



INSTITUTO POLITÉCNICO NACIONAL
Centro de Investigación en Computación



Modelos Celulares Alfa-Beta

TESIS

que para obtener el grado de

DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

presenta

BENJAMÍN LUNA BENOSO

Directores de Tesis: Dr. Cornelio Yáñez Márquez
Dr. Jesús Figueroa Nazuno

México, D. F.

Enero de 2011

Índice de Figuras

2.1	a) Vecindad de von Neumann. b) Vecindad de Moore. c) Vecino Izquierdo.	8
2.2	Triángulo de Sierpinsky-Gasket generado mediante un autómata celular (regla 90).	9
2.3	Dilatación de una región del plano real con un elemento estructurante en forma de disco euclideo centrado en el origen.	11
2.4	Erosión de una región del plano real con un elemento estructurante en forma de disco euclideo centrado en el origen.	11
2.5	Representación de una memoria asociativa en su fase a) de aprendizaje y b) de recuperación.	12
3.1	Ejemplo de un autómata celular con $S = \{0, 1\}$	25
3.2	Representación del conjunto A , el elemento estructurante $B = \{(0, 0), (1, 0), (2, 0)\}$ y la dilatación $A \oplus B$	26
3.3	Representación del conjunto A , el elemento estructurante $B = \{(-1, 0), (0, 0), (1, 0)\}$ y la erosión $A \ominus B$	27
3.4	Distancia euclidea	42
3.5	Ejemplo de la partición de un conjunto	43
4.1	Muestra la diferencia conceptual entre el elemento estructural y la vecindad de una célula. En a) se muestra el elemento estructural B , en b) la vecindad $v_{(0,0)}$, en c) el conjunto $(B)_{(2,2)}$ y en d) la vecindad $v_{(2,2)}$	45
4.2	Ejemplo de la composición de AC	48
4.3	Representaciones del espacio discreto	51
4.4	Distancia city-block, $d_{cb}((1, 4), (3, 1)) = 5$	52
4.5	Espacio discreto restringido a $A \times A$	53

4.6	Relación entre operador alfa y la función de anclaje sobre $x = (0, 1)$	54
4.7	Clases $\mathcal{C}_0 = \{(x, y) \in B \times A \alpha(0, y) = x\} = \{(1, 0), (0, 1)\}$ y $\mathcal{C}_1 = \{(x, y) \in B \times A \alpha(1, y) = x\} = \{(2, 0), (1, 1)\}$	55
4.8	Clasificación de los patrones $(0, 0)$ y $(2, 1)$	56
4.9	Lattice del autómata celular alfa-beta de tamaño $2m \times 2n$	57
4.10	Estados del AC alfa-beta.	57
4.11	Configuración del AC del ejemplo 4.2.1, en a) después de aplicar f_Q y en b) después de aplicar f_W	61
4.12	Configuración del AC del ejemplo 4.2.2, en a) después de aplicar f_Q y en b) después de aplicar f_W	64

Índice de Tablas

2.1	Fase de aprendizaje de la Lernmatrix	14
2.2	Fase de aprendizaje para la red asociativa	16
3.1	Operador α	31
3.2	Operador β	31
4.1	Muestra el funcionamiento del AC alfa-beta en su fase de recuperación <i>max</i>	65
4.2	Muestra el funcionamiento del AC alfa-beta en su fase de recuperación <i>min</i>	66
5.1	k -NN aplicado a la base de datos de: <i>Iris</i>	71
5.2	Memorias Asociativas alfa-beta aplicado a la base de datos: <i>Iris</i>	71
5.3	Rendimiento de clasificación de otros modelos para la base de datos: <i>Iris</i>	72
5.4	Modelo Celular alfa-beta aplicado a la base de datos: <i>Iris</i> ; con elemento estructurante: vecino izquierdo	73
5.5	Modelo Celular alfa-beta aplicado a la base de datos: <i>Iris</i> ; con elemento estructurante: vecindad de Neumann	73
5.6	Modelo Celular alfa-beta aplicado a la base de datos: <i>Iris</i> ; con elemento estructurante: vecinda de Moore	73
5.7	k -NN aplicado a la base de datos: <i>Abalone</i>	75
5.8	Memorias Asociativas alfa-beta aplicado a la base de datos: <i>Abalone</i>	75
5.9	Rendimiento de clasificación para la base de datos: <i>Abalone</i>	75
5.10	Modelo Celular alfa-beta aplicado aplicado a la base de datos: <i>Abalone</i> ; con elemento estructurante: vecino izquierdo	76
5.11	Modelo Celular alfa-beta aplicado a la base de datos: <i>Abalone</i> ; con elemento estructurante: vecindad de Neumann	76

5.12	Modelo Celular alfa-beta aplicado a la base de datos: <i>Abalone</i> ; con elemento estructurante: vecindad de Moore	76
5.13	k -NN aplicado a la base de datos: <i>Hepatitis</i>	78
5.14	Memorias Asociativas aplicado a <i>Hepatitis</i>	78
5.15	Rendimiento de clasificación para la base de datos: <i>Hepatitis</i> .	79
5.16	Modelo Celular alfa-beta aplicado a la base de datos: <i>Hepati-</i> <i>tis</i> ; con elemento estructurante: vecino izquierdo	80
5.17	Modelo Celular alfa-beta aplicado a la base de dato: <i>Hepatitis</i> ; con elemento estructurante: vecindad de Neumann	80
5.18	Modelo Celular alfa-beta aplicado a la base de datos: <i>Hepati-</i> <i>tis</i> ; con elemento estructurante: vecindad de Moore	80
5.19	k -NN aplicado a la clasificación de Retinopatía Diabética . . .	82
5.20	Memorias Asociativas alfa-beta aplicado a la clasificación de Retinopatía Diabética	82
5.21	Modelo Celular alfa-beta aplicado a <i>Retinopatía Diabética</i> con elemento estructurante: vecino izquierdo	82
5.22	Modelo Celular alfa-beta aplicado a <i>Retinopatía Diabética</i> con elemento estructurante: vecindad de Neumann	83
5.23	Modelo Celular alfa-beta aplicado a <i>Retinopatía Diabética</i> con elemento estructurante: vecindad de Moore	83

Contenido

Agradecimientos	v
Resumen	vi
Abstract	vii
Índice de Figuras	ix
Índice de Tablas	ix
1 Introducción	1
1.1 Antecedentes	1
1.2 Problemas Científicos y Tecnológicos	3
1.3 Hipótesis y Justificación	4
1.4 Objetivo	4
1.5 Aportaciones	5
1.6 Organización del Documento	5
2 Estado del Arte	7
2.1 Autómatas Celulares	7
2.2 Morfología Matemática	10
2.3 Memorias Asociativas	12
2.4 Memorias Asociativas relevantes	13
2.4.1 Lernmatrix de Steinbuch (1961)	14
2.4.2 Correlograph de Willshaw, Buneman & Longuet-Higgins (1969)	15
2.4.3 Linear Associator de Anderson-Kohonen (1972)	16
2.4.4 Memoria Asociativa Hopfield (1982)	17
2.4.5 Memoria Asociativa Bidireccional (1988)	19

2.4.6	Memorias Asociativas Morfológicas (1998)	20
2.4.7	Memorias Asociativas alfa-beta (2002)	21
2.5	Autómatas Celulares y Reconocimiento de Patrones	21
3	Materiales y Métodos	23
3.1	Autómatas Celulares	23
3.2	Morfología Matemática	25
3.2.1	Dilatación	25
3.2.2	Erosión	26
3.3	Memorias Morfológicas	27
3.3.1	Memorias morfológicas autoasociativas (<i>max</i> y <i>min</i>)	27
3.3.2	Memorias morfológicas heteroasociativas (<i>max</i> y <i>min</i>)	29
3.4	Memorias Asociativas alfa-beta	30
3.4.1	Memorias Heteroasociativas alfa-beta (<i>max</i> y <i>min</i>)	32
3.4.2	Memorias Autoasociativas alfa-beta (<i>max</i> y <i>min</i>)	36
3.5	Normas y Distancias	40
3.6	Partición sobre un Conjunto	42
3.7	Orden de Complejidad	43
4	Modelo Propuesto	44
4.1	Morfología Matemática Celular	44
4.1.1	Dilatación Celular	44
4.1.2	Erosión Celular	46
4.1.3	Apertura Morfológica y Apertura Celular	49
4.1.4	Cerradura Morfológica y Cerradura Celular	49
4.2	Modelos Celulares alfa-beta	50
4.2.1	Operadores alfa-beta	50
4.2.2	Autómata Celular alfa-beta	55
4.2.3	Modelos Celulares alfa-beta con Erosión y Dilatación	66
5	Experimentos y Resultados	70
5.1	Base de Datos: Iris	70
5.2	Base de Datos: Abalone	74
5.3	Base de Datos: Hepatitis	77
5.4	Base de Datos: Retinopatía Diabética	81
5.5	Discusión de Resultados	84

6 Conclusiones	87
6.1 Aportaciones	88
6.2 Trabajos a Futuro	89
Publicaciones	90
Referencias	92

Resumen

Los autómatas celulares, la morfología matemática y los modelos asociativos alfa-beta son tres áreas de investigación científica de interés actual; y cada una de estas tres áreas cuenta con un esquema teórico que sirve de soporte a importantes resultados que, a través del tiempo, se han aplicado con éxito en diversos ámbitos del quehacer humano.

Sin embargo, estos tres campos de interés científico se han mantenido ajenos; es decir, se han desarrollado de manera separada, y en cada uno de los tres esquemas de conocimientos se realizan avances, tanto teóricos como prácticos, pero sin relación alguna.

Este trabajo de tesis consiste en unir los destinos de estas tres interesantes y relevantes áreas de investigación. Por un lado, se ha creado una serie de algoritmos, sustentados teóricamente, que permiten representar las operaciones básicas de la morfología matemática como autómatas celulares, lo que ha traído consecuencias útiles: con esta nueva representación, será posible aplicar todo el bagaje de resultados de autómatas celulares a las operaciones morfológicas, dondequiera que se apliquen.

El nuevo esquema teórico permite representar a los autómatas celulares como una cadena de operaciones morfológicas, hecho que hará posible algo no imaginado hasta hoy: se pueden trabajar los autómatas celulares como cadenas de operaciones morfológicas, a través de la nueva representación.

Se logró establecer también una simbiosis entre autómatas celulares y los modelos asociativos alfa-beta; para ello, cada una de las operaciones alfa y beta se han representado como autómatas celulares y viceversa, con el correspondiente soporte teórico. Así, una aplicación que use autómatas celulares podrá aprovechar todos los teoremas alfa-beta, y aquellas aplicaciones de los modelos alfa-beta podrán ser trabajadas usando los resultados importantes de los autómatas celulares.

Además de que se ha creado el modelo teórico que soporta estas simbiosis entre áreas que estaban ajenas, se llevó a cabo una serie de experimentos de aplicación de los nuevos modelos celulares alfa-beta, cuyos resultados evidencian la superioridad de estos nuevos modelos en tareas de clasificación de patrones.

Abstract

Cellular automata, mathematical morphology, and Alpha-Beta associative memories are three areas of current scientific research. Each of these fields has a theoretical scheme that supports a growing number of important results, which have been successfully applied to different areas of human endeavor throughout the years.

How ever, these three fields of scientific interest have remained mutually exclusive. In other words, these fields have developed separately and independently: in each knowledge scheme advances are made, both on theory and applications, but with no relation to those advances of the other areas mentioned.

The present work of thesis consists of bringing together these three very interesting and relevant areas of research. On one hand, a series of algorithms has been created -each with its theoretical foundation- allowing the representation of basic mathematical morphology operations as cellular automata. This result offers some useful implications: with this new representation, it becomes possible to apply the whole set of results on cellular automata to morphological applications, wherever they are found.

The new theoretical perspective allows the representation of cellular automata as a succession of morphological operations, which opens the possibilities for something entirely unthought-of to this day: cellular automata can be applied as series of morphological operations under the new representation.

A symbiosis was built between cellular automata and Alpha-Beta associative models; for this, both operations alpha and beta have been represented as cellular automata and vice versa, with the corresponding theoretical foundation. Thus, an application using cellular automata can take advantage of all of the Alpha-Beta theorems, and those applications of the Alpha-beta models can incorporate the relevant results of cellular automata.

Besides creating the theoretical model which supports the symbiosis between areas that were foreign to each other, a series of experiments on the application of the new Alpha-Beta cellular models have been done. The results obtained exhibit the superiority of these new models on tasks of pattern classification.

Capítulo 1

Introducción

En este trabajo de tesis se describe y sustenta una forma novedosa de asociar dos áreas de investigación científica: los autómatas celulares y las memorias asociativas alfa-beta. Se presenta la propuesta de un modelo original de memoria asociativa implementado mediante autómatas celulares.

1.1 Antecedentes

El concepto de autómata celular (AC) fue desarrollado a finales de los años cuarenta por Jon von Neumann [1]. Con la ayuda de su amigo Stanislaw M. Ulam [23], von Neumann creó el primer autómata celular que surge de la idea de crear una máquina que pudiera autorreproducirse [4, 23, 24]. Los autómatas celulares tuvieron un gran éxito gracias a John Horton Conway en 1970, quien desarrolló un autómata celular conocido como *Life* (el *juego de la vida*) el cual fue publicado en la revista *Scientific American* [5].

Los autómatas celulares son herramientas matemáticas que sirven para modelar sistemas dinámicos discretos, en donde el comportamiento del sistema depende de la interacción de los individuos que la componen con individuos de su entorno. El tiempo se actualiza sincrónicamente para cada individuo del sistema [2, 3, 4, 5, 6, 7, 8, 9, 23, 25, 26, 27]. Los autómatas celulares han tenido aplicaciones a la criptografía [8, 10], en modelaciones de flujo vehicular [11, 28, 29], en modelaciones de virus biológicos [12, 13, 30], en

efectos psicológicos de personas ante ciertas situaciones [31], en la expansión de incendios forestales [32], por sólo mencionar algunas.

La morfología matemática es un campo basado sobre la teoría de conjuntos, y sus creadores fueron Georges Matheron [33] y Jean Serra [34, 35, 36, 37]. A mediados de los años sesenta Matheron da origen a la morfología matemática con sus trabajos referentes al estudio de análisis de imágenes en el ámbito de los medios porosos [33, 38]. Jean Serra dio un impulso más a la morfología matemática con la publicación de su libro en 1982 [34] y sus avances en 1988 [35], siendo una de sus mayores aplicaciones el tratamiento digital de imágenes [39, 40, 41]. Es importante hacer notar que, a través de la historia, la morfología matemática se ha mantenido ajena al desarrollo de los autómatas celulares.

En otro orden de ideas, tenemos a las memorias asociativas, las cuales son modelos matemáticos que tienen como propósito fundamental recuperar correctamente patrones completos a partir de patrones de entrada, los cuales pueden estar alterados con ruido aditivo, sustractivo o combinado [18]. Históricamente, se han creado modelos de memorias asociativas cuya operación se basa en distintos paradigmas. Las memorias asociativas clásicas (Lernmatrix (1961) [14], Correlograph (1969)[42], Linear Associator (1972)[43, 44], Memoria Asociativa Hopfield (1982)[45, 46], Memoria Asociativa Bidireccional (1988) [47]) se basan en las operaciones usuales de suma y multiplicación (sumas de productos). Este paradigma fue sustituido por otro más productivo cuando, en 1998, Ritter et al [48, 49] crearon las memorias asociativas morfológicas, las cuales se basan en las operaciones morfológicas de dilatación y erosión; por ello, usan máximos o mínimos de sumas para la fase de aprendizaje y máximos o mínimos de sumas para la fase de recuperación [48, 49]. En el año 2002 Yáñez-Márquez creó dos operadores a los que denominó alfa y beta, y a partir de éstos se crearon las memorias asociativas alfa-beta [18].

Actualmente se han realizado varias investigaciones en el campo de las memorias asociativas alfa-beta (memorias alfa-beta bidireccionales [19, 50], memorias asociativas alfa-beta basadas en el código Jhonson-Möbius modificado [51], redes neuronales alfa-beta [52], máquinas asociativas alfa-beta con soporte vectorial [53]); sin embargo, en lo que concierne a su relación con el área de los autómatas celulares, ambos campos se han mantenido ajenos.

1.2 Problemas Científicos y Tecnológicos

Consideremos estos hechos:

- Hecho 1: La Morfología Matemática (MM) y los Autómatas Celulares (AC) son dos áreas de investigación ajenas.
- Hecho 2: Los Modelos Asociativos alfa-beta (MAab) y los Autómatas Celulares son dos áreas de investigación ajenas.
- Hecho 3: La Morfología Matemática es un área de estudio madura y bien estructurada, con resultados formales específicos (cuya utilidad se ha mostrado ampliamente).
- Hecho 4: Los Autómatas Celulares representan un campo de investigación con un esquema teórico bien fundamentado, lo cual ha redundado en el desarrollo de ideas y de resultados expresados formalmente, que se han aplicado en diversas áreas de la ingeniería y de la ciencia durante siete décadas.
- Hecho 5: Desde su creación, los Modelos Asociativos alfa-beta han exhibido su utilidad en dos ámbitos: la investigación científica (por ser modelos teóricos bien fundamentados) y las aplicaciones (debido a su alto grado de rendimiento y exactitud).

Imaginemos que podamos representar, mediante algún modelo, la dilatación (que es una de las operaciones básicas de la MM) como un AC, ¿ qué pasaría si se aplica todo el esquema teórico de los AC en esta dilatación (que ya es un AC)? podríamos esperar resultados sorprendentes? útiles? interesantes?...

Uno de los problemas científicos a resolver en esta tesis es precisamente dilucidar las virtudes y las ventajas científicas y tecnológicas que potencialmente se perciben en la simbiosis MM-AC, puesto que, así como es posible representar las operaciones de la MM como AC, también es posible representar los AC como operaciones de la MM.

Algo similar ocurre en una posible simbiosis singular: MAab-AC, lo cual representa otro de los problemas científicos a resolver en este trabajo de

tesis, dado que es posible representar las operaciones alfa y beta como AC, y algunos AC como operaciones propias de los MAab.

Al resolver estos problemas científicos, los ámbitos de acción de los esquemas teóricos y los resultados mencionados en los hechos 1 a 3, se extenderán de manera importante, a fin de atacar problemas tecnológicos relacionados con la clasificación de patrones en cúmulos de datos de diferentes áreas de aplicación, entre ellas el área médica.

1.3 Hipótesis y Justificación

La justificación de esta tesis queda plenamente establecida dado que, con este trabajo de tesis se resuelven los problemas científicos y tecnológicos planteados en la sección previa.

La hipótesis de este trabajo de tesis consiste en lo siguiente: las simbiosis MM-AC y MAab-AC permitirán extender los ámbitos de acción de los esquemas teóricos propios de las MM, los AC y los MAab de modo que sea posible aplicar en AC las propiedades y los resultados que se encuentran en MMab o en MM, y recíprocamente, es posible aplicar el esquema teórico de los AC en la MM y en los MMab, lo cual redundará en clasificación de patrones con mejor rendimiento.

1.4 Objetivo

Objetivo General

Proponer nuevos modelos de memorias asociativas implementados con autómatas celulares.

Objetivos particulares

1. Desarrollar las operaciones básicas de la morfología matemática con autómatas celulares.
2. Implementar las operaciones α y β con autómatas celulares.

3. Implementar los modelos de las memorias asociativas alfa-beta *max* y *min* mediante autómatas celulares.
4. Ilustrar con ejemplos sencillos el funcionamiento de los modelos propuestos.
5. Aplicar los nuevos modelos de memorias asociativas creados, de manera que funcionen como clasificadores en bases de datos conocidas en la literatura científica actual.
6. Realizar estudios estadísticos comparativos en el rendimiento de los modelos propuestos en esta tesis al aplicarlo a bases de datos de la UCI MLR, contra otros algoritmos de clasificación de patrones.

1.5 Aportaciones

1. Implementación de las operaciones básicas de la morfología matemática con autómatas celulares: dilatación celular y erosión celular.
2. Implementación de las memorias asociativas alfa-beta mediante autómatas celulares.
3. Una asociación teórica entre los campos de los autómatas celulares y de las memorias asociativas alfa-beta.
4. Un nuevo modelo de memorias asociativas implementado mediante autómatas celulares alfa-beta y las nuevas operaciones morfológicas celulares.
5. Comparación del rendimiento del modelo propuesto con otros modelos aplicados a bases de datos de la UCI MLR.

1.6 Organización del Documento

Este capítulo describe los antecedentes, la justificación, el objetivo y las aportaciones a alcanzar en el trabajo de tesis.

El capítulo 2 presenta el estado del arte, donde se muestra un panorama general de la investigación realizada en cuanto a autómatas celulares, morfología matemática y memorias asociativas se refiere.

En el capítulo 3 se muestran los materiales y métodos, que incluyen los conceptos básicos y las herramientas matemáticas que se requieren para el desarrollo de este trabajo.

El capítulo 4 es la parte principal de la tesis, dado que aquí se presenta, se describe, se ejemplifica y se sustenta teóricamente el modelo original propuesto.

En el capítulo 5 se muestran los experimentos y resultados del nuevo modelo, y un estudio comparativo respecto del rendimiento de otros clasificadores.

El capítulo 6 incluye las conclusiones, las aportaciones y el trabajo a futuro que se propone desarrollar. Finalmente se incluyen las referencias.

Capítulo 2

Estado del Arte

2.1 Autómatas Celulares

El concepto de autómatas celulares (AC) fue desarrollado a finales de los años cuarenta por John von Neumann. Neumann se interesaba en crear una máquina que pudiera autoreproducirse [4, 23, 24] como carecía de los medios físicos y dispositivos técnicos que le permitieran hacer su proyecto, trató de hacerlo utilizando una forma muy simple y abstracta siguiendo una sugerencia de su amigo Stanislaw M. Ulam [23]; esencialmente, Ulam sugirió un espacio cuadrículado donde cada cuadro podía ser ocupado por una *célula* dada que podía tener un número finito de estados, el tiempo en este espacio avanzaba en lapsos discretos, cada célula hacía una transición a un nuevo estado dependiendo de su valor actual y el de las células en cada borde o vecinos. De este modo, von Neumann pudo desarrollar un AC que podía autorreproducirse, contaba con 29 estados diferentes para cada célula.

En 1970 John Horton Conway desarrolló un juego matemático llamado *Life*, éste apareció publicado en la columna de Martin Gardner en *Scientific American* [5]. *Life* es un autómatas celulares que evoluciona en el tiempo y el espectador observa cómo ocurre esta evolución, recibe su nombre debido a que cada célula tiene dos estados posibles, vivo o muerto.

Esencialmente, un AC d -dimensional consiste de un enrejado (*lattice*) regular de celdas (células o sitios) d -dimensional infinitamente extendido. Cada célula tiene asociada una variable de estado, llamada el *estado de la célula*

que fluctúa sobre un conjunto finito, llamado *conjunto de estados*. El sistema se actualiza sincrónicamente en cada instante de tiempo discreto por medio de una regla explícita, llamada *función de transición*; las reglas son determinísticas, sin embargo el resultado que se obtiene es aleatorio. La función de transición es utilizada para determinar cuál será el estado de una célula en el instante de tiempo $t + 1$ basada en los estados de sus n células vecinas (figura 2.1) en el instante de tiempo t . La asignación de estados a todas las células de la lattice es llamada una *configuración de la lattice*. [2, 3, 6, 7, 8, 9, 25, 54]

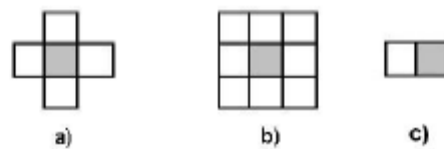


Figura 2.1: a) Vecindad de von Neumann. b) Vecindad de Moore. c) Vecino Izquierdo.

Los problemas que anteriormente se trataban desde un punto de vista continuo, ahora se pueden abordar desde el punto de vista discreto; por ejemplo, el problema de flujo vehicular o simulaciones de gases; una forma para atacar estos problemas es recurrir a ecuaciones diferenciales o ecuaciones cinéticas de tipo Boltzmann, ocasionando que la resolución del problema se vuelva compleja; una herramienta auxiliar son los autómatas celulares [55, 56, 57]. Los AC tienen la característica de ser sistemas discretizados en tiempo y espacio; y por su naturaleza, son sistemas de cómputo en paralelo, idea que surge aun antes de que existiera el cómputo secuencial. Los AC son sistemas que nos ayudan a entender cualquier problema físico en el que un individuo interactúa con otros en su vecindad, además son fácilmente de implementar cuando ya se conocen el problema a fondo y fácilmente de representar de manera visual [58]. Por ejemplo, el triángulo de Sierpinsky-Gasket es un fractal cuya similitud se ve reflejada en triángulos contenidos indefinidamente, este fractal se muestra de manera visual mediante una regla conocida como regla 90 [59] (figura 2.2).

