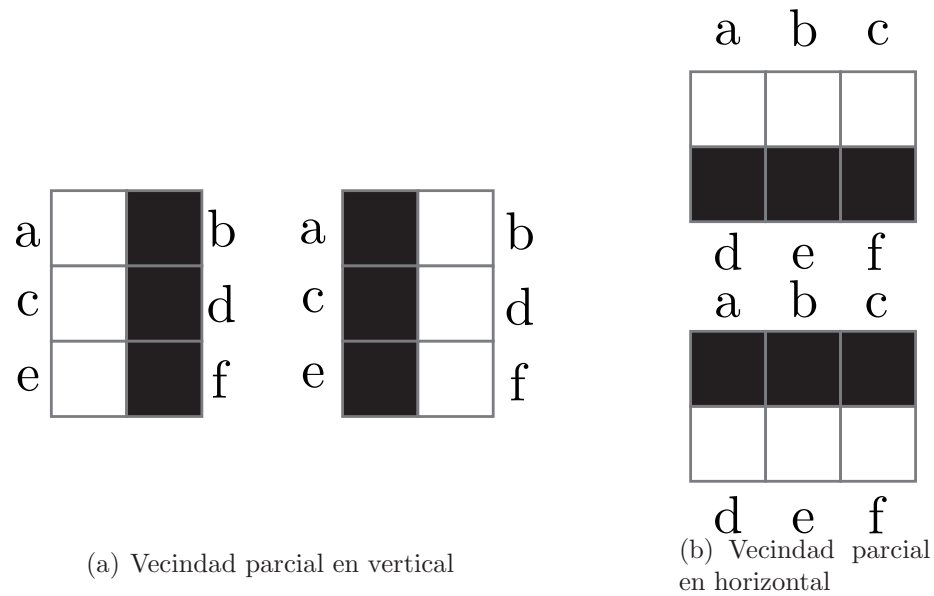


Figura 4.3: Vecindades parciales de la vecindad hexagonal

## 4.2. Extensión de vecindades parciales

En la sección anterior se explicaba las características de los elementos del diagrama de de Bruijn para evaluar una sólo célula. En la presente sección se describen las particularidades para evaluar más de una célula o evaluar a un tiempo  $t > 1$ .

Para los AC 1D bastaba con extender  $r$  células a la izquierda y  $r$  células a la derecha. Sin embargo, para los AC 2D se debe considerar que es posible evaluar a más de una célula en una o ambas dimensiones. Para poder realizar la extensión de la vecindad se deben considerar dos factores:

- El conjunto de células deben ser las necesarias para evaluar un conjunto  $E$  de células, las cuales deben existir en ambos nodos.
- La vecindad completa debe dividirse en dos vecindades parciales y tener un conjunto  $T$  de células que serán las que se intersecten y  $E \in T$ .

### Vecindad Moore

La extensión horizontal para la vecindad de Moore se da cuando se desea evaluar más de una célula y se forma una vecindad, como la que se muestra en la figura 4.4. El tamaño de la vecindad parcial es:

$$3 \times (e + 1) \quad (4.5)$$

Donde  $e$  es el número de células a evaluar.

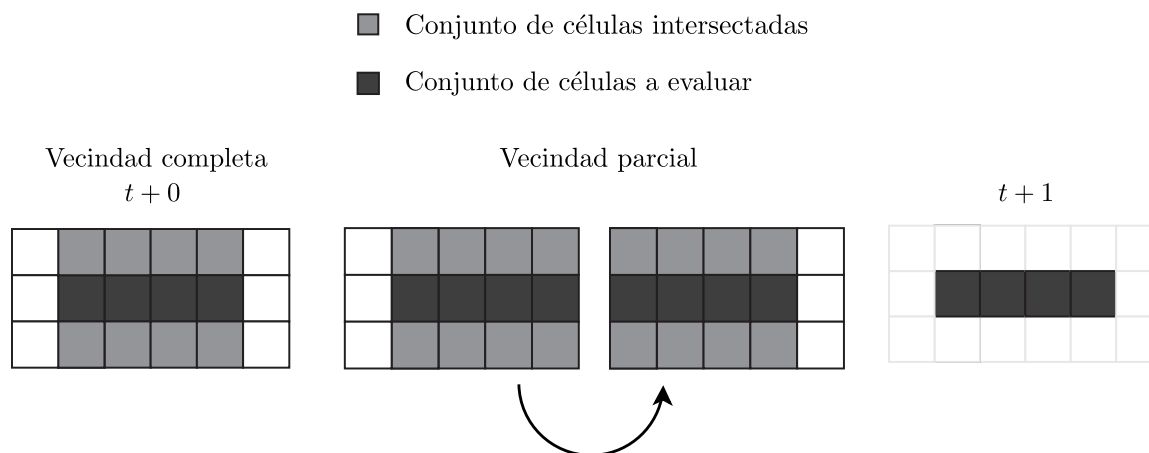


Figura 4.4: Vecindad de Moore extendida en horizontal

La extensión vertical para la vecindad de Moore se da cuando se desea evaluar más de una célula y se forman vecindades como la que se muestra en la figura 4.5. De esta manera el tamaño de la vecindad parcial es:

$$(e + 2) \times 3 \quad (4.6)$$

Donde  $e$  es el número de células a evaluar.

La extensión vertical para la vecindad de Moore se da cuando se desea evaluar más de una célula y se forman vecindades como la que se muestra en la figura 4.6. De esta manera el tamaño de la vecindad parcial es:

$$(e + 2) \times (e + 1) \quad (4.7)$$

Donde  $e$  es el número de células que a evaluar.

### Vecindad Hexagonal

Las vecindades parciales de la vecindad hexagonal, a diferencia de las vecindades parciales de Moore, requiere de una sólo célula para intersectarse; la vecindad hexagonal no tiene una simetría en dos dimensiones al formarse lattices más grandes, debido a la forma natural que se adquiere al unir hexagonos, por lo cual la extensión de esta

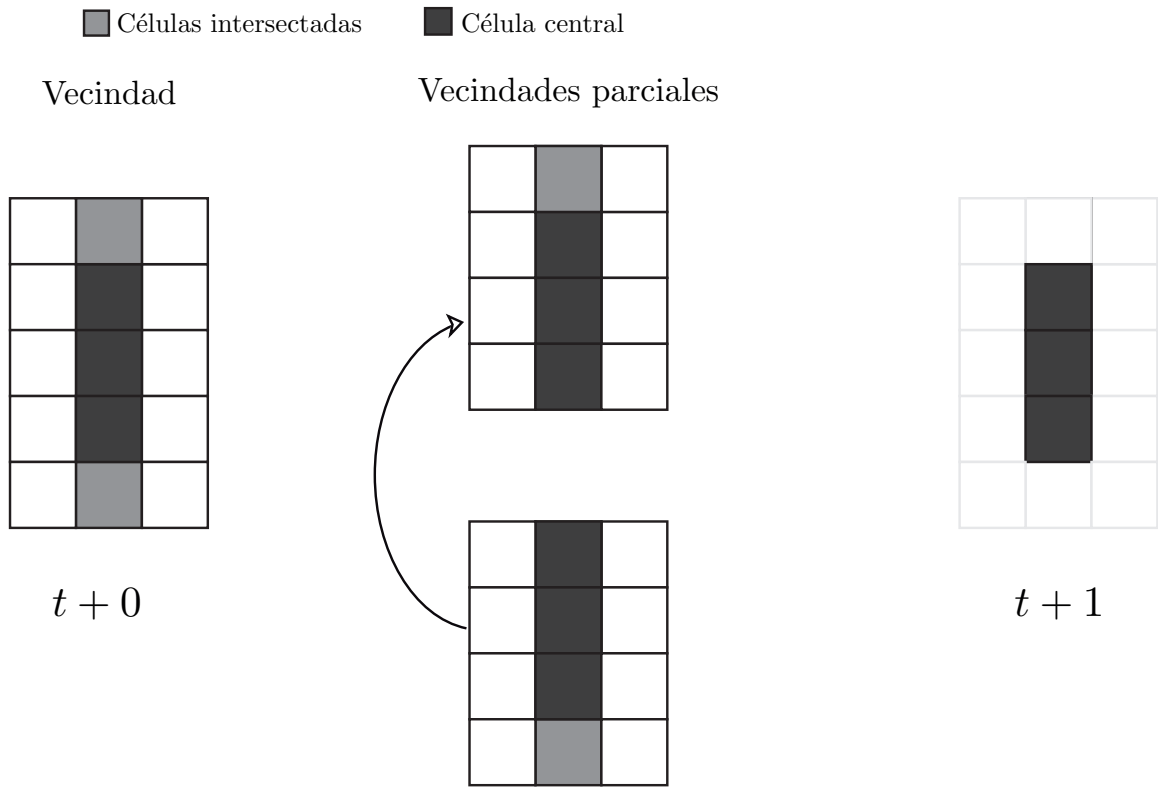


Figura 4.5: Vecindad de Moore extendida en vertical

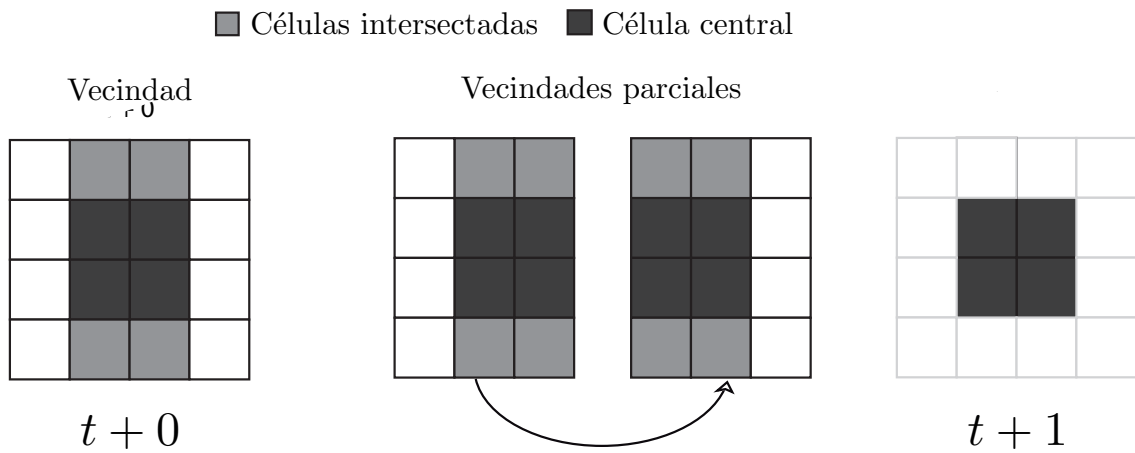


Figura 4.6: Vecindad de Moore extendida en dos dimensiones

vecindad tiene un tratamiento ligeramente distinto al de la vecindad de Moore.

La evaluación de más de una célula se da de dos formas: en forma vertical y en las dos dimensiones; a diferencia de la vecindad de Moore, no se pueden realizar extensiones para la vecindad hexagonal de forma horizontal, lo cual se debe a la forma que adquiere la rejilla y que no permite la simetría entre las vecindades parciales.

La extensión vertical para la vecindad hexagonal se da cuando se desea evaluar más de una célula y se forman vecindades, como las que se muestra en la figura 4.7. De esta manera el tamaño de la vecindad parcial es:

$$2 \times (e + 1) \quad (4.8)$$

Donde  $e$  es el número de células que a evaluar.

