

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Informática

Enumeración de árboles

Tesis

Que como parte de los requisitos para obtener el grado de

Maestro en

Ciencias Computacionales

Presenta

Verónica López Blancas

Querétaro noviembre del 20

No. Adq. H64849

No. Título

Clas. T3

511.5

L864e

Ej. 01



Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Ciencias Computacionales

ENUMERACIÓN DE ÁRBOLES

TESIS

Que como parte de los requisitos para obtener el grado de

Maestro en Ciencias

Presenta:

L.I. Verónica López Blancas

Dirigido por:

Dr. Jaime Rangel Mondragón

SINODALES

Dr. Jaime Rangel Mondragón
Presidente

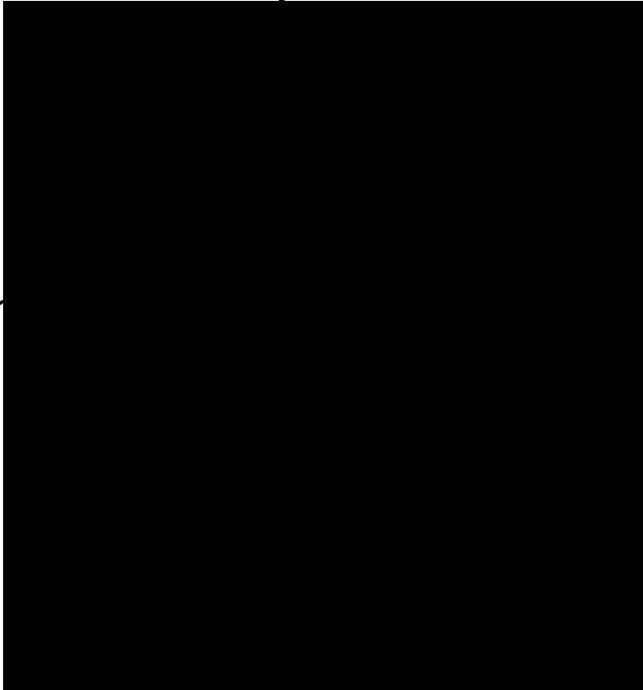
M. en C. Arturo González Gutiérrez
Secretario

M. en C. Rosa María Romero González
Vocal

Dr. Ivan Terol Villalobos
Suplente

M. en C. Vicente Rodríguez Hernández
Suplente


I.S.C. Teresa Guzmán Flores
Directora de la Facultad de Informática


Dr. Sergio Quesada Aldana
Director de Investigación y
Posgrado

Centro Universitario
Querétaro, Qro.
noviembre del 2000
México

RESUMEN

Existen tres razones para que la enumeración de árboles sea analizada: Primero, porque la teoría de grafos constituye el desarrollo más importante que ha tenido la combinatoria desde 1920, segundo, los grafos en general, sirven como modelos abstractos para explicar una gran variedad de relaciones entre objetos del mundo real y su estructura algorítmica; tercero, porque los árboles constituyen la estructura más importante en aplicaciones computacionales y su taxonomía influye en la asignación y análisis de recursos en dichas aplicaciones. El objetivo de esta tesis es analizar las estructuras arborescentes mediante su enumeración para conocer el número, tipo y el comportamiento de estas estructuras con el fin de incursionar en otras disciplinas que se benefician del enfoque informático. Con este propósito se hace un estudio del origen del concepto de árbol y se analiza la relación del modelo matemático con la representación computacional dentro de un paradigma funcional. A continuación se analiza y define la estructura de sus componentes así como de los árboles de mayor interés, programando las funciones más importantes e incluyendo la enumeración de los árboles libres, binarios y trivalentes, así como su eficiencia. La enumeración de los árboles es un tema poco explorado y su importancia radica en que permite una gama de diferentes decisiones, con base a un modelo fundamental. Al final de este trabajo se muestra una aplicación al diseño de estrategias por Elección Racional para el combate a la pobreza.

(Palabras clave: árbol, enumeración, paradigma, pobreza, elección racional)

SUMMARY

There are three reasons why the enumeration of trees is worth analyzing: first the graph theory comprises the most important development the field of combinatorics has undergone since 1920; secondly, in general graphs are useful as abstract models in a variety of relationships involving real-world objects and their algorithmic structure; finally, the trees constitute the most important structure in computational applications and their taxonomy influences the assignment and analysis of resources in those applications. The objective of this thesis is to analyze tree structures through their enumeration with the two fold aim of knowing the type and behavior of these structures and to apply them in other disciplines which would benefit from the computational point of view. With this, a study of the concept of tree is undertaken, along with the analysis of the relation of the mathematical model and the computational representation within a functional framework. Following this, the structure of its components is defined and analyzed along with the most interesting trees, programming the most important functions including the enumeration of tree, binary and trivalent trees and their efficiency. The enumeration of trees is narrowly explored theme and its importance hinges on the big amount of different decisions it allows based on a fundamental model. At the end of his work we show an application on the design of strategies towards, attacking poverty by means of rational elections.

Keyword: tree, enumeration, paradigm, poverty, rational election

DEDICATORIAS

Al Dulce niño de Praga

A mi madre:

Por que con tu ejemplo de amor a la vida, valor y entrega desinteresada, te convertiste en mi inspiración para lograr todas las metas de mi vida.

A mi esposo:

Al hombre que con sus defectos me ha enseñado mis fortalezas y porque con sus virtudes me ha hecho inmensamente feliz.

A los Martínez:

Por que Violeta es la flor de mi vida y Julio Alberto, la luz de alma.

A la familia

Luis, Carlos, Germán, María Fernanda, Poli, Mariana, Marisol, Víctor.

AGRADECIMIENTOS

Para terminar con éxito ésta tesis agradezco a los universitarios que desinteresadamente guiaron mi formación:

Al M. en C. Alfredo Zepeda Garrido, rector de nuestra máxima casa de estudios en el periodo lectivo 1994-2000 por su compromiso con los universitarios para apoyarlos en su constante superación.

A la M. en C. Rosalba Rodríguez Durán, Coordinadora de la Carrera de Ciencias Políticas y Administración Pública, maestra y amiga por brindarme su talento en todo el proceso de la maestría.

Al Dr. Carlos Dorantes González, Coordinador de la creación de la carrera de Sociología y exdirector de la Facultad de Ciencias Políticas y Sociales, Director del periódico Tribuna Universitaria, quien me impulsó a desarrollarme profesionalmente, así como por brindarme su apoyo para iniciar estudios de posgrado.

Al L.I. Jaime Alberto Martínez por su valiosa aportación a mi trabajo, así como por su invaluable ayuda.

Al Dr. Jaime Rangel Mondragón, Investigador internacional de las ciencias de la computación, por brindarme su talento en la dirección de ésta tesis y por su paciencia a lo largo de toda la formación de posgrado.

Al M. en C. Arturo González, Maestro investigador, Coordinador de la creación de la Maestría en Ciencias Computacionales, U.A.Q., le doy gracias por su fe depositada en mí.

A la M. en C. Rosa María Romero González, egresada de la primera generación y la primera en obtener el grado en Ciencias computacionales, en la U.A.Q., por su dedicación y compromiso en la revisión de esta tesis.

Al Dr. Iván Terol Villalobos, Investigador y conferencista, pionero de la morfología matemática y sus aplicaciones en México por sus comentarios al modelo de decisiones, los cuales representaron una contribución esencial al desarrollo de la presente tesis.

Al M. en C. Vicente Rodríguez Hernández, investigador de virus computacionales, por las reflexiones brindadas a la tesis en general, que redundaron en un trabajo de mayor calidad.

A Martha Laura Ocampo Rojas, por su ayuda a lo largo de la maestría.

Un agradecimiento de manera especial, reiterando mi admiración y respeto al

M. en C. José Luis Ruíz Gutiérrez

actual Secretario de Extensión Universitaria, de la U.A.Q.

Quien con su apoyo y consejo hizo posible la realización de esta tesis;

gracias por ser fuente permanente de inspiración,

al permitirme escuchar su particular visión de las ciencias sociales

y por dedicarme su tiempo y espacio,

para lograr un sueño común:

Un modelo matemático aplicado a las ciencias sociales.

ÍNDICE

	Página
Resumen	i
Summary	ii
Dedicatorias	iii
Agradecimientos	iv
Índice	vii
Índice de tablas	xii
Índice de figuras	xiv
Símbolos	xvii
I. INTRODUCCIÓN	1
II. ÁRBOLES, MODELO MATEMÁTICO Y REPRESENTACIÓN COMPUTACIONAL	
II.1 Origen del modelo matemático de los árboles	5
II.2 Árboles, estructura de información	8
II.3 Relación entre el modelo matemático de los árboles y su representación computacional	10
II.4 Comparación de las estructuras de datos tradicionales para representar árboles	12
II.4.1 Representación tradicional de los árboles	12
II.4.2 Comparación de las estructuras de datos en la representación de árboles.	13
II.4.3 Análisis de paradigmas de programación	13
II.4.3.1 Evaluación de postulados de los paradigmas	14
II.4.4 Lenguaje <i>Mathematica</i> ®	16
II.4.5 Comparación de las estructuras de datos para representar árboles.	16

III.	ÁRBOLES, CONCEPTOS Y ALGORITMOS BÁSICOS	
III.1	Fundamento matemático	17
III.2	Definiciones de árboles	18
III.3	Propiedades de los árboles	19
III.3.1	Trayectorias	19
III.3.1.1	Relación de alcanzabilidad	20
III.3.1.2	Relación de conectividad	22
III.4	Funciones estructurales	24
III.4.1	Número de nodos	26
III.4.2	Número de hojas	27
III.4.3	Altura	27
III.4.4	Ancho	28
III.4.5	Profundidad	28
III.5	Representación de árboles	29
III.5.1	Pares ordenados	29
III.5.2	Pares ordenados invertidos	29
III.5.3	Notación corta de pares ordenados	30
III.5.4	Representación matricial	30
III.5.5	Representación gráfica	32
IV	ÁRBOLES CON RAÍZ	
IV.1	Árbol binario	33
IV.1.1	Árboles de búsqueda inorden, preorden y posorden	33
IV.2	Árbol n-ario	36
IV.3	Árbol n-ario completo	36
IV.3.1	Árbol balanceado	37
IV.3.2	Árbol extendido	37
IV.4	Árboles usados en las ciencias sociales	37
IV.4.1	Árboles de decisión	37
IV.4.2	Árbol de Kuhn	38

V.	ENUMERACIÓN DE ÁRBOLES	
V.1	Análisis básico de componentes	39
V.1.1	Relación de los vértices y las aristas en un árbol	40
V.1.2	Relación del número de nodos en la construcción de árboles	40
V.1.3	Relación de los nodos ordenados por nivel	41
V.1.3.1	Relación de nodos etiquetados por nivel	42
V.2	Enumeración de árboles libres	44
V.2.1	Enumeración de árboles libres	44
V.2.2	Enumeración de arboles transversales	46
V.3	Enumeración de árboles con raíz y sus componentes	50
V.3.1	Número de árboles con raíz	50
V.3.2	Longitud de trayectoria interna en un árbol	52
V.3.2.1	Longitud media de trayectoria interna en un árbol	52
V.3.3	Longitud de trayectoria externa (E)	53
V.3.3.1	Longitud media de trayectoria externa en un árbol	54
V.3.4	Relación entre longitud interna y longitud externa para un árbol binario	54
V.3.5	Relación entre longitud interna y longitud externa para un árbol n-ario	55
V.3.6	Número máximo de nodos para un árbol binario completo, con altura i	56
V.3.7	Número máximo de nodos para un árbol n-ario completo, con altura i	57
V.3.8	Análisis del peor caso en las búsquedas de árboles binarios	58
V.3.9	Análisis del mejor caso en las búsquedas de árboles	59
V.4	Enumeración de árboles trivalentes	61
V.4.1	Triangulaciones	61
V.4.2	Asociación de factores	64
V.4.3	Correspondencia entre triangulaciones y Asociaciones	65

V.4.4	Enumeración de árboles trivalentes	67
V.4.5	Relación entre triangulaciones, árboles trivalentes y números binarios	70
V.5	Enumeración de árboles no isomorfos	72
VI.	LÍNEAS DE INVESTIGACIÓN	75
VI.1	Un método de análisis de variables sociológicas, basadas en árboles. Caso. El modelo de Elección Racional en el combate a la pobreza	76
VI.1.1	Síntesis de elección Racional	77
VI.2.1.1	Planteamiento del problema	78
VI.2.1.2	Justificación Teórica	78
VI.2.1.3	Objetivos	79
VI.2.1.4	Antecedentes	79
VI.2.1.5	Definición del problema	79
VI.2.1.6	Hipótesis	79
VI.2.1.7	Metodología	79
VI.2.1.7.1	Definición de pobreza	80
VI.2.1.7.2	Establecer el modelo	80
VI.2.1.7.2.1	Definir las variables que intervienen en el problema	80
VI.2.1.7.2.2	Delimitación del problema	81
VI.2.1.7.2.3	Construcción del modelo	81
VI.2.1.7.2.3.1	Proponer clases de familias	82
VI.2.1.7.2.3.2	Análisis de hipótesis. Construcción de hipótesis y calificación	86
VI.2.1.7.2.4	Pruebas.	95
VI.2.1.7.2.4.1	Determinar el comportamiento de las variables, por sistema de hipótesis	97
VI.2.1.7.2.5	Resultados	97
VI.2.1.7.2.5.1	Análisis de oportunidades	100
VI.2.1.8	Conclusiones	102

VII. CONCLUSIONES	103
VII. Trabajo futuro	105
BIBLIOGRAFÍA	106
APENDICES	
A. Revisión histórica de la Teoría de árboles	108
B. Logaritmos	110
C. Programa procedural de un árbol	111
D. Síntesis del manual PROGRESA	115
E. Muestra de combinación de variables.	117

INDICE DE TABLAS

Tabla	Página
1 <i>Relación del modelo matemático y la representación computacional</i>	11
2 <i>Autores más relevantes y lenguajes de programación que prefieren para representar árboles</i>	12
3 <i>Tabla comparativa de arreglos con apuntadores y Leftmost-child-right-sibling</i>	13
4 <i>Estructura de datos funcional para definir un árbol</i>	16
5 <i>Función MatrixPower</i>	22
6 <i>Función Potencias de una matriz</i>	23
7 <i>Notación extendida de arbolUno, arbolDos, arbolTres</i>	26
8 <i>Función Notación Corta</i>	26
9 <i>Función Número de Nodos</i>	27
10 <i>Función noHojas, la cual cuenta el número de hojas</i>	27
11 <i>Función altura</i>	28
12 <i>Función ancho</i>	28
13 <i>Función profundo</i>	29
14 <i>Función pares</i>	29
15 <i>Función paresinv</i>	30
16 <i>Función depares</i>	30
17 <i>Función matriz</i>	31
18 <i>Función dibuja árbol</i>	32
19 <i>Función postorden</i>	34
20 <i>Función preorder</i>	34
21 <i>Función inorder</i>	36
22 <i>Relación del número de vértices, grados internos y número de aristas</i>	40
23 <i>Relación del no. de vértices y del número de árboles no ordenados que se pueden generar</i>	41
24 <i>Número de vértices y relación de ordenación por permutación en la construcción de árboles</i>	41
25 <i>Función piso de logaritmos base 2</i>	43
26 <i>Relación de vértices con el número de árboles con raíz.</i>	50
27 <i>Composición de trayectorias</i>	52

INDICE DE TABLAS

Tabla		Página
28	<i>Cálculo de longitud de trayectoria interna</i>	52
29	<i>Cálculo de longitud de trayectoria externa</i>	53
30	<i>Relación de los nodos de un árbol binario y los nodos especiales</i>	54
31	<i>Relación del número máximo de nodos para un árbol binario completo, con altura i</i>	56
32	<i>Relación del número máximo de nodos para un árbol n-ario completo, con altura i</i>	57
33	<i>Relación de los nodos con el nivel y su relación con la trayectoria interna</i>	58
34	<i>Función para determinar el número de árboles no isomorfos</i>	72
35	<i>Función para representar árboles no isomorfos por matriz</i>	73
36	<i>Arboles no isomorfos de cuatro vértices en notación corta</i>	74
37	<i>Primera tipologías de familias</i>	83
38	<i>Segunda tipologías de familias</i>	84
39	<i>Tercera tipología de familias</i>	85
40	<i>Escala de calificación de variables</i>	86
41	<i>Tabla de hipótesis uno sobre tipologías de familias</i>	87
42	<i>Tabla de hipótesis dos sobre tipologías de familias</i>	88
43	<i>Tabla de hipótesis tres sobre tipologías de familias</i>	89
44	<i>Tabla de hipótesis cuatro sobre tipologías de familias</i>	90
45	<i>Tabla de hipótesis cinco sobre tipologías de familias</i>	91
46	<i>Tabla de hipótesis seis sobre tipologías de familias</i>	92
47	<i>Tabla de hipótesis siete sobre tipologías de familias</i>	93
48	<i>Tabla de hipótesis ocho sobre tipologías de familias</i>	93
49	<i>Función para graficar</i>	95
50	<i>Tabla de posibilidades</i>	97

INDICE DE FIGURAS

Figura		Página
1	<i>Red de comunicación negativa con una persona discapacitada</i>	7
2	<i>Las contiendas electorales modelada por Round-Robin</i>	7
3	<i>El fenómeno del poder, modelado por un modelo de árboles con raíz.</i>	8
4	<i>Árbol en la naturaleza</i>	8
5	<i>Estructura de Directorio</i>	9
6	<i>Tipología de red tipo estrella modelada por árbol</i>	10
7	<i>Representación del isobutano por un árbol libre</i>	10
8	<i>Representación gráfica de una relación</i>	11
9	<i>Universo de Paradigmas de Programación</i>	14
10	<i>Sociograma o grafo de relaciones de una comunidad</i>	21
11	<i>Relación de conectividad</i>	24
12	<i>Representación gráfica de arbolUno, arbolDos, arbolTres</i>	25
13	<i>Matriz que representa un árbol con raíz 4(1,2,5)</i>	31
14	<i>Resultado de la función dibuja un árbol</i>	32
15	<i>Relación de nodos ordenados por niveles de un árbol binario</i>	42
16	<i>Relación de nodos enumerados con el nivel correspondiente.</i>	43
17	<i>Ejemplo de 5 Hidrocarbonos saturados</i>	44
18	<i>Grafo de enlaces</i>	44
19	<i>Correspondencia de un árbol y una lista</i>	45
20	<i>Correspondencia de un árbol y una lista 111238</i>	45
21	<i>Función que determina el número de árboles libres</i>	46
22	<i>16 árboles construidos de 4 nodos</i>	47
23	<i>Grafo con cuatro nodos</i>	47
24	<i>Árboles transversales resultantes con cuatro nodos</i>	48
25	<i>Grafos que almacena la computadora para $n=6$</i>	49
26	<i>Árboles transversales construidos de 6 vértices</i>	49
27	<i>Ejemplo de un árbol T para calcular la longitud de trayectoria interna</i>	52