Abordemos el campo de los AC de manera filosófica. El trabajar con AC es una nueva manera de ver el mundo en el que nos desarrollamos. Final-

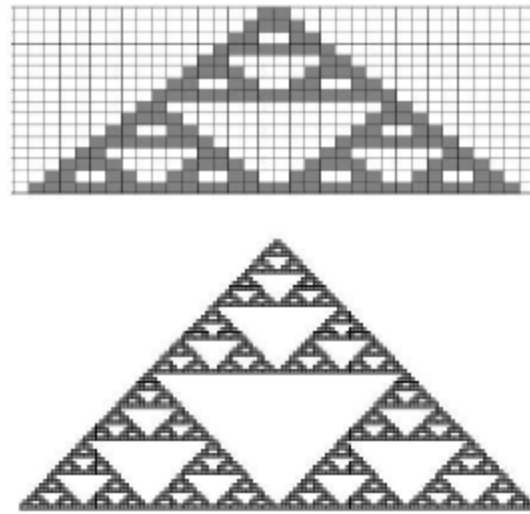


Figura 2.2: Triángulo de Sierpinsky-Gasket generado mediante un autómata celular (regla 90).

mente cada partícula que existe en el universo está fuertemente ligada con otras de su alrededor, todo funciona de manera organizada y en sociedad, es decir todo involucra a un individuo y a un conjunto de individuos en su vecindad. El universo está compuesto por millones y millones de astros que interactúan unos con otros; la sociedad como la vivimos ahora, cada individuo se desarrolla en su medio ambiente, sin embargo no deja de interactuar con otro conjunto de personas; un insecto, interactúa con otros de su misma especie en su hábitat; las bacterias, los virus, los micro-organismos en general requieren de convivir e interactuar con otros tipos de micro-organismos; el cerebro está compuesto por una enorme cantidad de neuronas que interactúan entre ellas; y así como estos ejemplos, basta observar el mundo que nos rodea para percatarnos de que cada ente interactúa con otros en su entorno. De esta manera, lo que se trata de exponer, es que la idea de lo que son los autómatas celulares va más allá de un simple sistema computacional para comprender un problema en particular [58].

2.2 Morfología Matemática

La morfología matemática de conjuntos es un marco de trabajo basado exclusivamente en la teoría de conjuntos. La teoría original desarrollada por Georges Matheron [33] y Jean Serra [34, 35, 36, 37] se ha utilizado con enorme éxito en el procesamiento digital de imágenes binarias. Sin embargo, su utilidad no termina en el procesamiento de imágenes o de señales, y dado que su única premisa de trabajo es la de utilizar conjuntos y sus propiedades, es de utilidad para cualquier clase de problema que se modele por medio de conjuntos [60, 61, 62, 63, 64, 65].

Existen dos operaciones fundamentales en la morfología matemática: la *dilatación* y la *erosión* [37, 66]. La dilatación es un término referido a aumentar, expandir, crecer, entre otras, de un cuerpo. Por otro lado, la erosión se refiere a la contracción, disminución, decrecimiento, adelgazamiento, entre otras, de un cuerpo.

Para hacer crecer un conjunto A (dilatarlo) cambiaremos su forma y aumentaremos su tamaño a través de un proceso controlado de añadido o agregación de puntos o elementos. Con el fin de establecer el control del proceso de crecimiento, tomaremos como referencia otro conjunto B con una estructura y forma simples, el cual será llamado indistintamente de varias maneras equivalentes: *elemento estructurante*, *estructural* o *de estructura*. El conjunto B permitirá un proceso de crecimiento controlado y estructurado. El tamaño y forma final del nuevo conjunto dependerá fuertemente del tamaño y forma del elemento estructurante B . Si se dilata una región del plano real con un elemento estructurante en forma de disco euclideo real con centro en el origen, la dilatación tiene el efecto de engrosar la región del plano real con una franja de anchura r , siendo r el radio del disco euclideo, como se muestra en la figura 2.3 [66].

La operación de erosión consiste en hacer decrecer un conjunto A a través de un proceso controlado de eliminación de elementos, tomando como referencia un elemento estructurante B . Al igual que sucede con la dilatación, el tamaño y forma finales del conjunto erosionado dependerá fuertemente del tamaño y forma del elemento estructurante B . Si se erosiona una región del plano real con un elemento estructurante en forma de disco euclideo real

con centro en el origen, la erosión tiene el efecto de adelgazar a la región del plano real con una franja de anchura r , siendo r el radio del disco euclideo, como se muestra en la figura 2.4 [66].

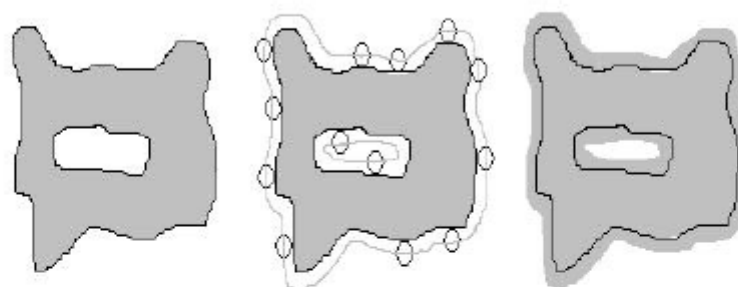


Figura 2.3: Dilatación de una región del plano real con un elemento estructurante en forma de disco euclideo centrado en el origen.

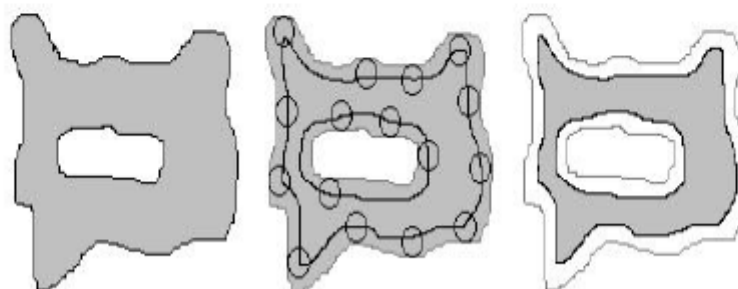


Figura 2.4: Erosión de una región del plano real con un elemento estructurante en forma de disco euclideo centrado en el origen.

2.3 Memorias Asociativas

Una memoria asociativa tiene como propósito fundamental recuperar correctamente patrones completos a partir de patrones de entrada, los cuales pueden estar alterados con ruido aditivo, sustractivo o combinado.

El funcionamiento de las memorias asociativas puede dividirse en dos fases claramente distinguibles:

1. *Fase de Aprendizaje* (generación de la memoria asociativa)
2. *Fase de recuperación* (operación de la memoria asociativa)

Una memoria asociativa puede formularse como un sistema de entrada y salida, idea que se esquematiza en la figura 2.5.

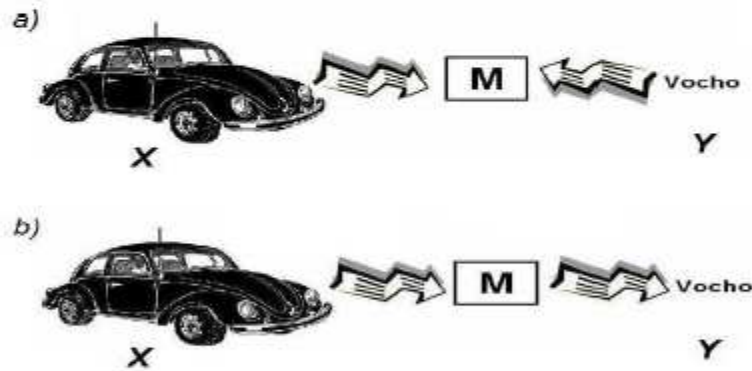


Figura 2.5: Representación de una memoria asociativa en su fase a) de aprendizaje y b) de recuperación.

El patrón de entrada está representado por un vector columna denotado por x y el patrón de salida, por un vector columna denotado por y . Cada uno de los patrones de entrada forma una asociación con el correspondiente patrón de salida. La notación para una asociación es similar a la de una pareja ordenada; por ejemplo, los patrones x y y del esquema forman la asociación (x, y) .

Dado un número entero positivo k específico, la asociación correspondiente será $(\mathbf{x}^k, \mathbf{y}^k)$.

La memoria asociativa M se representa generalmente mediante una matriz cuya componente ij -ésima es m_{ij} [67]; la matriz M se genera a partir de un conjunto finito de asociaciones conocidas de antemano: éste es el conjunto fundamental de asociaciones, o simplemente *conjunto fundamental*. Se denota por p la cardinalidad del conjunto fundamental (p es un número entero positivo).

Si μ es un índice, el conjunto fundamental se representa de la siguiente manera:

$$\{(\mathbf{x}^\mu, \mathbf{y}^\mu) | \mu = 1, 2, \dots, p\}$$

A los patrones que conforman las asociaciones del conjunto fundamental, se les llama *patrones fundamentales*, y la naturaleza de dicho conjunto proporciona un importante criterio para clasificar las memorias asociativas [68, 69]:

- a) Una memoria es *autoasociativa* si se cumple $\mathbf{x}^\mu = \mathbf{y}^\mu$ para cada $\mu = 1, 2, \dots, p$.
- b) Una memoria es *heteroasociativa* si se cumple que existe $\mu \in \{1, 2, \dots, p\}$ tal que $\mathbf{x}^\mu \neq \mathbf{y}^\mu$.

Se dice que la memoria M produce una *recuperación perfecta* si al presentarle un patrón alterado $\tilde{\mathbf{x}}^\omega$ como entrada ($\omega \in \{1, 2, \dots, p\}$), M responde con el correspondiente patrón fundamental de salida \mathbf{y}^ω .

2.4 Memorias Asociativas relevantes

A continuación, se presenta cronológicamente una selección de las memorias asociativas más representativas, con objeto de establecer el marco de referencia en que surgieron las memorias asociativas alfa-beta.

	x_1^μ	x_2^μ	\dots	x_j^μ	\dots	x_n^μ
y_1^μ	m_{11}	m_{12}	\dots	m_{1j}	\dots	m_{1n}
y_2^μ	m_{21}	m_{22}	\dots	m_{2j}	\dots	m_{2n}
\vdots	\vdots	\vdots		\vdots		\vdots
y_i^μ	m_{i1}	m_{i2}	\dots	m_{ij}	\dots	m_{in}
\vdots	\vdots	\vdots		\vdots		\vdots
y_p^μ	m_{p1}	m_{p2}	\dots	m_{pj}	\dots	m_{pn}

Tabla 2.1: Fase de aprendizaje de la Lernmatrix

2.4.1 Lernmatrix de Steinbuch (1961)

Karl Steinbuch [14], conocido como el padre alemán de la informática, fue el primero en desarrollar un modelo de memoria asociativa: la *Lernmatrix*.

La Lernmatrix es una memoria heteroasociativa que puede funcionar como un clasificador de patrones binarios si se escogen adecuadamente los patrones de salida; es un sistema de entrada y salida que al operar acepta como entrada un patron binario $\mathbf{x}^\mu \in A^n$, $A = \{0, 1\}$ y produce como salida la clase $y^\mu \in A^p$ que le corresponde (de entre p clases diferentes), codificada ésta con un método simple, a saber: para representar la clase $k \in \{1, 2, \dots, p\}$, se asignan a las componentes del vector de salida \mathbf{y}^μ los siguientes valores: $y_k^\mu = 1$ y $y_j^\mu = 0$ para $j = 1, 2, \dots, k-1, k+1, \dots, p$ [14, 70].

En la tabla 2.1 se esquematiza la fase de aprendizaje para la Lernmatrix de Steinbuch, al incorporar la pareja de patrones de entrenamiento $(\mathbf{x}^\mu, \mathbf{y}^\mu) \in A^n \times A^p$.

Cada uno de los componentes m_{ij} de M , la Lernmatrix de Steinbuch, tiene valor cero al inicio, y se actualiza de acuerdo con la regla $m_{ij} + \Delta m_{ij}$, donde:

$$\Delta m_{ij} = \begin{cases} +\epsilon & \text{si } x_j^\mu = 1 = y_i^\mu \\ -\epsilon & \text{si } x_j^\mu = 0 \text{ y } y_i^\mu = 1 \\ 0 & \text{en otro caso} \end{cases}$$

Siendo ϵ una constante positiva escogida previamente.

La fase de recuperación consiste en encontrar la clase a la que pertenece un vector de entrada $x^\omega \in A^n$ dado. Encontrar la clase significa obtener las coordenadas del vector $y^\omega \in A^p$ que le corresponde al patrón x^ω ; en virtud del método de construcción de los vectores y^μ la clase debería obtenerse sin ambigüedad.

La i -ésima coordenada y_i^ω del vector de clase $y^\omega \in A^p$ se obtiene como lo indica la siguiente expresión, donde \vee es el operador máximo:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega = \bigvee_{h=1}^p (\sum_{j=1}^n m_{hj} \cdot x_j^\omega) \\ 0 & \text{en otro caso} \end{cases}$$

2.4.2 Correlograph de Willshaw, Buneman & Longuet-Higgins (1969)

El Correlograph es un dispositivo óptico elemental capaz de funcionar como una memoria asociativa. En palabras de los autores "el sistema es tan simple, que podría ser construido en cualquier laboratorio escolar de física elemental" [42].

El Correlograph consta de tres pantallas opacas y una fuente de luz; el aparato genera correlogramas de puntos luminosos en una de las pantallas (la cual se perfora precisamente en esos puntos), a partir de pares de patrones de huecos realizados en las otras dos pantallas (fase de aprendizaje). Cada correlograma así generado se usa entonces para obtener cada uno de los patrones de entrada partiendo de su patrón asociado (fase de recuperación).

A pesar de las evidentes diferencias entre el Correlograph y la Lernmatrix, los autores Willshaw, Buneman & Longuet-Higgins tomaron como punto de partida este dispositivo óptico para crear un ente al que llamaron red asociativa, la cual no es otra cosa que un modelo de memoria asociativa tipo crossbar parecida a la Lernmatrix, pero con una regla de aprendizaje diferente a la regla de Steinbuch y con la característica adicional: esta memoria no se comporta sólo como un clasificador, sino como una memoria que asocia patrones de entrada con patrones de salida.

La red asociativa es un sistema de entrada y salida que al operar acepta como entrada un patrón binario $x^\mu \in A^n$, $A = \{0, 1\}$ y produce como salida un patrón $y^\mu \in A^m$.

	x_1^μ	x_2^μ	\dots	x_j^μ	\dots	x_n^μ
y_1^μ	m_{11}	m_{12}	\dots	m_{1j}	\dots	m_{1n}
y_2^μ	m_{21}	m_{22}	\dots	m_{2j}	\dots	m_{2n}
\vdots	\vdots	\vdots		\vdots		\vdots
y_i^μ	m_{i1}	m_{i2}	\dots	m_{ij}	\dots	m_{in}
\vdots	\vdots	\vdots		\vdots		\vdots
y_m^μ	m_{m1}	m_{m2}	\dots	m_{mj}	\dots	m_{mn}

Tabla 2.2: Fase de aprendizaje para la red asociativa

En la tabla 2.2 se esquematiza la fase de aprendizaje para la red asociativa, al incorporar la pareja de patrones de entrenamiento $(x^\mu, y^\mu) \in A^n \times A^m$.

Cada uno de los componentes m_{ij} de la red asociativa M tiene valor cero al inicio, y se actualiza de acuerdo con la regla:

$$m_{ij} = \begin{cases} 1 & \text{si } y_i^\mu = 1 = x_j^\mu \\ \text{valor anterior} & \text{en otro caso} \end{cases}$$

En la fase de recuperación se le presenta a la red asociativa M un vector de entrada $x^\omega \in A^n$. Se realiza el producto de la matriz M por el vector x^ω y se ejecuta una operación de umbralado, de acuerdo con la siguiente expresión:

$$y_i^\omega = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} \cdot x_j^\omega \geq u \\ 0 & \text{en otro caso} \end{cases}$$

Donde u es el valor de umbral.

2.4.3 Linear Associator de Anderson-Kohonen (1972)

En la Linear Associator la fase de aprendizaje consiste de dos etapas [44]:

1. Para cada una de las p asociaciones (x^μ, y^μ) se encuentra la matriz $y^\mu \cdot (x^\mu)^t$ de dimensiones $m \times n$.

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu \ x_2^\mu \ \dots \ x_n^\mu)^t$$

$$y^\mu \cdot (x^\mu)^t = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \cdots & y_2^\mu x_j^\mu & \cdots & y_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix}$$

2. Se suman las p matrices para obtener la memoria

$$M = \sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t = [m_{ij}]_{m \times n}$$

de manera que la ij -ésima componente de la memoria M se expresa así:

$$m_{ij} = \sum_{\mu=1}^p y_i^\mu x_j^\mu$$

La fase de recuperación consiste en presentarle a la memoria un patrón de entrada x^ω , donde $\omega \in \{1, 2, \dots, p\}$ y realizar la operación:

$$M \cdot x^\omega = \left[\sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t \right] \cdot x^\omega$$

Este modelo hace uso de las operaciones usuales del álgebra lineal para las fases de aprendizaje y recuperación, sin embargo, requiere que los patrones del conjunto fundamental sean ortonormales para que la recuperación sea correcta.

2.4.4 Memoria Asociativa Hopfield (1982)

La memoria Hopfield es autoasociativa, es decir, el conjunto fundamental es $\{(x^\mu, x^\mu) | \mu = 1, 2, \dots, p\}$ con [45] :

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad y \quad A = \{1, -1\}$$

La fase de aprendizaje para la memoria Hopfield es similar a la fase de aprendizaje del Linear Associator, con una diferencia relacionada con la diagonal principal en ceros, como se muestra en la siguiente regla para obtener la ij -ésima componente de la memoria Hopfield M :

$$M = \sum_{\mu=1}^p [x^{\mu} \cdot (x^{\mu})^t - I] = [m_{ij}]_{n \times m}$$

Donde I es la matriz identidad.

La forma en que se lleva a cabo la fase de recuperación para la memoria Hopfield cambia drásticamente respecto de lo que sucede en el Linear Associator. Al presentar un patrón de entrada \tilde{x} a la memoria Hopfield, ésta cambiará su estado con el tiempo, de modo que cada neurona x_i ajuste su valor de acuerdo con el resultado que arroje la comparación de la cantidad

$$\sum_{j=1}^n m_{ij} x_j$$

Si representemos el estado de la memoria Hopfield en el tiempo t por $x(t)$; entonces $x_i(t)$ representa el valor de la neurona x_i en el tiempo t y $x_i(t+1)$ el valor de x_i en el tiempo siguiente ($t+1$).

Dado un vector columna de entrada \tilde{x} , la fase de recuperación consta de tres pasos [45]:

1. Para $t = 0$, se hace $x(t) = \tilde{x}$; es decir, $x_i(0) = \tilde{x}_i, \forall i \in \{1, 2, 3, \dots, n\}$
2. $\forall i \in \{1, 2, 3, \dots, n\}$ se calcula $x_i(t+1)$ de acuerdo con la condición siguiente:

$$x_i(t+1) = \begin{cases} 1 & \text{si } \sum_{j=1}^n m_{ij} x_j(t) > 0 \\ x_i(t) & \text{si } \sum_{j=1}^n m_{ij} x_j(t) = 0 \\ -1 & \text{si } \sum_{j=1}^n m_{ij} x_j(t) < 0 \end{cases}$$

3. Se compara $x_i(t+1)$ con $x_i(t) \forall i \in \{1, 2, 3, \dots, n\}$. Si $x(t+1) = x(t)$ el proceso termina y el vector recuperado es $x(0) = \tilde{x}$. De otro modo, el proceso continuará de la siguiente forma: Los pasos 2 y 3 se iteran tantas veces como sea necesario hasta llegar a un valor $t = \tau$ para el cual $x_i(\tau+1) = x_i(\tau) \forall i \in \{1, 2, 3, \dots, n\}$; el proceso termina y el patrón recuperado es $x(\tau)$.

La importancia del trabajo de Hopfield radica en que sirvió como detonador para el resurgimiento, tanto de las memorias asociativas, como de las redes neuronales, dado que su modelo es básicamente una red neuronal recurrente que converge a estados localmente estables, análogamente a algunos modelos físicos conocidos (como la orientación de dominios en sistemas magnéticos). La memoria Hopfield es una memoria autoasociativa y su capacidad de recuperación es de apenas $0.15n$, siendo n la dimensión de los patrones almacenados.

2.4.5 Memoria Asociativa Bidireccional (1988)

La memoria BAM (Bidirectional Associative Memory) es un sistema de entrada y salida que al operar acepta como entrada un patrón bipolar $\mathbf{x}^\mu \in A^n$, y produce como salida un patrón $\mathbf{y}^\mu \in A^m$, $A = \{-1, 1\}$ [47].

La fase de aprendizaje para la BAM genera dos matrices diferentes, M y M^t , pero en cada caso es similar a la fase de aprendizaje de la memoria Hopfield y del Linear Associator [47].

1. Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se encuentra la matriz $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$ de dimensiones $m \times n$:

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu \quad x_2^\mu \quad \cdots \quad x_n^\mu)$$

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_i^\mu & \cdots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \cdots & y_2^\mu x_i^\mu & \cdots & y_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_i^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_i^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix}$$

2. Se suman la p matrices $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$ para obtener la matriz M :

$$M = \sum_{\mu=1}^p \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n}$$

3. La matriz M^t de $n \times n$ se define como la transpuesta de M ; es decir:

$$\begin{aligned} M^t &= \left[\sum_{\mu=1}^p y^\mu \cdot (x^\mu)^t \right]^t = \sum_{\mu=1}^p [y^\mu \cdot (x^\mu)^t]^t \\ &= \sum_{\mu=1}^p x^\mu \cdot (y^\mu)^t = [w_{ij}]_{m \times n} \end{aligned}$$

La fase de recuperación de la BAM, para cada uno de los dos tipos de patrones x^μ, y^μ es idéntica a la fase de recuperación de la memoria Hopfield. En la BAM se requiere verificar convergencia en ambos casos, para x^μ y para y^μ . Representemos el estado de la BAM en el tiempo t por $[x(t), y(t)]$; entonces $[x_i(t), x_i(t)]$ representa los valores de x_i y y_i en el tiempo t , y $[x_i(t+1), y_i(t+1)]$ los valores en el tiempo siguiente ($t+1$).

Se espera que la pareja recuperada $[x(\tau), y(\tau)]$ sea una de las parejas (x^μ, y^μ) del conjunto fundamental.

La desventaja fundamental de la BAM de Kosko es su capacidad de recuperación correcta de patrones, la cual es mucho menor al mínimo de las dimensiones de los patrones entrenados.

2.4.6 Memorias Asociativas Morfológicas (1998)

Ritter et al. [48], con sus *memorias Morfológicas* (MAMs), rompe con el esquema utilizado a través de los años en los modelos de memorias asociativas clásicas, que utilizan operaciones convencionales entre vectores y matrices para la fase de aprendizaje y suma de productos para la recuperación de patrones. Las memorias asociativas morfológicas cambian los productos por sumas y la suma por máximos o mínimos en ambas fases, tanto de aprendizaje como de recuperación. La diferencia fundamental entre las memorias asociativas clásicas (e.g. Linear Associator) y las morfológicas radica en sus fundamentos operacionales, los cuales son las operaciones morfológicas de dilatación y erosión. Estas memorias tienen 2 problemas fuertes: el primero es que aunque la versión autoasociativa asegura que el conjunto fundamental siempre va a ser recuperado correctamente, en la versión heteroasociativa no

sucede esto. Por otro lado, el tipo de ruido que soporta la tipo *max* es aditivo, pero no sustractivo, y la tipo *min* soporta sustractivo pero no aditivo. En ambos casos no soportan ruido mixto.

2.4.7 Memorias Asociativas alfa-beta (2002)

Las memorias asociativas alfa-beta se fundamentan en dos operadores binarios: el operador alfa y el operador beta. Estas memorias, al igual que las memorias morfológicas, hacen uso del operador máximo y mínimo en lugar de sumas. Y al igual que las memorias morfológicas, las memorias alfa-beta tienen los mismo problemas.

Las memorias asociativas alfa-beta serán expuestas en el capítulo siguiente de manera detallada, ya que es uno de los temas centrales de este trabajo.