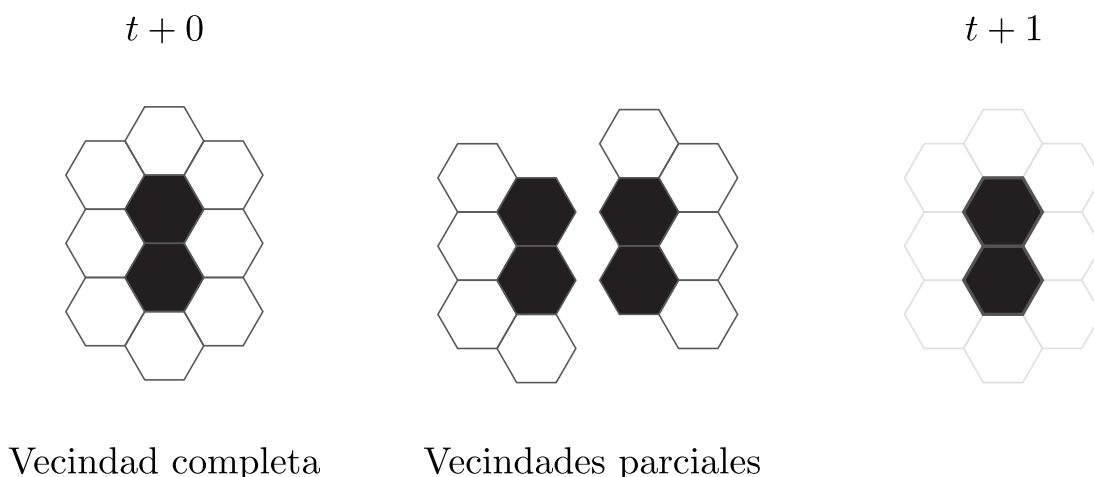


Figura 4.7: Vecindad hexagonal extendida en dos dimensiones verticalmente

En el caso de la extensión de dos dimensiones, se da cuando se desea evaluar más de una célula tanto en vertical, como en horizontal, como se muestra en la figura 4.8. De esta manera, el tamaño de la vecindad parcial es:

$$(e + 1) \times (e + 1) \quad (4.9)$$

### 4.3. Análisis de los AC 2D vía diagrama de de Bruijn

En la presente sección se explicará la metodología propuesta para construir los diagramas de de Bruijn para AC 2D. Para iniciar se debe recordar que:

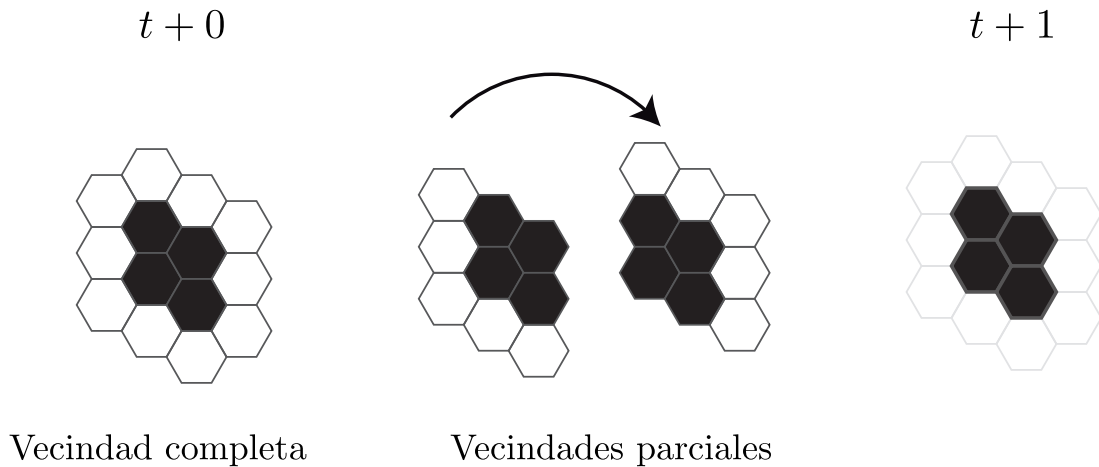


Figura 4.8: Vecindad hexagonal extendida en dos dimensiones

**Definición 45** *Un diagrama de de Bruijn es un grafo dirigido definido como  $G(V, A)$  etiquetado, donde  $V$  es el conjunto de nodos y  $A$  es el conjunto de aristas y las etiquetas de estas son los estados de las células en un tiempo  $t + 1$  de las células a evaluar de la vecindad formada.*

Para formar los diagramas de de Bruijn es necesario definir cada uno de sus componentes, así se define lo siguiente:

**Definición 46** *El conjunto de nodos  $V$  del diagrama de de Bruijn se define como el conjunto de todas las combinaciones de los estados que puede tener todas las células de la vecindad parcial.*

Por ejemplo, el conjunto de nodos de una vecindad parcial de  $2 \times 3$  células y 2 estados, se muestra en la figura 4.9. Para fines prácticos, se ha etiquetado a cada elemento del conjunto con  $n$  número binario asociado.

Una vez que se tiene el conjunto de nodos, se calcula el conjunto de relaciones o aristas  $A$ , las cuales tienen la forma  $a = (x, y)$ , donde  $x$  y  $y$  son elementos de  $V$ . Cada elemento de  $A$  debe cumplir con la definición 43. Por ejemplo, la figura 4.10 muestra dos vecindades parciales de una vecindad de Moore, cada una de las vecindades se une por medio de las células en color gris para formar una vecindad completa, cumpliendo de esta manera con la definición 43, con la cual se conoce el valor del estado de la célula en color negro en un tiempo  $t + 1$ . La figura 4.11 ejemplifica la unión de dos vecindades parciales, para formar una vecindad completa.

Los métodos para poder ver las relaciones son variados, van desde el método gráfico hasta la representación matricial donde las relaciones entre los nodos se representa por la matriz de incidencias del diagrama. Cada fila representa el primer nodo y las

columnas el segundo nodo; dicha matriz es cuadrada, es decir tiene el mismo número de columnas y de filas y la cual no es una matriz simétrica.

La matriz que representa al conjunto  $A$  del ejemplo de un conjunto  $V$  de vecindades parciales de  $2 \times 3$  y 2 estados se observa en la tabla 4.1, en la cual, el símbolo “-” representa la ausencia de relación, los símbolos 0 y 1, para este caso, representan el valor que se tiene en un tiempo  $t + 1$  la célula evaluada, como resultado de la vecindad que se forma al intersectar las dos vecindades parciales, correspondiendo las filas al nodo1 y las columnas al nodo2.

El análisis de los AC a través de la búsqueda de patrones de comportamiento se realiza por medio de filtros: la permanencia y los corrimientos, gracias a los cuales es posible encontrar *still lifes* y *gliders*, estructuras o patrones que emergen de la interacción entre células, como parte del comportamiento del AC.

Dadas las características de las vecindades de tamaño  $n \times m$  de células cuadradas existen 4 comportamientos que presenta la célula o células evaluadas, las cuales se observan en la figura 4.12 y se describen de la siguiente manera:

- Permanencia: Se presenta cuando la célula o células evaluadas tienen el mismo valor de estado que al ser evaluada en un tiempo  $t + 1$ .
- Corrimiento de izquierda a derecha: Se presenta cuando la célula o células evaluadas tienen el mismo valor que el estado de su célula vecina a la izquierda al ser evaluadas en un tiempo  $t + 1$ .
- Corrimiento de abajo a arriba: Se presenta cuando la célula o células evaluadas tienen el mismo valor que el estado de la célula vecina de abajo al ser evaluadas en un tiempo  $t + 1$ .
- Corrimiento diagonal: Se presenta cuando la célula o células evaluadas tienen el mismo valor que el estado de la célula vecina en diagonal al ser evaluadas en un tiempo  $t + 1$ .

Con lo anterior, la construcción de patrones más grandes, se da por medio de la búsqueda de caminos o rutas inherentes al grafo. Tal y como se realiza en los AC 1D las rutas pueden determinar construcciones más grandes con una periodicidad. Debido a que el número de nodos es muy grande, la matriz booleana es útil para determinar las propiedades que se buscan. La figura 4.13 muestra una parte del diagrama de de Bruijn para un AC de 2 estados, al cual se le ha realizado todo el procesos descrito en esta sección y se utilizó el filtro de corrimiento en diagonal, con el cual, se formo una serie de rutas y ciclos. En la parte superior de la imagen se observa una serie de patrones que se forman al seguir las rutas del diagrama. De dichas construcciones es posible conocer como será su comportamiento de la célula central en el tiempo  $t + 1$ , que se representa por la cadena que se encuentra abajo de cada estructura.

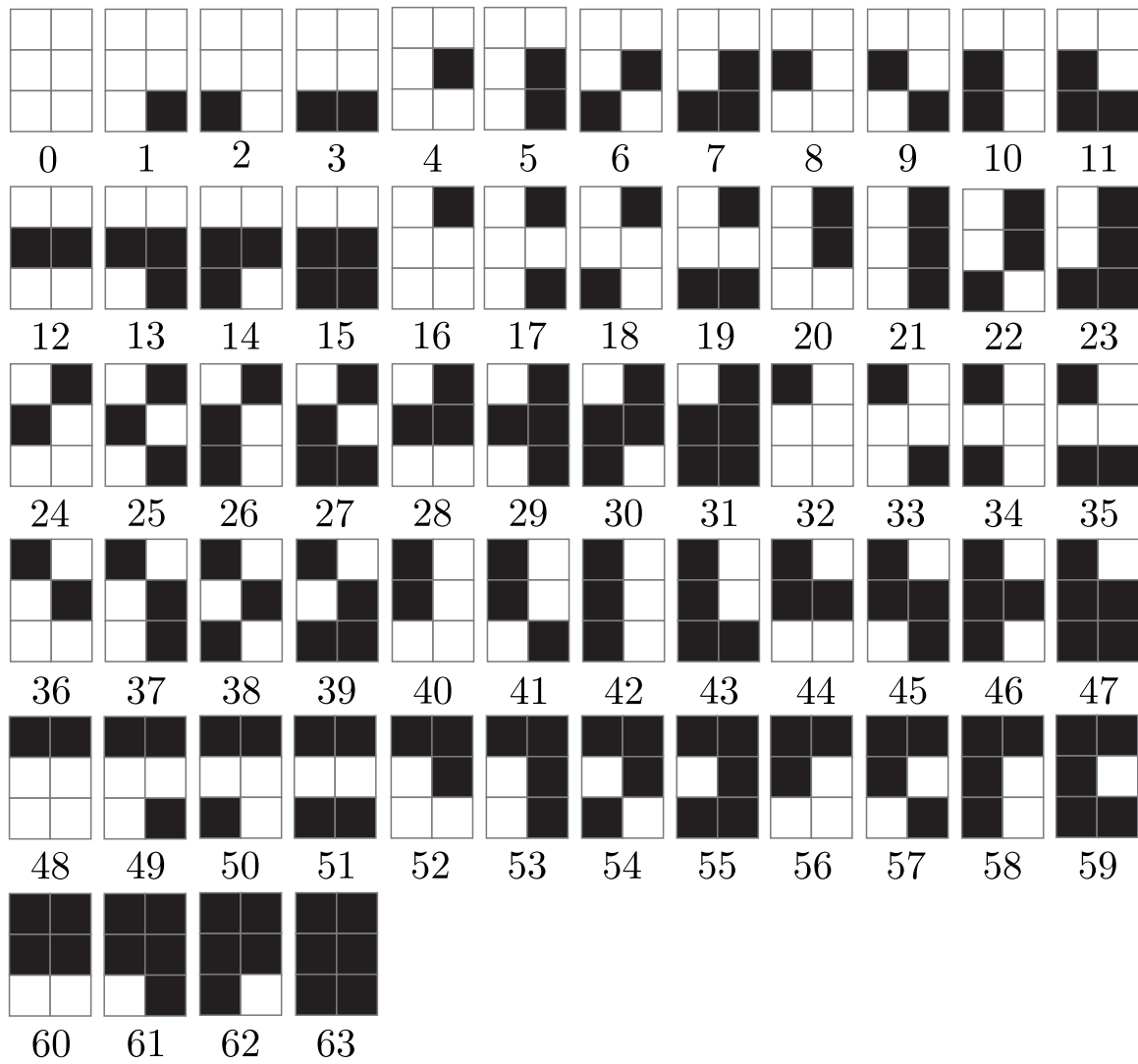


Figura 4.9: Conjunto  $V$  para una vecindad parcial de  $2 \times 3$  células y 2 estados.