INDICE DE FIGURAS

Figura		Página
28	<i>Árbol extendido con nodos especiales</i>	53
29	<i>Longitud interna</i>	54
30	<i>Longitud externa</i>	54
31	<i>Ejemplo de relación interna y externa de un árbol n-ario</i>	56
32	<i>Casos de longitud interna máxima</i>	58
33	<i>Generalización de ruta interna de longitud máxima.</i>	59
34	<i>Ejemplos del mejor caso en las búsquedas de árboles binarios.</i>	60
35	<i>Ejemplo de un árbol para calcular el mejor caso en las búsquedas de árboles binarios</i>	60
36	<i>Triangulación de un cuadrado</i>	61
37	<i>Triangulación de un heptágono</i>	61
38	<i>Triangulaciones de un octágono</i>	62
39	<i>Ejemplo de un n-ágono</i>	63
40	<i>Correspondencia entre triangulaciones y asociaciones</i>	66
41	<i>Correspondencia entre triangulaciones de un octágono y las asociaciones de una lista</i>	66
42	<i>Árbol trivalente con tres vértices</i>	67
43	<i>Árbol trivalente con cuatro vértices</i>	67
44	<i>Árbol trivalente con cinco vértices</i>	67
45	<i>Árbol trivalente con seis vértices</i>	68
46	<i>Árbol trivalente con siete vértices</i>	68
47	<i>Árbol trivalente con ocho vértices</i>	69
48	<i>Asociación de un triángulo, un árbol trivalente y la generación de un número binario</i>	70
49	<i>Triangulación de un cuadrado, un árbol ternario y la generación de números binarios</i>	70
50	<i>Relación entre un pentágono, su triangulación, árboles ternarios y cinco números binarios.</i>	71
51	<i>Proceso de elección racional</i>	77
52	<i>Iteración de la pobreza con deseos y oportunidades</i>	81
53	<i>Modelo de pobreza</i>	82

INDICE DE FIGURAS

Figura		Página
54	<i>Grafica clases de familias</i>	96
55	<i>Comportamiento del rango en tercera dimensión, de la asignación de clases de familias.</i>	96
52	<i>Árbol de decisiones, del modelo racional en el combate a la pobreza.</i>	96
56	<i>Árbol de decisión para caso 1</i>	99
57	<i>Árbol de Kuhn que describe las estrategias contra la pobreza</i>	100
55	<i>Matriz de oportunidades</i>	99

SÍMBOLOS

\subseteq	<i>Subconjunto de</i>
$V \times V$	<i>Producto cartesiano, en V</i>
\emptyset	<i>Conjunto vacío</i>
\neq	<i>Diferente de</i>
\cap	<i>Intersección</i>
\in	<i>Símbolo de pertenencia</i>
R	<i>Relación</i>
\otimes	<i>Producto booleano</i>
\cup	<i>Unión</i>
\vee	<i>Conector lógico Or</i>
I_n	<i>Matriz identidad</i>
$G=(V,E)$	<i>G es un grafo donde V y E son conjuntos de vértices y aristas.</i>
$ x $	<i>Valor absoluto de x</i>
$!$	<i>Factorial</i>

I. INTRODUCCIÓN

La presente tesis titulada "Enumeración de árboles", se ubica en la rama fundamental de la Maestría en Ciencias Computacionales, de la Facultad de Informática, de la Universidad Autónoma de Querétaro. El objetivo de ésta tesis es: analizar las estructuras arborescentes mediante su enumeración para conocer el número, tipo y el comportamiento de éstas estructuras para la solución de problemáticas sociales.

Los árboles son tipos de datos dinámicos, es decir, que durante la ejecución del programa, éstos pueden cambiar en tamaño y forma [WIRTH N., 1987], esto manifiesta su capacidad computacional de ser estructuras de información [KNUTH, D. E, 1973]; es decir, su representación, el orden y las operaciones que pueden realizar con sus elementos generan algoritmos para que la información pueda ser:

- Analizada: Los analizadores sintácticos se basan en árboles gramaticales.
- Recuperada: Por ejemplo, en los accesos a disco y en la paginación se basan en algoritmos de árboles B
- Procesada: Un ejemplo en la interpretación de la información, son los árboles de Böhm, en lógica combinatoria, los cuales analizan los elementos del árbol y su relación entre sí.
- Ordenada y buscada: Por ejemplo, las búsquedas en inorden, preorden y algoritmo SORT [AHO, V. A., ULLMAN J., 1995].
- Interpretada: Manipulación de fórmulas algebraicas.

Los árboles no son las únicas estructuras de información, existen otras, por ejemplo: las listas lineales [AHO V. A., HOPCROFT J. E., ULLMAN, J. D., 1988]. El ejemplo más interesante de la aplicación de las listas lineales es el método de almacenamiento Hashing, el cual se utiliza para almacenar grandes cantidades de información [GRIMALDI, R. P. 1994]. Esta técnica crea un número de listas, y decide en que lista debe insertarse un nuevo elemento; se crea una función Hashing del conjunto de claves al conjunto de números en la lista. Las funciones

Hashing con frecuencia utilizan un módulo, el resultado entra a una lista y se inserta al final [Grimaldi 1984]. Los factores para determinar una buena función de Hashing, entre otras son: el número de registros a insertar y la técnica para localizar la lista. Debe observarse que la determinación de una función Hashing adecuada para alguna función específica en particular, es una tarea desafiante.

La razón por la que seleccionó el tema de los árboles, se centra en tres puntos que se describen a continuación y que dan testimonio de la importancia del tema.

La primera razón, es que la teoría de grafos, nace de la necesidad de solucionar problemas de combinatoria y también del análisis matemático de algoritmos fundamentales, de lo que ahora se conoce como ciencias de la computación; además, es el desarrollo más importante que ha tenido la combinatoria desde 1920. La característica principal de esta disciplina, radica en el hecho de que con los grafos se hacen representaciones matemáticas de un proceso, que se utiliza para analizar (simplificar y planificar) una realidad. En otras palabras pueden servir como modelos abstractos para explicar una gran variedad de relaciones entre objetos de un conjunto.

Es importante señalar que enfoque matemático y computacional se selecciona partiendo de la premisa de que la programación es un arte [Wirth 1987] para hacer un buen programa es necesario que éste represente un análisis basado en la realidad, junto con un análisis a partir de razonamientos matemáticos; tal y como lo afirma desde 1968, Donald Knuth.

La segunda es que las ciencias computacionales, hacen uso exhaustivo de ésta estructura para generar algoritmos en el análisis sintáctico, ordenamientos y búsquedas.

La tercera, es que la computadora digital ha demostrado ser un impulso constante para las aplicaciones de árboles, [GRIMALDI, 1994] ofreciendo sus resultados y aplicaciones, no solo en el campo de estudio de la computación, sino también en la química, física, biología y ciencias sociales.

La cuarta razón de tipo personal, se definió en la estrategia de tomar materias optativas como: Programación Funcional, Fractales y Autómatas Infinitos, Teoría Algebraica de Códigos, así como materias del área fundamental y las de la especialidad de Ingeniería de Software, de tal forma que ésta formación, sirviera para tener los conocimientos necesarios para profundizar en modelos matemáticos y entender o generar en el futuro aplicaciones en las ciencias sociales.

La presente tesis se integra de la forma siguiente: el segundo capítulo, presenta una recapitulación del modelo matemático de los árboles y su representación computacional. Cabe mencionar que hasta ahora, la mayoría de los análisis de los árboles se traducen en programación procedural y dado que la evolución y desarrollo de paradigmas es permanente, se analiza la factibilidad de la programación funcional, en esta aplicación.

El tercer capítulo muestra los árboles con vistas funcionales, es decir; un estudio de los principales componentes de los árboles acompañados de un algoritmo computacional para su mejor comprensión.

El cuarto capítulo trata sobre árboles n-arios, definiciones de los principales árboles con raíz y sus aplicaciones más importantes; este capítulo se integra debido al interés a nuevas líneas nuevas de investigación, que se mencionan en su apartado correspondiente.

El quinto capítulo, se centra en la enumeración, desde los árboles libres hasta la enumeración de árboles trivalentes; analizando la enumeración de sus componentes, así como la enumeración de la mejor búsqueda, la peor o el caso promedio.

El sexto capítulo, analiza brevemente las líneas tradicionales y las modernas de investigación, precisamente; como parte de éste último, se propone un Modelo de Elección Racional en Combate a la Pobreza; analizando las variables sociales sistemáticamente con un enfoque computacional.

El modelo muestra, una explicación científica de un problema social para la planeación de estrategias; su diseño se basa en tres puntos fundamentales.

Primero, la teoría de grafos ayuda a modelar y a comprender el fenómeno social de la pobreza.

Segundo, los árboles de decisiones se utilizan para la búsqueda de soluciones y la planeación de estrategias,

Por último, la teoría de elección racional, se toma como marco de referencia porque es una teoría planificadora de las ciencias sociales y con ella se interpreta los resultados derivados del modelo, cumpliendo con el objetivo social para lo cual fue diseñado.

Finalmente el capítulo siete ofrece las conclusiones de la tesis y una síntesis del trabajo futuro que ha de realizarse.

II. ÁRBOLES, MODELO MATEMÁTICO Y REPRESENTACIÓN COMPUTACIONAL

En las ciencias computacionales, el término “árbol” y/o su representación es conocida y aplicada a varias disciplinas como: estructura de datos [Wirth 1987], matemáticas discretas [Johnsonbaugh 1997], teoría de grafos [Chartrand 1977], combinatoria [Reingold 1977], telemática, teoría de autómatas y lenguajes [Kelley 1995], investigación de operaciones [Prawda 1989] y teoría de juegos [Shubik 1996].

El presente capítulo presenta el modelo matemático basado en el concepto anteriormente descrito; se verá como el modelo representa una condición o un conjunto de operaciones del mundo real y como ésta da origen a una estructura computacional; también se hace un análisis de la representación tradicional de los árboles y la propuesta de un paradigma llamado funcional para analizar el concepto de árboles y sus aplicaciones.

II.1 Origen del modelo matemático de los árboles.

De acuerdo a Chartrand la teoría de grafos, que estudia conjuntos de puntos llamados vértices o nodos y la relación de su conectividad con un conjunto de aristas, nace de la necesidad de solucionar problemas de combinatorias y también en el planteamiento de algunos acertijos matemáticos. La característica más importante de esta disciplina, radica en el hecho de que los grafos pueden servir como modelos abstractos para explicar una gran variedad de relaciones entre objetos de un conjunto.

Haciendo un análisis de las referencias principales sobre los orígenes, la teoría de grafos, se resume de forma siguiente: Chartrand en su obra hace un estudio muy completo sobre los modelos de dígrafos; y Fred nos muestra en un solo capítulo los dos temas de árboles dirigidos y no dirigidos, la mayoría de los autores como Knuth, Aho, y Grimaldi, hacen referencia a los grafos dirigidos y no dirigidos en capítulos por separado. Es decir, los estudios anteriores basan las diferencias de los grafos básicamente en la relación de cómo se unen las aristas a

los nodos o vértices. Por ejemplo si lo hacen de manera unidireccional, es decir; que el nodo A ésta relacionado con un nodo B se llama dirigido, por el contrario si la relación es bidireccional se llamará no dirigido.

Como resultado de éste análisis bibliográfico, se puede deducir, que:

- Los árboles pueden ser representados en los modelos de grafos, aquí representan por si mismos un subconjunto de relaciones cuya característica principal es que éstos subgrafos son aciclos y conexos.
- También en el modelo de digrafos, estudia los árboles como problemas de conectividad [Chartrand 1977].

Bajo los puntos anteriores, se propone la siguiente clasificación para entender el origen de los árboles.

La teoría de grafos se divide en dos grandes ramas.

1. Modelo de grafos.

- Modelo de árboles libres. Este modelo estudia grafos acíclicos conexos o mejor conocidos como árboles libres. La característica principal, del modelo es que sirve para representar relaciones simétricas entre objetos. Gracias a él, los científicos sociales pueden modelar relaciones sociales y representar en ellos fenómenos sociales como la influencia, el poder y el status [Roberts 1986]. La base para éste análisis social, se basa en redes de comunicación y de dominación [Klieiman 1985].

Este modelo se abordará en forma amplia más adelante, solo mencionaremos que es óptimo para la construcción de árboles abarcador o transversal de expansión mínima (Consultar V.2).

Una aplicación del modelo de grafos, se encuentra en el problema sociológico sobre el estudio de las redes de comunicación, la figura 1 muestra un ejemplo, de la relación de una persona discapacitada con los seres que le rodean, en éste caso las líneas punteadas significan una cadena de rechazo o una relación negativa.

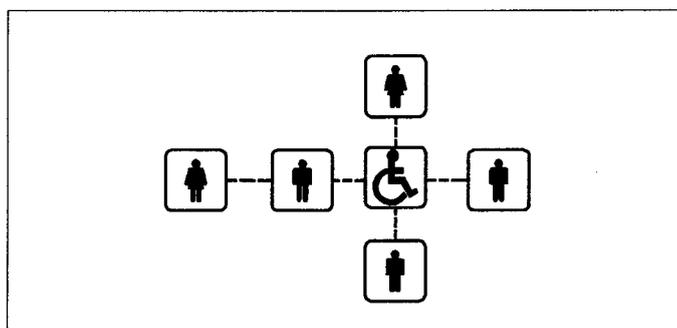


Figura 1. Red de comunicación negativa con una persona discapacitada.

2. Modelo de digrafos.

El modelo de digrafos se clasifica en dos grandes modelos, en Tournaments y en el modelo de digrafos.

- El Modelo de torneos. Básicamente puede verse como una competencia en la modalidad de *Round-Robin* (todos contra todos).

En las ciencias sociales puede verse la aplicación de éste modelo en el caso de las contiendas electorales. La figura 2 muestra un ejemplo de una contienda electoral entre tres partidos políticos. Éste modelo aparentemente sencillo al ser implementado en un modelo real de votaciones, tiende a tener un grado de complejidad mayor, ya que se ha comprobado que el resultado de una votación no sólo depende de una persona que desea una opción más que otra, sino también de segundas y de terceras preferencias; sin mencionar que, además, las relaciones se afectan por conductas insinceras o estratégicas de los políticos [Colomer 1990].

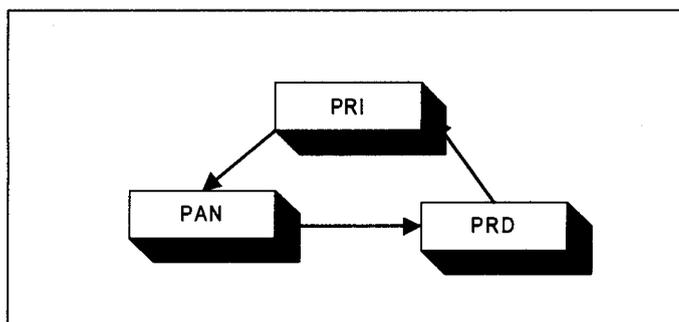


Figura 2. Las contiendas electorales modelada por Round-Robin

- Modelo de árboles con raíz. Representan estructuras de información mediante un grafo y la asignación de una función con un cierto orden. Cuando los fenómenos sociales como el poder y status [Roberts 1986], se derivan en una estructura jerárquica, éste fenómeno puede ser representado por éste modelo como lo muestra la figura 3.

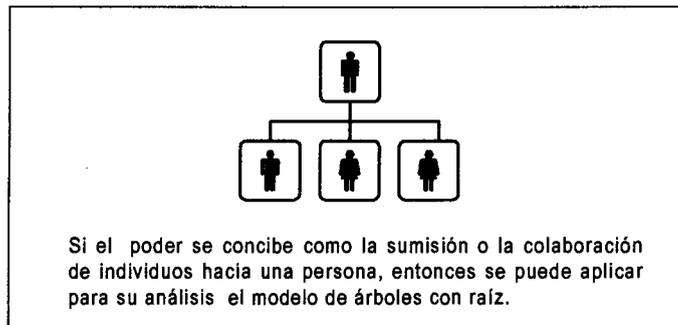


Figura 3. El fenómeno del poder modelado por un modelo de árboles con raíz.

II.2 Árboles, estructura de información.

En ciencias de la computación, un árbol tiene múltiples aplicaciones porque es una estructura información que se utiliza para representar relaciones entre datos. Cuando se hace referencia a éste término, se da la siguiente interpretación: Estructura de información dinámica; la cual por concepto de árbol, deriva una relación entre sus elementos [Aho 1995].

Como muestra la figura 4, los árboles en la naturaleza muestran dos tipos de elementos, las hojas unidas por una relación de conectividad representada por las ramas.



Figura 4. Árbol en la naturaleza

- Para G. H. Mealy, las estructuras de información son consideradas como colecciones de funciones, operaciones y transformaciones.

Como síntesis de las referencias más importantes, se puede decir que los árboles, son tipos de datos dinámicos, porque durante la ejecución del programa, éstos pueden cambiar en tamaño y forma [Aho 1988], esta característica manifiesta su capacidad computacional de ser estructuras de información [Aho 1995], es decir; su representación, el orden y las operaciones que se pueden realizar con sus elementos generan algoritmos, para que la información se pueda analizar, ordenar, recuperar, almacenar, procesar e interpretar.

Las estructuras de árboles se aplican por ejemplo, para representar fórmulas matemáticas y los sistemas operativos ofrecen una aplicación muy común en la representación de los directorios; la figura 5 muestra una estructura de directorio generada en el sistema operativo MS-DOS.

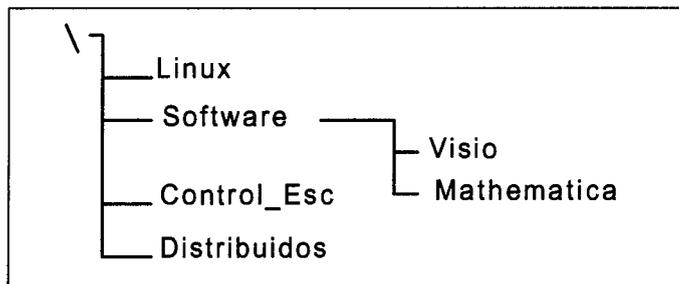


Figura 5. Estructura de Directorio

Otra aplicación en las ciencias de la computación se presenta en materias como telemática (Nombre colectivo de los servicios de información públicos para uso de casas y oficinas); en ésta disciplina algunas topologías o el patrón de interconexión de los nodos en las redes, pueden ser representadas por grafos, por ejemplo; las redes anulares o distribuidas suelen ser modeladas por un grafo, la red punto a punto es un árbol, las tipologías centralizadas pueden ser representadas por un árbol libre y las redes jerárquicas son un buen ejemplo de árbol con raíz [Madron 1992]. Un ejemplo se muestra en la figura 6, la cual ejemplifica la topología de estrella en una red de comunicación.