2.5 Autómatas Celulares y Reconocimiento de Patrones

Puesto que el trabajo que se presenta asocia el campo de los autómatas celulares con el de las memorias asociativas y por ende el de el reconocimiento de patrones, es de suma importancia conocer trabajos relacionados con el tema. Enseguida se presentan trabajos encontrados más cercanos al tema de investigación de este trabajo.

Ovaros y Austin presentan una red celular de procesadores asociativos capaz de procesar entradas simbólicas [71]. Los procesadores son colocados en un arreglo bidimensional y a partir de un conjunto de reglas simbólicas, definen el siguiente estado; el sistema que muestran es en realidad un AC y su principal objetivo es transformar descripciones simbólicas de entrada en cada instante de tiempo para posteriormente reconocer de qué objeto se trata ya sea en un ambiente con ruido o en un conjunto de varios objetos.

Maji P., et al., [72, 73, 74, 75, 76, 77] proponen un autómatata celular basado en una memoria asociativa para el reconocimiento de patrones por medio de puntos atractores. Suponga que se consideran dos patrones binarios

a clasificar x_1 y x_2 ; Maji et al., proponen crear árboles invertidos con raíz los patrones a clasificar, los cuales serán los puntos atractores, y como nodos interno y hojas proponen colocar patrones binarios de tal manera que su distancia de Hamming vaya disminuyendo. Consideran un AC híbrido, esto es, un AC donde las reglas que definen al AC no son aplicadas a todas las células, es decir, se aplican diferentes reglas a cada una de las células del sistema. Para decidir que regla aplicar, hacen uso de algoritmos genéticos.

Reynaga y Amthauer presentan en [78] un autómata celular de radio uno para clasificar si una imagen binaria presenta densidad menor, mayor o igual a $1/2$.

Capítulo 3

Materiales y Métodos

3.1 Autómatas Celulares

En esta sección se presenta la parte conceptual referente a los autómatas celulares el cual fue extraído de [59, 79].

Sean A_α una familia contable de intervalos cerrados en \mathbb{R} tal que cumplen las siguientes condiciones:

1. $\bigcup_{X \in A_\alpha} X = [a, b]$ para algún $a, b \in \mathbb{R}$ o $\bigcup_{X \in A_\alpha} X = \mathbb{R}$.
2. Si $[a_i, b_i] \in A_\alpha$ entonces $b_i - a_i > 0$.
3. Si $[a_i, b_i]$ y $[c_j, d_j]$ están en A_α , entonces $[a_i, b_i] \cap [c_j, d_j] = \emptyset$ ó $[a_i, b_i] \cap [c_j, d_j] = [c_j, d_j] = b_i = c_j$.

Definición 3.1.1 Sean $[a, b]$ un intervalo de \mathbb{R} con $a \neq b$ y A_α una familia de intervalos cerrados que cumplen 1, 2 y 3. Una *lattice cuadrada* de dimensión 1 ó *1-dimensional* es el conjunto $\mathcal{L} = \{x_i \times [a, b] \mid x_i \in A_\alpha\}$. Si $A_{\alpha_1}, A_{\alpha_2}, \dots, A_{\alpha_n}$ son familias de intervalos que cumplen 1, 2 y 3, entonces una *lattice cuadrada* de dimensión $n > 1$ es el conjunto $\mathcal{L} = \{x_{\alpha_1} \times x_{\alpha_2} \times \dots \times x_{\alpha_n} \mid x_{\alpha_i} \in A_{\alpha_i}\}$.

Definición 3.1.2 Sea $r \in \mathbb{R}$. Una *lattice cuadrada 1-dimensional* es *regular* si $[a_i, b_i] = r$ para cada $[a_i, b_i] \in A_\alpha$. Una *lattice cuadrada n-dimensional* es *regular* si $[a_{\alpha_i}, b_{\alpha_i}] = r$ para cada $[a_{\alpha_i}, b_{\alpha_i}] \in A_{\alpha_i}$ para $i = 1, \dots, n$.

Definición 3.1.3 Sea \mathcal{L} una lattice cuadrada. Una *célula*, *celda* o *sitio* es un elemento de \mathcal{L} . Esto es, una célula es un elemento de la forma $[a_{\alpha_{1k}}, b_{\alpha_{1k}}] \times \dots \times [a_{\alpha_{nk}}, b_{\alpha_{nk}}]$ con $[a_{\alpha_{ik}}, b_{\alpha_{ik}}] \in A_{\alpha_i}$ para $i = 1, \dots, n$.

Definición 3.1.4 Sea \mathcal{L} una lattice cuadrada, y sea r una célula de \mathcal{L} . Una *vecindad* de tamaño $n \in \mathbb{N}$ para r , es el conjunto $v(r) := \{\{k_1, k_2, \dots, k_n\} | k_j \text{ es célula de } \mathcal{L} \text{ para cada } j\}$.

La siguiente definición muestra lo que es un autómata celular con lattice cuadrada. Durante todo el trabajo se utilizan lattices cuadradas, es por ello, que de aquí en adelante escribiremos lattice en lugar de lattice cuadrada.

Definición 3.1.5 Sea $n \in \mathbb{N}$. Un *Autómata Celular* es una tupla $(\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$ tal que:

1. \mathcal{L} es una *lattice regular*.
2. \mathcal{S} es un *conjunto finito de estados*.
3. \mathcal{N} es un *conjunto de vecindades* definido de la siguiente manera:

$$\mathcal{N} = \{\mathcal{N}(r) \mid r \text{ es una célula y } \mathcal{N}(r) \text{ es una vecindad de } r \text{ de tamaño } n\}$$

4. $f : \mathcal{N} \rightarrow \mathcal{S}$ es una función llamada *función de transición*.

Definición 3.1.6 Una *configuración del autómata celular* $(\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$, es una función $C_t : \mathcal{L} \rightarrow \mathcal{S}$ que asocia a cada célula de la lattice \mathcal{L} (al tiempo t) un estado de \mathcal{S} .

Si $(\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$ es un AC y $r \in \mathcal{L}$, entonces la configuración C_t está relacionada con f mediante

$$C_{t+1}(r) = f(\{C_t(i) \mid i \in \mathcal{N}(r)\})$$

Generalmente la lattice es considerada como un enrejado regular de celdas en \mathbb{Z}^n , donde cada celda es una célula.

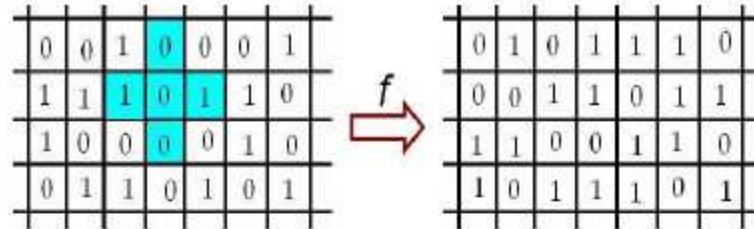


Figura 3.1: Ejemplo de un autómata celular con $S = \{0, 1\}$.

3.2 Morfología Matemática

En esta sección se presenta la parte correspondiente de la morfología matemática, el cual fue tomado de [59, 80].

3.2.1 Dilatación

Comenzaremos el estudio breve de la operación de dilatación. Para ello, consideraremos en todo lo que sigue, conjuntos discretos 2-dimensionales, esto es, subconjuntos de \mathbb{Z}^2 .

Definición 3.2.1 Sea $A \subseteq \mathbb{Z}^2$. Se define el *conjunto reflejado* de A denotado por A^- , mediante:

$$A^- = \{-x \mid x \in A\}$$

Definición 3.2.2 Sean $A \subseteq \mathbb{Z}^2$ y $x \in \mathbb{Z}^2$. La *traslación* de A por x denotada por $(A)_x$ se define así:

$$(A)_x = \{a + x \mid a \in A\}$$

La siguiente definición muestra el concepto formal de lo que es la operación dilatación.

Definición 3.2.3 Sean $A, B \subseteq \mathbb{Z}^2$, La *dilatación* de A por B , denotada por $A \oplus B$, es la *suma de Minkowski* de A y B ; esto es:

$$A \oplus B = \{a + b \mid a \in A \text{ y } b \in B\}$$

Al conjunto B de la definición anterior será llamado *elemento estructurante*.

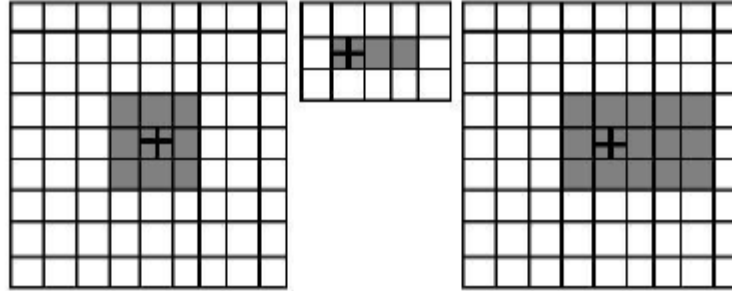


Figura 3.2: Representación del conjunto A , el elemento estructurante $B = \{(0, 0), (1, 0), (2, 0)\}$ y la dilatación $A \oplus B$.

Teorema 3.2.1 Sean $A, B \subseteq \mathbb{Z}^2$. Se cumple que:

$$A \oplus B = \{x | (B^-)_x \cap A \neq \emptyset\}$$

Demostración. $x \in A \oplus B$ sii $x = a + b$ para algún $a \in A$ y $b \in B$ sii $x - b = a$ para algún $a \in A$ y $b \in B$ sii $x - b = a$ para algún $a \in A$ y $-b \in B^-$ sii $a - x = -b$ para algún $a \in A$ y $-b \in B^-$ sii $a \in (B^-)_x$ y $a \in A$ sii $a \in (B^-)_x \cap A$ para algún a sii $(B^-)_x \cap A \neq \emptyset$. ■

El teorema anterior nos permite definir de una forma alternativa lo que es la dilatación de un conjunto A por el elemento estructurante B .

3.2.2 Erosión

Enseguida se muestra el concepto formal de lo que es la erosión.

Definición 3.2.4 Sean $A, B \subseteq \mathbb{Z}^2$. La *erosión* de A por B , denotada por $A \ominus B$, es la *resta de Minkowski* de A por B ; esto es:

$$A \ominus B = \{x \in \mathbb{Z}^2 | x + b \in A \text{ para cada } b \in B\}$$

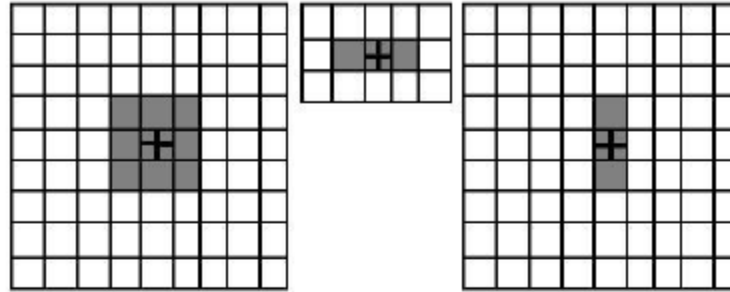


Figura 3.3: Representación del conjunto A , el elemento estructurante $B = \{(-1, 0), (0, 0), (1, 0)\}$ y la erosión $A \ominus B$.

Teorema 3.2.2 Sean $A, B \subseteq \mathbb{Z}^2$. Se cumple que:

$$A \ominus B = \{x | (B)_x \subseteq A\}$$

Demostración. $x \in A \ominus B$ sii $x + b \in A$ para cada $b \in B$ por la definición 3.2.4 sii $(B)_x \subseteq A$ por la definición 3.2.2. ■

El teorema anterior nos permite definir de una forma alternativa lo que es la erosión de un conjunto A por el elemento estructurante B .

3.3 Memorias Morfológicas

Enseguida se presenta el modelo de las memorias morfológicas tal y como aparecen en [15].

3.3.1 Memorias morfológicas autoasociativas (*max* y *min*)

Las memorias morfológicas autoasociativas *max* utilizan el producto mínimo para ambas fases: aprendizaje y recuperación de patrones. Además, en

la fase de aprendizaje juega un papel relevante el operador máximo; de ahí el nombre de memorias *max*. La fase de aprendizaje consiste de dos etapas [15]:

1.- Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se encuentra la matriz $\mathbf{x}^\mu \Delta (-\mathbf{x}^\mu)^t$ de dimensiones $n \times n$, donde $(-\mathbf{x}^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$. Se tiene:

$$\mathbf{x}^\mu \Delta (-\mathbf{x}^\mu)^t = \begin{pmatrix} 0 & x_1^\mu - x_2^\mu & \cdots & x_1^\mu - x_j^\mu & \cdots & x_1^\mu - x_n^\mu \\ x_2^\mu - x_1^\mu & 0 & \cdots & x_2^\mu - x_j^\mu & \cdots & x_2^\mu - x_n^\mu \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_i^\mu - x_1^\mu & x_i^\mu - x_2^\mu & \cdots & 0 & \cdots & x_i^\mu - x_n^\mu \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^\mu - x_1^\mu & x_n^\mu - x_2^\mu & \cdots & x_n^\mu - x_j^\mu & \cdots & 0 \end{pmatrix}$$

2.- Se aplica el operador máximo \bigvee las p matrices para obtener la memoria M .

$$M = \bigvee_{\mu=1}^p [\mathbf{x}^\mu \Delta (-\mathbf{x}^\mu)^t] = [m_{ij}]_{n \times n}$$

de manera que la ij -ésima componente de la memoria M se expresa así:

$$m_{ij} = \bigvee_{\mu=1}^p (x_i^\mu - x_j^\mu)$$

La fase de recuperación consiste en realizar el producto mínimo Δ de la memoria M con un patrón de entrada \mathbf{x}^ω , donde $\omega \in 1, 2, \dots, p$, para obtener un vector columna \mathbf{x} de dimensión n :

$$\mathbf{x} = M \Delta \mathbf{x}^\omega$$

La i -ésima componente del vector \mathbf{x} es:

$$x_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega)$$

Las memorias morfológicas autoasociativas *min* son el dual de las memorias morfológicas autoasociativas *max*, es decir [16]:

$$w_{ij} = \bigwedge_{\mu=1}^p (x_i^\mu - x_j^\mu)$$

$$x_i = \bigvee_{j=1}^n (w_{ij} + x_j^\omega)$$

3.3.2 Memorias morfológicas heteroasociativas (*max* y *min*)

Las memorias morfológicas heteroasociativas *max* utilizan el producto mínimo para ambas fases: aprendizaje y recuperación de patrones. Además, en la fase de aprendizaje juega un papel relevante el operador máximo; de ahí el nombre de memorias *max* [17].

1.- La fase de aprendizaje consiste de dos etapas:

Para cada una de las p asociaciones $(\mathbf{x}^\mu, \mathbf{y}^\mu)$ se encuentra la matriz $\mathbf{y}^\mu \Delta (-\mathbf{x}^\mu)^t$ de dimensiones $m \times n$, donde $(-\mathbf{x}^\mu)^t = (-x_1^\mu, -x_2^\mu, \dots, -x_n^\mu)$. Se tiene:

$$\mathbf{y}^\mu \Delta (-\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu - x_1^\mu & y_1^\mu - x_2^\mu & \cdots & y_1^\mu - x_j^\mu & \cdots & y_1^\mu - x_n^\mu \\ y_2^\mu - x_1^\mu & y_2^\mu - x_2^\mu & \cdots & y_2^\mu - x_j^\mu & \cdots & y_2^\mu - x_n^\mu \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_i^\mu - x_1^\mu & y_i^\mu - x_2^\mu & \cdots & y_i^\mu - x_j^\mu & \cdots & y_i^\mu - x_n^\mu \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_m^\mu - x_1^\mu & y_m^\mu - x_2^\mu & \cdots & y_m^\mu - x_j^\mu & \cdots & y_m^\mu - x_n^\mu \end{pmatrix}$$

2.- Se aplica el operador máximo \bigvee las p matrices para obtener la memoria M .

$$M = \bigvee_{\mu=1}^p [\mathbf{y}^\mu \Delta (-\mathbf{x}^\mu)^t] = [m_{ij}]_{m \times n}$$

de manera que la ij -ésima componente de la memoria M se expresa así:

$$m_{ij} = \bigvee_{\mu=1}^p (y_i^\mu - x_j^\mu)$$

La fase de recuperación consiste en realizar el producto mínimo Δ de la memoria M con un patrón de entrada x^ω , donde $\omega \in \{1, 2, \dots, p\}$, para obtener un vector columna y de dimensión m :

$$y = M \Delta x^\omega$$

La i -ésima componente del vector y es:

$$y_i = \bigwedge_{j=1}^n (m_{ij} + x_j^\omega)$$

Las memorias morfológicas heteroasociativas *min* son el dual de las memorias morfológicas heteroasociativas *max*, es decir:

$$w_{ij} = \bigwedge_{\mu=1}^p (y_i^\mu - x_j^\mu)$$

$$y_i = \bigvee_{j=1}^n (w_{ij} + x_j^\omega)$$

3.4 Memorias Asociativas alfa-beta

En el trabajo de la tesis doctoral del Dr. Cornelio Yáñez Márquez [18] se crearon y desarrollaron las memorias asociativas alfa-beta. Este es un modelo alternativo a las memorias asociativas morfológicas, el cual se fundamenta en bases teóricas ajenas a las operaciones morfológicas de dilatación y erosión. Las nuevas memorias asociativas poseen características similares a las morfológicas en cuanto a capacidad, eficiencia en respuesta e inmunidad al ruido.

Las herramientas matemáticas del nuevo modelo de memorias asociativas alfa-beta incluyen dos operaciones binarias inventadas en la citada tesis, cuyos operadores fueron bautizados arbitrariamente con las dos primeras grafías del alfabeto griego: alfa y beta [81].

Operaciones Binarias α y β

Las memorias asociativas alfa-beta utilizan dos operadores binarios α y β ; dichos operadores trabajan sobre dos conjuntos específicos.

Se definen los conjuntos A y B de la siguiente manera[18]:

$$A = \{0, 1\} \quad \text{y} \quad B = \{0, 1, 2\}$$

A partir de los conjuntos A y B se definen los operadores binarios $\alpha : A \times A \rightarrow B$ y $\beta : B \times A \rightarrow A$ como se muestran en las tablas 3.1 y 3.2 respectivamente.

x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

Tabla 3.1: Operador α

x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Tabla 3.2: Operador β

3.4.1 Memorias Heteroasociativas alfa-beta (*max* y *min*)

En [18] se proponen dos tipos de memorias heteroasociativas alfa-beta: tipo *max* (∇) y tipo *min* (Δ); en este trabajo se desarrollarán sólo las de tipo ∇ ya que las propiedades de las memorias tipo Δ se obtienen por dualidad.

En el diseño de ambos tipos de memoria se utilizan los operadores matriciales Δ_α , ∇_α , Δ_β , ∇_β definidos así:

1. Operación αmax : $P_{m \times r} \nabla_\alpha Q_{r \times n} = [f_{ij}^\alpha]_{m \times n}$, donde

$$f_{ij}^\alpha = \bigvee_{k=1}^r \alpha(p_{ik}, q_{kj})$$

2. Operación βmax : $P_{m \times r} \nabla_\beta Q_{r \times n} = [f_{ij}^\beta]_{m \times n}$, donde

$$f_{ij}^\beta = \bigvee_{k=1}^r \beta(p_{ik}, q_{kj})$$

3. Operación αmin : $P_{m \times r} \Delta_\alpha Q_{r \times n} = [h_{ij}^\alpha]_{m \times n}$, donde

$$h_{ij}^\alpha = \bigwedge_{k=1}^r \alpha(p_{ik}, q_{kj})$$

4. Operación βmin : $P_{m \times r} \Delta_\beta Q_{r \times n} = [h_{ij}^\beta]_{m \times n}$, donde

$$h_{ij}^\beta = \bigwedge_{k=1}^r \beta(p_{ik}, q_{kj})$$

Se tiene $y \Delta_\alpha x^t = y \nabla_\alpha x^t$ para dos vectores cualesquiera $x \in A^n$, $y \in A^m$. Se usará el operador \boxtimes para representar los dos operadores matriciales de forma indistinta.

El operador \boxtimes tiene la siguiente forma, para los índices $\mu \in \{1, 2, \dots, p\}$, $i \in \{1, 2, \dots, m\}$, y $j \in \{1, 2, \dots, n\}$:

$$[y^\mu \boxtimes (x^\mu)^t]_{ij} = \alpha(y_i^\mu, x_j^\mu)$$

Fase de aprendizaje.

1. Para cada $\mu = 1, 2, \dots, p$, a partir de la pareja (x^μ, y^μ) se construye la matriz

$$[y^\mu \boxtimes (x^\mu)^t]_{m \times n} = \begin{bmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_i^\mu \\ \vdots \\ y_m^\mu \end{bmatrix} \boxtimes [x_1^\mu \ x_2^\mu \ \dots \ x_j^\mu \ \dots \ x_n^\mu]$$

$$= \begin{bmatrix} \alpha(y_1^\mu, x_1^\mu) & \alpha(y_1^\mu, x_2^\mu) & \dots & \alpha(y_1^\mu, x_j^\mu) & \dots & \alpha(y_1^\mu, x_n^\mu) \\ \alpha(y_2^\mu, x_1^\mu) & \alpha(y_2^\mu, x_2^\mu) & \dots & \alpha(y_2^\mu, x_j^\mu) & \dots & \alpha(y_2^\mu, x_n^\mu) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \alpha(y_i^\mu, x_1^\mu) & \alpha(y_i^\mu, x_2^\mu) & \dots & \alpha(y_i^\mu, x_j^\mu) & \dots & \alpha(y_i^\mu, x_n^\mu) \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \alpha(y_m^\mu, x_1^\mu) & \alpha(y_m^\mu, x_2^\mu) & \dots & \alpha(y_m^\mu, x_j^\mu) & \dots & \alpha(y_m^\mu, x_n^\mu) \end{bmatrix}$$

2. Se aplica el operador binario *máximo* \bigvee a las matrices obtenidas en el paso 1:

$$V = \bigvee_{\mu=1}^p [y^\mu \boxtimes (x^\mu)^t]$$

La entrada ij -ésima está dada por la siguiente expresión:

$$\nu_{ij} = \bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu)$$

Es posible observar que $\nu_{ij} \in B, \forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}$

Fase de recuperación.

1. Se presenta un patrón \mathbf{x}^ω , con $\omega \in \{1, 2, \dots, p\}$, a la memoria heteroasociativa alfa-beta tipo \mathbf{V} y se realiza la operación Δ_β :

$$\begin{aligned} \mathbf{V} \Delta_\beta \mathbf{x}^\omega &= \begin{bmatrix} \nu_{11} & \nu_{12} & \cdots & \nu_{1j} & \cdots & \nu_{1n} \\ \nu_{21} & \nu_{22} & \cdots & \nu_{2j} & \cdots & \nu_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ \nu_{i1} & \nu_{i2} & \cdots & \nu_{ij} & \cdots & \nu_{in} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ \nu_{m1} & \nu_{m2} & \cdots & \nu_{mj} & \cdots & \nu_{mn} \end{bmatrix} \Delta_\beta \begin{bmatrix} x_1^\omega \\ x_2^\omega \\ \vdots \\ x_j^\omega \\ \vdots \\ x_n^\omega \end{bmatrix} \\ &= \begin{bmatrix} \bigwedge_{j=1}^n \beta(\nu_{1j}, x_j^\omega) \\ \bigwedge_{j=1}^n \beta(\nu_{2j}, x_j^\omega) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{mj}, x_j^\omega) \end{bmatrix} \end{aligned}$$

Dado que las dimensiones de la matriz \mathbf{V} son $m \times n$ y \mathbf{x}^ω es un vector columna de dimensión n , el resultado de la operación anterior debe ser un vector columna de dimensión m , cuya i -ésima componente es:

$$\begin{aligned} (\mathbf{V} \Delta_\beta \mathbf{x}^\omega)_i &= \bigwedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega) \\ &= \bigwedge_{j=1}^n \beta \left\{ \left[\bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \right], x_j^\omega \right\} \end{aligned}$$

2. Se presenta un patrón binario $\tilde{\mathbf{x}}$ (patrón alterado de algún patrón fundamental \mathbf{x}^ω) que es un vector columna de dimensión n , a la memoria

heteroasociativa alfa-beta tipo *max* y se realiza la operación Δ_β :

$$\mathbf{V} \Delta_\beta \tilde{\mathbf{x}} = \begin{bmatrix} \nu_{11} & \nu_{12} & \dots & \nu_{1j} & \dots & \nu_{1n} \\ \nu_{21} & \nu_{22} & \dots & \nu_{2j} & \dots & \nu_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ \nu_{i1} & \nu_{i2} & \dots & \nu_{ij} & \dots & \nu_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \nu_{m1} & \nu_{m2} & \dots & \nu_{mj} & \dots & \nu_{mn} \end{bmatrix} \Delta_\beta \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_j \\ \vdots \\ \tilde{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} \bigwedge_{j=1}^n \beta(\nu_{1j}, \tilde{x}_j) \\ \bigwedge_{j=1}^n \beta(\nu_{2j}, \tilde{x}_j) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{ij}, \tilde{x}_j) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{mj}, \tilde{x}_j) \end{bmatrix}$$

El resultado de la operación anterior es un vector columna de dimensión m , cuya i -ésima componente se expresa de la siguiente manera:

$$(\mathbf{V} \Delta_\beta \tilde{\mathbf{x}})_i = \bigwedge_{j=1}^n \beta(\nu_{ij}, \tilde{x}_j)$$

$$(\mathbf{V} \Delta_\beta \tilde{\mathbf{x}})_i = \bigwedge_{j=1}^n \beta \left\{ \left[\bigvee_{\mu=1}^p \alpha(y_i^\mu, x_j^\mu) \right], \tilde{x}_j \right\}$$

Las memorias heteroasociativas alfa-beta tipo *min* se desarrollan por dualidad, partiendo de los resultados obtenidos para las memorias heteroasociativas alfa-beta tipo *max*. Para ello, se realizan los siguientes cambios:

- Donde haya un operador \bigvee colocar un \bigwedge
- Donde haya un operador \bigwedge colocar un \bigvee
- Usar el operador ∇_β en lugar del operador Δ_β

Mientras que las memorias heteroasociativas alfa-beta tipo *max* tienen cierta inmunidad al ruido aditivo y son sensitivas a ruido sustractivo, con las memorias heteroasociativas alfa-beta tipo *min* sucede precisamente lo contrario: son inmunes a cierta cantidad de ruido sustractivo, pero sensitivas a ruido aditivo. Una pequeña cantidad de ruido aditivo puede tener efectos no deseados en la operación de este tipo de memorias asociativas alfa-beta; sin embargo, las memorias asociativas alfa-beta tipo Λ son capaces de soportar ciertas cantidades de ruido aditivo.