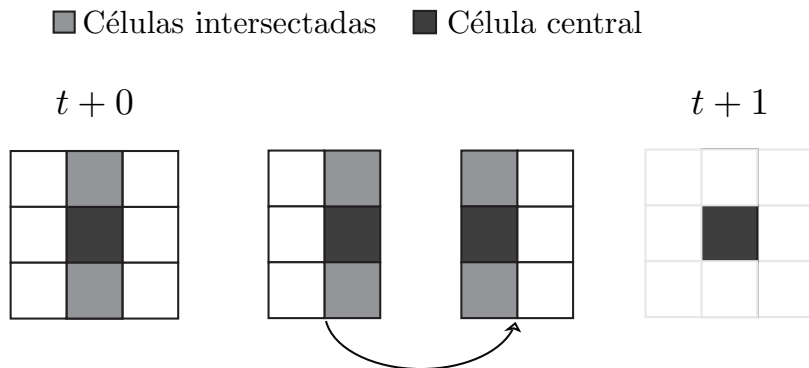


Figura 4.10: Relaciones entre dos nodos

$$\begin{array}{|c|c|c|} \hline \blacksquare & \square & \square \\ \hline \square & \blacksquare & \square \\ \hline \blacksquare & \square & \square \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline \square & \square & \square \\ \hline \blacksquare & \blacksquare & \square \\ \hline \square & \square & \square \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline \blacksquare & \square & \square \\ \hline \square & \blacksquare & \blacksquare \\ \hline \blacksquare & \square & \square \\ \hline \end{array}$$

38
12
38 - 12

Figura 4.11: Ejemplo de relaciones entre dos nodos

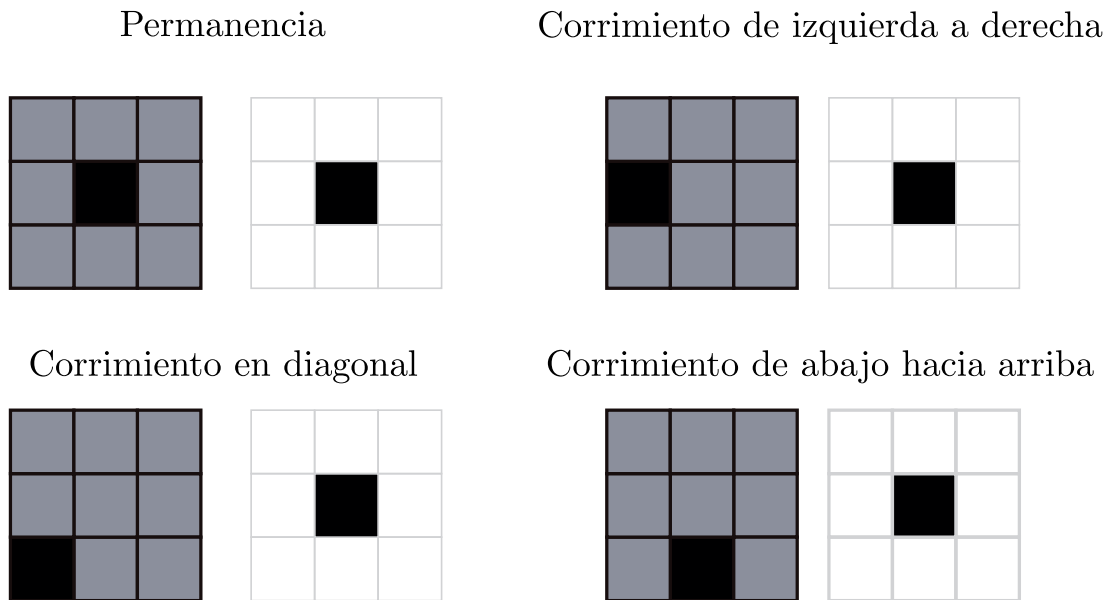


Figura 4.12: Tipos de corrimientos



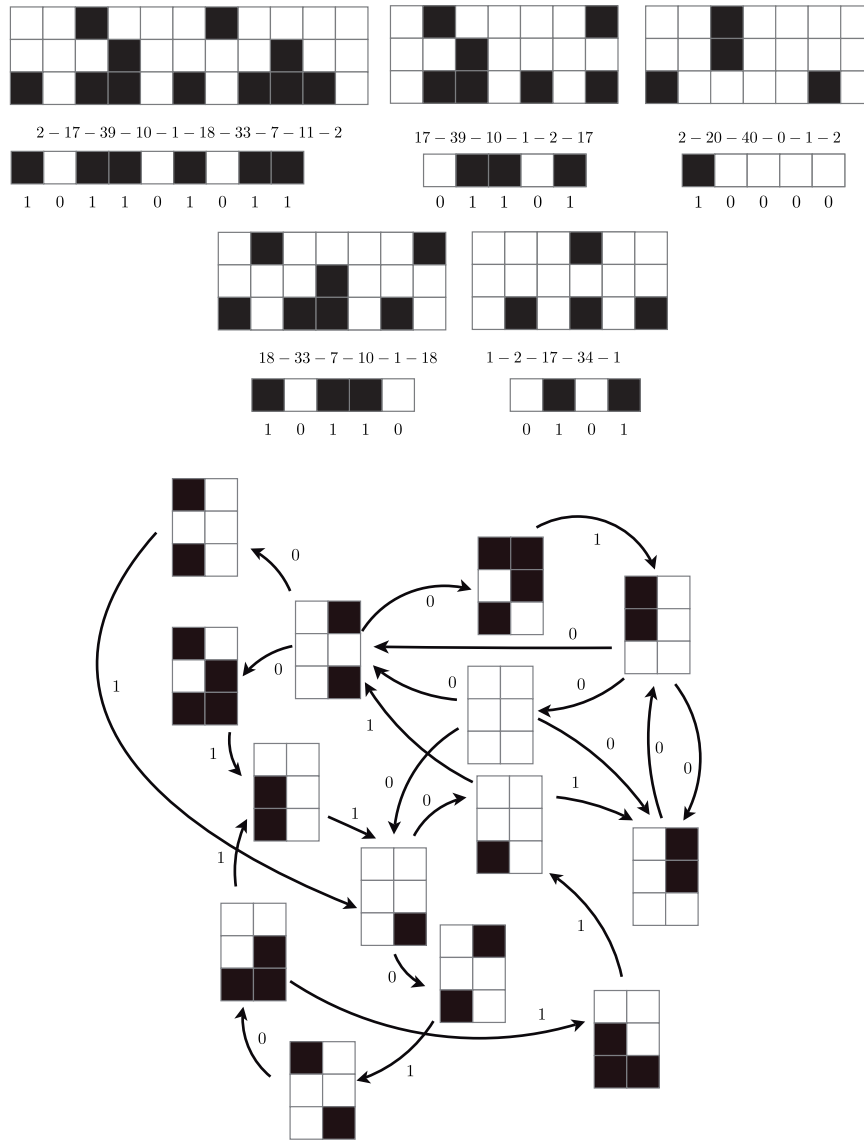


Figura 4.13: Ejemplo de diagrama de de Bruijn



# Capítulo 5

## Experimentación

Una vez hechas las definiciones para los AC 2D y desarrollada la metodología para realizar el análisis vía diagramas de de Bruijn de estos autómatas, se realizó un sistema de prueba. El objetivo de estos experimentos es verificar el uso de intersecciones para la construcción de los diagramas de de Bruijn y presentar los casos de estudio del Juego de la Vida y Regla de Difusión.

### 5.1. Análisis

La experimentación y comprobación del método descrito en el capítulo anterior, está basado en una vecindad de Moore y sus respectivas extensiones. En esta sección se mostrará el análisis realizado para determinar los casos de estudio.

La tabla 5.1 muestra el tamaño de la vecindad parcial para evaluar células a un tiempo  $t$ . Este estimado se realizó en vecindades parciales del tipo de la figura 4.3(b). En la misma tabla, se puede observar que existe una relación en el crecimiento de las vecindades parciales, y es, que por cada tiempo se suman dos células en cada dimensión.

Tiempo	Tamaño de la vecindad
1	$2 \times 3$
2	$4 \times 5$
3	$6 \times 7$
4	$7 \times 9$

Tabla 5.1: Evaluación de células en  $t$  tiempo vs tamaño vecindad parcial

De esta manera, se necesitan atacar dos problemas, primero la manipulación y generación de todas las combinaciones y la segunda, la manipulación de matrices que se generaran, así como la administración de los recursos computacionales.

## 5.2. Algoritmos utilizados

En esta sección se mostrarán los dos algoritmos principales para el desarrollo del método de los diagramas de de Bruijn. El primero calcula las relaciones que existen entre dos nodos, el cual se acopla a la definición 43. El segundo, es la evaluación de las células para un tiempo  $t + 1$ . El primer algoritmo es importante por la relación con las definiciones propuestas en la sección 4.1 y segundo algoritmo es importante para determinar el comportamiento global de la región formada por la relación entre dos nodos.

Las combinaciones resultantes de los nodos crecen de forma exponencial como se mostró en los capítulos anteriores, de esta manera, la cantidad de recursos computacionales, tanto memoria, como de procesamiento, también incrementaba exponencialmente. Debido a que se requería una cantidad muy grande de nodos, se propuso el uso de números binarios y la realización de operaciones a nivel bit; lo cual significa que se utilizaron cadenas binarias para formar vecindades de dos dimensiones.

El primer algoritmo es el cálculo de la relación entre vecindades que se intersectan. Este se describe en el algoritmo 1. El procesamiento de este algoritmo es lineal y se relaciona directamente con el número de nodos que tiene el conjunto  $N$ . Los parámetros que recibe son básicamente el número de células que tiene a lo ancho (columnas) y el número de células que tiene a lo alto (filas).

---

**Algoritmo 1** Algoritmo de intersección de vecindades parciales

---

- 1: Se selecciona el nodo1 del conjunto  $N$
  - 2: Se seleccionan los bits (células) que son de interés para realizar la intersección por medio de una mascara.
  - 3: Por cada elemento del conjunto  $N$  se realiza lo siguiente
  - 4: Se selecciona el nodo2 del conjunto  $N$
  - 5: Se seleccionan los bits (células) que son de interés para realizar la intersección por medio de una mascara.
  - 6: Se calcula la operación AND entre el nodo1 y el nodo2, Si el resultado es 1, existe la intersección y se almacenan como elementos del conjunto  $A$  (aristas).
  - 7: Continuar con el siguiente elemento de  $N$ .
- 

El siguiente es el cálculo de la evaluación de la función de transición de la regla de AC. En este punto, cabe mencionar que se debe considerar que se está utilizando una cadena de bits, por lo que es importante plantear correctamente cual es la distribución de los bits en dos dimensiones, de lo contrario la evaluación será errónea. Para lograr dicho objetivo se utilizan variables como el número de células a lo ancho (columnas) y el número de células a lo alto (filas). El algoritmo 2 describe como se evalúa la función de transición.