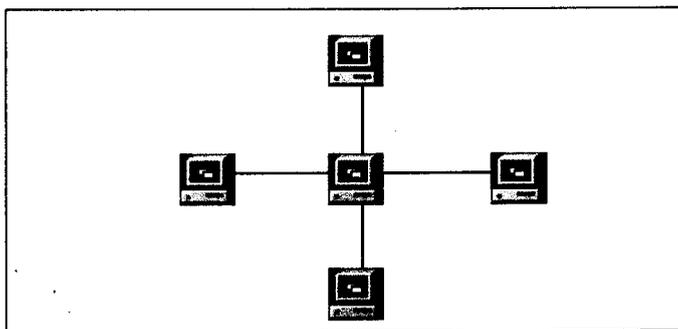


Figura 6. Tipología de red tipo estrella modelada por árbol

En el campo de la Química, también está presente la aplicación de los árboles en la enumeración de compuestos de hidrógeno y carbono llamados isómeros. La figura 7, muestra un ejemplo en donde puede observarse como un árbol libre representa a un isómero llamado isobutano

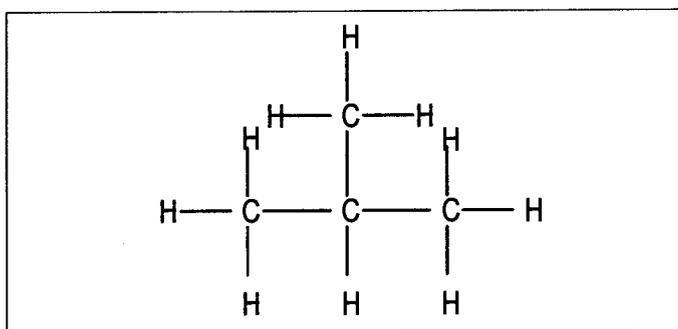


Figura 7. Representación del isobutano por un árbol libre.

II.3 Relación entre el modelo matemático de árboles y su representación computacional

Para mostrar la relación matemática entre el modelo de árboles y su representación computacional, partimos del siguiente ejemplo:

Sea A un conjunto de elementos y \mathfrak{R} una relación binaria [Grimaldi 1994]
 $A = \{a,b,c,d,e,f,g\}$ y una posible relación $\mathfrak{R} = \{[a,b],[a,c],[b,d],[b,e],[c,f],[c,g]\}$.

La representación gráfica de ésta relación se puede mostrar en la figura 8, en donde puede observarse el orden como se relacionan los elementos del conjunto

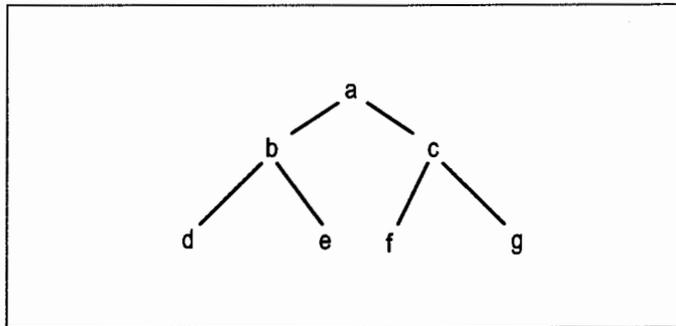


Figura 8. Representación gráfica de una relación.

De ésta representación gráfica de las estructuras arborescentes, se puede abstraer un modelo matemático, identificando los elementos que integran la estructura y definiendo la relación de los mismos.

Ahora bien la representación computacional se integra como sigue: La estructura de información, es interpretada por un lenguaje en una estructura de datos llamada registro, los nodos son celdas del registro, las flechas son siempre ligas de campo, la asignación son los datos del registro y el identificador es el apuntador del registro; a través de procedimientos o funciones esta estructura se enriquece hasta convertirse en una sistema de información.

Para resumir esta transformación del modelo matemático y la representación computacional, citaremos a [Plaft 1977] que lo resume como muestra la tabla 1.

Modelo matemático [abstracto]	Representación computacional
Nodos	Unidad mínima de la estructura de datos
Flechas	Ligas o apuntadores [relaciones]
Asignación	Datos, asignados a ese nodo
Identificador	Ligas, o apuntadores [relaciones] que tienen que ver con una celda en particular

Tabla 1. Relación del modelo matemático y la representación computacional

Se observa que ésta representación computacional debe modificarse; porque la representación fue elaborada en 1977 usando lenguaje ensamblador PL/1, FORTRAN, por lo que el modelo matemático está estrechamente unido con una estructura estática del registro.

II.4 Comparación de las estructuras de datos en la representación de árboles.

Para analizar la forma de representación de los árboles, en primera instancia se muestra una lista de los lenguajes que tradicionalmente se han utilizado para representar árboles, después se analizarán las estructuras de datos más importantes y se ubicará el paradigma de programación preferido de los autores. Posteriormente se comparará con los otros paradigmas y de esta forma poder elaborar un juicio sobre la forma de representación ideal para esta tesis.

La tabla 2, muestra una síntesis de los autores más relevantes y los lenguajes de programación que utilizan en el desarrollo de sus ejercicios para representar árboles.

Autor	Año	Lenguaje que usa para representar árboles
KNUTH	1968	Ensamblador
PLAFT	1977	Fortran
WIRTH	1987	Modula-2
AHO, ULLMAN	1988	Pascal
AHO, ULLMAN	1995	Lenguaje C

Tabla 2. Autores más relevantes y lenguajes de programación que prefieren para representar árboles.

De ésta tabla se observa que la mayoría de los autores utiliza un paradigma de programación procedural (consultar sección II.4.3); las tres secciones que se presentan a continuación hacen un análisis de los algoritmos de representación computacional más utilizados y su comparación con otros paradigmas.

II.4.1 Representación tradicional de los árboles

Las estructuras de datos más importantes para las representaciones de los árboles; independientemente del lenguaje y del tipo de árbol, son los arreglos con apuntadores y Leftmost-child-right-sibling(hijo más a la derecha) [Aho 1995]. Éstas estructuras que se detallan a continuación:

- Arreglos con apuntadores. Se declara una constante llamada branching factor (bf), que contiene el número máximo del arreglo a representar; una estructura que contenga a los datos y otra que se utiliza para manejar a los apuntadores.

La definición de la estructura de datos en C, es la siguiente:

```

Typedef struct Node *pNode;
Struct Node {
    Int info,
    PNode children[BF]};

```

- Leftmost-child-right-sibling (hijo más a la derecha). Ésta representación necesita apuntadores cuando se tiene a la derecha al menos un sucesor. Se puede crear de inmediato una lista ligada de los sucesores relacionados con ese nodo.

II.4.2 Comparación de las estructuras de datos para árboles del lenguaje procedural.

La tabla 3, muestra las ventajas y las desventajas de las estructuras antes mencionadas, en donde se puede observar por ejemplo que leftmost-child-right-sibling (hijo más a la derecha) necesita menos memoria y la manipulación de la estructura es más flexible.

Arreglos con apuntadores	Leftmost-child-right-sibling
Ventajas	
<ul style="list-style-type: none"> • Los arreglos necesitan un tiempo de $O[l]$ para encontrar a un sucesor, por lo que es muy fácil encontrar a los sucesores. 	<ul style="list-style-type: none"> • Necesitan menos memoria • La representación del arreglo de datos no tiene límite • La modificación de los arreglos de datos se hace sin modificar su estructura.
Desventaja	
<ul style="list-style-type: none"> • Requieren de gran cantidad de memoria. • Tienen una representación rígida de elementos, ya que el arreglo se predefine en las variables locales del programa. [branching factor] 	<ul style="list-style-type: none"> • Necesitan apuntadores para su manejo.

Tabla 3. Tabla comparativa de arreglos con apuntadores y Leftmost-child-right-sibling

II.4.3 Análisis de paradigmas de programación.

Se concluye que los lenguajes utilizados pertenecen al paradigma procedural. La evaluación sobre la forma tradicional de representar árboles

referida, nos obliga a cuestionar sobre la eficiencia de los lenguajes procedurales, para la representación computacional de los árboles.

El análisis que se propone se centra en los paradigmas. De acuerdo con [Sánchez, 1993] los seis paradigmas existentes son:

1. Lenguaje máquina y ensamblador
2. Procedural
3. Orientado a objetos
4. Funcional
5. Programación paralela
6. Herramientas de aplicación

La estructura utilizada por los paradigmas de programación, varía de acuerdo al lenguaje que se está utilizando, en la figura 9 es posible visualizar la definición de una estructura en un lenguaje funcional es diferente de su definición en un lenguaje procedural.

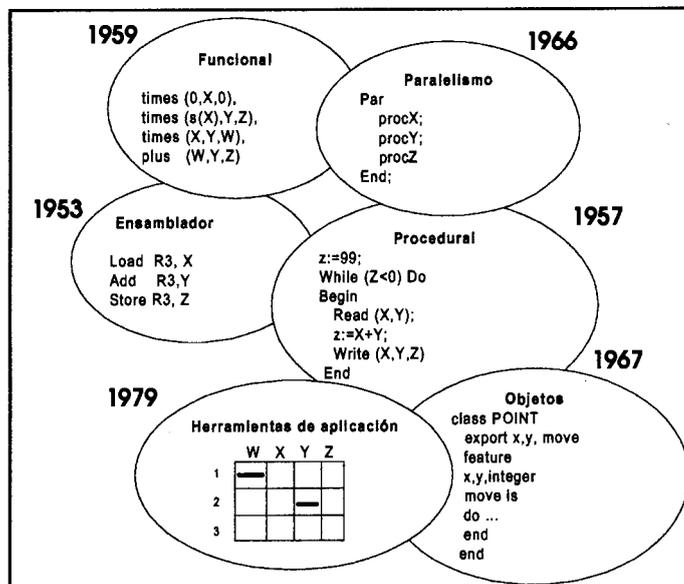


Figura 9. Universo de Paradigmas de Programación

II.4.3.1 Evaluación de postulados de los paradigmas

Los paradigmas son un conjunto de formas que sirven de modelo en una reflexión o pensamiento. Así pues, el paradigma de ensamblador no se abordará,

ya se necesitaría de un conocimiento más técnico para el manejo de apuntadores y sus direccionamientos a memoria, teniendo como desventaja hacer estática y rígida su manipulación de las estructuras arborescentes. Tampoco se estudia el paradigma de herramientas de aplicación, porque no se encontró ninguna que cubriera con los objetivos de la tesis. La programación paralela también se descarta, porque hasta el día de hoy es un verdadero reto su instrumentación.

Análisis de la evaluación de los paradigmas:

1. El paradigma procedural parte de la estructura de datos. En el caso de los árboles el tipo de datos es el registro, más algoritmos, da como resultado un programa. Es importante mencionar que éste paradigma se basa en una semántica, es decir; en las reglas lógicas que estudian el significado de las palabras, más que en la forma en que las palabras y los símbolos se unen en una sentencia (sintaxis), por eso existen lenguajes que aunque son típicamente procedurales es posible usar otro paradigma, como es el caso de C y sus variaciones como C++ y C Builder.
2. Orientado a Objetos. A partir de records + métodos = objetos. Los objetos son instrumentados como un tipo de dato complejo (record).
3. Funcional. Formas funcionales + expresión de expresiones= aplicaciones.

Los paradigmas procedural y orientado a objetos basan en su funcionamiento en registros, lo que hace rígida la definición de estructura. Este análisis se inclina por el paradigma funcional porque cuenta con las características que a continuación se describen:

- Recursividad. Hace uso extensivo de la recursividad.
- Las expresiones son el resultado del valor de sus subexpresiones y de los operadores correspondientes.
- Funciones primitivas. El lenguaje cuenta con un conjunto poderoso de funciones primitivas.
- Sintaxis sencilla. Uso de pocas reglas formales para la estructura de las sentencias.

- Equivalencia entre funciones y datos. Las funciones y los datos tienen características similares (valor, argumento de rutina, estructura).
- Modificación del código en ejecución. Por su naturaleza se requieren de pocas estructuras de programación secuencial, condicional y cíclica.

II.4.4 Lenguaje Mathematica

La alternativa óptima para representar las estructuras arborescentes es la programación funcional [Mclenan 1990], ya que proporciona las siguientes ventajas:

- No se necesita reservar memoria para cada inserción de un nodo.
- Los datos de los arreglos pueden variar, sin necesidad de modificar varias líneas del programa. Solo se modifica el argumento de entrada.
- Permite un manejo más eficiente de los recursos de la computadora. Por que Mathematica es por definición de los diseñadores de *Wolfram* un modelador y un analizador de datos.

II.4.5 Comparación de las estructuras de datos para representar árboles.

Para verificar las ventajas de utilizar programación funcional, obsérvese el programa elaborado en lenguaje C, que se muestra en el anexo D. El objetivo del programa es crear un árbol binario y leerlo en inorden, posorden y en orden. Como se puede observar el programa utiliza una librería llamada malloc, para reservar memoria, además; llama la librería llamada stdlib.h a memoria para ejecutar el programa.

En la programación funcional no sucede lo mismo, como puede observarse en la tabla 4, se presenta la estructura de un árbol en programación funcional.

Discriminador	<i>arbol = { {arbol..} } id</i>
Constructor	<i>arbol = { _, {arbol..} } id</i>
Selector	<i>preorden, posorden, inorden</i>

Tabla 4. Estructura de datos funcional para definir un árbol

III. ÁRBOLES, CONCEPTOS Y ALGORITMOS BÁSICOS

Los árboles son estructuras de información no lineales, analizadas en la ciencia de la computación se utilizan como modelo óptimo en el análisis matemático de relaciones.

En este capítulo se mostrarán definiciones de árboles y sus propiedades estructurales así como las representaciones más comunes de éstos; un enfoque computacional importante en ésta sección, es la enumeración o conteo de éstas estructuras arborecentes.

III.1 Fundamento matemático

La presente sección define los conceptos más importantes en el estudio de los árboles.

Definición 3.1.1 [Grimaldi 1994]

Sea V un conjunto no vacío y sea $E \subseteq V \times V$. El par (V, E) es llamado un **grafo dirigido** o **digrafo** donde V es el conjunto de vértices o nodos y E es un conjunto de aristas. Se escribe $G=(V, E)$ para denotar dicho digrafo.

Definición 3.1.2

Sean x, y dos vértices no necesariamente distintos de un grafo $G=(V, E)$. Una **trayectoria** x - y en G es una secuencia alternada finita donde:

$$x = x_0, e_1, x_1, e_2, x_2, e_3, \dots, e_{n-1}, x_{n-1}, e_n, x_n = y$$

De vértices y aristas de G , empezando en el vértice x y finalizando en y

Definición 3.1.3

Un **ciclo** es una trayectoria cerrada x - x . Un grafo **acíclico** no existen ciclos.

Definición 3.1.4

Un grafo $G=(V, E)$, se llama **conexo**, si existe un camino entre dos vértices distintos de G .

Definición 3.1.5

Si $G=(V,E)$ es un grafo (dirigido o no), entonces $G_1=(V_1,E_1)$ es llamado un **subgrafo** de G si $V_1 \subseteq V$ y $E_1 \subseteq E$, donde cada arista E_1 es incidente con los vértices de V_1 .

Definición 3.1.6

Si $G_1=(V_1,E_1)$ es un subgrafo y además cumple con las características de ser acíclico y conexo entonces decimos que es un **subárbol**.

Definición 3.1.7

Una **partición** o conjunto cociente de un conjunto no vacío A , es una colección de P subconjuntos no vacíos de A tales que:

1. Cada elemento de A pertenece a uno de los subconjuntos de P .
2. Si A_1 y A_2 son elementos distintos de P , entonces $A_1 \cap A_2 = \emptyset$

III.2 Definiciones de árboles.

Definición 3.2.1. [Roberts 1984]

Un **árbol**, es un grafo T , el cual es acíclico y conexo.

Definición 3.2.2 [Aho 1995]

Un **árbol** es un conjunto finito T de nodos tales que:

- 1) Hay un nodo especial llamado raíz
- 2) Los nodos restantes se particionan en subárboles

Definición III.3 BNF

De acuerdo a la notación BNF (Backus Normal Form, lenguaje de descripción de gramáticas) definida por [McLennan 1990] la estructura para expresar un **árbol** es la siguiente:

$$\text{arbol} = \{ \} \mid \{ \text{raiz}, \{ \text{arbol} _ \} \}$$

Como se puede observarse la primera definición no es recursiva; la segunda y la tercera definición de árbol lo son. La diferencia es que la definición BNF puede interpretarse computacionalmente como una lista de listas.

III.3 Propiedades de los árboles

En 1847, el físico Alemán Gustav Robert Kirchhoff examina un tipo de grafo muy especial, al cual denomina "árbol". Las propiedades que presenta este grafo son las siguientes:

1. Un grafo con n nodos y e ramas, es un árbol si y solo si $n=e+1$ [Grimaldi 1994]]
2. Si $x,y \in T$ existe solo una ruta o trayectoria de x a y .
3. Al remover una línea $e=(x_0, x_1) \in T$, éste se desconecta en dos subgrafos, los cuales también son árboles.

La tercera propiedad, por ejemplo, implica la partición de los árboles en subárboles; es decir, que los subárboles forman parte de un bosque. Para evitar ambigüedades, Harary en 1969, define a un grafo G como un árbol si y solo sí uno de los siguientes enunciados equivalentes es verdadero:

- a) G es conexo y acíclico.
- b) G es acíclico, pero al agregar una arista adicional se crea un circuito.
- c) G es acíclico y $|n|=|e|+1$.

III.3.1. Trayectorias

De la definición de árbol y sus propiedades; se observa que un árbol es una estructura, cuya característica principal es tener rutas (o trayectorias) únicas entre dos distintos vértices. En general, el concepto de trayectoria de alguna relación nos proporciona, herramientas para definir otras relaciones importantes, por ejemplo; la relación de conectividad y la relación de alcanzabilidad, las cuales se analizarán en las siguientes dos secciones.

III.3.1.1 Relación de alcanzabilidad

Ésta relación establece si existe una trayectoria de longitud n de x a y en R .

Definición 3.5.2 [Kolman, Busby, Ross 1995]

Relación de alcanzabilidad. Si n es un entero positivo, se define una relación R^n sobre un conjunto A como sigue: $xR^n y$ si hay una trayectoria de longitud n de x a y en R .

Para calcular la relación de alcanzabilidad, se observan los dos siguientes teoremas:

Teorema 1.

Si R es una relación sobre $A = \{a_1, a_2, \dots, a_n\}$, entonces $M_{R^n} = M_R \otimes M_R$

Teorema 2.