Se observa que en la fase de aprendizaje: dada la pareja de patrones (x^μ, y^μ) , se construye una matriz de tamaño $m \times n$; y esto se realiza para cada uno de los p elementos del conjunto fundamental, luego la fase de aprendizaje tiene orden de complejidad $O(n^3)$. Para la fase de recuperación se opera una matriz de tamaño $m \times n$ con un patrón de tamaño n , luego el orden de complejidad para la fase de recuperación es $O(n^2)$. Por tanto el orden de complejidad de las memorias alfa-beta heteroasociativas es $O(\max(n^3, n^2)) = O(n^3)$.

3.4.2 Memorias Autoasociativas alfa-beta (*max* y *min*)

Si a una memoria heteroasociativa se le impone la condición de que $y^\mu = x^\mu \forall \mu \in \{1, 2, \dots, p\}$, entonces deja de ser heteroasociativa y ahora se le denomina memoria *autoasociativa* [18, 81].

A continuación se enlistan algunas características de las memorias autoasociativas alfa-beta:

1. El conjunto fundamental toma la forma $\{(x^\mu, x^\mu) \mid \mu = 1, 2, \dots, p\}$
2. Los patrones fundamentales de entrada y de salida son de la misma dimensión; denotémosla por n .
3. La memoria es una matriz cuadrada, para ambos tipos, V y Λ . Si $x^\mu \in A^n$, entonces $V = [\nu_{ij}]_{n \times n}$ y $\Lambda = [\lambda_{ij}]_{n \times n}$

Fase de aprendizaje.

1. Para cada $\mu = 1, 2, \dots, p$, a partir de la pareja (x^μ, x^μ) se construye la matriz

$$\begin{aligned}
 [x^\mu \boxtimes (x^\mu)^t]_{n \times n} &= \begin{bmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_i^\mu \\ \vdots \\ x_n^\mu \end{bmatrix} \boxtimes [x_1^\mu \ x_2^\mu \ \dots \ x_j^\mu \ \dots \ x_n^\mu] \\
 &= \begin{bmatrix} \alpha(x_1^\mu, x_1^\mu) & \alpha(x_1^\mu, x_2^\mu) & \dots & \alpha(x_1^\mu, x_j^\mu) & \dots & \alpha(x_1^\mu, x_n^\mu) \\ \alpha(x_2^\mu, x_1^\mu) & \alpha(x_2^\mu, x_2^\mu) & \dots & \alpha(x_2^\mu, x_j^\mu) & \dots & \alpha(x_2^\mu, x_n^\mu) \\ \vdots & \vdots & & \vdots & & \vdots \\ \alpha(x_i^\mu, x_1^\mu) & \alpha(x_i^\mu, x_2^\mu) & \dots & \alpha(x_i^\mu, x_j^\mu) & \dots & \alpha(x_i^\mu, x_n^\mu) \\ \vdots & \vdots & & \vdots & & \vdots \\ \alpha(x_n^\mu, x_1^\mu) & \alpha(x_n^\mu, x_2^\mu) & \dots & \alpha(x_n^\mu, x_j^\mu) & \dots & \alpha(x_n^\mu, x_n^\mu) \end{bmatrix}
 \end{aligned}$$

2. Se aplica el operador binario *máximo* \bigvee a las matrices obtenidas en el paso 1:

$$V = \bigvee_{\mu=1}^p [x^\mu \boxtimes (x^\mu)^t]$$

La entrada ij -ésima de la memoria está dada así:

$$\nu_{ij} = \bigvee_{\mu=1}^p \alpha(x_i^\mu, x_j^\mu)$$

Se tiene que $\nu_{ij} \in B, \forall i \in \{1, 2, \dots, n\}, \forall j \in \{1, 2, \dots, n\}$

Fase de recuperación.

1. Se presenta un patrón \mathbf{x}^ω , con $\omega \in \{1, 2, \dots, p\}$, a la memoria autoasociativa $\alpha\beta$ tipo *max* y se realiza la operación β :

$$\mathbf{V} \Delta_\beta \mathbf{x}^\omega = \begin{bmatrix} \nu_{11} & \nu_{12} & \dots & \nu_{1j} & \dots & \nu_{1n} \\ \nu_{21} & \nu_{22} & \dots & \nu_{2j} & \dots & \nu_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ \nu_{i1} & \nu_{i2} & \dots & \nu_{ij} & \dots & \nu_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \nu_{n1} & \nu_{n2} & \dots & \nu_{nj} & \dots & \nu_{nn} \end{bmatrix} \Delta_\beta \begin{bmatrix} x_1^\omega \\ x_2^\omega \\ \vdots \\ x_j^\omega \\ \vdots \\ x_n^\omega \end{bmatrix}$$

$$= \begin{bmatrix} \bigwedge_{j=1}^n \beta(\nu_{1j}, x_j^\omega) \\ \bigwedge_{j=1}^n \beta(\nu_{2j}, x_j^\omega) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{nj}, x_j^\omega) \end{bmatrix}$$

El resultado de la operación anterior será un vector columna de dimensión n .

$$(\mathbf{V} \Delta_\beta \mathbf{x}^\omega)_i = \bigwedge_{j=1}^n \beta(\nu_{ij}, x_j^\omega)$$

$$(\mathbf{V} \Delta_\beta \mathbf{x}^\omega)_i = \bigwedge_{j=1}^n \beta \left\{ \left[\bigvee_{\mu=1}^p \alpha(x_i^\mu, x_j^\mu) \right], x_j^\omega \right\}$$

2. Se presenta un patrón binario $\tilde{\mathbf{x}}$ que es un vector columna de dimensión n , a la memoria autoasociativa alfa-beta tipo *max* y se realiza la ope-

ración Δ_β :

$$\mathbf{V} \Delta_\beta \tilde{\mathbf{x}} = \begin{bmatrix} \nu_{11} & \nu_{12} & \dots & \nu_{1j} & \dots & \nu_{1n} \\ \nu_{21} & \nu_{22} & \dots & \nu_{2j} & \dots & \nu_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ \nu_{i1} & \nu_{i2} & \dots & \nu_{ij} & \dots & \nu_{in} \\ \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \nu_{n1} & \nu_{n2} & \dots & \nu_{nj} & \dots & \nu_{nn} \end{bmatrix} \Delta_\beta \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_j \\ \vdots \\ \tilde{x}_n \end{bmatrix}$$

$$= \begin{bmatrix} \bigwedge_{j=1}^n \beta(\nu_{1j}, \tilde{x}_j) \\ \bigwedge_{j=1}^n \beta(\nu_{2j}, \tilde{x}_j) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{ij}, \tilde{x}_j) \\ \vdots \\ \bigwedge_{j=1}^n \beta(\nu_{nj}, \tilde{x}_j) \end{bmatrix}$$

Al igual que en el caso 1, el resultado de la operación anterior es un vector columna de dimensión n , cuya i -ésima componente se expresa de la siguiente manera:

$$(\mathbf{V} \Delta_\beta \tilde{\mathbf{x}})_i = \bigwedge_{j=1}^n \beta(\nu_{ij}, \tilde{x}_j)$$

$$(\mathbf{V} \Delta_\beta \tilde{\mathbf{x}})_i = \bigwedge_{j=1}^n \beta \left\{ \left[\bigvee_{\mu=1}^p \alpha(x_i^\mu, x_j^\mu) \right], \tilde{x}_j \right\}$$

Las memorias autoasociativas alfa-beta tipo *min* se desarrollan por dualidad, partiendo de los resultados obtenidos para las memorias autoasociativas alfa-beta tipo *max*; para ello, se realizan cambios similares a los que se indicaron para las memorias heteroasociativas. También, mientras que las memorias autoasociativas alfa-beta tipo *max* son inmunes a cierta cantidad de ruido aditivo pero sensibles a ruido sustractivo, con las memorias autoasociativas alfa-beta tipo *min* sucede lo contrario.

Se observa que en la fase de aprendizaje: dada la pareja de patrones $(\mathbf{x}^\mu, \mathbf{x}^\mu)$, se construye una matriz de tamaño $n \times n$; y esto se realiza para cada uno de los p elementos del conjunto fundamental, luego la fase de aprendizaje tiene orden de complejidad $O(n^3)$. Para la fase de recuperación se opera una matriz de tamaño $n \times n$ con un patrón de tamaño n , luego el orden de complejidad para la fase de recuperación es $O(n^2)$. Por tanto el orden de complejidad de las memorias alfa-beta autoasociativas es $O(\max(n^3, n^2)) = O(n^3)$.

3.5 Normas y Distancias

En esta sección se dan los conceptos de norma y distancia los cuales fueron extraídos de [82]. Estos conocimientos serán necesarios para la fundamentación del autómata celular alfa-beta. En lo que sigue consideremos M un subespacio vectorial de \mathbb{R}^n

Definición 3.5.1 Una *norma vectorial* en M es una función $\|\bullet\|$ de M a \mathbb{R} con las siguientes propiedades:

- i) $\|\mathbf{x}\| \geq 0$ para cada $\mathbf{x} \in M$
- ii) $\|\mathbf{x}\| = 0$ si y sólo si $\mathbf{x} = \mathbf{0}$,
- iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ para cada $\alpha \in \mathbb{R}$ y $\mathbf{x} \in M$
- iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ para cada $\mathbf{x}, \mathbf{y} \in M$

Ejemplos de normas, son la norma l_2 y l_∞ . Estas se definen de la siguiente

manera. Dado $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in M$, la norma l_2 se definen como

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n x_i^2 \right)^{1/2}$$

y la norma l_∞ se define como

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

La norma l_2 es llamada *norma euclidea* y la norma l_∞ es llamada *norma infinita*.

Definición 3.5.2 Una *distancia* o *métrica* sobre M es una función binaria $d : M \times M \rightarrow \mathbb{R}$ que satisface:

- i) $d(\mathbf{x}, \mathbf{y}) \geq 0$ para cada $\mathbf{x}, \mathbf{y} \in M$
- ii) $d(\mathbf{x}, \mathbf{y}) = 0$ si y sólo si $\mathbf{x} = \mathbf{y}$
- ii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$ para cada $\mathbf{x}, \mathbf{y} \in M$
- iii) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ para cada $\mathbf{x}, \mathbf{y}, \mathbf{z} \in M$

Dada una norma $\|\bullet\|$ sobre M , se induce una distancia sobre M dada por

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$$

Así, dada la norma euclidea, esta induce la distancia euclidea dada por

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

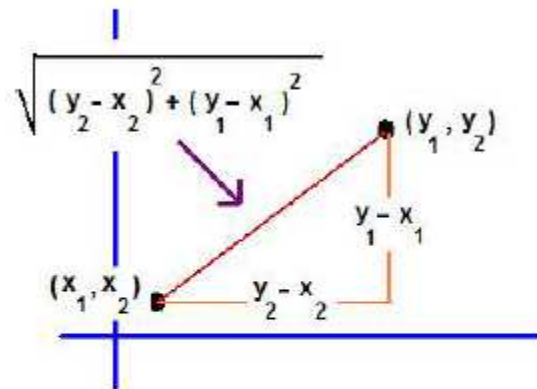


Figura 3.4: Distancia euclídeana

3.6 Partición sobre un Conjunto

Otra definición de importancia para fundamentar el concepto del autómata celular alfa-beta es el significado de una partición para un conjunto. Para ello, primero mostramos la definición de lo que es una familia de conjuntos. Los siguientes conceptos fueron tomados de [83].

Definición 3.6.1 Sean I y U dos conjuntos. Una *familia de conjuntos* de U con índices en I , es una función $f : I \rightarrow P(U)$, con $P(U)$ el conjunto potencia de U . La familia de conjuntos se representa mediante

$$\{A_i : i \in I\} \text{ o } \{A_i\}_{i \in I}$$

donde $A_i = f(i)$ para cada $i \in I$.

Definición 3.6.2 Sea P un conjunto. Se dice que la familia de conjuntos $\{A_i\}_{i \in I}$ es una *partición* sobre P si

- i) $A_i \neq \emptyset$ para cada $i \in I$
- ii) $\bigcup_{i \in I} A_i = P$
- iii) $A_i \cap A_j = \emptyset$ para cada $i, j \in I, i \neq j$

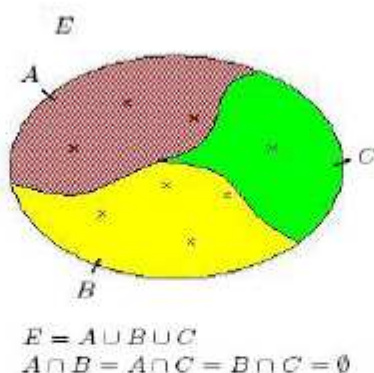


Figura 3.5: Ejemplo de la partición de un conjunto

3.7 Orden de Complejidad

Una de las principales medidas sobre la aplicación de algoritmos, es la *complejidad algorítmica*. Enseguida se muestra el concepto del orden de complejidad, el cual fue extraído de [84].

Definición 3.7.1 Sean $f(n)$ y $g(n)$ dos funciones en \mathbb{R} . Se dice que $f(n)$ está en el orden de $g(n)$, o que $f(n)$ es $O(g(n))$, si existen enteros positivos a y b tales que $f(n) \leq ag(n)$ para cada $n \geq b$.

Ejemplo 3.7.1 La función $f(n) = n^2 + 100n$ es $O(n^2)$.

Se tiene que $f(n) = n^2 + 100n \leq n^2 + 100n^2 = (100 + 1)n^2 = 101n^2 = 101g(n)$ para $n \geq 1$, luego tomando $a=101$ y $b = 1$, se tiene que $f(n) \leq ag(n)$ para cada $n \geq b$. Por lo tanto $f(n)$ es $O(g(n))$. ■

El siguiente teorema es de suma importancia para el análisis de la complejidad de algoritmos.

Teorema 3.7.1 Sean $f(n)$ y $g(n)$ dos funciones en \mathbb{R} . Si $f(n)$ es $O(h_1(n))$ y $g(n)$ es $O(h_2(n))$, entonces:

$$f(n) + g(n) \text{ es } O(\max(h_1(n), h_2(n)))$$

$$\text{y } g(n) * f(n) \text{ es } O(h_1(n)h_2(n)).$$

Capítulo 4

Modelo Propuesto

4.1 Morfología Matemática Celular

Este capítulo muestra los modelos propuestos, parte principal de este trabajo de tesis. Comenzamos presentando las operaciones fundamentales de la morfología matemática desarrolladas mediante autómatas celulares (AC).

4.1.1 Dilatación Celular

Enseguida se presentará un AC que exactamente opera sobre conjuntos tal y como lo hace la dilatación, después se demostrará que dicho AC y la definición de dilatación en el contexto clásico son equivalentes.

Nuestro objetivo es mostrar un AC que opere sobre conjuntos discretos exactamente igual a la dilatación y probar que dicho AC es equivalente a la definición 3.2.3.

Definición 4.1.1 Sea $\mathcal{R} = (\mathcal{L}, S, \mathcal{N}, f)$ un AC con $\mathcal{L} = \mathbb{Z}^2$. Si $A \subseteq \mathbb{Z}^2$ y $x \in S$, entonces $[A]_x$ denotará el número de elementos x que contiene A .

Sean $A, B \subseteq \mathbb{Z}^2$. En lo que sigue, $\mathcal{D} = (\mathcal{L}, S, \mathcal{N}, f)$ es un AC con configuración inicial A definido como sigue:

- $\mathcal{L} = \mathbb{Z}^2$.
- $S = \{0, 1\}$.

- $\mathcal{N} = \{v_x | x \in \mathcal{L}\}$ con $v_x = (B^-)_x$.
- La función de transición $f : \mathcal{N} \rightarrow S$ está dada como sigue:

$$f(v_x) = \begin{cases} 1 & \text{si } [v_x]_1 > 0 \\ 0 & \text{si } [v_x]_1 = 0 \end{cases}$$

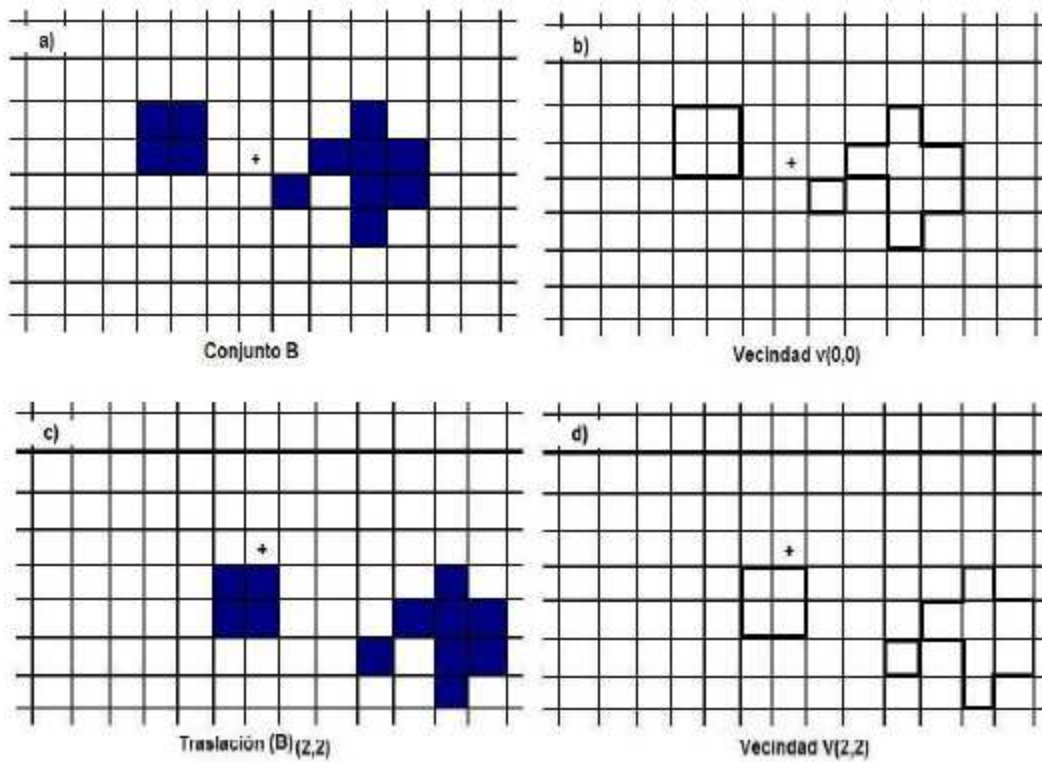


Figura 4.1: Muestra la diferencia conceptual entre el elemento estructural y la vecindad de una célula. En a) se muestra el elemento estructural B , en b) la vecindad $v_{(0,0)}$, en c) el conjunto $(B)_{(2,2)}$ y en d) la vecindad $v_{(2,2)}$

Observación. En la definición anterior, dada una célula x , su vecindad v_x está dada por la traslación del conjunto B por x . Obsérvese que cada

célula del conjunto B tiene valor 1, y cada traslación también; sin embargo v_x solamente considera la forma del conjunto $(B)_x$, en este caso, puede haber células con valor 1 o valor 0 (figura 4.1).

Teorema 4.1.1 El AC \mathcal{D} con configuración inicial A es equivalente a dilatar el conjunto A por B en la primera iteración.

Demostración. Si $x \in A \oplus B \subseteq \mathcal{L}$, entonces $(B^-)_x \cap A \neq \emptyset$ por el teorema 3.2.1. Por tanto, existe $y \in \mathcal{L}$ tal que $y \in (B^-)_x \cap A$, puesto que estamos manejando imágenes binarias, esto significa que y corresponde a un valor 1 tal que $y \in v_x$, por tanto $[v_x]_1 > 0$, luego $f(x) = 1$. Si $x \notin A \oplus B$, entonces $(B^-)_x \cap A = \emptyset$, puesto que estamos manejando imágenes binarias, esto significa que la vecindad v_x esta formada únicamente por células con valores igual a cero, luego $[v_x]_1 = 0$, y por tanto $f(x) = 0$. Recíprocamente, si $x \in \mathcal{L}$ con $f(x) = 1$, entonces $[v_x]_1 > 0$, luego, existe una célula con valor 1 que se encuentra en la vecindad v_x , es decir, existe $y \in \mathcal{L}$ tal que $y \in v_x$ y $y \in A$, pues A es la configuración inicial del AC, por tanto, existe $y \in (B^-)_x$ y $y \in A$, luego $y \in (B^-)_x \cap A$, se sigue del teorema 3.2.1 que $x \in A \oplus B$. Si $x \in \mathcal{L}$ con $f(x) = 0$, entonces $[v_x]_1 = 0$, esto significa que la vecindad v_x esta formada únicamente por células con valores igual a cero, es decir, $v_x \cap A = \emptyset$, entonces $(B^-)_x \cap A = \emptyset$, por tanto $x \notin A \oplus B$. ■

4.1.2 Erosión Celular

Enseguida se mostrará un AC que exactamente opera sobre conjuntos tal y como lo hace la erosión, después se demostrará que dicho AC y la definición de erosión en el contexto clásico son equivalentes.

Ahora, nuestro objetivo es mostrar un AC que opere sobre conjuntos discretos exactamente igual a la erosión y probar que dicho AC es equivalente a la definición 3.2.4.

Sean $A, B \subseteq \mathbb{Z}^2$. En lo que sigue, $\mathcal{E} = (\mathcal{L}, S, \mathcal{N}, f)$ es un AC con configuración inicial A definido como sigue:

- $\mathcal{L} = \mathbb{Z}^2$.

- $S = \{0, 1\}$.
- $\mathcal{N} = \{v_x | x \in \mathcal{L}\}$ con $v_x = (B)_x$.
- La función de transición $f : \mathcal{N} \rightarrow S$ está dada como sigue:

$$f(v_x) = \begin{cases} 1 & \text{si } [v_x]_1 = |B| \\ 0 & \text{si } [v_x]_1 < |B| \end{cases}$$

Teorema 4.1.2 El AC \mathcal{E} con configuración inicial A es equivalente a erosionar el conjunto A por B en la primera iteración.