---

**Algoritmo 2** Algoritmo de evaluación de las vecindades

---

- 1: Por cada bit que corresponde a una célula a evaluar, realizar lo siguiente
  - 2: La variable de *evaluacion* dar el valor 1
  - 3: Seleccionar el bit que corresponde a la célula a evaluar
  - 4: Seleccionar por medio de una mascara el bit que corresponde al vecino norte utilizando el nodo1.
  - 5: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 6: Seleccionar por medio de una mascara el bit que corresponde al vecino noreste utilizando el nodo1.
  - 7: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 8: Seleccionar por medio de una mascara el bit que corresponde al vecino noroeste utilizando el nodo1.
  - 9: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 10: Seleccionar por medio de una mascara el bit que corresponde al vecino este utilizando el nodo1.
  - 11: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 12: Seleccionar por medio de una mascara el bit que corresponde al vecino oeste utilizando el nodo1.
  - 13: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 14: Seleccionar por medio de una mascara el bit que corresponde al vecino sur utilizando el nodo2.
  - 15: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 16: Seleccionar por medio de una mascara el bit que corresponde al vecino sureste utilizando el nodo2.
  - 17: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 18: Seleccionar por medio de una mascara el bit que corresponde al vecino suroeste utilizando el nodo2.
  - 19: Si el valor del bit es 1, hacer un corrimiento a la izquierda a la variable *evaluacion*
  - 20: Evaluar conforme a la regla deseada.
-

## 5.3. Casos de estudio y resultados

### 5.3.1. Regla de comportamiento complejo: El Juego de la Vida

El estudio de la regla del Juego de la Vida ha sido muy variado, esto debido a la complejidad que presenta con un conjunto de reglas sencillas. El Juego de la Vida es un AC de dos estados, en una vecindad de Moore, es por esta razón por la cual lo utilizamos para probar lo propuesto en la presente tesis.

La regla del Juego de la Vida son las siguientes:

- El Juego de la vida tiene dos estados. Una célula viva se representa con el color negro, y una célula muerta se representa con el color blanco.
- Si una célula viva tiene 2 o 3 células vivas como vecinos, entonces permanecerá viva, en otro caso “morirá”.
- Si no está viva y tiene exactamente 3 células vivas, la siguiente evolución “nacerá”.

De esta manera, se utilizaron espacios para vecindades parciales, con los cuales se pretendía encontrar configuraciones conocidas de la regla de AC.

El primer experimento realizado fue a la vecindad de Moore, con la cual las vecindades parciales eran de  $3 \times 2$ , de lo cual se obtuvieron los diagramas de de Bruijn. Para fines prácticos, se tomó sólo una parte de los diagramas de de Bruijn, y se clasificaron de acuerdo a su comportamiento.

La figura 5.1 muestra el diagrama cuyo comportamiento denota permanencia. Es decir, la célula central al ser evaluada mantendrá su valor original. Este comportamiento ayuda a la búsqueda de estructuras que permanecen, en la literatura de los AC se les llama *still lifes*.

La figura 5.2 muestra el diagrama cuyo comportamiento muestra un desplazamiento en diagonal, es decir, que la célula central al ser evaluada tiene el valor de su vecino inferior izquierdo. Este comportamiento ayuda a la búsqueda de estructuras que se desplazan en forma diagonal, característica que tienen los *gliders* en el Juego de la Vida.

### 5.3.2. Regla de comportamiento caótico: Regla de Difusión

La regla de Difusión es una regla que nace como parte de la familia de los AC que simulan el fenómeno de reacción difusión. El estudio de esta regla es interesante debido a que pese a que se encuentra clasificada como una regla de comportamiento caótico, el comportamiento local, es decir en menor escala, muestra un comportamiento interesante, en donde se encuentra una riqueza inexplorada de patrones.

El estudio de la familia de AC 2D de dos estados, con una vecindad de Moore, con una célula central. La regla de transición es la siguiente:

$$x^{t+1} = \begin{cases} 1, & \text{si } x^t = 0 \text{ y} \\ 0 & \text{en otro caso.} \end{cases} \quad (5.1)$$

El primer experimento realizado fue a la vecindad de Moore, con la cual las vecindades parciales eran de  $3 \times 2$ , de lo cual se obtuvieron los diagramas de de Bruijn. Para fines prácticos, se tomo sólo una parte de los diagramas de de Bruijn, y se clasificaron de acuerdo a su comportamiento.

La figura 5.3 muestra el diagrama cuyo comportamiento denota permanencia. La figura 5.4 muestra el diagrama cuyo comportamiento muestra un desplazamiento en diagonal.

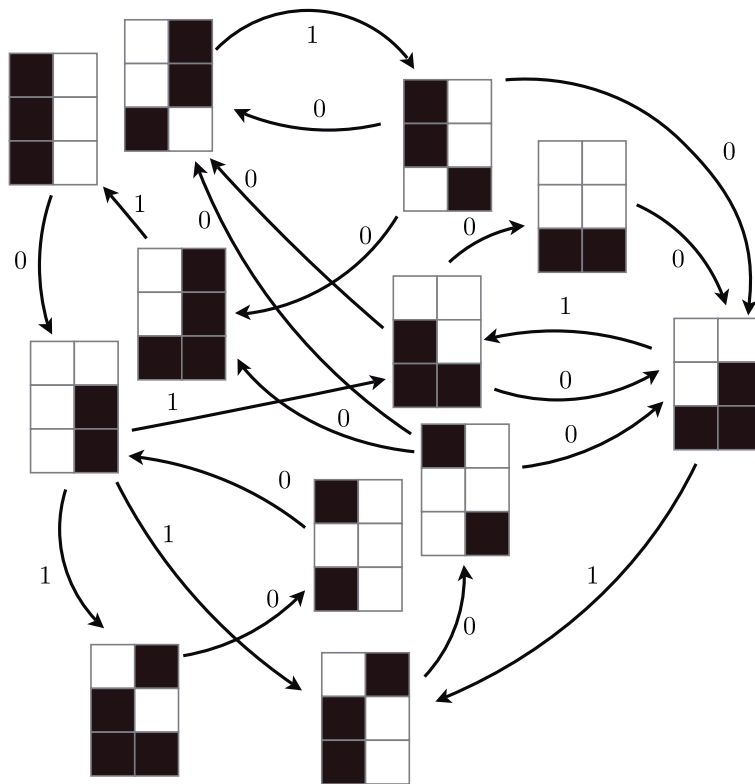
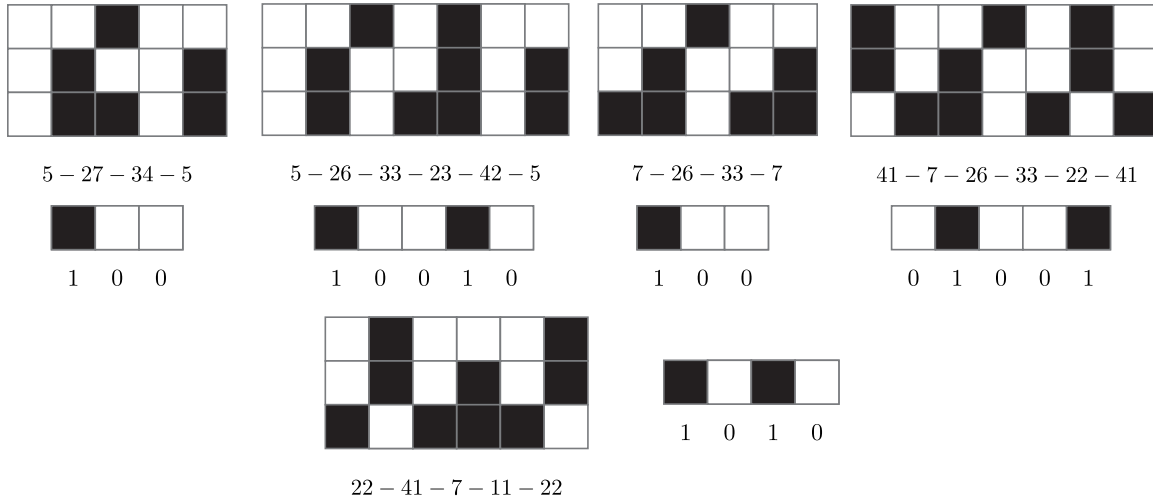


Figura 5.1: Permanencia el Juego de la Vida



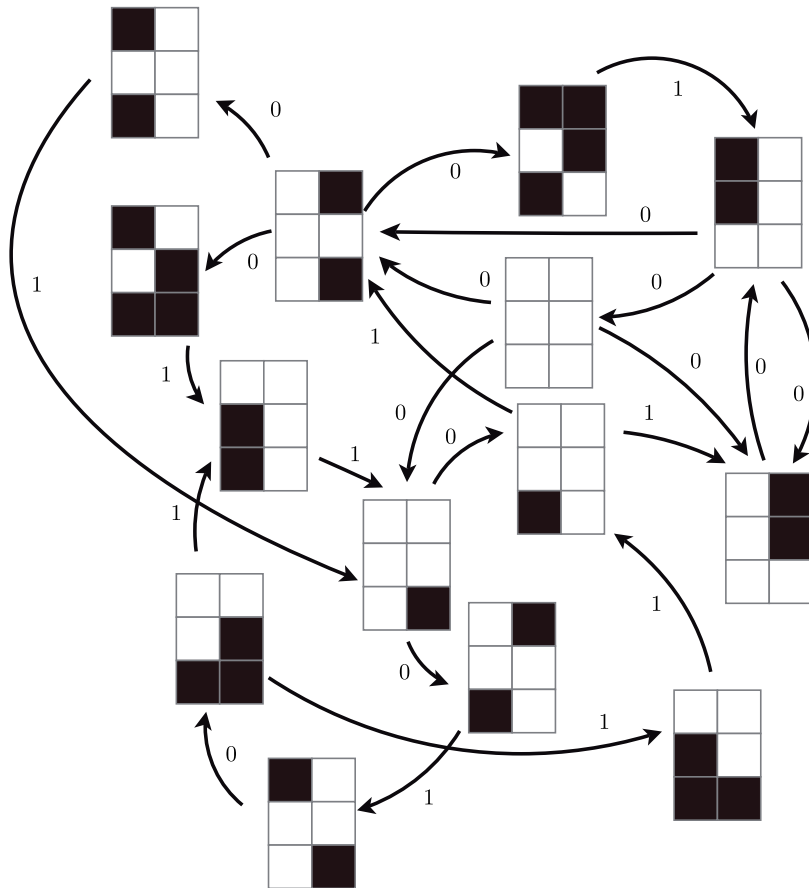
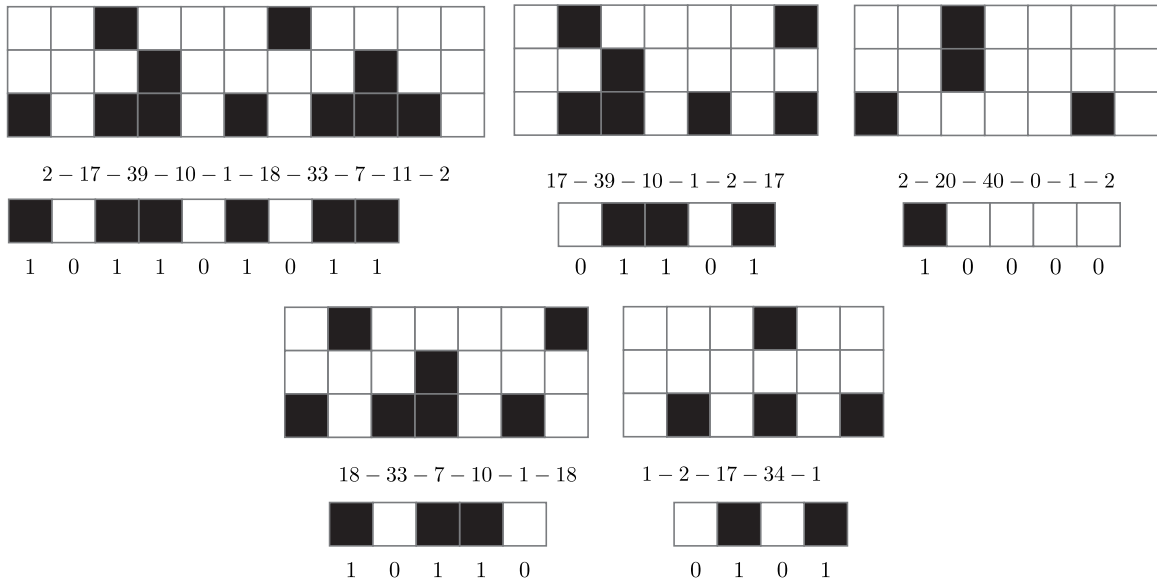