Para $n \geq 2$ y siendo R una relación sobre un conjunto finito A se tiene que $M_{R^n} = M_R \otimes M_R \otimes \dots \otimes M_R \otimes M_R$ (n factores)

Demostración

Por definición si el elemento c_{ij} en $M_R \otimes M_R$ es igual a 1 si y solo a_{ij} es igual a 1 y b_{ij} en M_R condición idéntica al producto booleano por lo que $M_{R^2} = M_R \otimes M_R$

Ahora, si $M_{R^2} = M_R \otimes M_R$ es verdadero, por inducción

$M_{R^n} = M_R \otimes M_R \otimes \dots \otimes M_R \otimes M_R$ (n factores) es verdadero [Kolman, Busby, Ross 1995].

Ejemplo:

Suponga que en una comunidad existen 9 personas, $p_1, p_2, p_3, \dots, p_9$, y se desea saber el grado de cohesión del grupo.

Se aplica un test sociométrico para identificar el grado de simpatía entre los individuos; el resultado se presenta en el sociograma mostrado en la figura 10. (Para más información sobre redes de comunicaciones consultar [Kleiman, Kleiman 1985])

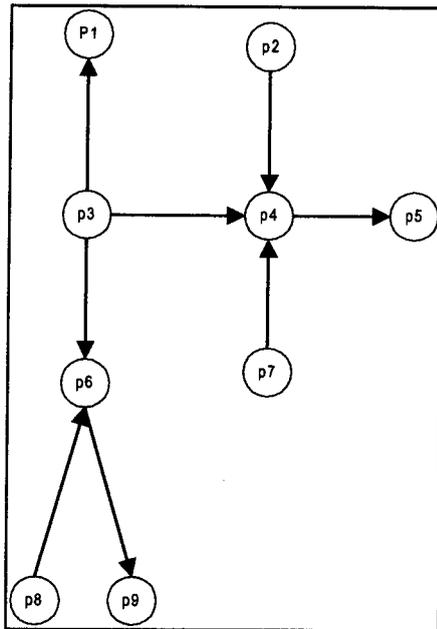


Figura 10. Sociograma o grafo de relaciones de una comunidad

Con el sociograma anteriormente descrito, deseamos saber:

- a. ¿Es un grupo articulado?
- b. ¿Quiénes son los puntos de articulación de la comunidad?

Respuesta.

Los puntos de articulación de un sistema, se definen como el vértice p_i que al ser removido forma una red disconexa que afectan a longitudes de alcanzabilidad mayor a 2; en otras palabras son las personas que al ser removidas del grupo tiene como resultado la pérdida de cohesión y por lo tanto son los puntos de vulnerabilidad del sistema [Kleiman, Kleiman 1985]

Para localizar los puntos de articulación, se calculará la alcanzabilidad para cada uno, la persona que tenga la trayectoria de alcanzabilidad mayor será el punto de articulación estratégico. Se calcula la potencia de la matriz n hasta que

resulten ceros y se toma la potencia n-1. Para el ejemplo anterior, la tabla 5 muestra la función MatrixPower que calcula directamente la matriz de n.

Código	<pre> {{0,0,0,0,0,0,0,0,0,0},{0,0,0,1,0,0,0,0,0,0},{1,0,0,1,0,1,0,0,0,0},{0,0,0,0,1,0,0,0,0,0},{0,0,0,0,0,0,0,0,0,0}, {0,0,0,0,0,0,0,0,0,1},{0,0,0,1,0,0,0,0,0,0},{0,0,0,0,0,1,0,0,0,0},{0,0,0,0,0,0,0,0,0,0}},2] //MatrixForm MatrixForm[MatrixPower[s,2]], MatrixForm[MatrixPower[s,3]], </pre>
Ejemplo	<div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;"> <pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 </pre> <p>Matriz R</p> </div> <div style="text-align: center;"> <pre> 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 </pre> <p>Matriz R cuadrada</p> </div> <div style="text-align: center;"> <pre> 0 </pre> <p>Matriz cúbica</p> </div> </div>

Tabla 5. Función MatrixPower de la figura 10.

El resultado de ésta matriz se interpreta de la manera siguiente: p_2, p_3, p_7, p_8 tienen longitud 2 (ver matriz R cuadrada), siendo los puntos de articulación; en otras palabras podemos decir que al remover a las personas 2,3,7, y 8 se fractura el grupo, convirtiéndolos en la vulnerabilidad del sistema.

La importancia de localizar puntos de vulnerabilidad en las redes de comunicación, tiene muchas aplicaciones, por ejemplo; una interpretación administrativa que puede dársele es que al renunciar alguna de éstas personas afectaría seriamente a la organización.

III.3.1.2 Relación de conectividad

En ésta relación se exhiben todos los vértices que pueden alcanzarse de x a y por alguna trayectoria en R , a continuación se presentan dos definiciones importantes y un ejemplo.

Definición.

Composición. Si A, B, C son conjuntos con $R_1 \subseteq A \times B$ y $R_2 \subseteq B \times C$, la relación compuesta $R_1 \circ R_2 = \{(x,z) | x \in A, z \in C \text{ y tal que } y \in B \text{ con } (x,y) \in R_1, (y,z) \in R_2\}$.

Definición 3.5.1

Relación de conectividad. Si i es un entero positivo, se define una relación $xR^i y$ si existe alguna trayectoria en R de x a y . La longitud de trayectoria i dependerá de x y y .

Demostración.

$$M_R^i = \bigcup_{k=1}^i M_k$$

La relación de conectividad depende de la longitud (i) de x a y si y solamente si

$$R^i = M_R \cup (M_R)^2 \cup (M_R)^3 \dots (i \text{ veces})$$

ó

$$M_R^i = I_n \cup M_R \cup (M_R)^2 \cup (M_R)^3 \dots (\text{hasta la longitud de trayectoria deseada})$$

Debe recordarse que I es una matriz de orden n , Si $A_{m \times n}$ $I_m A = A I_n = A$ por lo que $A^0 = I_n$. El proceso para trabajar con las potencias de una matriz fue concebida por Cayley-Hamilton [Howar 1988].

$$C_0 + C_1 \lambda + C^2 = 0$$

$$C_0 I + C_1 A + A^2 = 0$$

$$A^2 = -C_1 A - C_0 I$$

si se multiplica lo anterior por A

$$A^3 = -C_1 A^2 - C_0 A$$

lo que expresa a A^3 en términos de A^2 y A^1 . Por ejemplo sea una matriz de $n=8$ elementos, se elabora la función 6.

Código	<pre>n=8; A=Table[Random[Integer,{0,5}],{n},{n}];MatrixForm[A] b=Eigenvalues[A]//N; p=Chop[Expand[Times@@(x-b)]] 1524IdentityMatrix-341A+10A.A+78A.A.A-44A.A.A.A-16A.A.A.A.A- 17A.A.A.A.A.A+A.A.A.A.A.A.A//MatrixForm</pre>
Función	Función potencias de una matriz tienden a cero

Tabla 6. Función Potencias de una matriz

Ejemplos	$ \begin{array}{cccccccc c} & 2 & 0 & 3 & 0 & 1 & 3 & & \\ 3 & 0 & 1 & 4 & 3 & 5 & 2 & 3 & \\ 5 & 3 & 4 & 5 & 0 & 5 & 3 & 2 & \\ 5 & 1 & 2 & 1 & 2 & 2 & 0 & 2 & \\ 5 & 2 & 4 & 2 & 5 & 5 & 2 & 4 & \\ 1 & 4 & 4 & 1 & 4 & 2 & 2 & & \\ 4 & 5 & 3 & 1 & 5 & 3 & 0 & & \\ 2 & 2 & 1 & 5 & 0 & 0 & & & \\ \hline \end{array} $
	$ 1524. - 341.x + 10.x^2 + 78.x^3 - 44x^4 - 16.x^5 - 38.x^6 - 17.x^7 + x^8 $
	$ \begin{array}{cccccccc c} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \hline \end{array} $

Tabla 6. Función Potencias de una matriz (continuación)

En resumen, M_R^i trata de encontrar la composición de las relaciones que van de x a y.

Ejemplo:

Con el sociograma mostrado en la figura 10, se desea saber si existe un líder en el grupo.

Respuesta.

Si el liderazgo se define como la persona que influye en la mayoría para tomar decisiones, entonces se calculará la conectividad para cada uno, la persona que tenga la trayectoria de conectividad mayor será el líder. La figura 11 muestra el siguiente resultado de la relación de conectividad.

$ \begin{array}{cccccccc c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\ \hline \end{array} $	$ \begin{array}{cccccccc c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\ \hline \end{array} $	$ \begin{array}{cccccccc c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \hline \end{array} $	$ \begin{array}{cccccccc c} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \hline \end{array} $
--	--	--	--

Figura 11. Relación de conectividad

Interpretando la última matriz de la figura 11, se puede deducir que la persona 3 es el líder ya que con la composición de la su trayectoria, éste influye a la persona 1,4,6 y 9

III.4 Funciones estructurales.

Los algoritmos de los árboles con raíz, se basan en el conocimiento de sus funciones estructurales [Rangel, González 1999]. Así por ejemplo, en ésta

sección se analizarán las funciones más importantes como: número de nodos, número de hojas, altura, ancho y profundidad.

En este trabajo, se presentan tres ejemplos diferentes que servirán de ejemplo para la implementación de funciones que se analizarán a lo largo del trabajo. En la figura 12 se pueden observar los tres ejemplos mencionados con anterioridad llamados arbolUno, arbolDos y arbolTres.

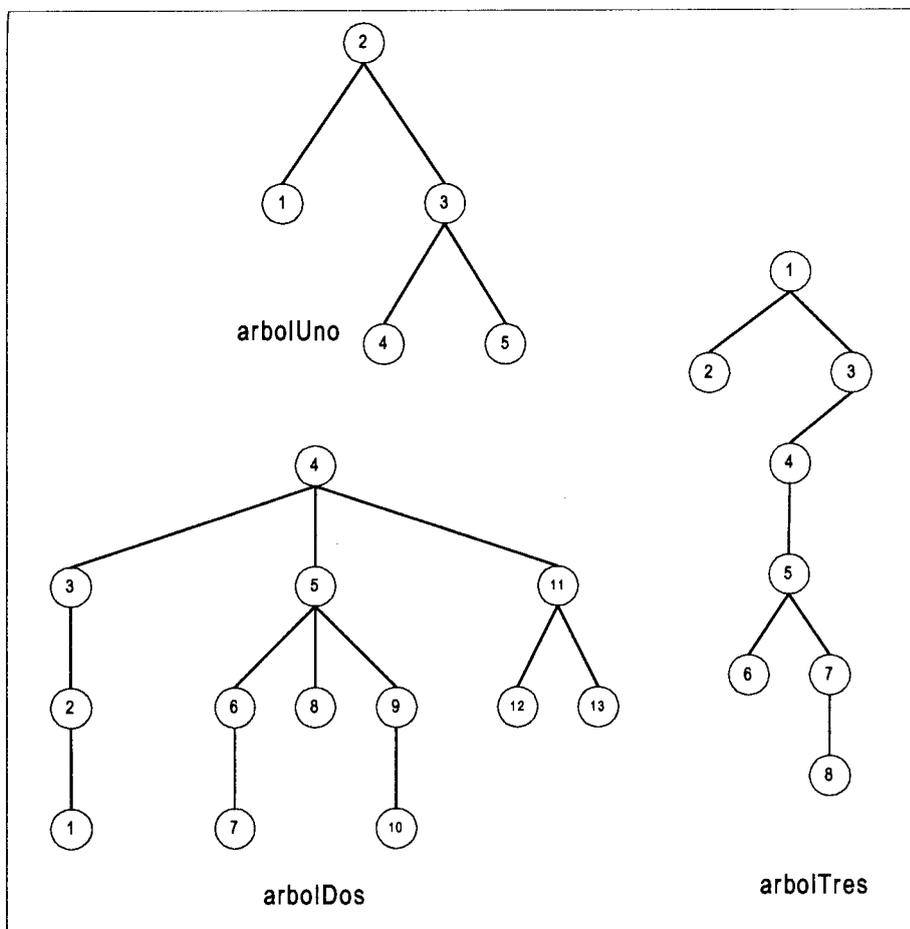


Figura 12. Representación gráfica de arbolUno, arbolDos, arbolTres

La tabla 7 que se muestra a continuación desarrolla la notación extendida de arbolUno, arbolDos y arbolTres.

arbolUno	<code>{2, {{1,0} , {3, {{4,0},{5,0}} } }</code>
arbolDos	<code>{4,{{3,{{2,{{1,0}}}},{5,{{6,{{7,0}}},{8,0},{9,{{10,0}}}},{11,{{12,0},{13,0}}}}</code>
ArbolTres	<code>{1,{{2,0},{3,{{4,{{5,{{6,0},{7,{{8,0}}}}}}}}}}</code>

Tabla 7. Notación extendida de arbolUno, arbolDos, arbolTres

Con el fin de simplificar la salida de la estructura arborecente, se muestra a detalle la función notacionCorta (mostrada en la tabla 8). La función `NotacionCorta[hoja_{ }] := ToString[hoja]` representa el caso base, la expresión se transforma en un string, con [r] raíz, sA toma los valores de los subárboles y MapAll o /@ aplica la función notacionCorta a todos los {sA}, reemplazando en el string las llaves por paréntesis.

Código	<code>NotacionCorta[hoja_{ }] := ToString[hoja] NotacionCorta[r_{sT_ }]:=ToString[r]<>StringReplace[ToString [notacionCorta /@ {sT}],{"{->"(,"->" " "->"}]</code>
Función	<i>NotacionCorta proporciona una expresión simplificada de una estructura arborecente.</i>
Ejemplos	<code>NotacionCorta[arbolUno] 2(1,3(4,5)) notacionCorta[arbolDos] 4(3(2(1)),5(6(7),8,9(10)),11(12,13)) notacionCorta[arbolTres] 1(2,3(4(5(6,7(8))))))</code>

Tabla 8. Función Notación Corta

III.4.1 Número de Nodos

De acuerdo a la definición recursiva del árbol un árbol está compuesto por un conjunto de nodos, la tabla 9 muestra la función para contar el número de nodos que existen en un árbol; flatten elimina los {} y de esa forma Length cuenta el número de elementos del árbol.

Código	<code>noNodos[a_]:=Length[Flatten[a]]</code>
Función	Función noNodos que cuenta el número de nodos.
Ejemplos	<code>NoNodos[arbolUno]</code> 5 <code>noNodos[arbolDos]</code> 13 <code>noNodos[arbolTres]</code> 8

Tabla 9. Función Número de Nodos

III.4.2 Número de Hojas

De acuerdo a [Aho 1995] las hojas son los nodos que no tienen descendientes, la tabla 10 llamada noHojas cuenta el número de hojas comenzando con un el caso base en éste caso la raíz y después en sA los subárboles se seleccionan las estructuras que no tienen descendientes de un árbol, cuando llega a una expresión $\{r_{_},\{\}\}$, llega a una hoja, y con Flatten elimina los paréntesis.

Código	<code>NoHojas[{r_{_},\{\}}]:=r</code> <code>noHojas[_,{sA_}]:= Flatten[noHojas /@{sA}]</code>
Función	Función número de Hojas (noHojas)
Ejemplos	<code>NoHojas[arbolUno]</code> {1, 4, 5} <code>noHojas[arbolDos]</code> {1, 7, 8, 10, 12, 13} <code>noHojas[arbolTres]</code> {2, 6, 8}

Tabla 10. Función noHojas, la cual cuenta el número de hojas

III.4.3 Altura

La altura del árbol se define como el máximo nivel que un nodo puede alcanzar; la tabla 11 muestra la función altura; el caso base `altura[_,\{\}]` asigna el valor de 0 en la raíz, posteriormente calcula la altura de un árbol $(1+ i)$ en: `altura[_,{sA_}]:=1+Max[altura /@ sA]`. Max obtiene el número mayor de la altura de cada subárbol $\{_{_},sA_{_}\}$, tomando la mayor altura de los subárboles, que lo integran.

Código	<pre> Altura[[_,{}]]:=0 altura[[_,sA_]]:=1+Max[altura /@ sA] </pre>
Función	Función altura (altura)
Ejemplos	<pre> Altura[arbolUno] 2 altura[arbolDos] 3 altura[arbolTres] 5 </pre>

Tabla 11. Función Altura

III.4.4 Ancho

El ancho es el grado máximo de todos los nodos de un árbol y el grado es el número de descendientes directos de un nodo.

La tabla 12 muestra como se calcula el ancho de un árbol; el caso base detecta la raíz (`ancho[[_,{}]]`) y le asigna el valor de 1 aumentando su valor en 1, el # significa que toma el primer valor que evalúa de la función ancho y & es la terminación de la función pura Map Apply ó @@ suma (Plus) los nodos hoja.

Código	<pre> ancho[[_,{}]]:=1 ancho[[_,a_]]:=Plus @@ Map[ancho[#]&,a] </pre>
Función	Función ancho
Ejemplos	<pre> ancho[arbolUno] 3 ancho[arbolDos] 6 ancho[arbolTres] 3 </pre>

Tabla 12. Función Ancho

III.4.5 Profundidad

Busca la profundidad de cada nodo, aumentando su nivel de profundidad en 1, para cada expresión anidada; la tabla 13 muestra la función profundo, asignando a w la representación del árbol por pares ordenados (Consultar III.5.1), después la función Map y flatten buscan la profundidad de cada nodo, analizando paréntesis anidados.