Demostración. Si $x \in A \ominus B$, entonces $(B)_x \subseteq A$ por el teorema 3.2.2, puesto que $(B)_x$ es el conjunto B trasladado por x , se tiene que $(B)_x$ tiene tantas células con valores 1 como B , por tanto, $[v_x]_1 = |B|$, por tanto $f(x) = 1$. Si $x \notin A \ominus B$, entonces $(B)_x \not\subseteq A$, es decir, existe $y \in (B)_x \subseteq \mathcal{L}$ tal que $y \notin A$, esto significa que existe una célula en $(B)_x$ con valor 1 cuyo valor es 0 en v_x , luego $[v_x]_1 < |B|$, por tanto $f(v_x) = 0$. Recíprocamente, si $x \in \mathcal{L}$ es una célula con $f(v_x) = 1$, entonces $[v_x]_1 = |B|$, luego $(B)_x \subseteq A$. Si $x \in \mathcal{L}$ es una célula con $f(v_x) = 0$, entonces $[v_x]_1 < |B|$, esto significa que existe una célula en v_x con valor 0 y por otro lado, esta célula tiene valor 1 en B_x , por tanto, existe $y \in (B)_x$ tal que $y \notin A$, por tanto $(B)_x \not\subseteq A$. ■

En un gran porcentaje de procesos de análisis de conjuntos (imágenes binarias), a pesar de que la dilatación y la erosión son operaciones básicas independientes, dentro de la morfología matemática se usan por parejas alternadas; es decir, normalmente se realizan procesos iterativos del tipo: erosión seguida de una dilatación o dilatación seguida de una erosión. El primero de estos procesos es conocido como *apertura* y el segundo como *cerradura*.

Lo que enseguida se presentará es mostrar los conceptos de la apertura y la cerradura en términos de los AC. Para ello, primeramente se definirá el concepto del AC *composición* de dos AC.

Definición 4.1.2 Sean $\mathcal{Q} = (\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$ y $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{N}', g)$ dos AC. Se define el AC *composición* de los AC \mathcal{Q} y \mathcal{W} en el tiempo $t = t_0$ denotada como $\mathcal{W} * \mathcal{Q}$ mediante el AC $\mathcal{W} * \mathcal{Q} = (\mathcal{L}, \mathcal{S}, \mathcal{N}, h)$ donde h , f y g están relacionadas de la siguiente manera:

$$C_{t_0+1}(r) = f(\{C_{t_0}(i) : i \in \mathcal{N}(r)\})$$

$$C_{t_0+2}(r) = g(\{C_{t_0+1}(i) : i \in \mathcal{N}'(r)\})$$

$$C_{t_0+2}(r) = h(\{C_{t_0}(i) : i \in \mathcal{N}(r)\})$$

La definición anterior muestra el concepto de lo que es la composición de dos autómatas celulares, esto es, dado dos autómatas celulares $\mathcal{Q} = (\mathcal{L}, \mathcal{S}, \mathcal{N}, f)$ y $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{N}', g)$, la composición de estos autómatas celulares equivale a realizar la composición de las funciones de transición; sin embargo, para ello, obsérvese que la función de transición f opera sobre vecindades definidas en \mathcal{N} y g opera sobre vecindades definidas en \mathcal{N}' . Dicho de otra manera, dado el AC \mathcal{Q} , primeramente se aplica la función de transición f a la configuración de la lattice \mathcal{L} obteniendo otra configuración de la lattice, a esta última configuración se le aplica la función de transición g del AC $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{N}', g)$, obteniendo finalmente la composición de los autómatas celulares.

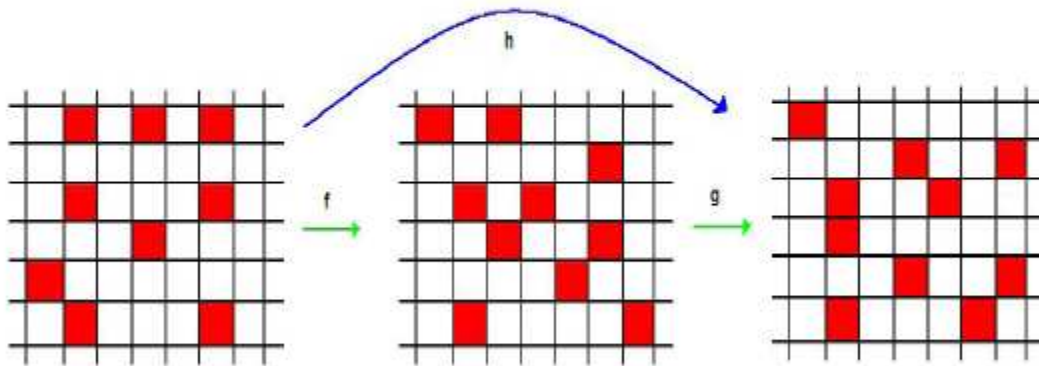


Figura 4.2: Ejemplo de la composición de AC

4.1.3 Apertura Morfológica y Apertura Celular

Enseguida se presentará una operación que se desprende de la dilatación y la erosión. La operación morfológica apertura consiste en aplicar a un conjunto dado, la operación erosión seguida de la operación dilatación. Se mostrará que la definición anterior es equivalente al de aplicar el AC $\mathcal{E} * \mathcal{D}$ al conjunto dado.

Definición 4.1.3 La *apertura* del conjunto A por el elemento estructurante B se denota como $A \circ B$ y se define así:

$$A \circ B = (A \ominus B) \oplus B$$

Teorema 4.1.3 El AC $\mathcal{E} * \mathcal{D}$ al tiempo $t = 0$ con configuración inicial A es equivalente a la apertura; esto es:

$$\mathcal{E} * \mathcal{D} = A \circ B.$$

Demostración. Se sigue de los teoremas 4.1.1 y 4.1.2. ■

4.1.4 Cerradura Morfológica y Cerradura Celular

Enseguida se presentará una operación más que se desprende de la dilatación y la erosión. La operación morfológica cerradura consiste en aplicar a un conjunto dado, la operación dilatación seguida de la operación erosión. Se mostrará que la definición anterior es equivalente al de aplicar el AC $\mathcal{D} * \mathcal{E}$ al conjunto dado.

Definición 4.1.4 La *cerradura* del conjunto A por el elemento estructurante B se denota por $A \bullet B$ y se define así:

$$A \bullet B = (A \oplus B) \ominus B$$

Teorema 4.1.4 El AC $\mathcal{D} * \mathcal{E}$ al tiempo $t = 0$ con configuración inicial A es equivalente a la cerradura; esto es:

$$\mathcal{D} * \mathcal{E} = A \bullet B.$$

Demostración. Se sigue de los teoremas 4.1.1 y 4.1.2. ■

4.2 Modelos Celulares alfa-beta

4.2.1 Operadores alfa-beta

En la sección 3.4 se definieron los operadores alfa y beta. Para ello se tomaron dos conjuntos $A = \{0, 1\}$ y $B = \{0, 1, 2\}$ de tal manera que los operadores binarios alfa y beta se definen de la siguiente manera respectivamente

$\alpha : A \times A \rightarrow B$		
x	y	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

y

$\beta : B \times A \rightarrow A$		
x	y	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

Lo siguiente es mostrar los operadores alfa-beta pero como autómatas celulares, para ello, es imprescindible fundamentar los operadores alfa-beta y de ahí partir para mostrarlos mediante autómatas celulares.

Consideremos el espacio discreto en \mathbb{Z}^2 , esto es, el espacio compuesto por valores enteros únicamente: En la figura 8 y 9 se muestran dos maneras de representar el espacio discreto de las que haremos uso. En la figura 8 a) se muestra el espacio discreto de los valores enteros de forma puntual, mientras que la figura 8 b) se muestran los valores enteros a manera de rejillas o

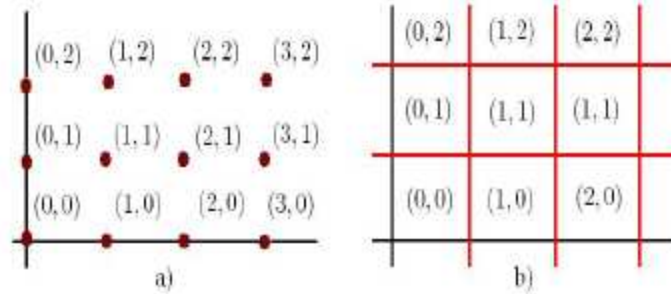


Figura 4.3: Representaciones del espacio discreto

células como se trabajan los autómatas celulares. Ambas representaciones son equivalentes.

Consideremos la representación del espacio discreto como en la figura 4.3 a). Sea

$$d_{cb} : \mathbb{Z}^2 \rightarrow \mathbb{Z} \subset \mathbb{R}$$

definida como sigue: si $\mathbf{x}^i = (x_1^i, x_2^i)$, $\mathbf{x}^j = (x_1^j, x_2^j) \in \mathbb{Z}^2$, entonces $d_{cb}(\mathbf{x}^i, \mathbf{x}^j) = |x_1^i - x_1^j| + |x_2^i - x_2^j|$. d_{cb} es una distancia sobre \mathbb{Z}^2 . En efecto:

Dados $\mathbf{x}^i = (x_1^i, x_2^i)$, $\mathbf{x}^j = (x_1^j, x_2^j) \in \mathbb{Z}^2$, se tiene:

1. $d_{cb}(\mathbf{x}^i, \mathbf{x}^j) = |x_1^i - x_1^j| + |x_2^i - x_2^j| \geq 0$.
2. $d_{cb}(\mathbf{x}^i, \mathbf{x}^j) = 0$ sí y sólo sí $|x_1^i - x_1^j| + |x_2^i - x_2^j| = 0$ sí y sólo sí $|x_1^i - x_1^j| = 0$ y $|x_2^i - x_2^j| = 0$ sí y sólo sí $x_1^i = x_1^j$ y $x_2^i = x_2^j$ sí y sólo sí $\mathbf{x}^i = \mathbf{x}^j$.
3. $d_{cb}(\mathbf{x}^i, \mathbf{x}^j) = |x_1^i - x_1^j| + |x_2^i - x_2^j| = |x_1^j - x_1^i| + |x_2^j - x_2^i| = d_{cb}(\mathbf{x}^j, \mathbf{x}^i)$.
4. Dado $\mathbf{z} = (z_1, z_2) \in \mathbb{Z}^2$, se tiene $d_{cb}(\mathbf{x}^i, \mathbf{x}^j) = |x_1^i - x_1^j| + |x_2^i - x_2^j| = |x_1^i - z_1 + z_1 - x_1^j| + |x_2^i - z_2 + z_2 - x_2^j| \leq |x_1^i - z_1| + |z_1 - x_1^j| + |x_2^i - z_2| + |z_2 - x_2^j| = |x_1^i - z_1| + |x_2^i - z_2| + |z_1 - x_1^j| + |z_2 - x_2^j| = d_{cb}(\mathbf{x}^i, \mathbf{z}) + d_{cb}(\mathbf{z}, \mathbf{x}^j)$.

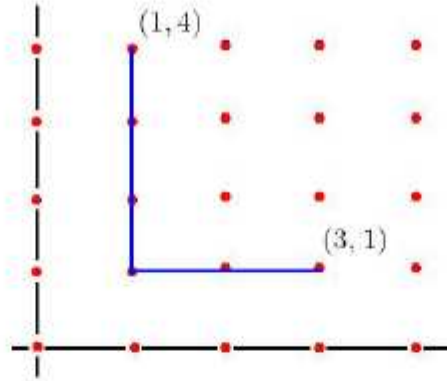


Figura 4.4: Distancia city-block, $d_{cb}((1, 4), (3, 1)) = 5$

■

A la distancia d_{cb} definida anteriormente se le conoce como distancia *city-block*. En la figura 4.4 se muestra la aplicación de esta distancia al espacio discreto.

Definición 4.2.1 Consideremos el espacio discreto. Sea $x \in \mathbb{Z}^2$, se define la *función de anclaje* sobre el punto x denotada por Anc_x como

$$\begin{aligned} Anc_x : \mathbb{Z}^2 &\rightarrow \mathbb{Z} \\ Anc_x(y) &= d_{cb}(x, y) \end{aligned}$$

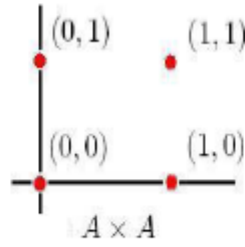
Por ejemplo, si $x = (1, 4)$, entonces $Anc_x(3, 1) = d_{cb}(x, (3, 1)) = d_{cb}((1, 4), (3, 1)) = |1 - 3| + |4 - 1| = 5$.

En lo que sigue consideremos el espacio discreto restringido al conjunto $A \times A$ como se muestra en la figura 4.5.

El siguiente teorema es de suma importancia para el desarrollo de las memorias asociativas ya que dan el sustento del operador alfa.

Teorema 4.2.1 Sea $x \in A \times A$, se tiene

$$\alpha(x) = Anc_{(0,1)}(x)$$

Figura 4.5: Espacio discreto restringido a $A \times A$.

Demostración Sea $\mathbf{x} = (x_1, x_2) \in A \times A$. Se tienen cuatro posibles casos para \mathbf{x} que se muestran enseguida:

1. $\mathbf{x} = (0, 0)$. Se tiene $Anc_{(0,1)}(\mathbf{x}) = Anc_{(0,1)}(0, 0) = dc((0, 1), (0, 0)) = 1 = \alpha(0, 0)$
2. $\mathbf{x} = (0, 1)$. Se tiene $Anc_{(0,1)}(\mathbf{x}) = Anc_{(0,1)}(0, 1) = dc((0, 1), (0, 1)) = 0 = \alpha(0, 1)$
3. $\mathbf{x} = (1, 0)$. Se tiene $Anc_{(0,1)}(\mathbf{x}) = Anc_{(0,1)}(1, 0) = dc((0, 1), (1, 0)) = 2 = \alpha(1, 0)$
4. $\mathbf{x} = (1, 1)$. Se tiene $Anc_{(0,1)}(\mathbf{x}) = Anc_{(0,1)}(1, 1) = dc((0, 1), (1, 1)) = 1 = \alpha(1, 1)$

■

La figura 4.6 muestra la relación existente entre la función alfa y la función de anclaje sobre $\mathbf{x} = (0, 1)$.

Ahora, nos proponemos a sustentar el operador beta. Para ello, consideremos lo siguiente. Sean $\mathcal{C}_0 = \{(x, y) \in B \times A \mid \alpha(0, y) = x\}$ y $\mathcal{C}_1 = \{(x, y) \in B \times A \mid \alpha(1, y) = x\}$. Esto es, $\mathcal{C}_0 = \{(1, 0), (0, 1)\}$ y $\mathcal{C}_1 = \{(2, 0), (1, 1)\}$. Consideremos las clases \mathcal{C}_1 y \mathcal{C}_2 (figura 4.7).

Teorema 4.2.2 Consideremos el clasificador 1-NN utilizando la distancia city-block. Sea $\mathbf{x} \in B \times A$. Sea $f : B \times A \rightarrow A$ definida como $f(\mathbf{x}) = 0$ si el clasificador 1-NN clasifica al patrón \mathbf{x} en la clase \mathcal{C}_0 y $f(\mathbf{x}) = 1$ si el clasificador 1-NN clasifica al patrón \mathbf{x} en la clase \mathcal{C}_1 . Entonces

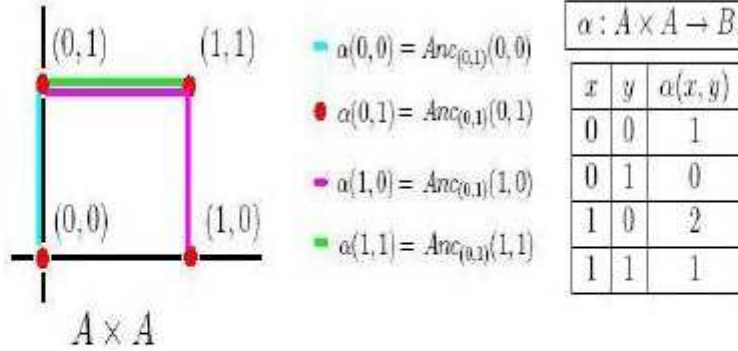


Figura 4.6: Relación entre operador alfa y la función de anclaje sobre $x = (0, 1)$.

$$\beta(x) = f(x)$$

Demostración Dada las clases \mathcal{C}_0 y \mathcal{C}_1 definidas anteriormente, consideremos el clasificador k -NN con $k = 1$, y la distancia que se empleará en este caso, es la distancia city-block. Sea $x \in B \times A$. Si $x = (0, 1)$ o $x = (1, 0)$, el clasificador clasifica a x en la clase \mathcal{C}_0 y por tanto $f(x) = 0 = \beta(x)$. Si $x = (1, 1)$ o $x = (2, 0)$, el clasificador clasifica a x en la clase \mathcal{C}_1 y por tanto $f(x) = 1 = \beta(x)$. Si $x = (0, 0)$ el clasificador clasifica al patrón x en la clase \mathcal{C}_0 ya que $\min\{Anc_x(y) | y \in \mathcal{C}_1 \cup \mathcal{C}_2\} = 1$ y se obtiene con $y = (0, 1)$ y con $y = (1, 0)$ los cuales pertenecen a la clase \mathcal{C}_0 , por tanto $f(x) = 0 = \beta(x)$. De manera similar, si $x = (2, 1)$ el clasificador clasifica al patrón x en la clase \mathcal{C}_1 ya que $\min\{Anc_x(y) | y \in \mathcal{C}_1 \cup \mathcal{C}_2\} = 1$ y se obtiene con $y = (1, 1)$ y con $y = (2, 0)$, los cuales pertenecen a la clase \mathcal{C}_1 , por tanto $f(x) = 1 = \beta(x)$ (figura 4.8). ■

El teorema 4.2.2 es de suma importancia para el desarrollo de las memorias asociativas alfa-beta, ya que muestra el sustento teórico del operador β .

La siguiente sección muestra la construcción de las memorias asociativas alfa-beta como AC.

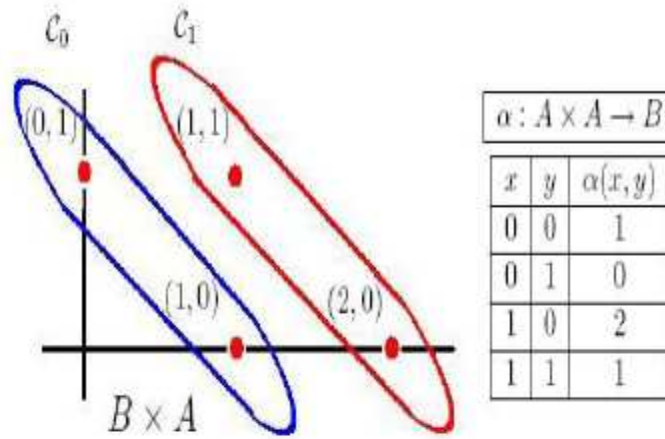


Figura 4.7: Clases $C_0 = \{(x, y) \in B \times A \mid \alpha(0, y) = x\} = \{(1, 0), (0, 1)\}$ y $C_1 = \{(x, y) \in B \times A \mid \alpha(1, y) = x\} = \{(2, 0), (1, 1)\}$.

4.2.2 Autómata Celular alfa-beta

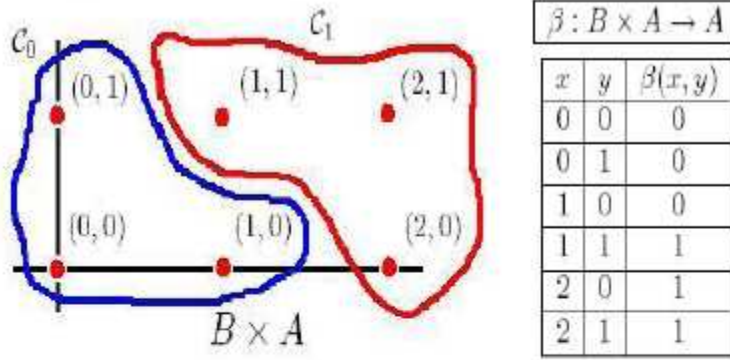
En esta sección se construirán las memorias asociativas alfa-beta mediante AC. Para ello, consideremos el espacio discreto como en la figura 4.3 b).

En lo que sigue, consideremos el conjunto $A = \{0, 1\}$ y el conjunto fundamental $\{(x^\mu, y^\mu) \mid \mu = 1, 2, \dots, p\}$ con $x^\mu \in A^n$ y $y^\mu \in A^m$. Para construir el AC alfa-beta, primeramente comenzaremos definiendo la lattice sobre la que vamos a trabajar.

La lattice \mathcal{L} para el AC alfa-beta estará compuesta por la matriz de tamaño $2m \times 2n$. Dicha lattice la trabajaremos a partir de la posición $(0, 0)$ como se muestra en la figura 4.9.

Consideremos $\mathcal{S} = \{0, 1\}$ como el conjunto finito de estados del AC alfa-beta. Representaremos al valor 0 mediante el color blanco y al valor 1, mediante el color verde (figura 4.10).

Sean $I = \{i \in \mathbb{Z} \mid i = 2k \text{ para algún } k = 0, 1, 2, \dots, n-1\} = \{0, 2, 4, \dots, 2(n-2)\}$ y $J = \{j \in \mathbb{Z} \mid j = 2k+1 \text{ para algún } k = 0, 1, 2, \dots, m-1\} = \{1, 3, 5, \dots, 2m-1\}$. Antes de definir el conjunto de vecindades sobre el que trabajará el AC alfa-beta, consideremos la siguiente proposición.

Figura 4.8: Clasificación de los patrones $(0, 0)$ y $(2, 1)$.

Proposición 4.2.1 La familia de subconjuntos $IJ = \{v_{(i,j)} | (i,j) \in I \times J\}$ con $v_{(i,j)} = \{(i,j), (i,j-1), (i+1,j), (i+1,j-1)\}$ es una partición de \mathcal{L} .

Demostración

1. Dado $v_{(i,j)} \in IJ$, se tiene que $(i,j) \in I \times J$ y $(i,j) \in v_{(i,j)}$, de donde $v_{(i,j)} \neq \emptyset$.

2. Sea $(i', j') \in \mathcal{L}$.

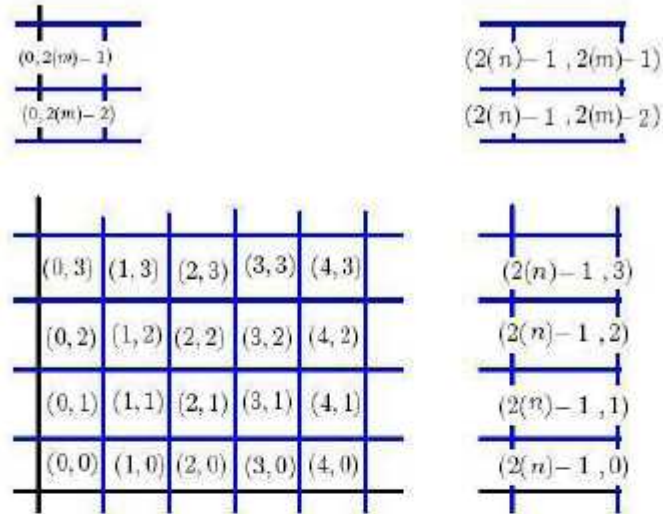
i) Si $(i', j') \in I \times J$, entonces $(i', j') \in v_{(i', j')}$, luego $(i', j') \in \bigcup_{(i,j) \in I \times J} v_{(i,j)}$.

ii) Si $i' \in I$ pero $j' \notin J$, entonces $j' + 1 \in J$ y $(i', j') \in v_{(i', j'+1)}$, luego $(i', j') \in \bigcup_{(i,j) \in I \times J} v_{(i,j)}$.

iii) Si $i' \notin I$ y $j' \in J$, entonces $i' - 1 \in I$ y $(i', j') \in v_{(i'-1, j')}$, luego $(i', j') \in \bigcup_{(i,j) \in I \times J} v_{(i,j)}$.

iv) Si $i' \notin I$ y $j' \notin J$, entonces $i' - 1 \in I$ y $j' + 1 \in J$ y $(i', j') \in v_{(i'-1, j'+1)}$, luego $(i', j') \in \bigcup_{(i,j) \in I \times J} v_{(i,j)}$.

i) a iv), implican que $\mathcal{L} \subseteq \bigcup_{(i,j) \in I \times J} v_{(i,j)}$

Figura 4.9: Lattice del autómata celular alfa-beta de tamaño $2m \times 2n$.
$$\mathcal{S} = \{0, 1\}$$


estado	color
0	
1	

Figura 4.10: Estados del AC alfa-beta.