Figura 5.2: Corrimiento en diagonal Juego de la Vida

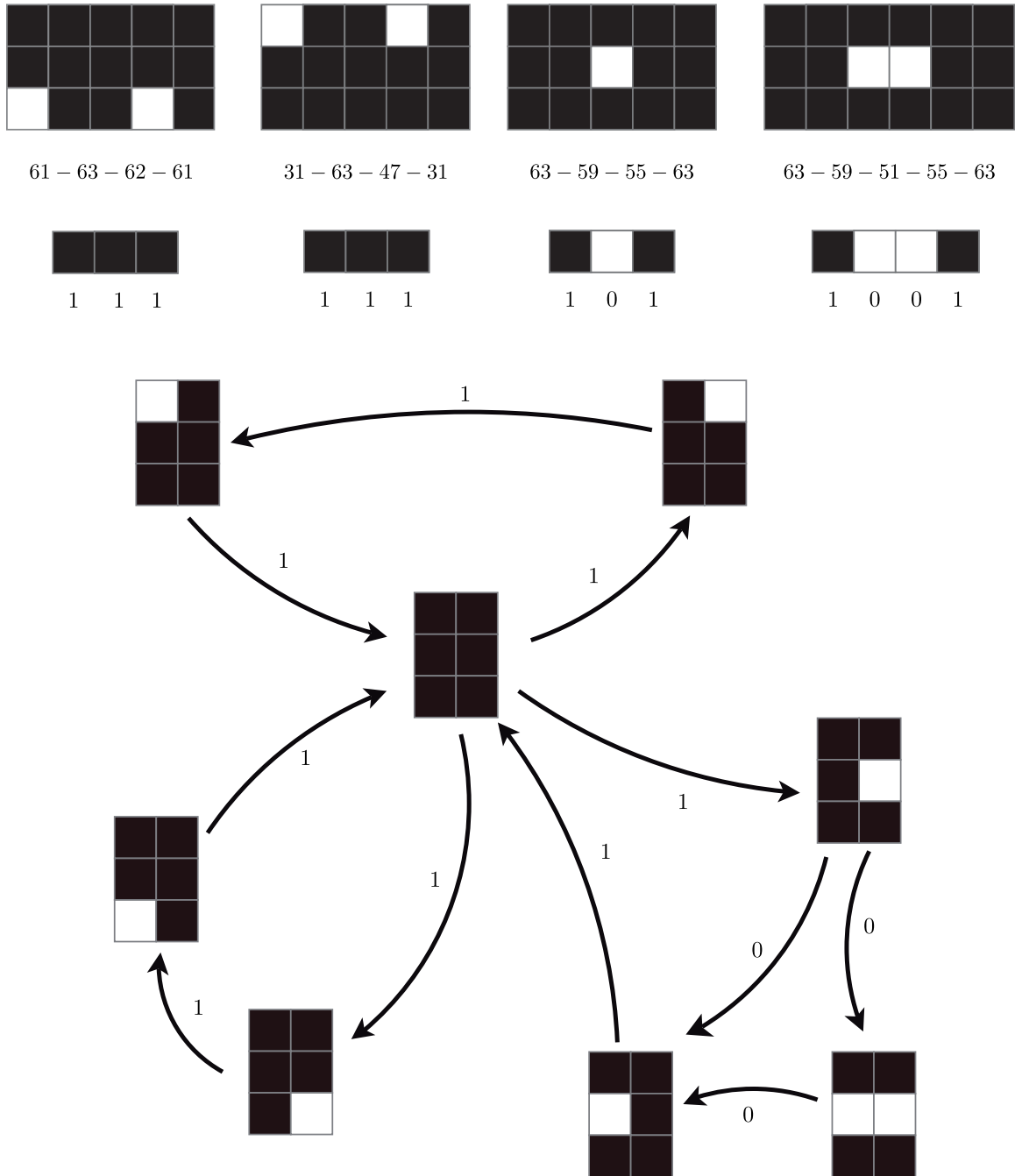


Figura 5.3: Permanencia Difusión

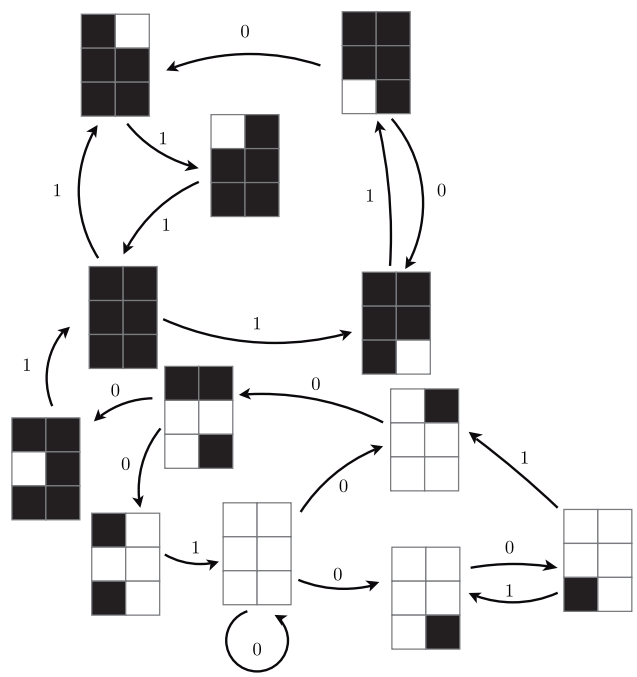
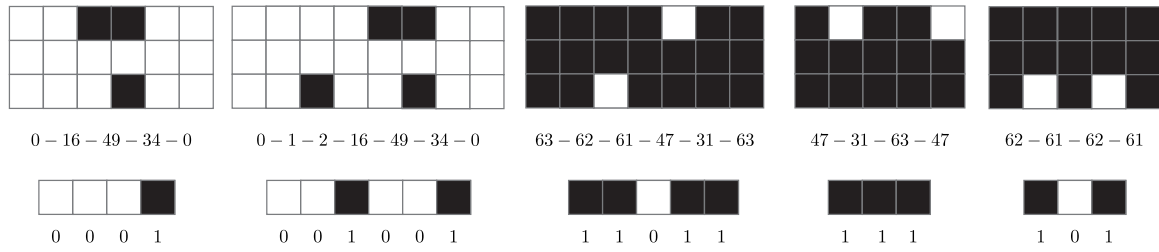


Figura 5.4: Corrimiento en diagonal Difusión



# Capítulo 6

## Resultados, discusión, conclusiones y trabajo a futuro

A lo largo de la presente tesis se expuso la introducción del uso de los diagramas de de Bruijn para analizar los AC 2D. Los temas importantes a discutir son: las definiciones de los componentes de los diagramas de de Bruijn para el caso especial de los AC 2D, la analogía que se hace con el análisis de los AC 1D vía diagramas de de Bruijn, y el método a seguir para la búsqueda de patrones. En la presente sección se concluye con la presentación de resultados obtenidos, las conclusiones y el trabajo a futuro.

### 6.1. Resultados

Como resultado del desarrollo de esta tesis, se obtuvo una serie de resultados. Estos resultados se pueden clasificar en dos: resultados formales y resultados experimentales. Los formales se relacionan a las estructuras propuestas para el tratamiento del problema. Los experimentales, son la consecuencia de aplicar las estructuras anteriores y con las cuales fue posible hacer un análisis de casos específicos. Los resultados se describen a continuación.

El desarrollo de esta tesis consistió en realizar el análisis de las definiciones básicas de los AC y de teorías, para proponer una definición formal de los elementos de los diagramas de de Bruijn. Como parte de los resultados se encuentra la definición de vecindad parcial 2D (definición 42) y la definición de aristas (definición 43) aplicadas para analizar los AC 2D.

Como parte de la propuesta del análisis de los AC 2D se diseñó un método basado en la extensión de las vecindades para evaluar un número de  $e \times f$  células de una lattice, tal y como se vio en la sección 4.2 de la presente tesis. Esta extensión se analizó para vecindades de Moore y vecindades hexagonales, y se probó que la vecindad de von Neumann requiere de un estudio con características diferentes.

Se implemento un método, el cual es capaz de encontrar las relaciones que existen entre todos los nodos, y es capaz de realizar filtros de comportamiento local; los cuales se apoyan de algoritmos que fueron propuestos para el desarrollo de esta tesis. Estos algoritmos son 1 y 2 como parte del programa de experimentación de los diagramas de de Bruijn, de los cuales se obtuvieron las matrices de de Bruijn de ambas reglas para cada una de las diferentes vecindades parciales.

El programa de experimentación abarca el análisis de las reglas de evolución Juego de la Vida y Regla de Difusión con vecindades parciales de diferentes tamaños.

Se encontró, que existen propiedades similares que se pueden observar a través de los diagramas de de Bruijn, a los que se observaban en los diagramas de 1 dimensión. Estas propiedades son: búsqueda de patrones de permanencia y desplazamientos; períodos.

## 6.2. Discusión

Una vez estudiadas las definiciones y conceptos de los elementos de los diagramas de de Bruijn para el análisis de los autómatas celulares de una dimensión, el proponer una definición para dos dimensiones se pensaría sencillo, sin embargo, se verificó que no todas las estructuras matemáticas pueden pasar fácilmente a más de una dimensión. En este sentido, se proponía realizar la extensión basándose en conceptos de espacios topológicos. El estudio de los espacios topológicos no resulto exitoso debido a que al definir el conjunto de nodos, no se cumplía con las propiedades básicas. Sin embargo, se propone que en base a las reglas y excepciones de estructuras que se derivan de los espacios topológicos se puede dar una definición diferente a la propuesta en la presente tesis.

La definición propuesta en este trabajo mostró tener sus limitantes. Primero, se analizaron dos tipos de vecindades simétricas y que rodean por completo la célula central, como la vecindad de Moore y la hexagonal, con las cuales se tuvo éxito para definir el conjunto de nodos y extensiones para las mismas. Posteriormente al analizar el caso de la vecindad de von Neumann no se tuvo el mismo éxito, debido a que su forma no permite hacer una división simétrica en la cual se incluya la célula central. No obstante, después de observar el comportamiento de la vecindad de von Neumann, se propone que existe una vecindad parcial, mostrada en la figura 6.1, donde la célula que se intersecta es únicamente la célula central, las células en gris claro son, por llamarles de alguna forma, información adicional y es en este punto donde el tratamiento de estas vecindades se propone que debe ser diferente.