Código	Profundo[n_,{a___},{b_,n_},d___]:=1+profundo[b,{a,d}] profundo[_,_]:=0
Función	Función profundo
Ejemplos	w=pares[arbolUno]; Map[#{#,profundo[#,w]}&,Flatten[arbolUno]] {{2, 0}, {1, 1}, {3, 1}, {4, 2}, {5, 2}} w=pares[arbolDos]; Map[#{#,profundo[#,w]}&,Flatten[arbolDos]] {{4, 0}, {3, 1}, {2, 2}, {1, 3}, {5, 1}, {6, 2}, {7, 3}, {8, 2}, {9, 2}, {10, 3}, {11, 1}, {12, 2}, {13, 2}} w=pares[arbolTres]; Map[#{#,profundo[#,w]}&,Flatten[arbolTres]] {{1, 0}, {2, 1}, {3, 1}, {4, 2}, {5, 3}, {6, 4}, {7, 4}, {8, 5}}

Tabla 13. Función profundo

III.5 Representación de árboles

La representación se hace por los tres métodos que se mencionan a continuación:

III.5.1 Pares ordenados

Los árboles pueden ser representados por un conjunto de pares ordenados, en una lista de relaciones de un conjunto de nodos. Esta representación marca el antecedente para la representación geométrica de los árboles como se verá más adelante. La tabla 14 muestra la representación de una estructura arborecente, por medio de pares ordenados. Partition[list, n] arregla la lista en una sublista de 2 elemnetos; flatten elimina los paréntesis anidados y Map toma el primer elemento raíz y lo relaciona con los primeros elementos del nivel correspondiente

Código	Pares[]:={} Pares[{r_,{sA_}}]:=Partition[Flatten[{Map[{r,#}&,First/@{sA}],Pares /@{sA}}],2]
Función	Función pares (pares)
Ejemplos	Pares[arbolUno] {{2, 1}, {2, 3}, {3, 4}, {3, 5}} Pares[arbolDos] {{4, 3}, {4, 5}, {4, 11}, {3, 2}, {2, 1}, {5, 6}, {5, 8}, {5, 9}, {6, 7}, {9, 10}, {11, 12}, {11, 13}} Pares[arbolTres] {{1, 2}, {1, 3}, {3, 4}, {4, 5}, {5, 6}, {5, 7}, {7, 8}}

Tabla 14. Función pares

Código	<pre> aPares[{r_,{sT_}}]:=Partition[Flatten[{Map[{r,#}&,First {sT}],aPares /@ {sT}]],2] aPares[_]:={} ordena[p_]:=Sort[p,(dep2[First[#1],p]>=dep2[First[#2],p])&] dePares[p_]:= Module[{q=ordena[p],t={},i,j}, Map[{(i,j)=#; t=t/.{a___,{i,{k___}},b___}->{a,{i,{k,{j,{}},}},b}, {a___}->{a,{i,{j,{}},}}}&,q]; First[t //. {a___,{b_,c_},d___,{e_,{f___,{b_,{}},g___}},h___}- >{a,d,{e,{f,{b,c},g}},h}]] aMatriz[{r_,sT_}]:=Module[{p=aPares[{r,sT}],m,i,j}, m=DiagonalMatrix[Table[0,{Length[p]+1}]]; Map[{(i,j)=#; </pre>
	<pre> m[[i,j]]=m[[j,i]]=1)&,p]; {r,m} filtraPares[r_,p_]:={}/; Cases[p,{r,_}]=={} filtraPares[r_,p_]:=Module[{t=Cases[p,{r,_}],c,q}, c=Cases[p,{_,r}]; q=Complement[p,t,c]; Partition[Flatten[{t,Map[filtraPares[#,q]&,Last /@ t]}],2]] DeMatriz[{r_,m_}]:=dePares[filtraPares[r,Position[m,1]]] NotacionCorta[{4,{{1,{}},{5,{{3,{}},{6,{}},{2,{}},}}}}] t=toMatrix[{4,{{1,{}},{5,{{3,{}},{6,{}},{2,{}},}}}}]; {First[t], Last[t]//MatrixForm} notacionCorta[deMatriz[t]] ListDensityPlot[-Reverse[Last[t]],Frame->False]; </pre>
Función	Amatriz, representación de un árbol en una matriz
Ejemplos	<pre> 4(1,5(3,6),2) 4, 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 1 0 1 0 0 0 0 1 0 </pre>

Tabla 17. Función matriz

La figura 13 muestra la representación gráfica de una matriz correspondiente al árbol 4(1,2,5)

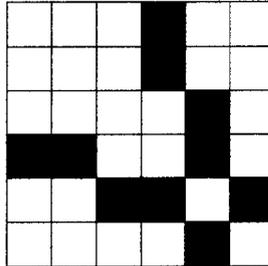


Figura 13. Matriz que representa un árbol con raíz 4(1,2,5)

La representación gráfica de una matriz, del conjunto de pares ordenados que conforman una estructura arborecente, observa las siguientes propiedades: irreflexiva, asimétrica y no transitiva.

III.5.5 Representación gráfica

Mathematica posee herramientas para hacer la representación gráfica de un árbol, a continuación se muestra la función que desarrolla, tal función. La tabla 18 muestra el código y la figura 14 el ejemplo de un árbol.

Código	<pre> dibujaArbol[_,{x_,y_]:={} DibujaArbol[_,t_]{x_,y_]:=Module[{h,w,wHijos=Map[ancho[#]&,t], n=Length[t],hijos,i,j,arbolGraf=nuevosHijos={}, {h,w}={altura[{1,t}],ancho[{1,t}]}; hijos=Table[{x-w/2,y-1}+{wHijos[[i]]/2+Plus@@ Take[wHijos,i-1],0},{i,n}]; Map[(AppendTo[arbolGraf,{Circle[#,0.05],Line[{#,x,y}]}]; AppendTo[nuevosHijos,#]&,hijos]; Do[AppendTo[arbolGraf,dibujaArbol[t[[i]],nuevosHijos[[i]]],{i,n}]; Flatten[arbolGraf]] dt[t]:=Show[Graphics[dibujaArbol[t,{0,0}],AspectRatio->Automatic]; dt[t] </pre>
Función	Dt Dibuja un árbol.

Tabla 18. Función dibuja árbol

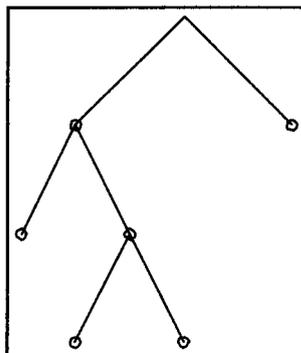


Figura 14. Resultado de la función dibuja un árbol

IV ÁRBOLES CON RAÍZ

La abstracción de modelos jerárquicos de datos, es la característica principal de los árboles con raíz. Por ser una estructura de información no lineal, los árboles con raíz son una alternativa para solucionar el problema de recursos computacionales en la representación de información como en el caso de los árboles binarios y de búsqueda.

En éste capítulo se presenta la terminología básica de los principales árboles con raíz, que será necesaria para entender el siguiente tema de enumeración. También se introduce a conceptos de estructuras arborecentes utilizados en las ciencias sociales.

IV.1 Árbol binario

Una estructura importante de los árboles, son los árboles binarios; éstas estructuras dan lugar algoritmos de búsqueda como se verá más adelante.

Definición 4.1

Un árbol binario T , consiste en un nodo especial llamado raíz y dos subárboles t_1 llamado subárbol derecho y T_2 llamado subárbol izquierdo. En un árbol binario cada nodo puede tener 0,1 ó 2 nodos.

IV.1.1 Árboles de búsqueda: inorden, preorden y posorden

Muchas son las aplicaciones, en las que es necesario recorrer los nodos de un árbol, de modo sistemático, para clasificar, recuperar, comparar o buscar información, como por ejemplo: clasificación lexicográfica, notación polaca, entre otros. Los árboles de búsqueda, son algoritmos para recorrer los nodos de árbol, en un cierto orden; los tres tipos de recorridos básicos en un árbol binario son: inorden, preorden y postorden.

POSTORDEN

Éste algoritmo básicamente hace un recorrido hacia arriba del árbol y se basa en el siguiente algoritmo:

1. Recorre los subárboles, el primer árbol en postorder
2. Recorre los demás en postorder
3. Visita la raíz

La función postorden y los ejemplos se presentan en la tabla 19.

Código	Postorden[r_]:=r latten[{{postorden/@ sA, r}}
Función	Función postorden
Ejemplos	Postorden[arbolUno] {1, 4, 5, 3, 2} postorden[arbolDos] {1, 2, 3, 7, 6, 8, 10, 9, 5, 12, 13, 11, 4} postorden[arbolTres] {2, 5, 7, 8, 6, 4, 3, 1}

Tabla 19. Función postorden

Un punto de interés es que el postorder de una expresión, representa una expresión algebraica. El postorder de tal expresión, da una instrucción en posfijo ó polaca inversa para su evaluación en un árbol de expresiones. De ésta manera la definición recursiva de **árboles de Expresiones**, se define de la forma siguiente:

Base : El operando atómico es una expresión, etiquetada en el nodo. Los operandos atómicos pueden ser datos como: Enteros, char, string o una variable.

Inducción : Las operaciones de una expresión, se arman una a una de acuerdo a la anidación más interna de paréntesis ó respetando el orden de evaluación.

Siguiendo con la lógica, al arbolUno representado por la notación corta $2(1,3(4,5))$ se modificará por siguientes: $+(1,*(4,5))$. Obsérvese que en éste caso los nodos

tienen un operador matemático y las hojas los números u operandos. De ésta manera recorriendo el árbolUno en postorder tenemos la siguiente lista:

{1, 4, 5, +, *} , evaluando es igual a 21

PREORDEN

El algoritmo de búsqueda o recorrido por profundidad, se basa en los siguientes pasos.

1. *Visita la raíz del primer árbol*
2. *Recorre los subárboles , en preorder*
3. *Recorre los demás árboles en preorder*

La función preorder y sus ejemplos se muestran en la tabla 20.

Código	preorden[r_]:=r preorden[{{r_,sA_}}:=Flatten[{{r, preorden /@ sA}}
Función	Función preorder
Ejemplos	preorden[arbolUno] {2, 1, 3, 4, 5} preorden[arbolDos] {4, 3, 2, 1, 5, 6, 7, 8, 9, 10, 11, 12, 13} preorden[arbolTres] {1, {{2, {}}, {3, {{4, {5, {{6, {}}, {7, {8, {}}}}}}}}}}}

Tabla 20. Función preorder

Un ejemplo es la notación polaca o notación infija, siguiendo con nuestro ejemplo el árbol representado por la notación corta $+(1,*(4,5))$.; al ser evaluado en proeder tenemos la siguiente lista: {+,1,* 4, 5} , evaluando es igual a 21

RECORRIDO INORDEN

El recorrido inorder también es conocido como simétrico ó lexicográfico; una de sus aplicaciones se da en el análisis de las gramáticas[Hopcrof 1997, Kelley 1995]; su algoritmo se da en los siguientes pasos:

1. Recorrer el árbol izquierdo en inorden
2. Visitar la raíz del primer árbol
3. Recorrer el árbol derecho en inorden

La función inorden y sus ejemplos se presentan a continuación en la tabla 21.

Código	<pre> Tree={a,{b,{c,d,{0,e}},f,{0,g,h,{i,j}}}} inorden[tree_]:=Flatten[ino[tree]] ino[{raiz_,ltree_,rtree_}]:=List[ino[ltree], raiz, ino[rtree]] ino[raiz_,0]:=List[ino[ltree],raiz,0] ino[raiz_,0,rtree_]:=List[0,raiz,ino[rtree]] </pre>
Función	función inorden
Ejemplos	<pre> inorden[tree] {a, b, c, d, 0, e, f, 0, g, h, i, j} </pre>

Tabla 21. Función inorden

IV.2 Árbol n-ario

Definición 4.2

En un grafo $G=(V,E)$, para cualquier vértice v de G , el **grado** de v es el número de aristas de G incidentes en v . El **grado de salida** de v es el número de aristas que salen de v .

Definición 4.3

Un árbol $T=(V,A)$ es un árbol con raíz y $n \in \mathbb{Z}^+$ entonces T se denomina **n-ario** si el grado de salida de v es n para todos los vértices internos de v .

Es decir, todos los nodos tienen a lo más n hijos. De ésta manera si $n=2$, el árbol es llamado **árbol binario**, si $n=3$ es llamado terciario y así sucesivamente. Los árboles con grado mayor a 2 se llaman multcaminos o multimodales.

IV.3 Árbol n-ario completo

Definición 4.4

Un árbol $T=(V,A)$ es un árbol con raíz y $n \in \mathbb{Z}^+$ entonces T se denomina **n-ario**

completo si el grado de salida de v es exactamente n para todos los vértices internos de v .

En otras palabras, los árboles n -arios completos o también llamados homogéneos son árboles, tales que cada nodo distinto de las hojas tiene exactamente n hijos.

IV.3.1 Árbol balanceado

Definición 4.5

Si $T=(V,A)$ es un árbol con raíz y a es el mayor número de nivel alcanzado por una hoja de T , entonces T tiene una altura a . Se dice que un árbol T con raíz de altura a está **balanceado** si el número de cualquier hoja de T es $a-1$ ó a .

IV.3.2 Árbol extendido

En un árbol $T=(V,A)$, se insertan elementos especiales, tales como cuadros pequeños blancos \square , que hacen explícito los vacíos de un árbol. Éstos elementos se insertan en un árbol para convertirlo T_1 en un árbol balanceado completo.

IV.4 Árboles usados en las ciencias sociales

Los árboles, por su capacidad, pueden reflejar una gran cantidad de relaciones, los árboles que más se utilizan en las Ciencias Sociales son los árboles de decisiones y árboles de Kuhn.

IV.4.1 Árboles de decisión

Los árboles de decisión, son algoritmos de comparación de dos claves. La representación gráfica de éste algoritmo es un árbol binario, donde cada nodo representa un estado del programa; es decir, cada nodo representa disposiciones iniciales de los elementos analizados hasta ese momento del programa, representando el conocimiento, de la disposición inicial hasta ese punto del programa. En las ciencias de la computación su aplicación principal se encuentra en la clasificación de información por comparación [Aho, Hopcroft, Ullman 1988]
En las ciencias sociales los árboles de decisión se utilizan en sistemas de análisis.

IV.4.2 Árbol de Kuhn

En el árbol de Kuhn, cada nodo representa una posición o estado posible del juego. Este árbol se utiliza en las ciencias sociales para modelar estrategias de forma extendida. Cada hoja evalúa el recorrido hecho hasta el momento y se marca con un resultado final; de ésta manera cada ruta del árbol representa un escenario posible de juego.

La Teoría de juegos es un modelo matemático de comportamiento de individuos u organizaciones, enfrentando procesos de interacción específica, negociación, competencia. Éste enfoque de análisis organizacional no debe ser examinado, ni juzgado a la ligera, por que en su nombre conlleva la palabra *juego*, al contrario es una herramienta para los especialistas interesados en conocer el estudio de las organizaciones [Shubik 1996].

En las ciencias computacionales, el árbol de juego se utiliza cuando se desea conocer una solución óptima a una serie de subproblemas y la única forma de hacerlo es mediante una búsqueda exhaustiva, mejor conocida como *backtraking*, algunos ejemplos, son los juegos como tres en raya, y gato. Cualquiera que sea su aplicación debe especificarse una función de utilidad, la cual nos señala las formas en que se ha de ganar un juego, de ésta manera la búsqueda exhaustiva puede hacerse implementando podas alfa-beta [Aho, Hopcrof, Ullman 1988].

V. ENUMERACIÓN DE ÁRBOLES

La enumeración es un concepto fascinante en el desarrollo de algoritmos; básicamente se aplica para medir la eficiencia de algoritmos de búsqueda, ordenación o análisis. Por lo tanto el presente capítulo comienza enumerando los componentes de los árboles, relacionando el número de nodos con las aristas; la enumeración de árboles libres incluye a los árboles transversales y la enumeración de árboles con raíz abarca la longitud externa e interna, por último se analiza la enumeración de árboles isomorfos.

V.1 Análisis básico de componentes.

La presente sección muestra las definiciones que se aplicarán en el presente capítulo.

Definición 5.1 [Grimaldi]

Sea $G=(V,E)$, para cualquier $v \in V$ (definido en índice de símbolos)

- a) El **grado de entrada o interno** de v es el número de aristas (en G) que son incidentes en v , y éstas son denotados por $id(v)$.
- b) El **grado de salida o externo** de v es el número de aristas que incidentes en otros vértices y éstos salen de v , y éstas son denotados por $od(v)$.

Definición 5.2

Un árbol T , se dice que es un **árbol ordenado** si las aristas que salen de los vértices internos, están ordenados de izquierda a derecha.

Definición 5.3

Dada una colección de n objetos, cualquier disposición de ellos se denomina permutación de la colección.

$$P(n,r) = \frac{n!}{(n-r)!}; P(n,r) \text{ representa el número de } \mathbf{permutaciones} \text{ posibles}$$

de n objetos tomando r a la vez.

Definición 5.4[knuth 1973]

En un árbol T, existe un nodo llamado raíz, que por definición tiene un nivel con valor de 0, los sucesores incrementan el nivel en 1.

V.1.1 Relación de los vértices y las aristas en un árbol.

Para deducir el número de aristas que tiene un árbol de n nodos, primero se debe recordar que la raíz de un árbol tiene grado interno 0 (definición 5.1) y los demás vértices grado interno 1 [Wirth1987]. La tabla 22, muestra como al incrementar el número de vértices recursivamente, el grado interno del vértice aumenta en uno y el número de aristas, por consiguiente se incrementa en uno.

Vértices (v)	Aristas (e)	
Número de Vértices	Grado interno	No. De aristas
1	0	0
2	1	1
3	1	2
.	.	.
.	.	.
Vn		n-1

Tabla 22. Relación del número de vértices, grados internos y número de aristas.

De la tabla anterior se deriva que la fórmula para determinar el número de vértices de un árbol es igual a: $v = e-1$

Por lo tanto, como se había determinado en las propiedades de los árboles el número de aristas de un árbol, es igual $e = v+1$.

Así por ejemplo, un árbol de 30 nodos tendrá 29 aristas, de acuerdo a la tabla y fórmula anteriormente descrita.

V.1.2 Relación del número de nodos en la construcción de árboles

En la sección anterior, se habló de la relación que tiene el número de vértices y el número de aristas de un árbol cualquiera. Un concepto interesante es la inversa, es decir ¿Cuántos árboles ordenados se pueden construir con n

vértices?. La tabla 23, muestra el número de árboles ordenados que pueden ser trazados a partir n vértices.

No de vértices	Total de árboles
1	1
2	2
3	12

Tabla 23. Relación del no. de vértices y del número de árboles ordenados que se pueden generar

Así por ejemplo, recordando la definición de un árbol binario con nodos $n=2$, la raíz, puede tener $n-1$ hijos. De ésta forma los primeros árboles que se generarán es un árbol con raíz y $n-1$ hijos, después los árboles generados sean de la forma $n-2$ hijos y así sucesivamente. Esto nos recuerda la fórmula de permutación.

$$P(n,r) = \frac{n!}{(n-r)!}$$

Siguiendo con la lógica anterior, la tabla 24 muestra el número de vértices y la relación en como pueden ser ordenados por permutación en la construcción de árboles.

No. De vértices	Fórmula	Total de árboles
1	1	1
2	$P(2,1)$	2
3	$P(3,2) + P(3,1)$	12
4	$P(4,3) + P(4,2) + P(4,1)$	60
.	.	.
N	$P(n,n-1) + \dots + P(n,1)$	$\sum_{r=1}^{r=n-1} P(n,r)$

Tabla 24. Número de vértices y relación de ordenación por permutación en la construcción de árboles

De ésta manera, en un conjunto de 3 vértices $T=\{a,b,c\}$ ¿Cuántos árboles en diferente orden se pueden construir, con tres vértices?. Respuesta. Se pueden trazar 12 árboles.