Por otra parte, si $x \in \bigcup_{(i,j) \in I \times J} v_{(i,j)}$, entonces $x \in v_{(i,j)}$ para algún

$(i, j) \in I \times J$, de la definición de $v_{(i,j)}$ se sigue que $x \in \mathcal{L}$. Por lo tanto

$$\mathcal{L} = \bigcup_{(i,j) \in I \times J} v_{(i,j)}.$$

- Sean $v_{(i,j)}$ y $v_{(i',j')}$ elementos de IJ con $(i, j) \neq (i', j')$. Supongamos $v_{(i,j)} \cap v_{(i',j')} \neq \emptyset$. Sea $(k, l) \in v_{(i,j)} \cap v_{(i',j')}$. Puesto que $(i, j) \neq (i', j')$, entonces $i \neq i'$ o $j \neq j'$. Sin pérdida de generalidad, supongamos $i \neq i'$, puesto que $(k, l) \in v_{(i,j)} \cap v_{(i',j')}$, entonces $(k, l) \in v_{(i,j)}$ y $(k, l) \in v_{(i',j')}$, luego $(k = i \text{ o } k = i + 1)$ y $(k = i' \text{ o } k = i' + 1)$ y $(l = j \text{ o } l = j - 1)$ y

($l = j' \circ l = j' - 1$). Si $i = k = i'$, esto implica una contradicción pues $i \neq i'$. Si $i + 1 = k = i' + 1$, entonces $i = i'$, lo cual es una contradicción con la hipótesis. Si $i = z = i' + 1$, entonces $i = i' + 1$ y puesto que i' es par ya que $i' \in I$, entonces $i = i' + 1$ es impar, es decir, $i \notin I$, lo cual es una contradicción con el hecho de que $i \in I$. De manera similar, se tiene que si $i + 1 = k = i'$, entonces se llega a una contradicción igual a lo anterior. Por tanto se concluye que $v_{(i,j)} \cap v_{(i',j')} = \emptyset$ si $(i, j) \neq (i', j')$.

1, 2 y 3 implican que IJ es una partición de \mathcal{L} . ■

Por ejemplo, si consideramos los vectores $(0, 1)$ y $(2, 5)$, entonces $v_{(0,1)} = \{(0, 1), (0, 0), (1, 1), (1, 0)\}$ y $v_{(2,5)} = \{(2, 5), (2, 4), (3, 5), (3, 4)\}$.

Del teorema anterior se concluye, que dado $l \in \mathcal{L}$ existe un único $(i, j) \in I \times J$ tal que $l \in v_{(i,j)}$. Denotaremos mediante v^l a este elemento único de IJ , esto es, $v^l = v_{(i,j)}$. Por ejemplo, si $l = (3, 0)$, entonces $l \in v^{(3,0)} = v_{(2,1)} = \{(2, 1), (2, 0), (3, 1), (3, 0)\}$.

A partir del hecho anterior se define el conjunto de vecindades para el AC alfa-beta de la siguiente manera:

$$\mathcal{N} = \{v^l | l \in \mathcal{L}\}$$

Antes de comenzar a definir la función de transición, consideremos el siguiente corolario.

Corolario 4.2.1 Si $k \in v_l$, entonces $Anc_l(k) \in \{0, 1, 2\}$.

Demostración Sea $k \in v_{l=(i,j)} = \{(i, j), (i, j - 1), (i + 1, j), (i + 1, j - 1)\}$. Si $k = (i, j)$, entonces $Anc_l(k) = 0$, si $k = (i, j - 1)$ o $k = (i + 1, j)$, entonces $Anc_l(k) = 1$, y si $k = (i + 1, j - 1)$, entonces $Anc_l(k) = 2$. ■

Definición 4.2.2 Consideremos el conjunto A^k . Se define la *función de proyección de la i -ésima componente* ($1 \leq i \leq k$) $Pr_i : A^k \rightarrow A$ como

$$Pr_i(\mathbf{z}) = z_i, \text{ con } \mathbf{z} = (z_1, z_2, \dots, z_k)$$

Enseguida se construya la función de transición para el AC alfa-beta. Para ello, en lo que sigue: sean $\mathbf{x} \in A^n$ y $\mathbf{y} \in A^m$. Sea $Pr_{\mathbf{y}\mathbf{x}} = \{(y_i, x_j) | y_i = Pr_i(\mathbf{y}) \text{ y } x_j = Pr_j(\mathbf{x})\}$.

Proposición 4.2.2 Si $(y_i, x_j) \in Pr_{\mathbf{y}\mathbf{x}}$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) \in v_{(2j-2, 2i-1)}$.

Demostración Se tiene que $v_{(2j-2, 2i-1)} = \{(2j-2, 2i-1), (2j-2, 2i-2), (2j-1, 2i-1), (2j-1, 2i-2)\}$. Puesto que $(y_i, x_j) \in Pr_{\mathbf{y}\mathbf{x}}$, entonces $y_i = Pr_i(\mathbf{y})$ y $x_j = Pr_j(\mathbf{x})$, y puesto que $\mathbf{x} \in A^n$ y $\mathbf{y} \in A^m$, entonces $y_i, x_j \in \{0, 1\}$, luego

1. si $y_i = x_j = 0$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 2, 2i - 2) \in v_{(2j-2, 2i-1)}$.
2. si $y_i = 0$ y $x_j = 1$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 2, 2i - 1) \in v_{(2j-2, 2i-1)}$.
3. si $y_i = 1$ y $x_j = 0$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 1, 2i - 2) \in v_{(2j-2, 2i-1)}$.
4. si $y_i = x_j = 1$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) = (2j - 1, 2i - 1) \in v_{(2j-2, 2i-1)}$.

■

Teorema 4.2.3 La relación $f_{(\mathbf{x}, \mathbf{y})\mathcal{S}} : Pr_{\mathbf{y}\mathbf{x}} \rightarrow \mathcal{S}$ definida como

$$\begin{aligned} f_{(\mathbf{x}, \mathbf{y})\mathcal{S}}(y_i, x_j) &= Anc_{(2j-2, 2i-1)}((2j - 2, 2i - 1) - (0, 1) + (y_i, x_j)) \\ &= Anc_{(2j-2, 2i-1)}(2j - 2 + y_i, 2i - 2 + x_j). \end{aligned}$$

es una función.

Demostración Sea $(y_i, x_j) \in Pr_{\mathbf{y}\mathbf{x}}$, entonces $(2j - 2 + y_i, 2i - 2 + x_j) \in v_{(2j-2, 2i-1)}$ por la proposición 1, luego $Anc_{(2j-2, 2i-1)}(2j - 2 + y_i, 2i - 2 + x_j) \in \{0, 1, 2\}$ por el corolario 4.2.1, esto es, $f_{(\mathbf{x}, \mathbf{y})\mathcal{S}}(y_i, x_j) \in \mathcal{S}$. Ahora bien, si $(y'_i, x'_j) \in Pr_{\mathbf{y}\mathbf{x}}$ es tal que $f_{(\mathbf{x}, \mathbf{y})\mathcal{S}}(y_i, x_j) = f_{(\mathbf{x}, \mathbf{y})\mathcal{S}}(y'_i, x'_j)$, se sigue de la definición de $v_{(2j-2, 2i-1)}$ que $(y_i, x_j) = (y'_i, x'_j)$. ■

Consideremos el conjunto fundamental $CF = \{(\mathbf{x}^\mu, \mathbf{y}^\mu) | \mu = 1, 2, \dots, p\}$. Se define el conjunto $\mathcal{L}_{CF} = \{(2j - 2 + y_i^\mu, 2i - 2 + x_j^\mu) | 1 \leq \mu \leq p, 1 \leq i \leq m \text{ y } 1 \leq j \leq n\} \subseteq \mathcal{L}$.

Consideremos el AC $\mathcal{Q} = (\mathcal{L}, \mathcal{S}, \mathcal{N}, f_{\mathcal{Q}})$ con $f_{\mathcal{Q}} : \mathcal{N} \rightarrow \mathcal{S}$ definida como sigue:

$$f_{\mathcal{Q}}(v^{(i,j)}) = \begin{cases} 1 & \text{si } (i,j) \in \mathcal{L}_{CF} \\ 0 & \text{si } (i,j) \notin \mathcal{L}_{CF} \end{cases}$$

Ahora, consideremos el AC $\mathcal{W} = (\mathcal{L}, \mathcal{S}, \mathcal{N}', f_{\mathcal{W}})$ con $\mathcal{N}' = IJ$ y $f_{\mathcal{W}} : \mathcal{N}' \rightarrow \mathcal{S}$ definida como sigue:

$$f_{\mathcal{W}}(v_{(i,j)}) = \begin{cases} 1 & \text{en la posición } (i+1, j) \quad \text{si } (i, j-1) = 1 \\ 1 & \text{en la posición } (i, j-1) \quad \text{si } (i+1, j) = 1 \end{cases}$$

Se define el AC alfa-beta de aprendizaje como $\mathcal{W} * \mathcal{Q} = (\mathcal{L}, \mathcal{S}, \mathcal{N}, f_A)$.

La configuración del AC alfa-beta de aprendizaje después de aplicar la función contiene la fase de aprendizaje para las memorias asociativas alfa-beta *max* y *min* a la vez. Esto es, la configuración corresponde a la fase de aprendizaje de las memorias asociativas, pero sin distinguir, de esta manera, en la fase de aprendizaje del AC alfa-beta, no hay diferencia entre la fase de aprendizaje para las memorias asociativas alfa-beta *max* y *min*.

Ahora bien, observemos que el conjunto \mathcal{L}_{CF} tiene cardinalidad $m \times n \times p$; luego, para formar el conjunto se requieren $m \times n \times p$ recorridos, lo cual significa que tiene orden de complejidad $O(m \times n \times p)$, esto es, tiene orden de complejidad $O(n^3)$. Ahora bien, puesto que $f_{\mathcal{Q}}$ y $f_{\mathcal{W}}$ operan sobre \mathcal{L} , se concluye que $f_{\mathcal{Q}}$ y $f_{\mathcal{W}}$ tienen orden de complejidad $O(2m \times 2n)$ cada uno; esto es, tienen orden de complejidad $O(n^2)$. Por lo tanto el orden de complejidad para formar el AC alfa-beta de aprendizaje es $O(\max(n^3, n^2, n^2)) = O(n^3)$.

Ejemplo 4.2.1 Sean $m = 3$, $n = 2$, $p = 2$. El conjunto fundamental $CF = \{(x^1, y^1), (x^2, y^2)\}$ está dado por

$$x^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad y^1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad y^1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- La lattice \mathcal{L} está compuesta por la matriz de tamaño $2m \times 2n = 6 \times 4$.
- El conjunto de estados es como quedó definida en la construcción, esto es $S = \{0, 1\}$ y con sus respectivas representaciones de colores para mostrarlo en la lattice.
- El conjunto de vecindades está definido por $\mathcal{N} = \{v^l | l \in \mathcal{L}\}$.
- Apliquemos la función f_Q . Para ello, el siguiente conjunto es aquel en el que f_Q toma el valor de 1 y 0 en su complemento (figura 4.11 a).

$$\begin{aligned} \mathcal{L}_{CF} = \{ & (2j - 2 + y_i^\mu, 2i - 2 + x_j^\mu) | 1 \leq \mu \leq 2, 1 \leq i \leq 3 \text{ y } 1 \leq j \leq 2\} = \\ & \{((2(1)-2+y_1^\mu, 2(1)-2+x_1^\mu), ((2(2)-2+y_1^\mu, 2(1)-2+x_2^\mu)), ((2(1)-2+y_2^\mu, 2(2)- \\ & 2+x_1^\mu), ((2(2)-2+y_2^\mu, 2(2)-2+x_2^\mu), ((2(1)-2+y_3^\mu, 2(3)-2+x_1^\mu), ((2(2)-2+ \\ & y_3^\mu, 2(3)-2+x_2^\mu) | \mu = 1, 2\} = \{(0, 1), (2, 0), (1, 3), (3, 2), (0, 5), (2, 4), (0, 0), \\ & (2, 1), (0, 2), (2, 3), (1, 4), (3, 5)\} \end{aligned}$$

- Aplicando la función f_W a la configuración, el AC alfa-beta de aprendizaje queda como se muestra en la figura 4.11 b).

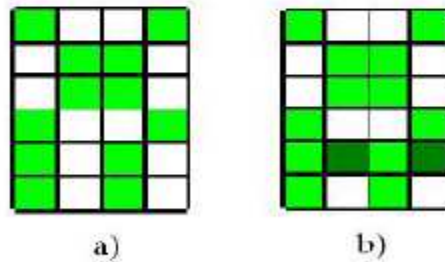


Figura 4.11: Configuración del AC del ejemplo 4.2.1, en a) después de aplicar f_Q y en b) después de aplicar f_W .

Una vez obtenido el AC alfa-beta de aprendizaje del conjunto fundamental, ahora se muestran los algoritmos para la fase de recuperación de patrones

de entrada. El primer algoritmo que se mostrará, es el equivalente a las memorias asociativas *max* y después se describirá el algoritmo equivalente a las memorias asociativas *min*.

Fase de recuperación

Algoritmo 4.2.1: Algoritmo para recuperación de patrones *max*

ENTRADA: Patrón a reconocer $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$

SALIDA: Patrón recuperado $\tilde{y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}$

PROCESO:

```

for  $i = 1, 2, \dots, m$ 
   $\tilde{y}_i = 1$ 
  for  $j = 1, 2, \dots, n$ 
    if  $\tilde{x}_j == 0 \ \&\& \ (2j - 1, 2i - 2) == 1$ 
      continue
    if  $\tilde{x}_j == 1 \ \&\& \ ((2j - 2, 2i - 2) == 1 || (2j - 1, 2i - 2) == 1)$ 
      continue
    else
       $\tilde{y}_i = 0$ 
      break
  fin segundo for
fin primer for

```

Algoritmo 4.2.2: Algoritmo para recuperación de patrones *min*

ENTRADA: Patrón a reconocer $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$

SALIDA: Patrón recuperado $\tilde{y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}$

PROCESO:

```

for  $i = 1, 2, \dots, m$ 
   $\tilde{y}_i = 0$ 
  for  $j = 1, 2, \dots, n$ 
    if  $\tilde{x}_j == 1 \ \&\& \ (2j - 2, 2i - 1) == 1$ 
      continue
    if  $\tilde{x}_j == 0 \ \&\& \ ((2j - 2, 2i - 1) == 1 \ || \ (2j - 1, 2i - 1) == 1)$ 
      continue
    else
       $\tilde{y}_i = 1$ 
      break
  fin segundo for
fin primer for

```

Se observa que los algoritmos 4.2.1 y 4.2.2 constan de dos for anidados. El for externo va de 1 hasta m y el for interno va de 1 hasta n , por tanto el orden de complejidad tanto del algoritmo 4.2.1 como del 4.2.2 es $O(m \times n)$, o lo que es lo mismo, tienen orden de complejidad $O(n^2)$.

Ejemplo 4.2.2 Sean $m = 4$, $n = 3$ y $p = 3$. El conjunto fundamental $CF = \{(x^1, y^1), (x^2, y^2), (x^3, y^3)\}$ esta dado por

$$x^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad y^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$x^2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad y^2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$x^3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad y^3 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

- La lattice \mathcal{L} está compuesta por la matriz de tamaño $2m \times 2n = 8 \times 6$.
- El conjunto de estados $S = \{0, 1\}$.
- El conjunto de vecindades está dado por $\mathcal{N} = \{v^l : l \in \mathcal{L}\}$.
- El siguiente conjunto \mathcal{L}_{CF} , es aquel en donde f_Q toma el valor de 1 y 0 en su complemento. (figura 4.12 a)

$$\begin{aligned} \mathcal{L}_{CF} = \{ & (2j - 2 + y_i^\mu, 2i - 2 + x_j^\mu) \mid 1 \leq \mu \leq 3, 1 \leq i \leq 4, 1 \leq j \leq 3\} = \\ & \{(0, 0), (1, 0), (2, 0), (4, 0), \\ & (5, 0), (0, 1), (3, 1), (4, 1), (0, 2), (1, 2), (2, 2), (3, 2), (4, 2), (0, 3), (2, 3), (5, 3), (0, 4), (2, 4), (4, 4), \\ & (0, 5), (2, 5), (4, 5), (0, 6), (2, 6), (3, 6), (4, 6), (5, 6), (1, 7), (2, 7), (4, 7)\} \end{aligned}$$

- Aplicando f_W al AC anterior, se obtiene el AC alfa-beta de aprendizaje que se muestra en la figura 4.12 b.
- Ahora apliquemos el algoritmo para recuperación de patrones *max* y *min* a partir del AC alfa-beta de aprendizaje.

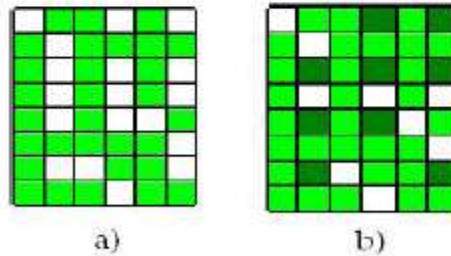


Figura 4.12: Configuración del AC del ejemplo 4.2.2, en a) después de aplicar f_Q y en b) después de aplicar f_W .

Consideremos el patrón de entrada

i	y_i^1	j	x_j^1	condición	$\rightarrow y_i^1$
1	1	1	1	$(2j - 2, 2i - 2) == 1$	1
		2	0	$(2j - 1, 2i - 2)! = 1$	0
2	1	1	1	$(2j - 2, 2i - 2) == 1$	1
		2	0	$(2j - 1, 2i - 2) == 1$	1
		3	0	$(2j - 1, 2i - 2)! = 1$	0
3	1	1	1	$(2j - 2, 2i - 2) == 1$	1
		2	0	$(2j - 1, 2i - 2)! = 1$	0
4	1	1	1	$(2j - 2, 2i - 2) == 1$	1
		2	0	$(2j - 1, 2i - 2) == 1$	1
		3	0	$(2j - 1, 2i - 2) == 1$	1

Tabla 4.1: Muestra el funcionamiento del AC alfa-beta en su fase de recuperación *max*

$$x^1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

La siguiente tabla muestra el valor inicial y final para cada componente del vector de salida y^1 cuando se aplica el algoritmo para recuperación *max*. La primera columna muestra el valor para la variable i considerando el primer *for* del algoritmo, la segunda columna es el valor de y_i^1 de default, el cual es 1. La tercera columna son los diferentes valores para el ciclo de la variable j , la cuarta columna es el valor respectivo que tiene x_j^1 , la quinta columna es la condición que se debe de cumplir en el algoritmo dependiendo si el valor de x_j^1 es 0 o 1. Finalmente la sexta columna muestra el valor final de la componente y_i^1 .

De la tabla anterior se concluye que $y^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. Por lo tanto recupera

el patrón y^1 cuando se le presenta el patrón de entrada x^1 al utilizar el algoritmo para recuperación de patrones *max*.

De manera similar a la tabla anterior, la siguiente tabla muestra el valor inicial y final para cada componente del vector de salida y^1 cuando se aplica el algoritmo para recuperación *min*.

i	y_i^1	j	x_j^1	condición	$\rightarrow y_i^1$
1	0	1	1	$(2j - 2, 2i - 1) == 1$	0
		2	0	$(2j - 1, 2i - 1) == 1$	0
		3	0	$(2j - 1, 2i - 1) == 1$	0
2	0	1	1	$(2j - 2, 2i - 1) == 1$	0
		2	0	$(2j - 2, 2i - 1) == 1$	0
		3	0	$(2j - 1, 2i - 1) == 1$	0
3	0	1	1	$(2j - 2, 2i - 1) == 1$	0
		2	0	$(2j - 2, 2i - 1) == 1$	0
		3	0	$(2j - 2, 2i - 1) == 1$	0
4	0	1	1	$(2j - 2, 2i - 1)! = 1$	1

Tabla 4.2: Muestra el funcionamiento del AC alfa-beta en su fase de recuperación *min*

De la tabla anterior se concluye que $y^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$. Por lo tanto recupera

el patrón y^1 cuando se le presenta el patrón de entrada x^1 al utilizar el algoritmo para recuperación de patrones *min*.

4.2.3 Modelos Celulares alfa-beta con Erosión y Dilatación

En la sección anterior se presentó el modelo del AC alfa-beta en su fase de aprendizaje y de recuperación.

El AC alfa-beta de aprendizaje es un autómata celular cuya configuración contiene la fase de aprendizaje tanto de las memorias asociativas alfa-beta *max* como de las memorias asociativas alfa-beta *min*. Ahora bien, enseguida se presenta una variante mostrado en la sección anterior, el cual hace uso de la dilatación celular, la erosión celular y la composición de autómatas celulares. Puesto que en la fase de recuperación, se divide en recuperación de patrones *max* (algoritmo 4.2.1) y recuperación de patrones *min* (algoritmo 4.2.2), se hace notar como consecuencia el modelo celular alfa-beta *max* y el modelo celular alfa-beta *min*.

Algoritmo 4.2.3: Modelo Celular alfa-beta *max (min)* con Erosión y Dilatación Celular aplicado a la Recuperación de Patrones

ENTRADA: Conjunto fundamental $CF = \{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$, elemento estructurante B , valor entero ne (Número de Erosiones), valor entero

nd (Número de Dilataciones) y patrón a reconocer $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$

SALIDA: Patrón recuperado $\tilde{y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_m \end{pmatrix}$

PROCESO:

1. Construir el AC alfa-beta de aprendizaje para CF .
2. Aplicar la composición del autómata celular erosión \mathcal{E} con elemento estructurante B a la configuración inicial del AC alfa-beta de aprendizaje ne veces. Esto es aplicar a la configuración del AC alfa-beta $\mathcal{E} * \mathcal{E} * \dots * \mathcal{E}$, ne veces.
3. Aplicar la composición del autómata celular dilatación \mathcal{D} con elemento estructurante B a la configuración final del punto 2 nd veces. Esto es aplicar a la configuración final del punto 2 $\mathcal{D} * \mathcal{D} * \dots * \mathcal{D}$, nd veces.
4. Aplicar el algoritmo para recuperación de patrones *max (min)* para el

valor de entrada $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$, la salida será el patrón \tilde{y} .

Se observa lo siguiente del algoritmo 4.2.3 respecto al orden de complejidad: el paso 1 tiene orden de complejidad $O(n^3)$ (ver sección anterior); por cada erosión o dilatación, se recorre toda la lattice de tamaño $2m \times 2n$, luego la aplicación de una erosión o dilatación tiene orden de complejidad $O(n^2)$,

luego aplicar ne veces una erosión o aplicar nd veces una dilatación nos lleva a que este conjunto de operaciones tienen orden de complejidad $ne \times O(n^2)$ y $nd \times O(n^2)$ respectivamente; por lo tanto el orden de complejidad del punto 2 y del punto 3 es $O(n^2)$. El orden de complejidad del punto 4 es $O(n^2)$ (ver sección anterior). En conclusión el orden de complejidad del algoritmo 4.2.3 es: $O(\max(n^3, n^2, n^2, n^2)) = O(n^3)$.

Generalmente el problema que se pretende atacar en el reconocimiento de patrones, es la clasificación de patrones. Esto es, se tienen un conjunto de clases disjuntas $C = \{C_{i_1}, C_{i_2}, \dots, C_{i_k}\}$ y un conjunto fundamental $CF = \{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ de tal manera que se sabe que x^μ pertenece a la clase C_μ para alguna $\mu = i_1, i_2, \dots, i_k$. Si se ingresa un nuevo patrón \tilde{x} al que se desconoce a que clase pertenece, el objetivo de la problemática consiste en asignarle una clase a el patrón \tilde{x} de tal manera que sea la que le corresponde. Es por ello, que enseguida se presenta un algoritmo celular alfa-beta enfocado en la problemática de la clasificación de patrones.

Algoritmo 4.2.4: Modelo Celular alfa-beta *max (min)* con Erosión y Dilatación Celular aplicado a la Clasificación de Patrones

ENTRADA: Conjunto de clases $C = \{C_{i_1}, C_{i_2}, \dots, C_{i_r}\}$, Conjunto fundamental $CF = \{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ del que se sabe que x^μ pertenece a la clase C_μ para alguna $\mu = i_1, i_2, \dots, i_r$; elemento estructurante B , valor entero ne (Número de Erosiones), valor entero nd (Número de Dilataciones)

y patrón a clasificar $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$

SALIDA: Clase C_{i_k} a la que pertenece el patrón \tilde{x}

PROCESO:

1. Construir el AC alfa-beta de aprendizaje para CF .
2. Aplicar la composición del autómata celular erosión \mathcal{E} con elemento estructurante B a la configuración inicial del AC alfa-beta de aprendizaje ne veces. Esto es aplicar a la configuración del AC alfa-beta $\mathcal{E} * \mathcal{E} * \dots * \mathcal{E}$, ne veces.