Una vez teniendo las definiciones de los nodos, se prosiguió a la búsqueda de las relaciones posibles entre todos los nodos. Las relaciones en una dimensión se daban por medio de la teoría de corrimientos de secuencias, y también como operaciones propias

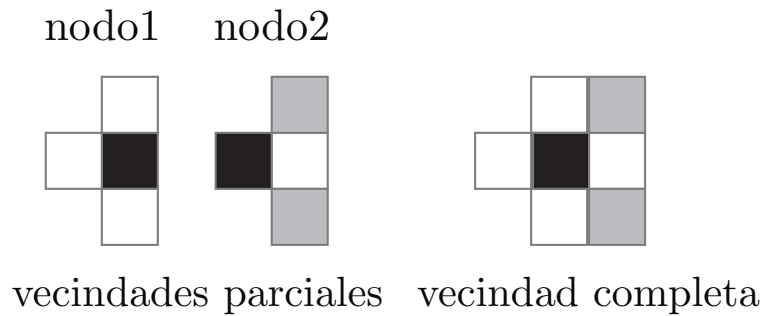


Figura 6.1: Vecindad parcial de von Neumann

de la teorías de campos finitos. Para el caso de dos dimensiones se determinó que se necesitaba de una estructura que se ajustará a las necesidades. Dado que la búsqueda de estructuras no fue exitosa, se decidió que se utilizarían cadenas grandes, fraccionadas modularmente de acuerdo al tamaño de la vecindad parcial. Esto es, si se tiene una vecindad parcial de Moore de  $2 \times 3$ , donde el 2 es el número de filas y 3 el número de columnas, entonces se tendrá una cadena de 6 elementos, donde los 3 símbolos que se encuentran más a la derecha representan las células de abajo y los que se encuentran más a la izquierda, representan a las células de arriba.

Para fines prácticos, se evaluaron únicamente reglas de autómatas celular de 2 estados. Con lo cual, el problema se reducía a operaciones booleanas y a corrimientos de secuencias. Esta manipulación se lleva a cabo por medio de la definición de operaciones para los campos finitos. Por lo cual, se estima que para manipular cadenas para reglas de autómatas celular de más de 2 estados se puede definir una operación para el campo finito propio.

El método propuesto para la evaluación de espacio grandes de células tiene como finalidad encontrar de manera más rápida estructuras o patrones que emerjan de la regla de evolución a evaluar. Se analizó que para la vecindad de Moore se encuentran dos vecindades parciales posibles. Para cada una de ellas se implementaron algoritmos para la búsqueda de intersecciones entre los nodos, sin embargo, se encontró que, dada la naturaleza de las secuencias de bits y las operaciones requeridas, el cálculo se daba más rápido en las vecindades parciales como la que se muestra en la figura 4.3(b).

Del análisis experimental se observó que existen propiedades de los AC 1D que también se presentan en 2D. La primer propiedad es el cálculo del comportamiento global en un tiempo  $t+1$ , este se realiza por medio del cálculo de los filtros, propuestos en esta tesis (ver sección 4.3). La razón de utilizar estos filtros es debido a que en la teoría de autómatas se buscan dos comportamientos específicos, la permanencia y el desplazamiento. La segunda propiedad que se encontró fue que es posible calcular los periodos por medio de los diagramas, sólo que existen limitantes, ya que los números

de células requeridas para probarlo experimentalmente son grandes. Sin embargo, se han encontrado las estructuras que tienen periodo 1, y se ha comprobado de forma experimental; por lo cual se propone que es posible encontrarlo para periodos mayores a 2.

### 6.3. Conclusiones

Después de presentado lo anterior, se llegó a las siguientes conclusiones.

Los diagramas de de Bruijn son herramientas útiles para conocer el comportamiento global de los AC; otras herramientas como la evaluación de puntos fijos de la teoría de sistemas dinámicos entre otros, no han sido llevado a cabo debido a la complejidad que implica, así como el costo computacional. Por tal razón, el cálculo de los diagramas de de Bruijn sirve para realizar un análisis de propiedades, tales como los Jardines del Eden y los Ancestros.

Después del análisis formal de las vecindades representativas de los AC 2D, se concluyó que por medio de las definiciones 42 y la definición 43 se deduce que la vecindad de von Neumann requiere de un tratamiento diferente y no es posible calcular las vecindades parciales para realizar el análisis por medio de los diagramas de de Bruijn.

Las definiciones 42 y la definición 43 son básica para realizar extensiones de las vecindades de Moore y Hexagonal y se puede analizar un regiones de  $e \times f$  células; considerando que la complejidad y el costo computacional que implica.

El costo computacional de este problema, aunque es grande, se puede solucionar al utilizar algoritmos que pueden ser paralelizados, lo cual implica un costo menor en tiempo de procesamiento.

Por lo anterior, se concluye que el uso de los diagramas de de Bruijn para AC 2D es posible realizarse de manera exitosa, y se utilizan principalmente para conocer el comportamiento global de un espacio de evolución.

### 6.4. Trabajo a futuro

El trabajo realizado en la tesis, plantea las bases para el análisis de los autómatas celulares en dos dimensiones vía diagramas de de Bruijn; lo cual, da pauta a que se proponga una serie de estudios que pueden complementar este trabajo y por ende dar un análisis de las propiedades de los autómatas celulares más completo.



El trabajo a futuro, se puede clasificar en tres áreas: la formal, la cual se encuentra directamente ligada a los conceptos matemáticos de los autómatas celulares; la computacional, la cual se relaciona a todo el procesamiento de información, optimización y visualización de los datos a procesar; y finalmente, el análisis experimental, la cual se enfoca a las propiedades específicas de los autómatas celulares y las características que se pueden conocer mediante los diagramas de de Bruijn.

En cuanto al área formal se proponen los siguientes trabajos a realizar:

- Encontrar una definición de vecindades parciales para incluir el tratamiento de la vecindad de von Neumann.
- Realizar el análisis de la vecindad de von Neumann para la construcción de los diagramas de de Bruijn.
- Buscar una definición basándose en teoría de corrimientos y campos finitos.

En el área computacional se propone:

- Encontrar un algoritmo que realice el cálculo de intersección de células para la vecindad hexagonal.
- Encontrar un algoritmo que realice el cálculo de intersección de células para la vecindad de von Neumann.
- Encontrar mecanismos de manejo de memoria y manipulación de archivos para encontrar espacios de evolución más grandes.
- Realizar una herramienta visual, como apoyo para el análisis de los diagramas de de Bruijn.
- Diseñar un algoritmo para el cálculo de matrices hexagonales.

En la parte del análisis experimental se propone:

- Encontrar estructuras de un periodo mayor a 1.
- Encontrar los Jardines del Eden.
- Encontrar patrones nuevos en cada una de las reglas.
- Encontrar patrones en reglas que utilicen vecindades hexagonales.
- Realizar el análisis de las matrices para vecindades hexagonales.



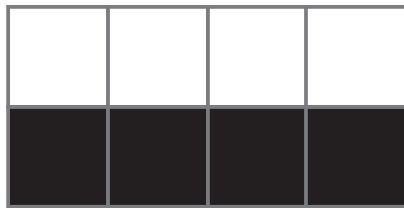
# Apéndice A

## Matrices de de Bruijn para Juego de la Vida

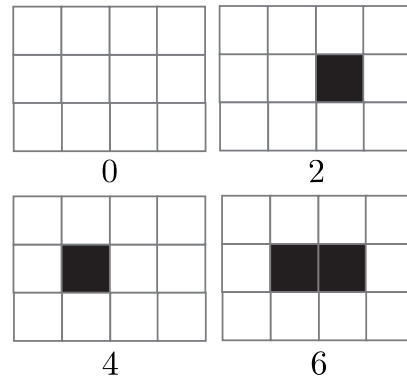
Este apéndice presenta la matriz de de Bruijn para vecindades de diferentes tamaños, para la regla de Juego de la Vida. Estos resultados se obtuvieron de la experimentación.

### A.1. Vecindades parciales de $4 \times 2$

En esta sección se presenta la matriz de de Bruijn para la vecindad parcial de  $4 \times 2$ , la cual se muestra en la figura B.1(a). En ella se evalúan dos células en horizontal. La matriz de la tabla B.2 representa las relaciones. Las filas representan el valor del nodo1 y las columnas el valor del nodo2. El número que se encuentra en la intersección de las filas y columnas representa el valor que tendrá las células en un tiempo  $t + 1$ ; los valores que toma se muestran en la figura B.1(b).



(a) Vecindad parcial de  $4 \times 2$  células



(b) Valores en  $t + 1$  de células, en vecindad parcial  $4 \times 2$

Figura A.1: Vecindad parcial  $4 \times 2$









## A.2. Vecindades parciales de $4 \times 3$

En esta sección se presenta la matriz de de Bruijn para la vecindad parcial de  $4 \times 3$ , la cual se muestra en la figura B.2. En ella se evalúan dos células en horizontal y dos células en vertical, formando un espacio de  $2 \times 2$ . La matriz de la tabla B.2 representa las relaciones. Las filas representan el valor del nodo1 y las columnas el valor del nodo2. El número que se encuentra en la intersección de las filas y columnas representa el valor que tendrá las células en un tiempo  $t + 1$ ; los valores que toma se muestran en la figura B.3.

Por razones de espacio, solo se presentará un fragmento de la matriz de de Bruijn.

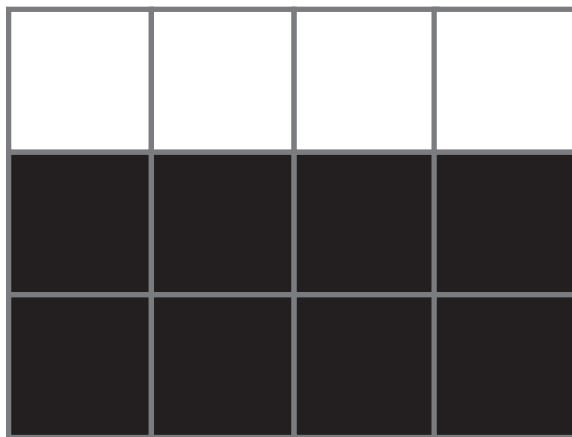


Figura A.2: Vecindad parcial de  $4 \times 3$



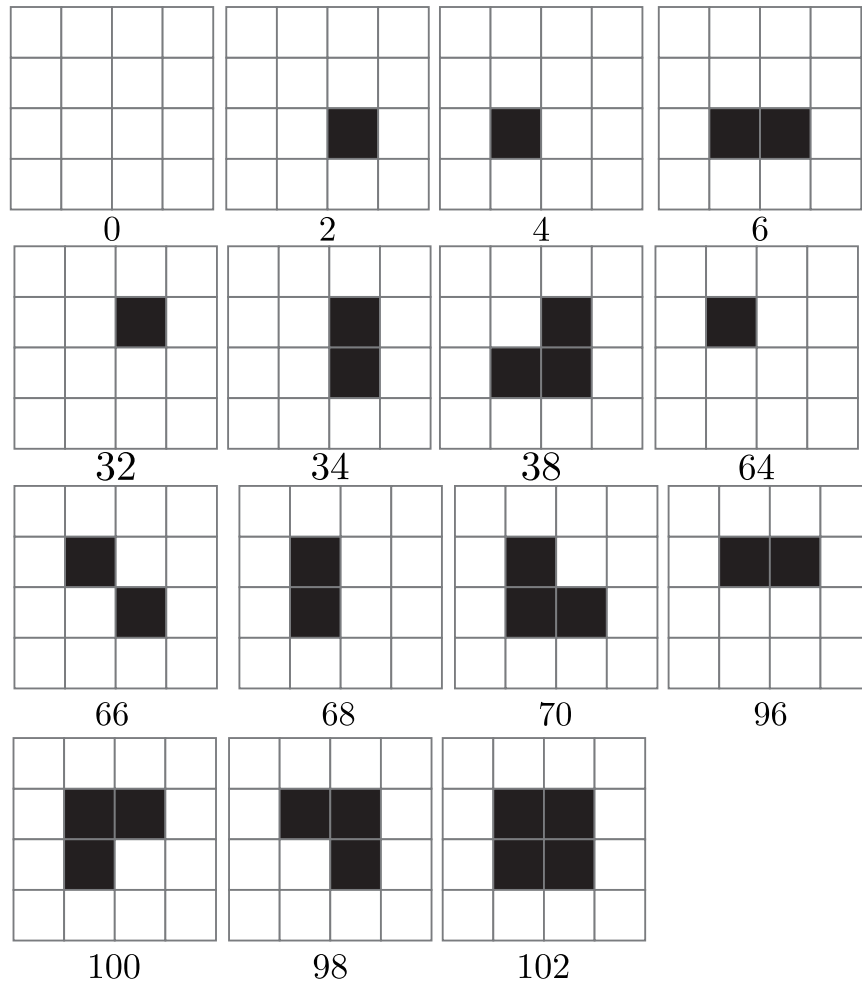


Figura A.3: Posibles valores que toman las células evaluadas en  $t + 1$

















# Apéndice B

## Matrices de de Bruijn la Regla Difusión

Este apéndice presenta la matriz de de Bruijn para vecindades de diferentes tamaños, para la regla de Difusión. Estos resultados se obtuvieron de la experimentación.