V.1.3 Relación de los nodos ordenados por nivel

Para conocer cuantos nodos ordenados por nivel corresponden a un nivel /

y cuales son cada uno de ellos, obsérvese la figura 15, donde se muestra la relación de los nodos ordenados por niveles de un árbol binario.

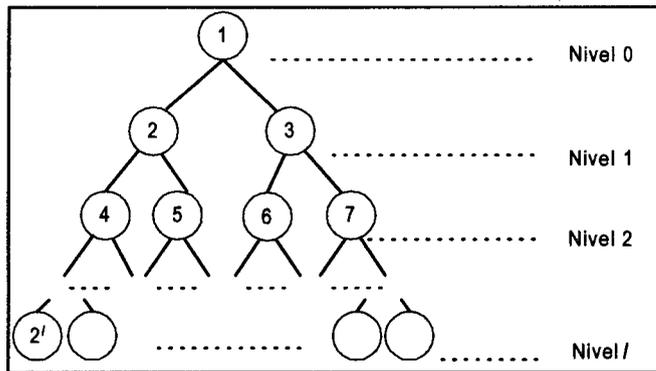


Figura 15. Relación de nodos ordenados por niveles de un árbol binario

Obsérvese por una parte que el número de nodos existentes en un nivel l es 2^l , por otra parte éste número coincide con el número del primer nodo del árbol ordenado por nivel. Por lo que en un nivel se encuentran los nodos:

$$2^{l+0}, 2^{l+1} \dots\dots\dots 2^{l+(2^l-1)}$$

Así por ejemplo, pensando en un nivel 4 de un árbol binario, tenemos que:

- a) existen 16 nodos en ese nivel
- b) el primer nodo de un árbol binario completo es 16
- c) los nodos existentes son: 16, 17, 18, ... 31

Generalizando podemos decir que en un árbol n -ario completo, de grado d , en un nivel l , existen d^l nodos correspondientes a éste nivel y estos son:

$$d^{l+0}, d^{l+1} \dots\dots\dots d^{l+(d^l-1)}$$

V.1.3.1 Relación de nodos etiquetados con el nivel.

Una característica importante se observa cuando se desea sumar los nodos enumerados con el mismo número del nivel correspondiente.

Por ejemplo ¿Qué pasa si se desea sumar los números existentes dentro de los nodos de un árbol binario?. La figura 16 muestra la relación de nodos enumerados con el nivel correspondiente.

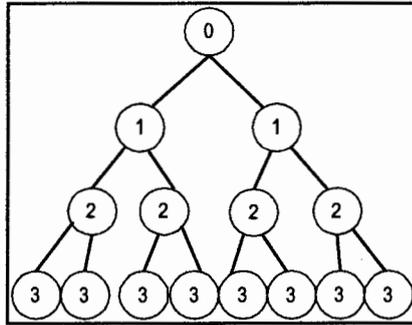


Figura 16. Relación de nodos enumerados con el nivel correspondiente.

Primero observe que los nodos tiene un comportamiento como el piso de logaritmos i (consultar apéndice B). La tabla 25 muestra la función piso de logaritmos base 2 y se observa que corresponden a los nodos etiquetados en la figura 16.

Función piso de logaritmos base 2		
i	Lg_i	$\lfloor \lg_i \rfloor$
1	0.000	0
2	1.000	1
3	1.585	1
4	2.000	2
5	2.322	2
6	2.585	2
7	2.807	2
8	3.000	3
9	3.170	3

Tabla 25. Función piso de logaritmos base 2

Por lo tanto la ecuación para sumatoria de los nodos del árbol binario correspondiente a la figura 13 es:

$$S = 0+1+1+2+2+2+2+2.....+3$$

Generalizando [Knuth 1973] la fórmula anterior se obtiene:

$$L_n = \sum_{i=1}^n \lfloor \lg i \rfloor$$

V.2 Enumeración de árboles libres

Dentro de la clasificación de árboles libres, se analiza la enumeración en dos partes: árboles libres y árboles transversales.

V.2.1 Enumeración de árboles libres

En 1857, Arthur Cayley, trabajaba en la enumeración de isómeros; los cuales son compuestos químicos de la forma C_kH_{2k+2} . Analizando las propiedades combinatorias los isómeros y al observar la representación de la unión química de éstos compuestos representando al Hidrógeno con H y al Carbono con C, Cayley descubrió unas estructuras a las cuales llamó "árboles"; la figura 17 muestra algunos ejemplos de isómeros o hidrocarburos saturados.

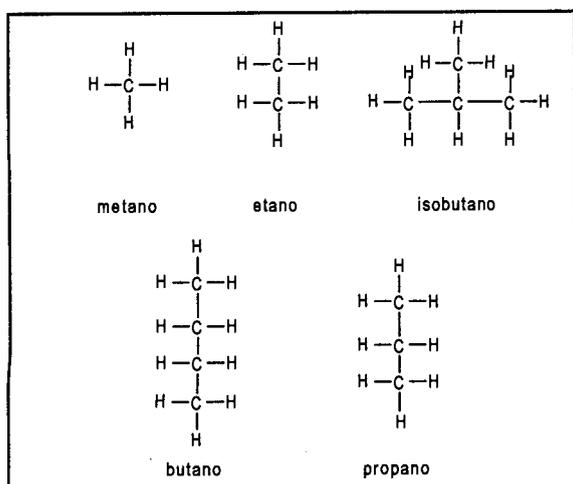


Figura 17. Cinco Hidrocarburos saturados

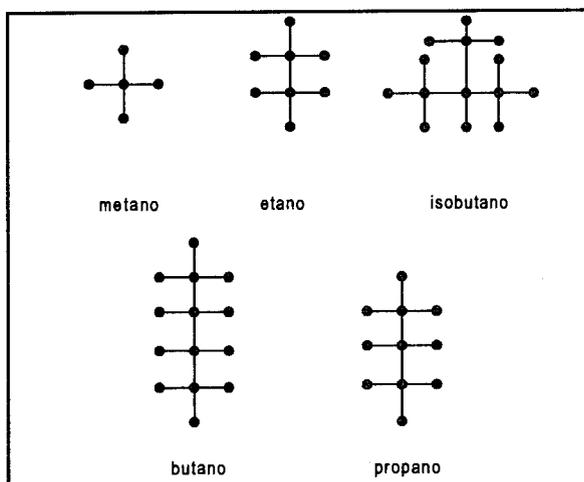


Figura 18. Grafo de enlaces

Para generalizar la representación de los isómeros como árboles libres, elaboró un grafo de enlaces; la figura 18 muestra los grafos de enlaces de los ejemplos mostrados en la figura 17. Es fácil observar quienes son los elementos del carbono, por que sus vértices correspondientes siempre tienen un grado 4, y los vértices de grado 1 corresponden al hidrógeno.

Los grafos de enlaces, por si solos no son suficientes para mostrar ampliamente la definición de árboles libres, para demostrarlo se comprueba por tres hechos conocidos:

Primero. El hidrocarburo saturado, es de la forma C_kH_{2k+2} , por lo que el grado de los árboles libres, en los grafos de enlace es: $K+2k+2=3k+2$

Segundo. Se muestra que el Carbono tiene grado 4 y el hidrógeno grado 1, resultando en nuestra ecuación $4k+1(2k+2)=6k+2$

Tercero. Recordamos que si G es un grafo e aristas, entonces $\sum \text{grado}(u) = 2e$ que el grado es el doble, que el número de aristas; Por lo que el número de aristas es siempre la mitad, $3k+1$ es uno menor que los vértices.

La demostración de que el grafo de enlaces es un árbol, estaba resuelto.

Para continuar sobre la enumeración de los árboles libres, observa la figura 19, que muestra la correspondencia de un árbol y una lista, la figura explica paso a paso la forma de como se construye la lista.

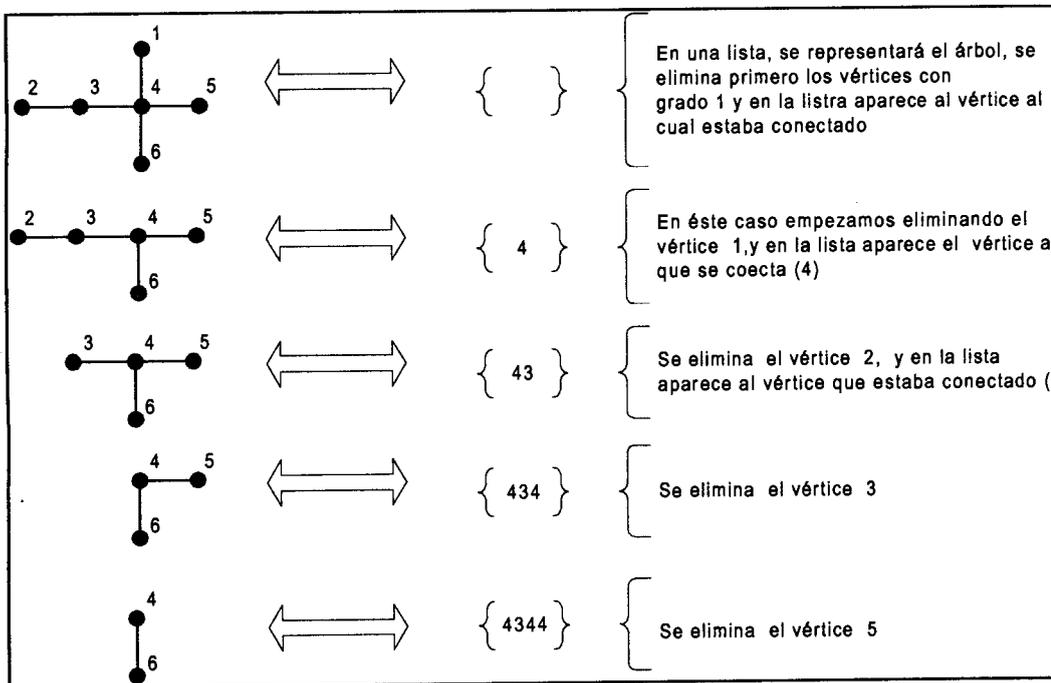


Figura 19. Correspondencia de un árbol y una lista

La figura 20, muestra otro ejemplo simplificado de ésta correspondencia.

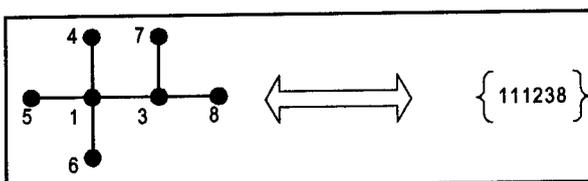


Figura 20. Correspondencia de un árbol y una lista 111238

En resumen: los vértices v aparecen $\text{grado}(v) - 1$ veces en una lista de $n-2$ vértices, por lo tanto si para cada árbol existe la representación de una lista $n - 2$ entonces existen n^{n-2} . La figura 21 muestra la función que determina el número de árboles libres.

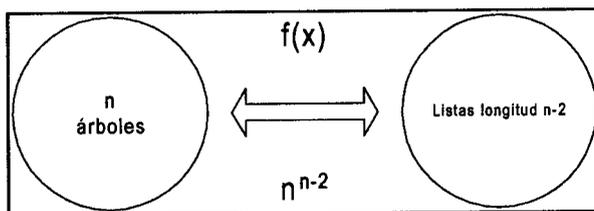


Figura 21. Función que determina el número de árboles libres

De las observaciones anteriores Cayley, dedujo que el número de árboles libres, se puede obtener aplicando el siguiente teorema:

Teorema. Si $n \geq 2$, existen n^{n-2} árboles distintos etiquetados de n vértices.

V.2.2 Enumeración de arboles transversales

Los árboles transversales (spanning tree), son conocidos como árboles abarcadores minimales; éste tema se centra en el área de las matemáticas denominada investigación de operaciones, su aplicación está básicamente en la optimización de redes, las cuales tienen un número asociado a cada arista llamado peso; éstos pueden ser tiempos, cantidad de transporte permitidos, entre otros.

La investigación de operaciones es un método que permite encontrar las relaciones óptimas que mejor operan un sistema, dado un objetivo específico.

Algunos problemas que se analizan en éste tema son:

1. En un árbol abarcador minimal, encontrar la suma de los mínima. [Grimaldi 1994]
2. Problemas de transporte. [Chartrand 1977] [Grimaldi 1994]
3. Solución de PERT [Chartrand 1977] y agente viajero [Prawda 1989]

La enumeración de los árboles transversales, ayuda a soportar una gama de diferentes decisiones que se pueden tomar en la minimización de la red.

Un árbol transversal es un grafo G_1 el cual contiene todos los vértice de G . Para enumerarlos, primero se establece por la fórmula de Cayley, que existen $\tau(K_n) = n^{n-2}$, árboles con n vértices.

Por ejemplo, la figura 22 muestra los 16 árboles, que se generan de $n=4$

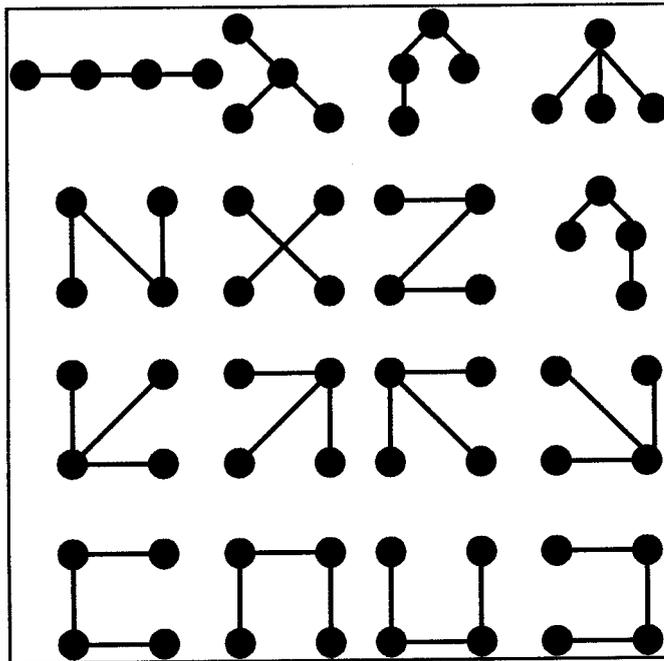


Figura 22. 16 árboles construidos de 4 nodos

De acuerdo con [Rosen 1999] la fórmula para obtener para obtener, los posibles árboles transversales, considerando el grado menor de los vértices de un grafo, es la siguiente:

$$\tau(I_s + K_{n-s}) = n^{n-2} \left(1 - \frac{s}{n}\right)^{s-1}$$

La figura 23 muestra un ejemplo para cuatro nodos y para $s=2$ (grado menor de un vértice).

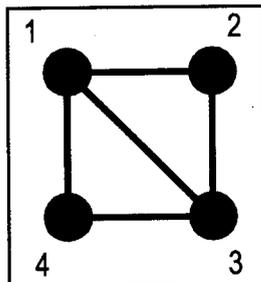


Figura 23. Grafo con cuatro nodos

Los árboles posibles a generar son $\tau(I_2 + K_{4-2}) = 4^2(1 - \frac{2}{4})^1 = 8$

Para comprobar el número árboles por medio del teorema de matriz, tenemos:

$$\begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}$$

para calcular el determinante a la matriz anterior le quitamos el renglón 3 columna 1, por lo que

$$(-1)^4 \begin{vmatrix} -1 & -1 & -1 \\ 2 & -1 & 0 \\ 0 & -1 & 2 \end{vmatrix} = 8$$

La figura 24 muestra cuales son árboles transversales resultantes

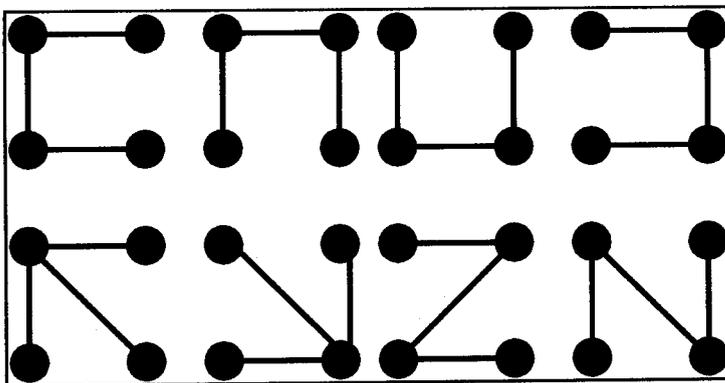


Figura 24. Árboles transversales resultantes con cuatro nodos

Es importante señalar que la computadora, almacena los grafos en forma recursiva. La fórmula recursiva, para el almacenamiento de los grafos es $\tau(G) = \tau(G - e) + \tau(G / e)$ [Rosen 1999]

Por ejemplo, los grafos dibujados por $\tau(G)$ para seis vértices se muestran en la figura 25.

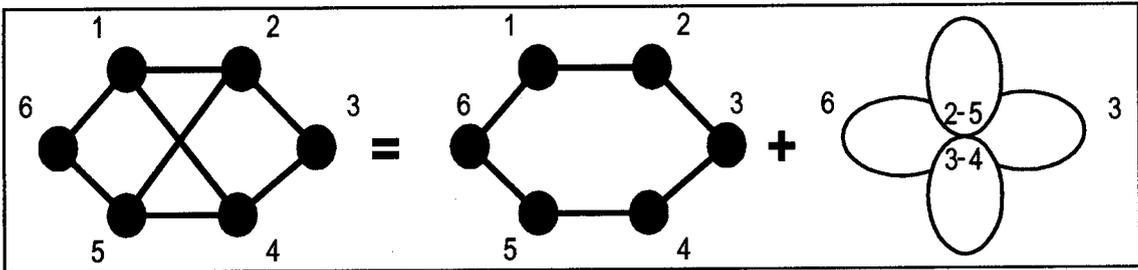


Figura 25. Grafos que almacena la computadora para $n=6$

Para enumerar árboles transversales con un número de grado definidos para cada uno de sus vértices, se muestra el siguiente ejemplo, basado en los vértices de un hexágono K_6 .

El número de árboles transversales que tenga un grado para v_1, v_2, \dots, v_6 igual a $3, 2, 1, 1, 1, 2$; en donde el grado 3 corresponde a v_1 y así sucesivamente, son dados por la fórmula [Rosen1999]

$$\left(\frac{(n-2)!}{(d_1-1)! \dots (d_n-1)!} \right), \text{ generando el siguiente resultado:}$$

$$\left(\frac{(6-2)!}{2!1!1!0!0!1!} \right) = \frac{4}{2} = 12 \text{ árboles transversales, la figura 26 muestra los árboles}$$

transversales obtenidos.