3. Aplicar la composición del autómata celular dilatación \mathcal{D} con elemento estructurante B a la configuración final del punto 2 nd veces. Esto es aplicar a la configuración final del punto 2 $\mathcal{D} * \mathcal{D} * \dots * \mathcal{D}$, nd veces.
4. Aplicar el algoritmo para recuperación de patrones *max* (*min*) para el valor de entrada $\tilde{x} = \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_n \end{pmatrix}$, la salida será un patrón \tilde{y} del CF .
5. Si \tilde{y} pertenece a la clase C_{i_k} , entonces el algoritmo clasificará el patrón \tilde{x} a la clase C_{i_k} .

Se observa lo siguiente del algoritmo 4.2.4 respecto al orden de complejidad: el paso 1 tiene orden de complejidad $O(n^3)$ (ver sección anterior); por cada erosión o dilatación, se recorre toda la lattice de tamaño $2m \times 2n$, luego la aplicación de una erosión o dilatación tiene orden de complejidad $O(n^2)$, luego aplicar ne veces una erosión o aplicar nd veces una dilatación nos lleva a que este conjunto de operaciones tienen orden de complejidad $ne \times O(n^2)$ y $nd \times O(n^2)$ respectivamente; por lo tanto el orden de complejidad del punto 2 y del punto 3 es $O(n^2)$. El orden de complejidad del punto 4 es $O(n^2)$ (ver sección anterior). El orden de complejidad del punto 5 es $O(n)$ ya que se trata de una búsqueda lineal dentro del conjunto de clases. En conclusión el orden de complejidad del algoritmo 4.2.4 es: $O(\max(n^3, n^2, n^2, n^2, n)) = O(n^3)$. Se hace notar que el orden de complejidad del algoritmo propuesto y el de las memorias asociativas alfa-beta es el mismo; esto es, para ambos casos el orden de complejidad es $O(n^3)$.

Capítulo 5

Experimentos y Resultados

En este capítulo se muestra la parte experimental de los modelos propuestos en el capítulo anterior. Para ello, se abordó el problema de la clasificación de patrones de bases de datos proporcionadas por el Centro de Aprendizaje Automático y Sistemas Inteligentes de la Universidad de California Irving (UCI) que se encuentran disponibles en <http://archive.ics.uci.edu/ml/>, también se hizo uso del problema de clasificación de presencia de retinopatía diabética, el cual mediante el tratamiento digital de imágenes de la red vascular retiniana se extrajeron las características de los patrones. En cada situación se hizo un estudio comparativo entre el modelo propuesto y una gama de modelos del bagaje de reconocimiento de patrones, entre los que se encuentran las memorias asociativas alfa-beta.

5.1 Base de Datos: Iris

La base de datos *Iris* es la base de datos más conocida y utilizada dentro del campo del reconocimiento de patrones. La base de datos esta compuesta por 3 clases de distintas plantas de tipo Iris: Iris Setosa, Iris Versicolor e Iris Virgínica. La base de datos consta de 150 instancias, donde cada instancia representa un tipo de planta iris. Cada instancia esta compuesta de 4 atributos: 1. longitud del sépalo en cm, 2. ancho del sépalo en cm, 3. longitud del pétalo en cm y 4. ancho del pétalo en cm.

Se tomaron los datos y se les aplicó el algoritmo de clasificación k -NN, con k de 1 a 10. En la tabla 5.1 se muestran los resultados obtenidos.

Se convirtieron los datos a su representación binaria mediante el código

Algoritmo k -NN	
k	Rendimiento (%)
1	96.66
2	94.66
3	96.66
4	96
5	97.33
6	97.33
7	96.66
8	98
9	96.66
10	96.66

Tabla 5.1: k -NN aplicado a la base de datos de: *Iris*

Memorias Asociativas	
alfa-beta <i>max</i> (%)	alfa-beta <i>min</i> (%)
91.69	93.33

Tabla 5.2: Memorias Asociativas alfa-beta aplicado a la base de datos: *Iris*

Johnson Möbius Modificado. A los datos codificados, se les aplicó el modelo de memorias autoasociativas *max* y *min*. La tabla 5.2 muestra los resultados obtenidos.

Por otro lado, la tabla 5.3 muestra otros modelos de clasificación de patrones aplicados a la base de datos iris y su rendimiento obtenido respectivamente.

Se aplicó el modelo celular alfa-beta *max* y *min*, considerando los datos en su codificación Johnson Möbius Modificado. Como elemento estructurante se tomó: la vecindad de von Neumann, la vecindad de Moore y la definida por la célula en estudio y su vecino izquierdo como se muestra en la figura 2.1. Se experimentó con una erosión celular seguida de una dilatación celular; dos erosiones celulares seguida de una dilatación celular; dos erosiones celulares seguidas de dos dilataciones celulares; tres erosiones celulares seguidas de dos dilataciones celulares; y tres erosiones celulares seguidas de tres dilataciones

Modelo	Iris Plant (%)
LogitboostNB [88]	94.87
NB [88]	95.53
Bayesian Network (K2) [88]	93.20
Adaboost NB [88]	94.80
Bagging NB [88]	95.53
DECORATE NB [88]	94.87
NBTree [88]	93.53
Boost C4.5 [88]	94.33
Bagging C4.5 [88]	94.67
LogitBostDS [88]	94.93
K-Means [89]	89
Neural Gas [89]	91.7
Kernel Method [89]	94.7
MLP [90]	95.99
SVM [90]	95.33
Ibk [90]	95.33
J48 [90]	95.33
NBT [90]	93.99
oneR [90]	93.33
PART [90]	94.66

Tabla 5.3: Rendimiento de clasificación de otros modelos para la base de datos: *Iris*

celulares. Se registraron los resultados. Las tablas 5.4, 5.5 y 5.6 muestran los resultados obtenidos, en donde \mathcal{E} significa que se realizó una erosión celular y \mathcal{D} una dilatación celular.

Modelo Celular alfa-beta Vecino Izquierdo		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	99.33	99.33
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	33.33	100
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	99.33	100
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	33.33	100
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	99.33	100

Tabla 5.4: Modelo Celular alfa-beta aplicado a la base de datos: *Iris*; con elemento estructurante: vecino izquierdo

Modelo Celular alfa-beta Vecindad: von Neumann		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	99.33	99.33
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	33.33	30
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	38.66	34
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	38	34
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	34	34

Tabla 5.5: Modelo Celular alfa-beta aplicado a la base de datos: *Iris*; con elemento estructurante: vecindad de Neumann

Modelo Celular alfa-beta Vecindad: Moore		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	33.33	30
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	33.33	30
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	34	34
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	35.33	34
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	34	34

Tabla 5.6: Modelo Celular alfa-beta aplicado a la base de datos: *Iris*; con elemento estructurante: vecinda de Moore

5.2 Base de Datos: Abalone

La base de datos *Abalone* tiene como objetivo predecir la edad de este molusco. La edad del abalone se determina mediante la reducción de la cubierta a través del cono, las manchas, el número de anillos que tiene entre otras. La base de datos consta de 4177 instancias, cada instancia con 8 atributos y una clase por predecir: la edad. Las instancias que forman cada atributo son: el sexo, la longitud, el diámetro, la altura y los anillos entre otros.

Se tomaron los datos y se les aplicó el algoritmo de clasificación k -NN, con k de 1 a 10. En la tabla 5.7 se muestran los resultados.

Se convirtieron los datos a su representación binaria mediante el código Johnson Möbius Modificado. A los datos codificados, se les aplicó el modelo de memorias autoasociativas *max* y *min*. En la tabla 5.8 se muestran los resultados.

Por otro lado, la tabla 5.9 muestra otros modelos de clasificación de patrones aplicados a la base de datos abalone y su rendimiento obtenido respectivamente.

Se aplicó el modelo celular alfa-beta *max* y *min*, considerando los datos en su codificación Johnson Möbius Modificado. Como elementos estructurantes se tomaron: la vecindad de von Neumann, la vecindad de Moore y la definida por la célula en estudio y su vecino izquierdo. Se experimentó con una erosión celular seguida de una dilatación celular; dos erosiones celulares seguida de una dilatación celular; dos erosiones celulares seguidas de dos dilataciones celulares; tres erosiones celulares seguidas de dos dilataciones celulares; y tres erosiones celulares seguidas de tres dilataciones celulares. Se registraron los resultados. Las tablas 5.10, 5.11 y 5.12 muestran los resultados obtenidos, en donde \mathcal{E} significa que se realizó una erosión celular al AC alfa-beta de aprendizaje y \mathcal{D} una dilatación celular.

Algoritmo k -NN	
k	Rendimiento (%)
1	48.58
2	48.68
3	52.42
4	52.15
5	52.27
6	52.32
7	52.11
8	52.25
9	52.44
10	52.18

Tabla 5.7: k -NN aplicado a la base de datos: *Abalone*

Memorias Asociativas	
alfa-beta <i>max</i> (%)	alfa-beta <i>min</i> (%)
46	45.45

Tabla 5.8: Memorias Asociativas alfa-beta aplicado a la base de datos: *Abalone*

Modelo	Abalone (%)
kNN [91]	60.82
MLP [91]	60.05
Adaboost/Bagging [91]	63
Cascading [91]	63.51
ANN-RBF [92]	46.21
FM3 [92]	45.93
SVM-poly [92]	45.87
SVM-rbf [92]	45.84

Tabla 5.9: Rendimiento de clasificación para la base de datos: *Abalone*

Modelo Celular alfa-beta Vecindad: Vecino Izquierdo		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	66.38	62.07
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	32.31	61.47
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	66.38	61.47
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	32.31	60.56
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	66.38	60.56

Tabla 5.10: Modelo Celular alfa-beta aplicado a la base de datos: *Abalone*; con elemento estructurante: vecino izquierdo

Modelo Celular alfa-beta Vecindad: von Neumann		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	66.38	62.34
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	31.31	36.84
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	32.65	31.36
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	32.31	31.36
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	34.04	31.36

Tabla 5.11: Modelo Celular alfa-beta aplicado a la base de datos: *Abalone*; con elemento estructurante: vecindad de Neumann

Modelo Celular alfa-beta Vecindad: Moore		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	32.31	36.84
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	32.31	36.84
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	32.31	31.36
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	32.31	31.36
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	34.04	31.36

Tabla 5.12: Modelo Celular alfa-beta aplicado a la base de datos: *Abalone*; con elemento estructurante: vecindad de Moore

5.3 Base de Datos: Hepatitis

La base de datos *Hepatitis* cuenta con 155 instancias, cada instancia de 19 atributos y dos clases por predecir: muerto o vivo. La base de datos contiene información de pacientes que padecen hepatitis tal como: la edad, el sexo, si consume esteroides, si consume antivirales, si tiene fatiga, si tiene anorexia y vórices entre otros atributos.

Se tomaron los datos y se les aplicó el algoritmo de clasificación k -NN, con k de 1 a 10. En la tabla 5.13 se muestran los resultados.

Se convirtieron los datos a su representación binaria mediante el código Johnson Möbius Modificado. A los datos codificados, se les aplicó el modelo de memorias autoasociativas *max* y *min*. En la tabla 5.14 se muestran los resultados.

Por otro lado, la tabla 5.15 muestra otros modelos de clasificación de patrones aplicados a la base de datos de la hepatitis y su rendimiento obtenido respectivamente.

Se aplicó el modelo celular alfa-beta *max* y *min*, considerando los datos en su codificación Johnson Möbius Modificado. Como elementos estructurantes se tomaron: la vecindad de von Neumann, la vecindad de Moore y la definida por la célula en estudio y su vecino izquierdo. Se experimentó con una erosión celular seguida de una dilatación celular; dos erosiones celulares seguida de una dilatación celular; dos erosiones celulares seguidas de dos dilataciones celulares; tres erosiones celulares seguidas de dos dilataciones celulares; y tres erosiones celulares seguidas de tres dilataciones celulares. Se registraron los resultados. Las tablas 5.16, 5.17 y 5.18 muestran los resultados obtenidos, en donde \mathcal{E} significa que se realizó una erosión celular al AC alfa-beta de aprendizaje y \mathcal{D} una dilatación celular.

Algoritmo k -NN	
k	Rendimiento (%)
1	72.66
2	71.33
3	82.66
4	78
5	81.33
6	81.33
7	76.66
8	73.33
9	74.66
10	72.66

Tabla 5.13: k -NN aplicado a la base de datos: *Hepatitis*

Memorias Asociativas	
alfa-beta <i>max</i> (%)	alfa-beta <i>min</i> (%)
80.30	93.75

Tabla 5.14: Memorias Asociativas aplicado a *Hepatitis*

Modelo	Hepatitis (%)
C4.5 (S=15) [93]	80.78
double C4.5 (S=15) [93]	66.16
C4.5/GA-small (S=15) [93]	79.36
C4.5/GA-large-SN (S=15) [93]	82.52
DIMLP [94]	79.1
MLP [94]	79.7
CN2 [94]	81.8
E_{pc} [94]	83.4
$Y2_{pe}$ [94]	83.0
P_{pe} [94]	84.4
Naïve Bayes [95]	82.58
IFN [95]	78.97
DOG [95]	80
GOV [95]	81.29
GEFS [95]	83.92
Adaboos [95]	81.29
AB [95]	81.3
Wrapper Simple [95]	79.1
VC upper bound Simple [95]	79.9
Wrapper New [95]	83.67
VC lower bound New [95]	76.15
VC upper bound New [95]	81.29
LogitBoostNB [88]	84.67
NB [88]	83.81
Bayesian Network [88]	84.18
AODE [88]	85.36
Adaboost NB [88]	84.67
Bagging NB [88]	84.13
DECORATE NB [88]	82.99
BoostFSNB [88]	84.81
NBTree [88]	81.36
LBR [88]	84.97

Tabla 5.15: Rendimiento de clasificación para la base de datos: *Hepatitis*

Modelo Celular alfa-beta Vecindad: Vecino Izquierdo		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	99.35	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	99.35	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	99.35	21.29

Tabla 5.16: Modelo Celular alfa-beta aplicado a la base de datos: *Hepatitis*; con elemento estructurante: vecino izquierdo

Modelo Celular alfa-beta Vecindad: von Neumann		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	99.35	99.35
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	78.06	21.29

Tabla 5.17: Modelo Celular alfa-beta aplicado a la base de dato: *Hepatitis*; con elemento estructurante: vecindad de Neumann

Modelo Celular alfa-beta Vecindad: Moore		
.	<i>max</i> (%)	<i>min</i> (%)
$\mathcal{E} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	78.06	21.29
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	78.06	21.29

Tabla 5.18: Modelo Celular alfa-beta aplicado a la base de datos: *Hepatitis*; con elemento estructurante: vecindad de Moore

5.4 Base de Datos: Retinopatía Diabética

La Retinopatía Diabética (RD) es una de las complicaciones de la exposición prolongada a la Diabetes [85], este padecimiento deteriora la vista y puede llevar a su pérdida total.

Esta enfermedad es difícil de diagnosticar cuando se encuentra en su primera etapa porque los daños que presenta el ojo son muy pequeños para percibirlos a simple vista.

A partir de un banco de imágenes de la red vascular retiniana de distintos pacientes otorgada por el Hospital de Especialidades Centro Médico La Raza Gaudencio González Garza del Instituto Mexicano del Seguro Social, se desarrolló en colaboración con la Escuela Superior de Cómputo un sistema capaz de predecir si un paciente padece o no retinopatía diabética a partir su red vascular retiniana [86]. Para ello primeramente se realizó el tratamiento digital de cada imagen del banco de imágenes, para obtener sus características y formar el patrón que lo representa.

La base de datos consta de 50 instancias, cada instancia de 14 atributos, los cuales representan los momentos de Hu para exudados y para hemorragias ya codificados.

Se tomaron los datos y se les aplicó el algoritmo de clasificación k -NN, con k de 1 a 10. En la tabla 5.19 se muestran los resultados.

A los datos, se les aplicó el modelo de memorias autoasociativas max y min . En la tabla 5.20 se muestran los resultados obtenidos.

Se aplicó el modelo celular alfa-beta max y min . Como elementos estructurantes se tomaron: la vecindad de von Neumann, la vecindad de Moore y la definida por la célula en estudio y su vecino izquierdo. Se experimentó con una erosión celular seguida de una dilatación celular; dos erosiones celulares seguida de una dilatación celular; dos erosiones celulares seguidas de dos dilataciones celulares; tres erosiones celulares seguidas de dos dilataciones celulares; y tres erosiones celulares seguidas de tres dilataciones celulares. Se registraron los resultados. Las tablas 5.21, 5.23 y 5.18 muestran los resultados obtenidos, en donde \mathcal{E} significa que se realizó una erosión celular al AC alfa-beta de aprendizaje y \mathcal{D} una dilatación celular.

Algoritmo k -NN	
k	Rendimiento
1	93.75
2	93.75
3	93.75
4	93.75
5	95.83
6	95.83
7	95.83
8	95.83
9	95.83
10	95.83

Tabla 5.19: k -NN aplicado a la clasificación de Retinopatía Diabética

Memorias Asociativas	
alfa-beta <i>max</i>	alfa-beta <i>min</i>
97.91	97.91

Tabla 5.20: Memorias Asociativas alfa-beta aplicado a la clasificación de Retinopatía Diabética

Modelo Celular alfa-beta Vecindad: Vecino Izquierdo		
.	<i>max</i>	<i>min</i>
$\mathcal{E} * \mathcal{D}$	95.91	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	95.91	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	95.91	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	95.91	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	95.91	95.91

Tabla 5.21: Modelo Celular alfa-beta aplicado a *Retinopatía Diabética* con elemento estructurante: vecino izquierdo

Modelo Celular alfa-beta Vecindad: von Neumann		
.	<i>max</i>	<i>min</i>
$\mathcal{E} * \mathcal{D}$	91.83	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	95.91	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	77.55	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	77.55	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	97.95	95.91

Tabla 5.22: Modelo Celular alfa-beta aplicado a *Retinopatía Diabética* con elemento estructurante: vecindad de Neumann

Modelo Celular alfa-beta Vecindad: Moore		
.	<i>max</i>	<i>min</i>
$\mathcal{E} * \mathcal{D}$	97.95	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D}$	97.95	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	97.95	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$	97.95	95.91
$\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$	97.95	95.91

Tabla 5.23: Modelo Celular alfa-beta aplicado a *Retinopatía Diabética* con elemento estructurante: vecindad de Moore

5.5 Discusión de Resultados

En las secciones anteriores, se utilizó el modelo celular alfa-beta en distintas bases de datos. Para sustentar la eficiencia del modelo celular alfa-beta, se consideró una gran gama de modelos de clasificación de patrones aplicados a la misma base de datos. En el caso de los experimentos, para validar los resultados del modelo celular alfa-beta, se utilizó el método k -fold cross validation tomando $k=10$, además se utilizaron diversas combinaciones de erosiones celulares seguida de dilataciones celulares considerando tres tipos de elementos estructurantes: el vecino izquierdo, la vecindad de von Neumann y la vecindad de Moore.

Se observa, que en el caso de la base de datos de la Iris, se alcanza un máximo de 97.37% de rendimiento con el algoritmo k -NN con $k = 5$ y $k = 6$ (tabla 5.1), con las memorias asociativas alfa-beta *max* y *min* se alcanza un rendimiento de 91.69% y 93.33% respectivamente (tabla 5.2), la tabla 5.3 muestra el rendimiento de varios modelos de clasificación de patrones aplicado a la base de datos iris, de donde se observa que el modelo MLP muestra el mejor rendimiento de entre todos los modelos expuestos en la tabla, alcanzando un rendimiento del 95.99(%), por otro lado, el modelo celular alfa-beta *mín* alcanza un máximo de 100% cuando se toma como elemento estructurante el vecino izquierdo y se consideran las configuraciones: $\mathcal{E} * \mathcal{E} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$ y $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$ (tabla 5.4). De esta manera se observa como el algoritmo celular alfa-beta supera a cada uno de los algoritmos con los que es comparado.

En la base de datos Abalone, el algoritmo k -NN encuentra su máximo rendimiento de 52.44% con $k = 9$ (tabla 5.7), las memorias asociativas alfa-beta *max* y *min* muestran un rendimiento de 46% y 45.45% respectivamente (tabla 5.8), la tabla 5.9 muestra el rendimiento de varios modelos de clasificación de patrones aplicado a la base de datos abalone, de donde se observa que el modelo Cascading muestra el mejor rendimiento de entre todos los modelos expuestos en la tabla, alcanzando un rendimiento de 63.51(%), por otro lado, el modelo celular alfa-beta *max* encuentra su máximo rendimiento de 66.38% cuando se considera la vecindad de vecino izquierdo y configuraciones: $\mathcal{E} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$ y $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$ o bien $\mathcal{E} * \mathcal{D}$ con vecindad de von Neumann (tabla 5.10) o bien vecindad de Neumann (tabla

5.11). Nuevamente se observa como el algoritmo celular alfa-beta supera a los otros algoritmos de clasificación con los que es comparado.

En el caso de la base de datos de la hepatitis, el algoritmo k -NN mostró su máximo rendimiento de 82.66% con $k = 3$ (tabla 5.13), las memorias asociativas alfa-beta *max* y *min* mostraron un rendimiento de 80.30% y 93.75% (tabla 5.14), el modelo con mejor rendimiento de la tabla 5.15 es el modelo AODE con un rendimiento de 85.36%, mientras que el modelo celular alfa-beta *max* y *min* con elemento estructurante la vecindad de vecino izquierdo y configuraciones: $\mathcal{E} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$ y $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$ o bien $\mathcal{E} * \mathcal{D}$ y vecindad de von Neumann motró su máximo rendimiento de 99.35% (tabla 5.17). De esta manera, se muestra nuevamente como el modelo celular alfa-beta supera a cada uno de los algoritmos con los que es comparado.

Cabe destacar que en cada una de las situaciones mostradas, el modelo celular alfa-beta con elemento estructurante el vecino izquierdo y configuraciones: $\mathcal{E} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$ y $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$ muestran un rendimiento superior (pero no siempre el mejor) al de otros modelos del bagaje del reconocimiento de patrones; entre ellos, las memorias asociativas alfa-beta.

Por otra parte en el caso de la base de datos de la retinopatía diabética, el algoritmo k -NN encontró su máximo de 95.83% con $k = 5, 6, 7, 8, 9, 10$ (tabla 5.19), las memorias asociativas alfa-beta mostraron un rendimiento de 97.91% cada uno (tabla 5.20), y el modelo celular alfa-beta *max* mostró un máximo de 97.95% considerando como elemento estructurante la vecindad de Moore y cualquiera de sus combinaciones de erosiones y dilataciones mostradas (tabla 5.23).

El modelo celular alfa-beta muestra un mejor rendimiento respecto a las memorias asociativas alfa-beta y a otros modelos del bagaje de reconocimiento de patrones. Se observa que en aquellos datos en los que las componentes de cada patrón representan características medibles del objeto en cuestión (por ejemplo la longitud del pétalo de una flor o la edad de una persona) o características de su esencia (como el sexo de una persona), entonces el modelo celular alfa-beta con vecindad el vecino izquierdo y configuraciones: $\mathcal{E} * \mathcal{D}$, $\mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D}$ y $\mathcal{E} * \mathcal{E} * \mathcal{E} * \mathcal{D} * \mathcal{D} * \mathcal{D}$ supera en rendimiento a los otros modelos de reconocimiento de patrones. Por otro lado, si los patrones han pasado por una serie de preprocesamientos de tal manera que las componentes del patrón no tienen un significado directo con el objeto

en estudio (por ejemplo para obtener las componentes del patrón de una imagen aplicando tratamiento digital de imágenes y momentos de Hu), entonces el modelo celular con vecindad de Moore muestra mejores resultados independientemente de la configuración.

Capítulo 6

Conclusiones

Este trabajo mostró el desarrollo de investigación del campo de los autómatas celulares asociado al de las memorias asociativas. Para ello, primeramente se presentó el desarrollo de las operaciones fundamentales de la morfología matemática; esto es, las operaciones de dilatación, erosión, apertura y cerradura por medio de autómatas celulares. Se definió el autómata celular erosión, dilatación y composición de autómatas celulares. Enseguida, se mostraron los operadores fundamentales de las memorias asociativas alfa-beta (operador alfa y operador beta) mediante métricas, para posteriormente mostrar el autómata celular alfa-beta en su fase de aprendizaje y en su fase de recuperación. Cabe destacar que la configuración el autómata celular alfa-beta en su fase de aprendizaje captura tanto la configuración de las memorias asociativas alfa-beta *max* como de las alfa-beta *min*. Finalmente se mostró un modelo de autómatas celulares para la clasificación de patrones al que se le nombro como modelo celular alfa-beta *max(min)* con erosión y dilatación celular.