### B.1. Vecindades parciales de $4 \times 2$

En esta sección se presenta la matriz de de Bruijn para la vecindad parcial de  $4 \times 2$ , la cual se muestra en la figura B.1(a). En ella se evalúan dos células en horizontal. La matriz de la tabla B.2 representa las relaciones. Las filas representan el valor del nodo1 y las columnas el valor del nodo2. El número que se encuentra en la intersección de las filas y columnas representa el valor que tendrá las células en un tiempo  $t + 1$ ; los valores que toma se muestran en la figura B.1(b).

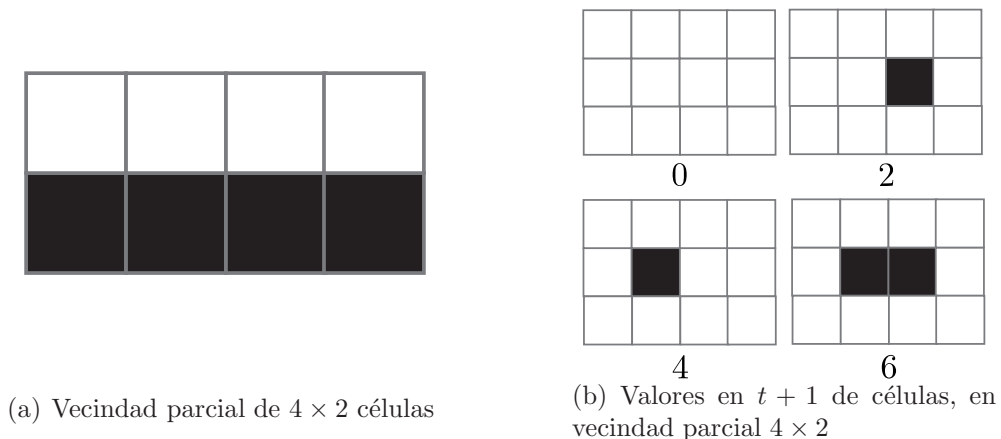


Figura B.1: Vecindad parcial  $4 \times 2$









## B.2. Vecindades parciales de $4 \times 3$

En esta sección se presenta la matriz de de Bruijn para la vecindad parcial de  $4 \times 3$ , la cual se muestra en la figura B.2. En ella se evalúan dos células en horizontal y dos células en vertical, formando un espacio de  $2 \times 2$ . La matriz de la tabla B.2 representa las relaciones. Las filas representan el valor del nodo1 y las columnas el valor del nodo2. El número que se encuentra en la intersección de las filas y columnas representa el valor que tendrá las células en un tiempo  $t + 1$ ; los valores que toma se muestran en la figura B.3. Por razones de espacio, solo se mostrará un fragmento de la matriz de de Bruijn.

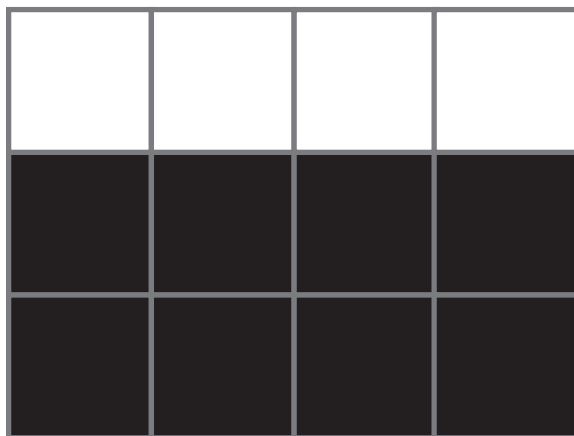


Figura B.2: Vecindad parcial de  $4 \times 3$

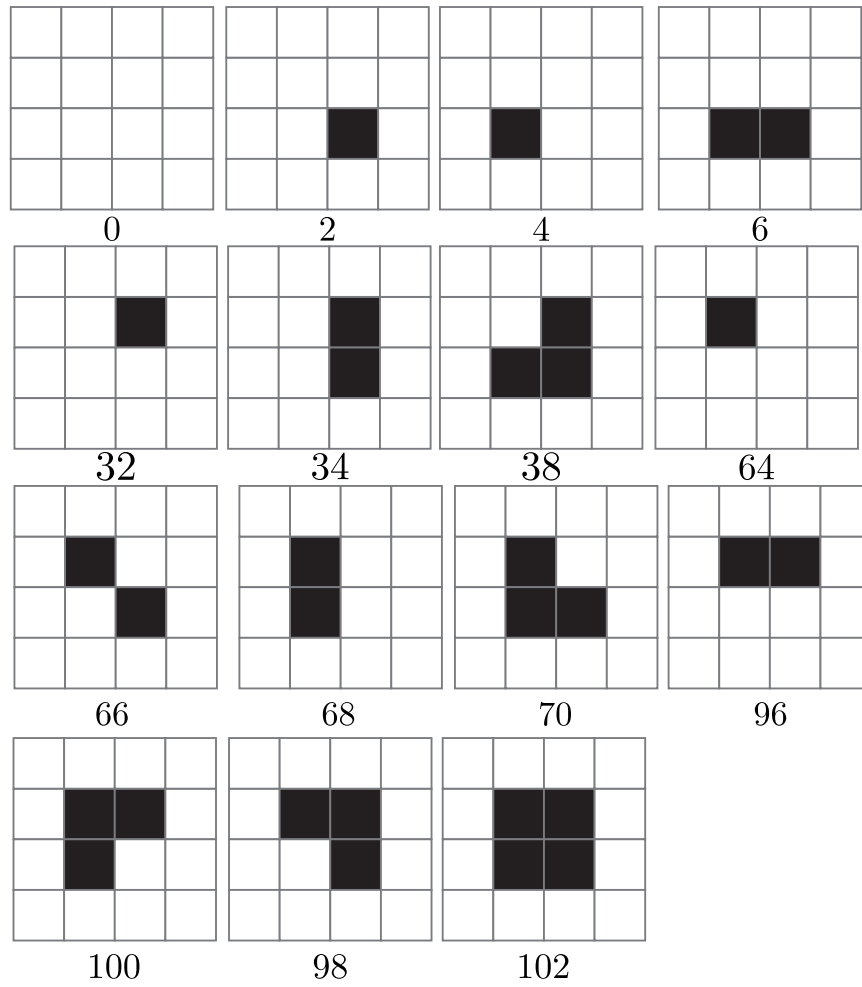


Figura B.3: Posibles valores que toman las células evaluadas en  $t + 1$

















# Apéndice C

## Manual técnico

El programa para obtener las matrices de de Bruijn fue desarrollado en el lenguaje de programación C. Esto se debe a la facilidad de uso de operadores binarios, y la gestión de memoria dinámica.

Por otro lado, se utilizó la librería OpenMP, para paralelizar la evaluación de la células y el cálculo de las intersecciones. Gracias a ésto, se obtuvieron resultados de evaluación de vecindades parciales de  $5 \times 3$  células, en un tiempo menor a 20 minutos; se probó que sin la paralelización tomaba en promedio más de tres horas.

```
1
2 #include <stdio.h>
3 #include <math.h>
4 #include <stdlib.h>
5 #include <omp.h>
6
7 /*la estructura de datos almacena el nodo1(filas)->nodo2(columna) de
8    manera direccionada
9    y celulaEvaluada almacena el valor que tendra el conjunto de celulas a
10   evaluar de las
11   vecindades formadas por nodo1 y nodo2 al ser evolucionada por una regla
12   de automata
13   celular en el tiempo t+1*/
14 struct dato
15 {
16     unsigned long int nodo1; //nodo1 en valor entero
17     unsigned long int nodo2; // nodo2 en valor entero
18     unsigned long int celulaEvaluada; // guarda el valor de las celulas
19     evaluadas
20     struct dato *next; //apuntador al siguiente nodo
21 };
22
23 /*en esta estructura unicamente se guarda una lista de los nodos que
24    contendran los datos*/
25 struct lista
26 {
```

```

    int alto; //guarda el numero de celulas que compone la subvecindad en
        alto, lo que equivaldria a filas
23 int ancho; //guarda el numero de celulas que compone la subvecindad en
        ancho, equivaldria a columnas
    struct dato *primerElemento; //direccion del primer elemento
25 struct dato *ultimoElemento; //direccion del segundo elemento
};
27
unsigned long int regionEncendida(int pos1, int pos2);
29 char *entero2Bits(unsigned long int num, int al, int an);
struct dato* agregarDato(unsigned long int n1, unsigned long int n2,
    unsigned long int cE);
31 struct lista* inicializaLista(int alto, int ancho);
unsigned long int evaluaVecindad(unsigned long int cc, unsigned long
    cont, int regla2D);
33 void evaluaCelulas(struct dato *aux, int alto, int ancho, int regla2D);

```

Listing C.1: Encabezado y declaración de estructuras utilizadas

```

1 #include "pruebasPau.h"
3 int main(int argc, char *argv[])
{
5     if(argc!=5)
    {
7         printf("Faltan parametros, Error -1\n Formato ./programa ancho
            alto regla archivo\n 1. Life 2. Difusion\n");
        return -1;
9     }
    int ancho, alto, regla; //ancho celulas en horizontal, alto celulas en
        vertical
11 ancho = atoi(argv[1]);
    alto = atoi(argv[2]);
13 regla = atoi(argv[3]);
    unsigned long int i,j; //contadores
15 unsigned long int nodo1;
    unsigned long int nodo2;
17 unsigned long int regiontraslape;
    unsigned long int limite;
19 struct lista *l; // lista de traslape
    struct dato *aux;
21 l=inicializaLista(alto, ancho);
    // el limite es 2^(ancho*alto), son 2 estados y (ancho*alto) numero de
        celulas
23 limite = (unsigned long)pow((double)2, (double)ancho*alto);
    // calcula los traslapes de todas las combinaciones
25 omp_set_num_threads(10);
    #pragma omp parallel for shared(l, limite, ancho, alto) firstprivate(
        nodo1, nodo2, regiontraslape) private(i, j)
27 for (j=(unsigned long)0; j<limite; j++) {
    printf(" j: %d \n", j );
29     nodo1=j;