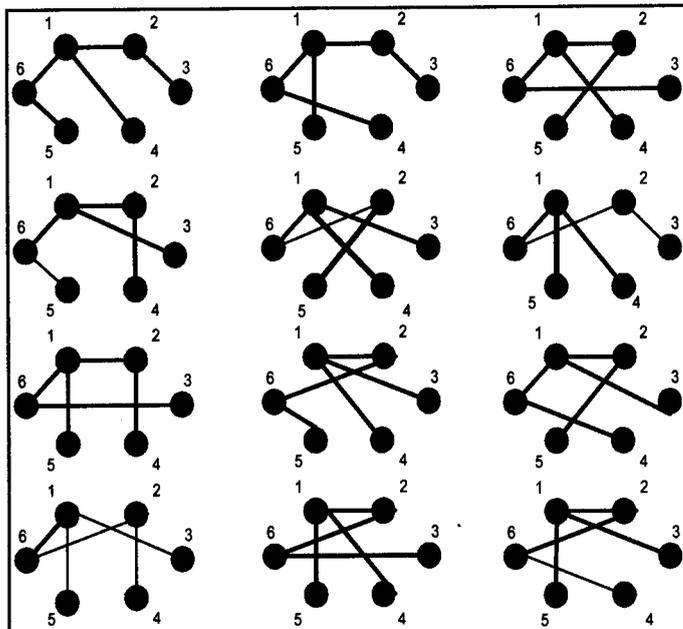


Figura 26. Árboles transversales construidos de 6 vértices

V.3 Enumeración de árboles con raíz y sus componentes

Esta sección analizará en general la enumeración de árboles con raíz comenzando con la longitud media de trayectoria interna y externa en un árbol, por último el número máximo de nodos para un árbol binario completo, con altura i .

V.3.1 Número de árboles con raíz

La ecuación para encontrar el número de árboles con raíz, de acuerdo a [Rosen 1999] es la siguiente:

$$r(x) = \sum_{n=1}^{20} R_n x^n = x + x^2 + 2x^3 + 4x^4 + 9x^5 + 20x^6$$

donde, el coeficiente de cada variable representa el número de arboles y los exponentes los n nodos del árbol, por ejemplo, existe un árbol de un nodo raíz, existe un árbol con raíz de dos nodos y así sucesivamente. La tabla 26 muestra gráficamente las variables de la ecuación.

Nodos	Árboles con raíz
1	1
2	1
3	2
4	4
5	9
	..

Tabla 26. Relación de vértices con el número de árboles con raíz.

Para encontrar la función generadora de los árboles con raíz, se observa que:

$$1 + x + 2x^2 + 4x^3 + 9x^4 + 20x^5 = e^{\sum}$$

Por lo tanto el número de árboles con raíz, se ve generado, por las siguientes ecuaciones:

$$A(z) = a + bx + cx^2 + \dots x^{10}$$

$$A(z) = a + bx^2 + cx^4 + \dots x^{10}$$

$$A(z) = a + bx^3 + cx^6 + \dots x^{10}$$

$$A(z) = a + \dots x^{10}$$

$$A(z) = a + bx^{20}$$

A continuación se muestra la función desarrollada en *Mathematica* para encontrar la función generadora de los árboles con raíz:

$$s1 = \text{Series}[(1-x)^{-a}, \{x, 0, 5\}]$$

$$1 + ax - \frac{1}{2}((-1-a)a)x^2 + \frac{1}{6}(-2-a)(-1-a)ax^3 - \frac{1}{24}((-3-a)(-2-a)(-1-a)a)x^4 + \frac{1}{120}(-4-a)(-3-a)(-2-a)(-1-a)ax^5 + 0[x]^6$$

$$s2 = \text{Series}[(1-x^2)^{-b}, \{x, 0, 5\}]$$

$$1 + bx^2 - \frac{1}{2}((-1-b)b)x^4 + 0[x]^6$$

$$s3 = \text{Series}[(1-x^3)^{-c}, \{x, 0, 5\}]$$

$$1 + cx^3 + 0[x]^6$$

$$s4 = \text{Series}[(1-x^4)^{-d}, \{x, 0, 5\}]$$

$$1 + dx^4 + 0[x]^6$$

$$s5 = \text{Series}[(1-x^5)^{-e}, \{x, 0, 5\}]$$

$$1 + ex^5 + 0[x]^6$$

$$r = s1s2s3s4s5 /. \{a \rightarrow 1, b \rightarrow 1, c \rightarrow 2, d \rightarrow 4, e \rightarrow 9\}$$

$$1 + x + 2x^2 + 4x^3 + 9x^4 + 20x^5 + 0[x]^6$$

Con lo cual queda demostrado

V.3.2. Longitud media de trayectoria interna en un árbol

La longitud de la trayectoria (definido en el capítulo II) es el número de aristas que se recorren para acceder desde la raíz al nodo x. La tabla 27 muestra la composición de trayectorias en un árbol T.

Nivel(i)	Longitud de trayectoria(l)
0	0 (Por definición la raíz tiene longitud de trayectoria 0)
1	1
2	2
...	...
l	l

Tabla 27. Composición de trayectorias

La longitud de trayectoria interna, es la suma de las longitudes de trayectorias de todos sus componentes. La figura 27 muestra un árbol T cuya longitud de trayectoria interna es igual 24.

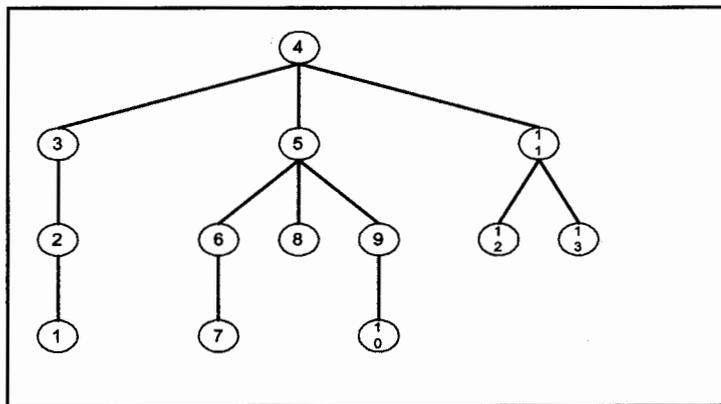


Figura 27. Ejemplo de un árbol T para calcular la longitud de trayectoria interna

La tabla 28 muestra como calcular la longitud de trayectoria interna del árbol mostrado en la figura 27 y en el renglón final de la tabla; se encontrará la fórmula general para encontrar la longitud interna de un árbol de nivel i con n nodos.

Nivel(i)	Longitud de trayectoria(l)	Número de nodos Correspondientes al nivel (i)	Longitud de trayectoria en un nivel i con n nodos
0	0	1	0*1
1	1	3	1*3
2	2	6	2*6
3	3	3	3*3
...
l	L	N	$l = \sum_{i=0}^{l-1} i * n_i$

Tabla 28. Cálculo de longitud de trayectoria interna

Por lo que la longitud promedio de trayectoria interna (I) es $\frac{I}{n}$

Para el caso anterior el promedio longitud interna de trayectoria es $24/13=1.8$, es decir; que en promedio se recorrerán 1.8 nodos para localizar cualquier nodo.

V.3.3 Longitud de trayectoria externa (E)

Para calcular la trayectoria externa, primero se amplía la estructura a un árbol extendido llenando las aristas vacías con los nodos especiales □, de forma que todos los nodos tengan el mismo grado. La figura 28 muestra como la estructura arborecente se convierte en un árbol extendido. Observe que los nodos especiales no tienen más descendientes [Wirth 1987].

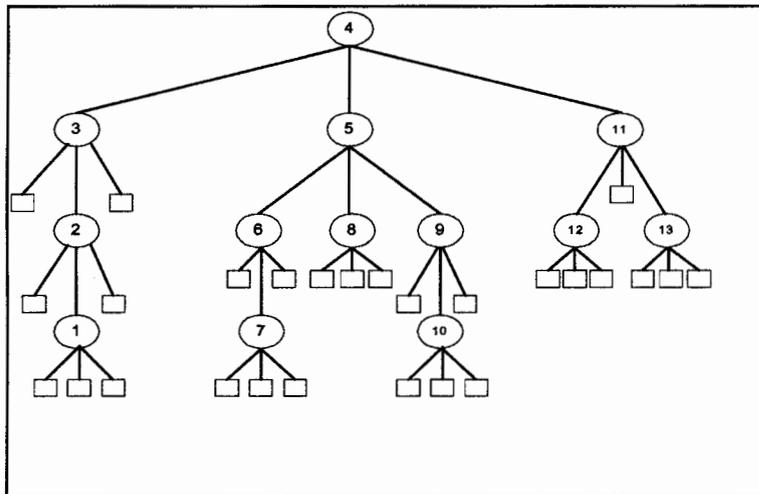


Figura 28. Árbol extendido con nodos especiales

Por definición [Reingold 1977] la trayectoria de longitud externa es igual a la suma de las longitudes de los nodos especiales. La tabla 29, muestra como calcular la trayectoria externa del árbol correspondiente a la figura 28.

Nivel(i)	Longitud de trayectoria(l)	Número de nodos especiales correspondientes al nivel (m)	Longitud de trayectoria en un nivel i con n nodos
1	1	0	1*3
2	2	3	2*6
3	3	15	3*15
4	4	9	4*9
I	L	M	$E = \sum_{i=1}^{i=L} i * m_i = 96$

Tabla 29. Cálculo de longitud de trayectoria externa

Como puede observarse, la longitud promedio externa depende del número de nodos originales inscritos en el árbol, por lo tanto el promedio es igual

$$a \text{ PE} = \frac{E}{m}$$

Obsérve que en los árboles binarios, los nodos especiales son siempre $n+1$, la tabla 30 muestra la relación de los nodos y de los nodos especiales en un árbol binario que demuestran que el promedio de la longitud externa es igual a la fórmula mencionada con anterioridad.

Nivel(i)	Nodos(n) en un árbol binario	Nodos (m) especiales que pueden ser insertados	Diferencia entre nodos originales y especiales
0	1	2	1
1	2	3	1
1	3	4	1
..
		$n+1$	

Tabla 30. Relación de los nodos de un árbol binario y los nodos especiales

Generalizando la fórmula es:
$$\frac{\sum_{i=m}^{i=1} i * m_i}{m}$$
 [Knuth 1973]

V.3.4 Relación entre la longitud interna y la longitud externa de un árbol binario.

Para definir la relación entre la longitud interna y la longitud externa de un árbol binario, primero observe las figuras 29 y 30. La figura 29 define recursivamente la longitud interna y la figura 30 la longitud externa.

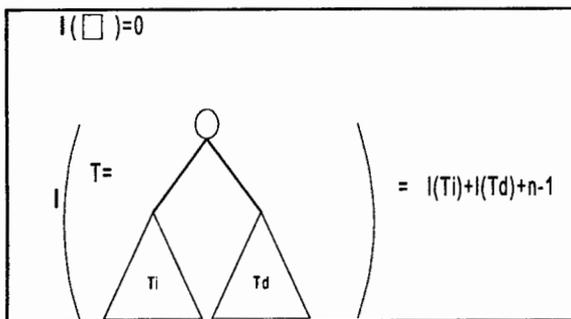


Figura 29. Longitud interna

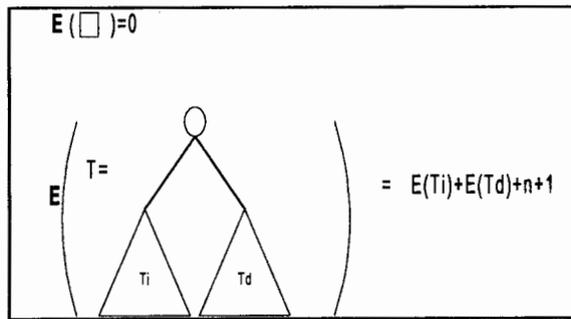


Figura 30. Longitud externa

Por lo tanto, la relación entre la longitud interna y externa es:

$$\begin{aligned} &I(Ti)+I(Td)+n-1 \quad (n-1 = \text{nodos internos entre } Ti \text{ y } Td \text{ menos la raíz}) \\ &E(Ti)+E(Td)+n+1 \end{aligned}$$

Simplificando

$$\frac{I(T)+n-1}{E(T)+n+1}$$

Sumando los términos, se genera el siguiente resultado: $E(T)+I(T)+2n$

Por lo que la diferencia entre $E(T)$, $I(T)$, es la diferencia expresada por $2n$

$D(T)=2n$; expresado en otras palabras, la relación entre la longitud interna y externa de un árbol está dado, por la siguiente fórmula: $E(T) = I(T) + 2n$

V.3.5. La Relación entre la longitud interna y la longitud externa de un árbol n-ario.

La fórmula anterior puede ser generalizada, bajo las siguientes consideraciones:

Primero se debe considerar que m es el número de nodos especiales y que en un árbol extendido existen n nodos originales y m especiales.

Segundo, para crear un árbol extendido el grado(d) afecta directamente a n nodos por lo que se tienen $d*n+1$ (este uno es la raíz)

Por lo que:

$$d*n+1=m+n$$

$$dn+1-n=m$$

$$dn-n+1=m$$

$$(d-1)n+1=m$$

$$m = (d - 1)n + 1 ; \text{ es la relación entre la longitud interna y externa de un árbol } n\text{-ario.}$$

Por ejemplo, si existen 27 nodos especiales, en un árbol ternario y 13 nodos originales, la relación se comprueba, bajo las siguientes operaciones:

$$27=(3-1)13+1$$

$$27=26+1$$

$$27=27$$

La figura 31 ejemplifica la relación mostrada anteriormente.

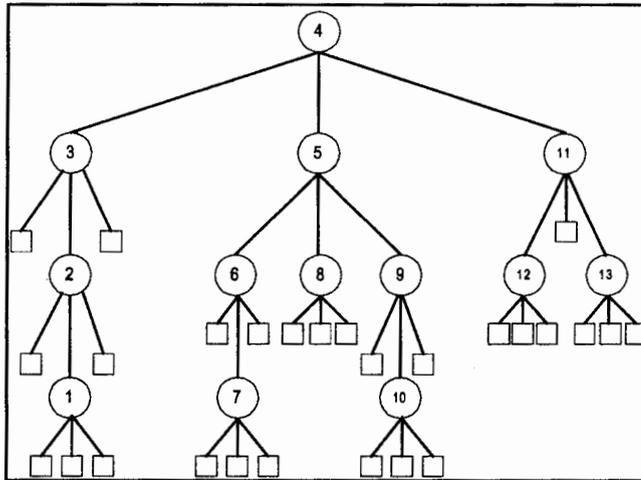


Figura 31. Ejemplo de relación interna y externa de un árbol n-ario

V.3.6 Número máximo de nodos para un árbol binario completo, con altura i

Para calcular el número máximo de nodos para un árbol binario completo de altura i , observe la tabla 31 que muestra la relación del nivel i , el grado del árbol binario y los números de nodos del árbol binario en el nivel i ; en el renglón final se observa la fórmula para encontrar el número máximo de nodos.

Nivel(i)	Grado (d)	Nodos(n)
0	2	1
1	2	2
2	2	4
3	2	8
..		...
N		d^i

Tabla 31. Relación del número máximo de nodos para un árbol binario completo, con altura i

Por lo tanto, el número de nodos en un árbol binario completo, en una altura (i) es

$$n_2 = 2^i, \text{ para } 0 \leq i$$

Ejemplo.

¿Cuántos nodos tiene un árbol como máximo si conocemos su altura y el grado del árbol?

$$\text{Altura}(i) = 3$$

$$\text{Grado del árbol}(d) = 3$$

$$\text{Fórmula } n_d = d^i$$

Resolviendo: $n_3 = 3^3 = 27$ nodos

$$\sum_{i=0}^{i=3} d^i = 3^0 + 3^1 + 3^2 + 3^3 = 1 + 3 + 9 + 27 = 40.$$

Es decir; 40 nodos tiene un árbol como máximo si su altura es 3 y es trivalente.

V.3.7 Número máximo de nodos para un árbol n-ario completo, con altura i

Para generalizar la fórmula para calcular el número de vértices en un árbol de grado (d), se recuerda que el número de nodos totales para árboles binarios balanceados son: $2^{n+1}-1$; donde $n=niveles$

Para árboles de grado mayor a 2, se observa que los nodos se incrementan según el grado del árbol ó el número de ramas llamadas Factor branching (b); por lo que se puede generalizar diciendo que el número de nodos en un árbol de grado (d), en una altura (i) es: $n_d = d^i$

Nivel(i)	Grado (d)	Nodos(n)
0	b	1
1	b	b
2	b	b ²
..		...
L		b ^l

Tabla 32. Relación del número máximo de nodos para un árbol n-ario completo, con altura i

La relación de los nodos, de la tercera columna de la tabla 30, puede representarse por:

$$S = 1 + b + b^2 + \dots + b^l$$

S es una progresión geométrica, para obtener el valor de la suma se multiplica por su factor b, entonces

$$b(S) = b + b^2 + \dots + b^l + b^{l+1}$$

Manipulando aritméticamente:

Restamos $b(S) = b + b^2 + \dots + b^l + b^{l+1}$

Menos $S = 1 + b + b^2 + \dots + b^l$

Igual a $bS - S = b^{l+1} - 1$

factorizando $S(b-1) = b^{l+1} - 1$

Por lo que la fórmula es:
$$S = \frac{b^{l+1} - 1}{b - 1}$$

Ejemplo.

¿Cuál es el número máximo de nodos para un árbol trivalente si tiene 3 niveles?
 $3^{3+1}-1/(3-1)=40$. Un árbol trivalente, con un nivel 3 tiene a lo más 40 vértices.

CQC

V.3.8 Análisis del peor caso en las búsquedas de árboles binarios

El peor caso en las búsquedas de árboles binarios, se presenta cuando existe una ruta interna de longitud máxima; la figura 32 por ejemplo, muestra en los incisos (a) y (b) dos ejemplos de longitud interna máxima y en el inciso (c) el peor caso de longitud máxima para un árbol binario.

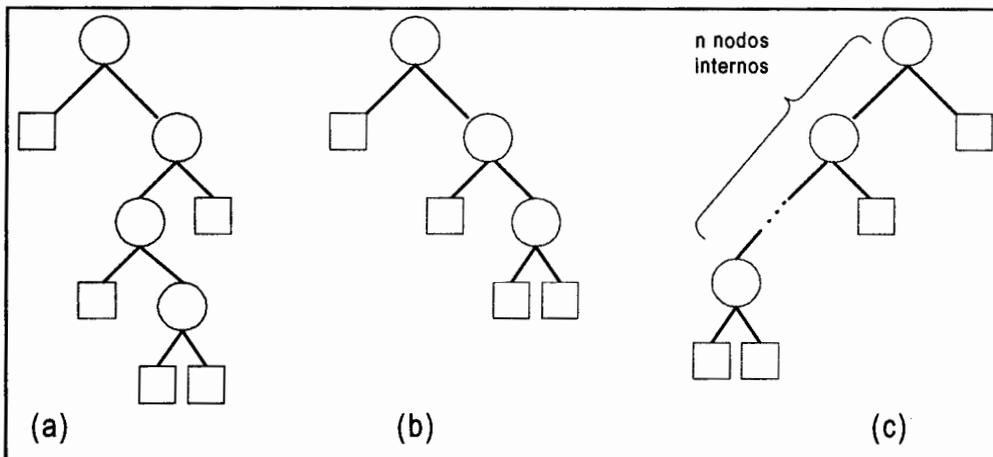


Figura 32. Casos de longitud interna máxima

Para calcular el peor caso, observe la tabla 33, la cual muestra la relación de los nodos con el nivel y su relación con la trayectoria interna. La relación resultante de la longitud de trayectoria interna con los nodos igual a $n(n-1)$.