Para medir y comparar el rendimiento del modelo celular alfa-beta, se utilizaron bases de datos proporcionadas por el Centro de Aprendizaje Automático y Sistemas Inteligentes de la Universidad de California Irving, que se encuentran disponibles en [87]. Las bases de datos utilizadas de este repertorio fueron: Iris Data Set, Abalone Data Set y Hepatitis data Set. También se utilizaron datos para clasificar si un paciente padece o no retinopatía diabética a partir de un banco de imágenes proporcionada por el Hospital de Especialidades Centro Médico La Raza Gaudencio González Garza del Instituto Mexicano del Seguro Social. El tratamiento digital de las imágenes del

banco de imágenes fue desarrollado en colaboración con la Escuela Superior de Cómputo [85].

Se observó que el modelo celular muestra un mejor rendimiento respecto al bagaje de algoritmos de clasificación de patrones presentados, incluyendo a las memorias asociativas alfa-beta. En las situaciones en donde las características de los patrones representan métricas del objeto en estudio o cualidades de su esencia, el modelo celular alfa-beta con vecindad del vecino izquierdo y configuraciones \mathcal{ED} , \mathcal{EEDD} y \mathcal{EEEDDD} muestran resultados más altos que el de los otros modelos del bagaje de algoritmos de clasificación de patrones. Por otro lado, si las características de los patrones son valores que se obtuvieron de un preprocesamiento en donde su significado no tiene que ver directamente con el objeto en estudio, entonces el modelo celular alfa-beta muestra un mejor rendimiento con la vecindad de Moore en cualquiera de sus configuraciones que se considero.

Lo anterior muestra que que el modelo celular alfa-beta es una nueva propuesta de un algoritmo de clasificación de patrones cuyo rendimiento muestra resultados mejores al de los mostrados en el bagaje de los algoritmos de reconocimiento de patrones bajo las condiciones dichas en el párrafo anterior.

6.1 Aportaciones

- Se han implementado las operaciones fundamentales de la morfología matemática (dilatación, erosión, apertura y cerradura) por medio de autómatas celulares; esto, con la finalidad de utilizarlas posteriormente para el desarrollo de los modelos celulares alfa-beta.
- Se ha logrado crear una asociación entre el campo de la morfología matemática y el de los autómatas celulares.
- Se han desarrollado las operaciones fundamentales de las memorias asociativas alfa-beta: el operador alfa y el operador beta por medio de autómatas celulares, punto necesario para el desarrollo de los modelos celulares alfa-beta que están basados en dichos operadores.
- Se han desarrollado las memorias asociativas alfa-beta por medio de autómatas celulares.

- Se ha creado una asociación entre el campo de las memorias asociativas y el de los autómatas celulares, de donde surgen los nuevos modelos de memorias asociativas.
- Se han propuesto nuevos modelos para el reconocimiento y clasificación de patrones por medio de autómatas celulares: Modelos Celulares alfa-beta, los cuales han mostrado un mejor rendimiento respecto a las memorias asociativas alfa-beta y otros modelos del bagaje de reconocimiento de patrones con los que fue comparado.

6.2 Trabajos a Futuro

- Hacer un análisis para conocer a fondo por qué el modelo celular alfa-beta responde con diferentes rendimientos cuando se aplican diferentes erosiones y dilataciones celulares; y encontrar la combinación de las operaciones dilatación y erosión celular para optimizar el rendimiento.
- Aplicar el modelo celular alfa-beta a más bases de datos de la UCI MLR.
- Encontrar mejoras al modelo celular alfa-beta.
- Comparar el rendimiento del modelo celular alfa-beta con una gran gama de modelos para clasificación de patrones.

Publicaciones

Trabajos en proceso de revisión ISI-JCR

- Luna-Benoso B., Yáñez-Márquez, C., Figueroa-Nazuno J., López-Yáñez. *Alpha-Beta Associative Memory Implemented with Cellular Automata*. Enviado a la revista: Pattern Recognition. ISSN: 0031-3203. ELSEVIER SCI LTD.
- Luna-Benoso B., García-Mejía J., Flores-Carapia R., Yáñez-Márquez. *Análisis del Modelo Cognitivo de Aprendizaje Automático "Memorias Asociativas" para Recuperación de Patrones*. Enviado a la revista: Interciencia. ISSN: 0378-1844.

Otros Trabajos Relacionados con la Tesis

- Luna Benoso, B., Yáñez Márquez, C., Figueroa Nazuno, J., López Yáñez, I., (2007). *Cellular Mathematical Morphology*, IEEE Computer Society, ISBN: 978-0-7695-3124-3, pp. 105-112.
- Luna Benoso, B., Yáñez Márquez, C., Figueroa Nazuno, J., López Yáñez, I., (2007). *Sixth Mexican International Conference on Artificial Intelligence*. Aguascalientes, México.
- Luna Benoso B., (2008). Conferencia: *Nuevo enfoque de la Morfología Matemática desde el punto de vista de los Autómatas Celulares*. UAEM Valle de México.
- Luna Benoso B., (2007). Conferencia: *Modelos Celulares Alfa-Beta*. Cuarta Semana de Ingeniería en Sistemas Computacionales. Tecnológico de Estudios Superiores de Jocotitlan, Edo. de México.
- Gómez Martínez M, Maldonado Mendoza P, Luna Benoso B., (2010). Conferencia: *Aplicaciones de las Memorias Asociativas alfa-beta en la resolución de problemas que requieran la clasificación de datos*. 3era Semana de Ciencias de la Informática. Instituto Leonardo Bravo.

- Luna Benoso B., (2009). Conferencia: *Clasificación de patrones por medio de memorias asociativas*. Universidad icel campus la villa. 5ta semana de conferencias de ciencia y tecnología. Noviembre.

Otros Trabajos

- Martínez Perales J., Barrales López A., Luna Benoso B., (2009). *Modelación y Simulación de Flujo Vehicular por medio de Autómatas Celulares*. XI congreso nacional de ingeniería electromecánica y de sistemas. Noviembre.
- Martínez Chávez A., Luna Benoso B., Flores Carapía R., (2009). *Simulación de un Nuevo Modelo del Virus del VIH en tres dimensiones implementado mediante autómatas celulares*. 5to. Congreso Internacional de Tendencias Tecnológicas en Computación. Celebrado en el CIDETEC del IPN. México D.F. Octubre.

Referencias

- [1] Von Neumann, J. (1966). Theory of Self-Reproducing Automata. Universidad de Illionis Press. Urbana and London. Consulta en línea: <http://www.walenz.org/vonNeumann/page0001.html>
- [2] Das, A. K., Ganguly, A., Dasgupta, A., Bhawmik, S., Chaudhuri, P. P. (1990). Efficient characterisation of cellular automata. Computers and Digital Techniques, IEE Proceedings E, Issue 1, ISSN: 0143-7062, pp. 81-87.
- [3] Martin, O., Odlyzko, A., Wolfram, S. (1984). Algebraic properties of cellular automata. Communications in Mathematical Physics 93 (1984) 219-258, Reprinted in Theory and Applications of Cellular Automata, (1986), pp. 51-90.
- [4] Hedlund, G. A. (1969). Endomorphisms and automorphisms of full shift dynamical system. Theory of Computing Systems, Springer New York, ISSN: 1432-4350, pp. 320-375.
- [5] Gardner, M. (1970). Mathematical Games - The fantastic combinations of John Conway's new solitaire game Life. Scientific American, pp. 120-123.
- [6] Guo, Y., Walters, G. A., Khu, S. T., Keedwell, E. (2007). A novel cellular automata based approach to storm sewer design. Engineering Optimization 39 (3), pp. 345-364.
- [7] Fúster-Sabater, A., Guía-Martínez. (2007). Modelling nonlinear sequence generators in terms of linear cellular automata. Applied Mathematical Modelling 31 (2), pp. 226-235.

-
- [8] Nandi, S., Kar, B. K., Chaudhuri, P. P. (1994). Theory and applications of cellular automata in cryptography. *IEEE Transactions on Computers* 43, pp. 1346-1357.
- [9] Wolfram, S. (1986). Random sequence generation by cellular automata. *Advances in Applied Mathematics* 7, pp. 123.
- [10] Seredynski, P., Bouvry, P., Zomaya, A. (2004). Cellular Automata Computations and Secret Key Cryptography. *Parallel computing* 30, Elsevier, pp. 753-766. .
- [11] Schadschneider, A. (2006). Cellular automata next term models of highway previous term traffic next term. *Physica A, Statistical Mechanics and its Applications*, Volume 372, Issue 1, pp. 142-150.
- [12] Corne, D. W., Frisco, P. (2008). Dynamics of HIV infection studied with cellular automata and conformon-P systems. *Biosystems*, Volume 91, Issue 3, pp. 531-544.
- [13] Gerlee, P., Anderson, A. R. A. (2008). A hybrid cellular automaton model of clonal evolution in cancer: The emergence of the glycolytic phenotype. *Journal of Theoretical Biology*, Volume 250, Issue 4, pp. 705-722.
- [14] Steinbuch, K. (1961). Die Lernmatrix. *Kybernetik*, vol. 1, no. 1, pp. 36-45.
- [15] Yáñez-Márquez, C., Díaz-de-León, J. L. (2001). *Memorias Morfológicas Autoasociativas*. No. 58 Serie VERDE, CIC-IPN, México.
- [16] Yáñez-Márquez, C., Díaz-de-León, J. L. (2003). *Memorias Autoasociativas Morfológicas min: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones*. IT-177, Serie Azul, ISBN 970-36-0036-0, CIC-IPN, México.
- [17] Yáñez-Márquez, C., Díaz-de-León, J. L. (2003). *Memorias Heteroasociativas Morfológicas max: condiciones suficientes para convergencia, aprendizaje y recuperación de patrones*. IT-174, Serie Azul, ISBN 970-36-0033-6, CIC-IPN, Mexico.

-
- [18] Yáñez-Márquez, C. (2002). *Memorias Asociativas basadas en Relaciones de Orden y Operadores Binarios*. Tesis doctoral, CIC-IPN, México.
- [19] Acevedo-Mosqueda, M.E., Yáñez-Márquez, C., López-Yáñez, I. (2007). Alpha-Beta Bidirectional Associative Memories: Theory and applications. *Neural Processing Letters*, 26(1), pp. 1-40.
- [20] Román-Godínez, I., Yáñez-Márquez, C. (2007). Complete recall on alpha-beta heteroassociative memory. *Lecture Notes in Computer Science*, Springer Heidelberg, LNCS 4827, pp. 193-202.
- [21] Yáñez-Márquez, C., Cruz-Meza, M. E., Sánchez-Garfias, F. A., López-Yáñez, I. (2007). Using alpha-beta associative memories to learn and recall RGB images. *Lecture Notes in Computer Science*, Springer Heidelberg, LNCS 4493, pp. 828-833.
- [22] Yáñez-Márquez, C., Sánchez-Fernández, L.P., López-Yáñez, I. (2006). Alpha-beta associative memories for gray level patterns. *Lecture Notes in Computer Science*, Springer Heidelberg, LNCS 3971, pp. 818-823.
- [23] Akin, H., Siap, I. (2007). On cellular automata over Galois rings. *Elsevier*, 103, pp. 24-27.
- [24] Poundstone, W. (1985). *The Recursive Universe*. William Morrow and Company Inc.
- [25] Wolfram, S. (1984). Cellular automata as models of complexity. *Nature* 311, pp. 419.
- [26] Korniyak V. V.. (2006). *Cellular Automata with Symmetric Local Rules*. *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, Volume 4194.
- [27] Wuensche, A. (1999). Classifying Cellular Automata Automatically: Finging Gliders, Filtering, and Relating Space-Time Patterns, Attractor Basins, and the Z Parameter. *Complexity* 4, pp. 47-66.

-
- [28] Jian-ping M., Shi-qiang D., Li-yun D., Jie-fang Z. (2007). Cellular automaton next term model for mixed previous term traffic flow next term with motorcycles. *Physica A: Statistical Mechanics and its Applications*, Volume 380, pp. 470-480.
- [29] Maerivoet, S., De Moor, B. (2005). Cellular automata term models of road traffic. *Physics Reports*, Volume 419, Issue 1, Pages 1-64.
- [30] Xiao, X., Shi-Huang S, Kuo-Chen C. (2006). A probability cellular automaton model for hepatitis B viral infections. *Biochemical and Biophysical Research Communications*, Volume 342, Issue 2, pp. 605-610.
- [31] Ying-Fan, Shang-Jun Y., Bing-Hong W., Yi-Ming W. (2008). The effect of investor psychology on the complexity of stock market: An analysis based on cellular automaton model. *Computers & Industrial Engineering*, In Press, Corrected Proof.
- [32] Hernández-Encinas, A., Hernández-Encinas, L., Hoya-White, S., Martín-del-Rey, A., Rodríguez-Sánchez, E. (2007). Simulation of forest fire fronts using cellular automata. *Advances in Engineering Software*, Volume 38, Issue 6, pp 372-378.
- [33] Matheron, G. (1975). *Random Sets and Integral Geometry*. Wiley, New York.
- [34] Serra, J. (1982). *Image Analysis and Mathematical Morphology*. Vol. I, Academic Press, London.
- [35] Serra, J. (1988). *Image Analysis and Mathematical Morphology*. Vol. 11: Theoretical Advances, Academic Press, London.
- [36] Dougherty, E. R., Lotufo, R. A. (2003). *Hands-on morphological image processing*. Vol TT59, spie press.
- [37] Aptoula, E., Lefèvre, S. (2007). A comparative study on multivariate mathematical morphology. *Pattern Recognition*.
- [38] Matheron, G. (1965). *Elements Pour une Tiorre des Mulieux Poreux*. Paris: Masson.

-
- [39] Bloch, I. (2008). Defining belief functions using mathematical morphology Application to image fusion under imprecision. *International Journal of Approximate Reasoning*, Volume 48, Issue 2, pp. 437-465.
- [40] Serra, J. (2008). *Advances in Mathematical Morphology: Segmentation*. *Advances in Imaging and Electron Physics*, Volume 150, pp. 185-219
- [41] Yu, Y. Y., Chen, H., Lin, C. H., Chen, C. M., Oviir, T., Chen, S. K. (2007). Digital imaging analysis with mathematical morphology and box counting for evaluation of reactive bone following endodontic treatment. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology, and Endodontology*, Volume 103, Issue 2, pp. e54-e55.
- [42] Willshaw, D., Buneman, O., Longuet-Higgins, H. (1969). Non-holographic asociative memory. *Nature*, no. 222, pp. 960-962.
- [43] Anderson, J. A. (1972). A simple neural network generating an interactive memory. *Mathematical Biosciences*, vol. 14, pp. 197-220.
- [44] Kohonen, T. (1972). Correlation matrix memories. *IEEE Transactions on Computers*, C-21, vol. 4, pp. 353-359.
- [45] Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, vol. 79, pp. 2554-2558.
- [46] Wickramasinghe, L. K., Alahakoon, L. D., Smith-Miles, K. (2007) . A novel Episodic Associative Memory model for enhanced classification accuracy. *Pattern Recognition Letters*, Volume 28, Issue 10, pp. 1193-1202.
- [47] Kosko, B., (1988). Bidirectional associative memories. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, no. 1, pp. 49-60.

-
- [48] Ritter, G. X., Sussner, P., Díaz-de-León, J. L. (1998). Morphological associative memories. *IEEE Transactions on Neural Networks*, 9, pp. 281-293.
- [49] Ritter, G. X., Dóaz-de-León, J. L., Sussner, P. (1998). Morphological bidirectional associative memories. *Neural Networks*, 12, pp. 851-867.
- [50] Acevedo-Mosqueda, M. E., Yáñez-Márquez, C., López-Yáñez, I. (2006). A new model of BAM: Alpha-beta bidirectional associative memories. *Lecture Notes in Computer Science*, Springer Heidelberg, LNCS 4263, pp. 286-295.
- [51] Flores-Carapia, R. (2006). Memorias Asociativas alfa-beta basadas en el código Jhonson-Möbius modificado. Tesis de Maestría. CIC-IPN.
- [52] Argüelles-Cruz, A. J., Lindig-Bos, M., Yáñez-Márquez, C. (2007). Redes Neuronales Alfa-Beta Weightless, "Tópicos Selectos en Ciencias de la Computación", Colección CIDETEC, ISBN: 978-970-94696-5-3, Instituto Politécnico Nacional, México, pp. 173-226.
- [53] López-Leyva, L. O. (2008). Máquinas Asociativas alfa-beta con soorte vectorial. Tesis Doctoral. CIC-IPN.
- [54] Ahmed, E., Elgazzar, A.S. (2007). Some mathematical results for cellular automata. *Physica A: Statistical Mechanics and its Applications* 373, pp. 354-362.
- [55] Adamatzky A., Costello B., Bull L., Stepney S., Teuscher C. (2007). *Unconventional Computing 2007*. ISBN-10: 1-905986-05-X Luniver Press.
- [56] Klar, A., Kuhne, D., Wegener, R. (1996). Mathematical Models for Vehicular Traffic. *Surv. Math. Ind.*, 6, pp. 215-239.
- [57] Karim-Daoudia, A., Najem-Moussa. (2003). Numerical Simulations of a Three-Lane Traffic Model Using Cellular Automata. *Chinese Journal of Physic*. Vol. 41, No. 6.
- [58] Wolfram, S. (2002). *A New Kind of Science*. ISBN: 1-57955-008-8.

-
- [59] Luna-Benoso B. (2006). Análisis y Generalización de un Modelo de Tráfico Vehicular usando Autómatas Celulares. Tesis de Maestría. CIC-IPN.
- [60] Comer, M., Delp, E. (1999). Morphological operations for color image processing. *J. Electron. Imaging* 8 (3), pp. 279289.
- [61] Goutsias, J., Heijmans, H. J. A. M, Sivakumar, K. (1995). Morphological operators for image sequences. *Comput, Vision Image Understanding* 62 (3), pp. 326346.
- [62] Heijmans, H. J. A. M., Ronse, C. (1990). The algebraic basis of mathematical morphology. Part I: dilations and erosions, *Comput. Vision Graphics Image Process.* 50 (3), pp. 245295.
- [63] Ronse, C., Heijmans. H. J. A. M. (1991). The algebraic basis of mathematical morphology. Part II: openings and closings, *Comput. Vision Graphics Image Process.* 54 (1), pp. 7497.
- [64] Ledda, A., Philips, W. (2005). Majority ordering for colour mathematical morphology. In: *Proceedings of the 13th European Signal Processing Conference, Antalya, Turkey.*
- [65] Burgeth, B., Bruhn, A., Papenberg, N., Welk, N., Weickert, N. (2007). Mathematical morphology for matrix fields induced by the loewner ordering in higher dimensions. *Signal Process.* 87 (2), pp. 277290.
- [66] Díaz-de-León, J. L., Yáñez-Márquez, C. (2003). *Introducción a la morfología matemática de conjuntos.* México.
- [67] Wennekers T., Sommer, F. T., Palm G. (1995). Iterative retrieval in associative memories by threshold control of different neural models. In *Supercomputers in Brain Research: From Tomography to Neural Networks*, Eds. H. J. Herrmann, D. E. Wolff, E. Pöppel. World Scientific, Singapoore, pp. 301-319.
- [68] Hassoun, M. H. (1993). *Associative Neural Memories.* Oxford University Press, New York.

-
- [69] Kohonen, T. (1989). *Self-Organization and Associative Memory*. Springer-Verlag, Berlin.
- [70] Sánchez-Garfias, F.A., Díaz-de-León, J.L, Yáñez-Márquez, C. (2005). A new theoretical framework for the Steinbuch's Lernmatrix. In *Proc. Optics & Photonics 2005, Conference 5916 Mathematical Methods in Pattern and Image Analysis*, SPIE (International Society for Optical Engineering), San Diego, CA.
- [71] Orovas, C., Austin, J. (1998). Cellular Associative Symbolic Processing for Pattern Recognition. *MFCS Workshop on Grammar Systems*, pp. 269-279.
- [72] Maji P., Chaudhuri, P. (2008). Non-uniform cellular automata based associative memory: Evolutionary design and basins of attraction. *Information Science*. Elsevier.
- [73] Ganguly N., Maji P., Das A., Silkdar B., Chaudhuri P. (2002). *Characterización of Non-linear Cellular Automata Model for Pattern Recognition*. Springer-Verlag, Berlin Heidelberg.
- [74] Maji P., Ganguly N., Saha S., Roy A., Chaudhuri P. (2002). *Cellular Automata Machine for Pattern Recognition*. Springer-Verlag, Berlin Heidelberg.
- [75] Maji P., Chaudhuri P., (2005). Fuzzy Cellular Automata Based Associative Memory for Pattern Recognition. *Proceedings of ICISIP*.
- [76] Maji P., Ganguly N., Chaudhuri P., (2003). Error Correcting Capability of Cellular Automata Based Associative Memory. *Systems and Humans*. IEEE, vol 33, No 4.
- [77] Maji P., Chaudhuri P. (2005). Basins of Attraction of Cellular Automata based Associative Memory and Its Rule Space. *Systems and Humans*. IEEE Computer Society Washington, DC, USA, ISBN:0-7803-9588-3, pp. 115-120.
- [78] Reynaga R., Amthauer. (2003). Two-dimensional cellular automata of radius one for density classification task $\rho = 1/2$. *Pattern Recognition Letters* 24, ELSEVIER, pp. 2849-2856.

-
- [79] Luna-Benoso, B., Yáñez-Márquez, C., Figueroa-Nazuno, J., López-Yáñez, I., (2007). Sixth Mexican International Conference on Artificial Intelligence - special session-. Edited by Alexander Gelbukh and Angel Fernando Kuri Morales. ISBN: 978-0-7695-3124-3, pp 105.
- [80] Díaz-de-León, J. L., Yáñez Márquez, C. (2003). Introduccin a la morfología matemática de conjuntos. Colección de Ciencia de la Computación, CIC-IPN-UNAM-FCE, México. ISBN: 970-36-0075-1.
- [81] Yáñez-Márquez, C., Díaz-de-León, J.L. (2003). Memorias Asociativas Basadas en Relaciones de Orden y Operaciones Binarias. Computación y Sistemas, Vol. 6, No. 4, México, ISSN 1405-5546, pp. 300-311.
- [82] Richard L. Burden, J. Douglas, F. (2002). Análisis Numérico. Thomson Learning, 7a. edición,
- [83] Kosniowski, C. (2003). Topología Algebraica. Editorial Reverte, ISBN: 84-291-5098-6.
- [84] Tenenbaum, A. M., Langsam, Y., Augenstein, M. A. (1997) Estructuras de Datos con C y C++. Prentice Hall, México. 2 edición, ISBN: 968-880-798-2.
- [85] Vila, L., Viguera, J., Alemán, R. (2008). Retinopata diabtica y ceguera en Espaa. Epidemiologa y prevencin.
- [86] Bustos, M., Quiñones, N., Flores, R., Luna-Benoso, B. (2010). Clasificación De Padecimiento De Retinopatía Diabética Mediante Segmentación De Imágenes y Memorias Asociativas. Trabajo Terminal: 20090141.
- [87] Center for Machine Learning and Intelligent Systems. Machine Learning Repository. <http://archive.ics.uci.edu/ml/>. Irvine Machine Learning Repository. University of California, Irvine.
- [88] Kotsiantis, S. B., Pintelas, P. E., (2005). Logitboost of Simple Bayesian Classifier. Informatica.

-
- [89] Camastra, F., Verri, A., (2005). A Novel Kernel Method for Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 27, NO. 5.
- [90] Wickramasinghe, L. K., Alahakoon, L. D., Smith-Miles, K., (2007). A novel Episodic Associative Memory model for enhanced classification accuracy. *Pattern Recognition Letters* 28, ELSEVIER, pp. 11931202.
- [91] Kaynak, C., Alpaydin, E., (2000). MultiStage Cascading of Multiple Classifiers: One Man's Noise is Another Man's Data. *ICML 2000*, pp. 455-462.
- [92] Abdullah, M.R., Toh, K. A., Srinivasan, D., (2006). A Framework for Empirical Classifiers Comparison. *1st IEEE Conference on Industrial Electronics and Applications ICIEA 2006*.
- [93] Carvalho, D. R., Freitas, A. A., (2004). A hybrid decision tree/genetic algorithm method for data mining. *Inf. Sci.* 163(1-3), ELSEVIER, pp. 13-35.
- [94] Luukka, P., (2009). Similarity classifier using similarities based on modified probabilistic equivalence relations. *Knowl.-Based Syst.* 22(1), ELSEVIER, pp. 57-62.
- [95] Rokach, L., (2008). Genetic algorithm-based feature set partitioning for classification problems. *Pattern Recognition* 41(5), ELSEVIER, pp. 1676-1700.