```



```

regiontraslape =regionEncendida (((ancho*alto)-ancho-1),0); //region
  de traslape contiene los bits q se van a evaluar,
31 nodo1= nodo1 & regiontraslape;
for (i=0; i<limite; i++) {
33   nodo2=i;
   regiontraslape =regionEncendida (alto*ancho-1,ancho);
35   nodo2 = nodo2 & regiontraslape;
   nodo2 = nodo2 >>ancho;
37   if (!(nodo1 ^ nodo2))
   {
39     //inserta los nodos de traslape
     #pragma omp critical
41     {
       if(l->primerElemento == NULL)
43       {
         l->primerElemento = agregarDato(j, i, 2);
45         l->ultimoElemento = l->primerElemento;
       }
47       else
       {
49         aux = agregarDato(j, i, 2);
         l->ultimoElemento->next= aux;
51         l->ultimoElemento= aux;
       }
53     }
   }
55 }
}

//evalua los nodos resultantes
59 aux= l->primerElemento;
while(aux!=NULL)
61 {
   evaluaCelulas(aux, alto , ancho , regla );
63   aux= aux->next;
}
65 free(l);
}

67 void evaluaCelulas(struct dato *aux, int alto, int ancho, int regla2D)
69 {
   int i, j;
71   int posCelEval; //guarda la posicion de la celula que se esta
     evaluando
   unsigned long int mask;
73   unsigned long int cont; // ayuda a obener el valor
   unsigned long int res; // resultado
75   unsigned long int evaluacion;

77   // guarda la posicion de la celula a evaluarse... es -2 porq es -1 por
     la celular vecina y -1 por la posicion en binario
   mask=1;

```

```
79  res = 0;
80  for(i=(ancho * (alto -1))-2; i>0; i--)
81  {
82      posCelEval=i;
83      cont =1;
84      if(posCelEval%ancho != 0 && posCelEval%ancho != ancho-1)
85      {
86          //vecino norte
87          mask=1;
88          mask = mask << (posCelEval+ancho);
89          if(aux->nodo1 & mask)
90          {
91              cont = cont<<1;
92          }
93          //vecino este
94          mask=1;
95          mask = mask << (posCelEval-1);
96          if(aux->nodo1 & mask)
97          {
98              cont = cont<<1;
99          }
100         //vecino oeste
101         mask=1;
102         mask = mask << (posCelEval+1);
103         if(aux->nodo1 & mask )
104         {
105             cont = cont<<1;
106         }
107         //vecino noreste
108         mask=1;
109         mask = mask << (posCelEval+ancho-1);
110         if(aux->nodo1 & mask )
111         {
112             cont = cont<<1;
113         }
114         //vecino noroeste
115         mask=1;
116         mask = mask << (posCelEval+ancho+1);
117         if(aux->nodo1 & mask )
118         {
119             cont = cont<<1;
120         }
121
122         if(posCelEval > ancho)
123         {
124             //vecino sur
125             mask=1;
126             mask = mask << (posCelEval-ancho);
127             if(aux->nodo1 & mask )
128             {
129                 cont = cont<<1;
130             }
131         }
```

```
131     //vecino sureste
132     mask=1;
133     mask = mask << (posCelEval-ancho-1);
134     if(aux->nodo1 & mask )
135     {
136         cont = cont <<1;
137     }
138     //vecino suroeste
139     mask=1;
140     mask = mask << (posCelEval-ancho+1);
141     if(aux->nodo1 & mask )
142     {
143         cont = cont <<1;
144     }
145 }
146 else
147 {
148     posCelEval = posCelEval+ ancho;
149     //vecino sur
150     mask=1;
151     mask = mask << (posCelEval-ancho);
152     if(aux->nodo2 & mask )
153     {
154         cont = cont <<1;
155     }
156     //vecino sureste
157     mask=1;
158     mask = mask << (posCelEval-ancho-1);
159     if(aux->nodo2 & mask )
160     {
161         cont = cont <<1;
162     }
163     //vecino suroeste
164     mask=1;
165     mask = mask << (posCelEval-ancho+1);
166     if(aux->nodo2 & mask )
167     {
168         cont = cont <<1;
169     }
170     posCelEval = i;
171 }
172 //celula central
173 mask=1;
174 mask = mask << posCelEval;
175 if(aux->nodo1 & mask )
176 {
177     evaluacion=1; // valor de la celula central
178 }
179 else
180 {
181     evaluacion =0;
182 }
```

```
183     evaluacion= evaluaVecindad(evaluacion,cont,regla2D);
184     evaluacion = evaluacion << i;
185     res = res | evaluacion;
186 }
187 }
188 aux->celulaEvaluada = res;
189 }
190
191 unsigned long int evaluaVecindad(unsigned long int cc, unsigned long
192     cont, int regla2D)
193 {
194     // printf(" %i, %i \n",cc,cont);
195     switch(regla2D)
196     {
197     case 1: //evaluacion de life
198         if(cc==1 && cont > 2 && cont <= 8)
199         {
200             return 1;
201         }
202         else
203         {
204             if(cont == 8)
205             {
206                 return 1;
207             }
208             return 0;
209         }
210     case 2: //evaluacion de difusion
211         if(cc==1 && cont == 128)
212         {
213             return 1;
214         }
215         else
216         {
217             if(cont == 4)
218                 return 1;
219             }
220         return 0;
221     }
222 }
223 }
224
225 char *entero2Bits(unsigned long int num, int al, int an)
226 {
227     unsigned long int mask=1;
228     int i;
229     int limite=an*al-1;
230     char *cadena= malloc(sizeof(char *)*(limite+1));
231
232     for (i=limite; i>=0; i--) {
233         if(num&(mask<<i))
```

```
235     {
236         cadena[limite-i]= '1';}
237     else {
238         cadena[limite-i]= '0';
239     }
240 }
241 return cadena;
242 }
243 unsigned long int regionEncendida(int pos1, int pos2) //la posicion 1 es
244     la posicion maxima, la posicion 2 es la minima
245 {
246     unsigned long int a,b;
247     int i;
248     a=1;
249     b=0;
250     for(i=pos1; i>=pos2; i--)
251     {
252         b = b+ (a<<i);
253     }
254     return b;
255 }
256 struct dato* agregarDato(unsigned long int n1, unsigned long int n2,
257     unsigned long int cE)
258 {
259     struct dato *d = malloc(sizeof(struct dato *));
260     d->nodo1= n1;
261     d->nodo2 = n2;
262     d->celulaEvaluada = cE;
263     d->next = NULL;
264     return d;
265 }
266 struct lista* inicializaLista(int alto, int ancho)
267 {
268     struct lista *l= malloc(sizeof(struct lista*));
269     l->alto = alto;
270     l->ancho= ancho;
271     l->primerElemento = NULL;
272     l->ultimoElemento = NULL;
273     return l;
274 }
```

Listing C.2: Implementación de funciones y main



# Bibliografía

- [Abellanas(1991)] Abellanas, M., 1991: *Matemática discreta*. Macrobit.
- [Basurto and León(2011)] Basurto, R. and P. León, 2011: *Sistemas complejos como modelos de computación*, chap. Estudio de la dinámica y análisis de complejidad de la regla Espiral, 83–107. Spring.
- [Boykett(1997)] Boykett, T., 1997: Comparism of radius 1/2 and radius 1 paradigms in one dimensional reversible cellular automata.
- [Calced(2010)] Calced, A., 2010: *Introducción a la teoría de grafos*. Ediciones Elizcom.
- [Chaté and Manneville(1996)] Chaté, H. and P. Manneville, 1996: Non-trivial collective behavior in extensively-chaotic dynamical systems: an update. *Physica A*, **224**, 447–457.
- [Codd(1968)] Codd, E. F., 1968: *Cellular Automata*. Academic Press.
- [Eppstein(2002)] Eppstein, D., 2002: Searching for spaceships. *MSRI Publications*, **42**, 433–452.
- [Gardner(1970)] Gardner, M., 1970: Mathematical games: The fantastic combinations of john conway’s new solitaire game “life”. *Scientific American*, **223** (4), 120–123.
- [Giudici and Bris(1997)] Giudici, R. and A. Bris, 1997: *Introducción a la teoría de grafos*. Equinoccio Ediciones de la Universidad Simon Bolivar.
- [Golomb(1967)] Golomb, S. W., 1967: *Shift Register Sequences*. Holden-Day.
- [Gómez(2008)] Gómez, J. M., 2008: Computation of explicit preimages in one-dimensional cellular automata applying the de bruijn diagram. *Journal of Cellular Automata*, **3** (3), 219–230.
- [Gutowitz(1991)] Gutowitz, H., 1991: *Cellular automata, Theory and experiment*. The MIT Press.
- [Hillman(1991)] Hillman, D., 1991: The structure of reversible one-dimensional automata. *Physic*, **D** (52), 277–292.

- [Ilachinski(2001)] Ilachinski, A., 2001: *Cellular Automata. A discrete universe*. World Scientific Press.
- [Jen(1986)] Jen, E., 1986: Global properties of cellular automata,. *Journal of Journal of Statistical Physics*, **43** (1), 219–242.
- [Jen(1987)] Jen, E., 1987: Scaling of preimages in cellular automata. *Complex Systems*, **1**, 1045–1062.
- [Jen(1988)] Jen, E., 1988: Cylindrical cellular automata. *Communication in Mathematical Physics*, **118** (4), 569–590.
- [Marshall(1971)] Marshall, C., 1971: *Applied Graph Tehory*. Wiley-Interscience.
- [McIntosh(2009)] McIntosh, H. V., 2009: *One Dimensional Cellular Automata*. Luniver Press.
- [McIntosh(2010a)] McIntosh, H. V., 2010a: *Life’s Still Lives*, chap. 4, 35–50. Springer.
- [McIntosh(2010b)] McIntosh, H. V., 2010b: *A Zoo of Life forms*, chap. 5, 51–70. Springer.
- [Morita(2008)] Morita, K., 2008: Reversible computing and cellular automata—a survey. *Theoretical Computer Science*, **395**, 101–131.
- [Ralston(1982)] Ralston, A., 1982: De bruijn sequences-a model example of the interaction of discrete mathematics and computer science. *Mathematics Magazine*, **55** (3), 131–143.
- [Rodríguez and Chapa(2004)] Rodríguez, R. and S. Chapa, 2004: Using de bruijn diagrams to analyze 1d cellular automata traffic models. *Cellular Automata 7th International Conference*, Vol. 3305, 306–315.
- [Sutner(1991)] Sutner, K., 1991: De bruijn graphs and linear cellular automata. *Complex Systems*, **5** (1), 19–30.
- [Ullman and Aho(1995)] Ullman, J. and A. Aho, 1995: *Foundations of Computer Science: Principles of Computer Science Series*. Computer Science Press.
- [von Neumann(1966)] von Neumann, J., 1966: *Theory of self-reproducing automata*. University of Illinois.
- [Voorhees(1996)] Voorhees, B. H., 1996: *Computational analysis of one-dimensional cellular automata*, A, Vol. 15. World Scientific Series on Nonlinear Science,.
- [Voorhees(2006)] Voorhees, B. H., 2006: Discrete baker transformation for binary valued cylindrical cellular automata. *Cellular Automata 7th International Conference*, Vol. 4173, 182–191.



- [Wolfram(1983)] Wolfram, S., 1983: Statistical mechanics of cellular automata. *Rev. Modern Pshysics*, **5**, 601–644.
- [Wolfram(1984)] Wolfram, S., 1984: Computation theory of cellular automata. *Communication in Mathematical Physics*, **96 (1)**, 15–57.
- [Wolfram(2002)] Wolfram, S., 2002: *A New Kind of Science*. Wolfram Media Inc.
- [Wuensche(2011)] Wuensche, A., 2011: *Sistemas complejos como modelos de computación*, chap. Sistemas dinámicos complejos y caóticos conjuntos de atractores, memorias y redes discretas, 3–22. Spring.