Nivel (l)	Nodos (n)	Trayectoria Interna
0	1	0
1	1	1
2	1	2
3	1	3
		$n(n-1)$

Tabla 33. Relación de los nodos con el nivel y su relación con la trayectoria interna

Como puede observarse en ésta relación se considera al árbol binario completo; generalizando el peor caso de longitud de trayectoria interna, véase la figura 33 la cual considera solo la mitad del árbol binario.

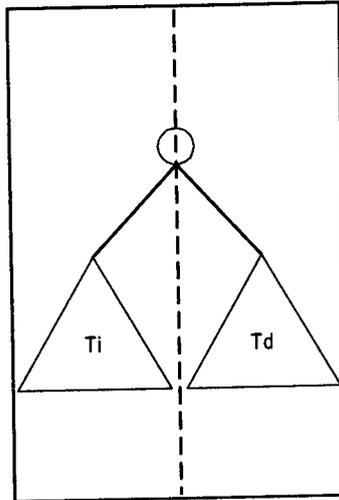


Figura 33. Generalización de ruta interna de longitud máxima.

Por las consideraciones anteriores tenemos que la relación de la longitud máxima de la ruta interna es igual a $I = n(n-1)/2$

Por tanto la ruta interna de longitud máxima se calcula con la siguiente fórmula.

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

Recordando que la ruta externa es igual a $E(T) = I(T) + 2n$, entonces

$$E(T) = \frac{1}{2}n(n-3)$$

V.3.9 Análisis del mejor caso en las búsquedas de árboles binarios

Si se pudieran extender los árboles a n nodos con un mínimo de trayectoria interna, éstos serían tales que no tuvieran nodos especiales en más de un nivel adyacente, la figura 34 en sus incisos a y b muestran dos ejemplos del mejor caso en las búsquedas de árboles binarios.

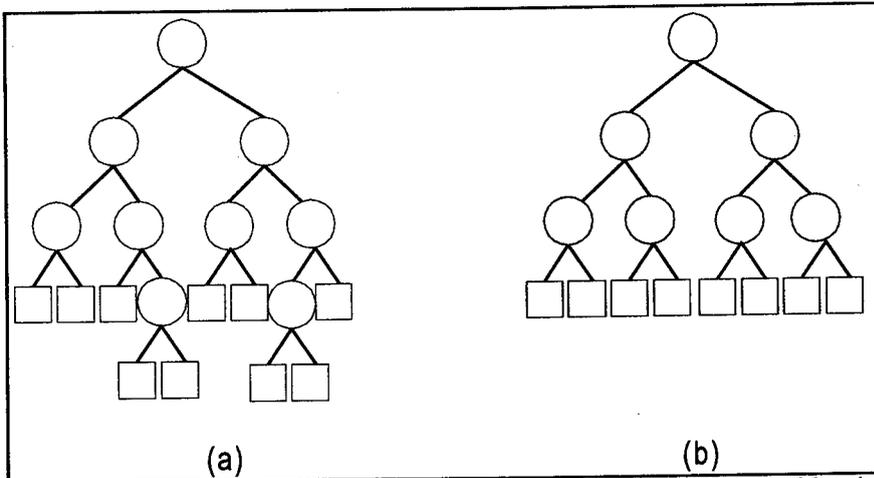


Figura 34. Ejemplos del mejor caso en las búsquedas de árboles binarios.

Para calcular, la suma de las longitudes internas, observamos la siguiente figura 35.

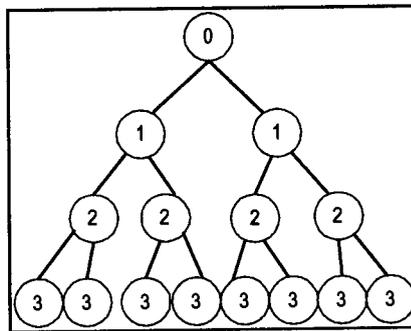


Figura 35. Ejemplo de un árbol para calcular el mejor caso en las búsquedas de árboles binarios.

Como se había mencionado anteriormente la enumeración de los nodos, como la figura 32 tiene un comportamiento como el piso de logaritmos i , por lo es igual a (definido en V.1.3.1)

el mejor caso de ruta interna mínima tenemos que,

$$C = 2 * l + 1$$

substituyendo en la ecuación, tenemos que

$$C_n = 2L_n + n/n$$

V.4 Enumeración de árboles trivalentes

Analizando un trabajo sobre combinatoria, realizado por Eugene Charles Catalán (1814-1894), Arthur Cayley, descubrió que podía ser aplicado a la enumeración de árboles trivalentes; éste trabajo consistía en dividir un polígono convexo en triángulos por diagonales que no se intersectan.

La presente sección muestra como triangular un n-ágono, la asociación de factores, la enumeración de árboles trivalentes y finalmente la relación entre éstos tres temas.

V.4.1 Triangulaciones

Catalán observó que todos los polígonos convexos con $n+2$ lados, podían ser triangulados, por diagonales que no se intersectan, así por ejemplo; un cuadrado puede ser dividido en 2 triangulaciones diferentes, como lo muestra la figura 36.

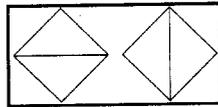


Figura 36. Triangulación de un cuadrado

De la misma manera, un heptágono puede ser triangulado en 5 formas y un octágono puede ser dividido en 132 formas diferentes; las figuras 37 y 38 muestran tales triangulaciones respectivamente.

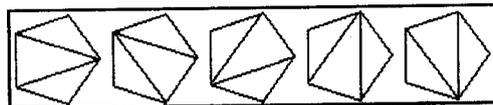


Figura 37. Triangulación de un heptágono

Por tanto a un polígono convexo de n lados ($n=1,2,3,\dots,12$, etc.) le corresponde el siguiente número de triangulaciones:

$T_1=0$, $T_2= 1$, $T_3=1$, $T_4=2$, $T_5=5$, $T_6=14$, $T_7=42$, $T_8=132$, $T_9=429$, $T_{10}=1430$, $T_{11}=4862$, $T_{12}=16796$, ...

Los números de las triangulaciones anteriormente descritos, satisfacen la siguiente fórmula recursiva no-lineal [Rangel, González 1998]

$$\text{Ecuación 5.1} \quad T_{n+1} = T_2 T_n + T_3 T_{n-1} + \dots + T_n T_2$$

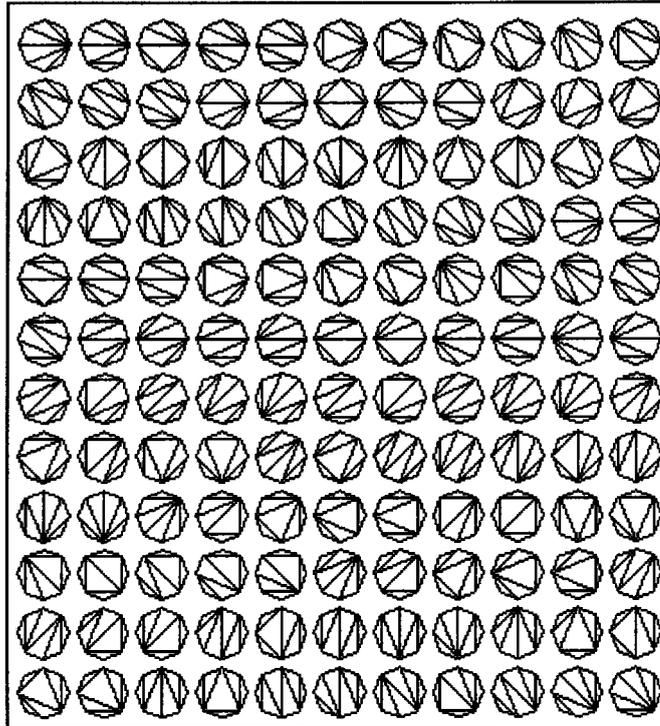


Figura 38. *Triangulaciones de un octagonono*

Los números que se generan de la ecuación recursiva no-lineal son llamados: números de Catalán; así por ejemplo, para calcular el noveno número de Catalán (T_9) por medio de la fórmula anteriormente descrita, se efectúan el siguiente cálculo:

$$T_9 = T_2 T_8 + T_3 T_7 + T_4 T_6 + T_5 T_5 + T_6 T_4 + T_7 T_3 + T_8 T_2 = 132 + 42 + 28 + 25 + 28 + 42 + 132 = 42$$

Estudios posteriores encontraron que la ecuación 5.2 mostrada a continuación, define a éstos números directamente [Rangel, González 1998]:

$$\text{Ecuación 5.2} \quad (n-3)T_n = \frac{n}{2}(T_3 T_{n-1} + T_4 T_{n-2} + \dots + T_{n-1} T_3)$$

Demostración de la ecuación 5.2 para encontrar la fórmula directa de triangulación.

Primero considere un n -ágono convexo con $n > 2$, como lo ilustra la figura 39; el primer triángulo se forma por v_2 y por una diagonal que parte de un punto V_1 a un punto V_i ; éste triángulo (V_1, V_2, V_i) divide en dos partes al n -ágono, al primero lo llamaremos i -ágono y al segundo $(n-i+2)$ -ágono.

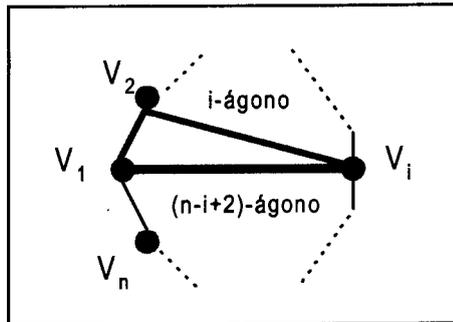


Figura 39. Ejemplo de un n -ágono

El i -ágono se puede triangular de T_i formas, mientras que el n -ágono se puede dividir de T_{n-i+2} formas, a partir de la diagonal. Se sabe que las diagonales que dividen un n -ágono son siempre $n-3$, por lo que tenemos $n-2$ diagonales posibles saliendo del mismo nodo V_1 , éstas pueden ser $V_1, V_3, V_1, V_4, \dots, V_1, V_{n-1}$

Por lo que la suma para V_1 es

$$T_3 T_{n-1} + T_4 T_{n-2} + \dots + T_{n-1} T_3$$

Considere que la suma es la misma para los n vértices, por lo que se genera la siguiente ecuación:

$$n(T_3 T_{n-1} + T_4 T_{n-2} + \dots + T_{n-1} T_3)$$

Por la simetría del polígono en las diagonales, éstas se usan dos veces una, y una por cada diagonal, afectando la ecuación en la forma siguiente:

$$\frac{n}{2}(T_3 T_{n-1} + T_4 T_{n-2} + \dots + T_{n-1} T_3)$$

Seguendo con la lógica, se observa que cada triangulación usa $n-3$ diagonales y para cada triangulación se obtiene el teorema original. Con lo cual queda demostrado.

V.4.2. Asociación de factores.

La triangulación puede usarse como analogía en la asociación de factores; ésta asociación debe tener como característica que cada subproducto de una lista sea la multiplicación de dos factores.

En otras palabras, podemos decir que la asociación de factores, es el número de formas en que los paréntesis pueden ser colocados dentro de una secuencia de elementos, de forma que cada subproducto es la multiplicación de dos elementos. Por ejemplo para 3 elementos (1,2,3), existen 2 formas de asociar los factores como secuencia de dos elementos. Las formas de asociación se observaban así:

$$(1 (2 3)) \quad ((1 2) 3)$$

Para una lista de 4 elementos, corresponden las siguientes 5 formas de asociación:

$$\begin{aligned} &(1 (2 (3 4))) \quad (1 ((2 3) 4)) \\ &((1 2) (3 4)) \quad ((1 (2 3)) 4) \\ &(((1 2) 3) 4) \end{aligned}$$

En una lista de 5 elementos, existen 14 maneras de asociar los elementos de dos en dos, por ejemplo:

$$\begin{aligned} &(1 (2 (3 (4 5)))) \quad (1 (2 ((3 4) 5))) \\ &(1 ((2 3) (4 5))) \quad (1 ((2 (3 4)) 5)) \\ &(1 (((2 3) 4) 5)) \quad ((1 2) (3 (4 5))) \\ &((1 2) ((3 4) 5)) \quad ((1 (2 3)) (4 5)) \\ &((1 (2 (3 4))) 5) \quad ((1 ((2 3) 4)) 5) \\ &(((1 2) 3) (4 5)) \quad (((1 2) (3 4)) 5) \\ &(((1 (2 3)) 4) 5) \quad (((((1 2) 3) 4) 5)) \end{aligned}$$

La lista de asociación de 6 elementos, se incrementa a 42 formas, éstas son:

$$\begin{aligned} &(1 (2 (3 (4 (5 6)))))) \quad (1 (2 (3 ((4 5) 6)))) \\ &(1 (2 ((3 4) (5 6)))) \quad (1 (2 (((3 4) 5) 6))) \\ &(1 (2 (((3 4) 5) 6))) \quad (1 ((2 3) (4 (5 6)))) \end{aligned}$$

(1 ((2 3) ((4 5) 6))) (1 ((2 (3 4)) (5 6)))
 (1 ((2 (3 (4 5))) 6)) (1 ((2 ((3 4) 5)) 6))
 (1 (((2 3) 4) (5 6))) (1 (((2 3) (4 5)) 6))
 (1 (((2 (3 4)) 5) 6)) (1 (((((2 3) 4) 5) 6)))
 ((1 2) (3 (4 (5 6)))) ((1 2) (3 ((4 5) 6)))
 ((1 2) ((3 4) (5 6))) ((1 2) ((3 (4 5)) 6))
 ((1 2) (((3 4) 5) 6)) ((1 (2 3)) (4 (5 6)))
 ((1 (2 3)) ((4 5) 6)) ((1 (2 (3 4))) (5 6))
 ((1 (2 (3 (4 5)))) 6) ((1 (2 ((3 4) 5))) 6)
 ((1 ((2 3) 4)) (5 6)) ((1 ((2 3) (4 5))) 6)
 ((1 ((2 (3 4)) 5)) 6) ((1 (((2 3) 4) 5)) 6)
 (((1 2) 3) (4 (5 6))) (((1 2) 3) ((4 5) 6))
 (((1 2) (3 4)) (5 6)) (((1 2) (3 (4 5))) 6)
 (((1 2) ((3 4) 5)) 6) (((1 (2 3)) 4) (5 6))
 (((1 (2 3)) (4 5)) 6) (((1 (2 (3 4))) 5) 6)
 (((1 ((2 3) 4)) 5) 6) (((((1 2) 3) 4) (5 6)))
 (((((1 2) 3) (4 5)) 6)) (((((1 2) (3 4)) 5) 6))
 ((((((1 2 3) 4) 5) 6)) ((((((1 2) 3) 4) 5) 6))

Es importante observar que esta forma de insertar paréntesis, el orden de los elementos no cambia y que los paréntesis exteriores multiplican siempre dos términos. Por tanto, siendo el primer producto de x_1 a x_p , para alguna posición; el segundo producto de $x_{p+1} \dots x_n$. Se dice que el primer término de las x 's (p) puede ser asociado de a_r formas y las últimas posiciones en $n-r$ de las x 's de a_{n-r} .

Por la regla del producto, es igual a $a_r a_{n-r}$, la sumatoria comienza desde $r=1$ hasta $n-1$, por lo que se genera la ecuación:

$$a_n = a_1 a_{n-1} + a_2 a_{n-2} + \dots + a_{n-1} a_1$$

Ecuación que se analizó en la sección anterior (Ecuación 5.1).

V.4.3. Correspondencia entre triangulaciones y asociaciones

Como puede observarse en las dos secciones anteriores, la fórmula para calcular el número de triangulaciones de un n -ágono y el número de asociaciones

és la misma. La relación que guardan las triangulaciones y las asociaciones puede verse claramente en la figura 40.

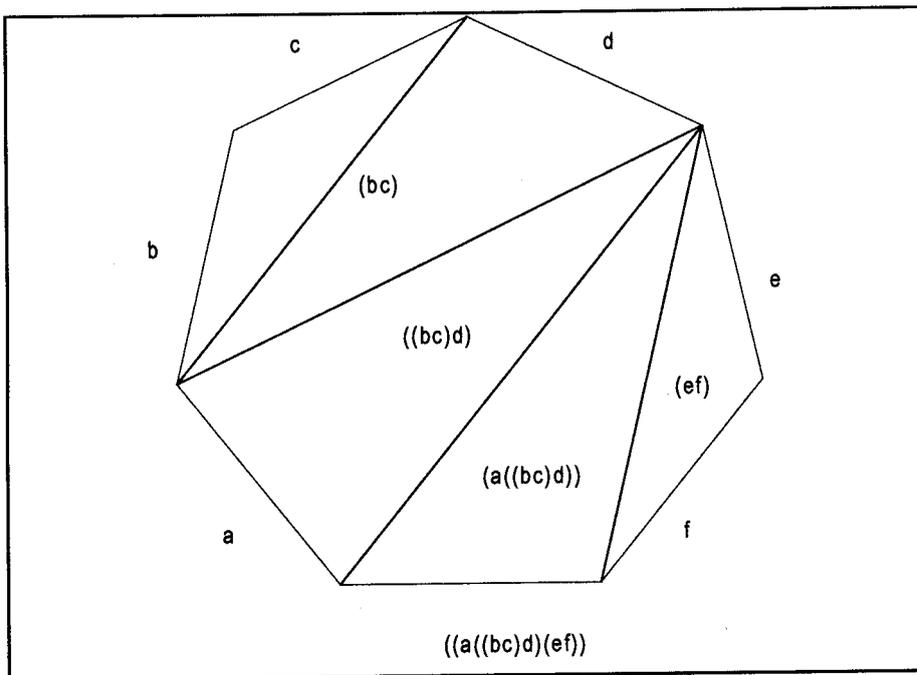


Figura 40. Correspondencia entre triangulaciones y asociaciones

La figura 41 muestra la correspondencia entre las triangulaciones de un octágono y las asociaciones de una lista.

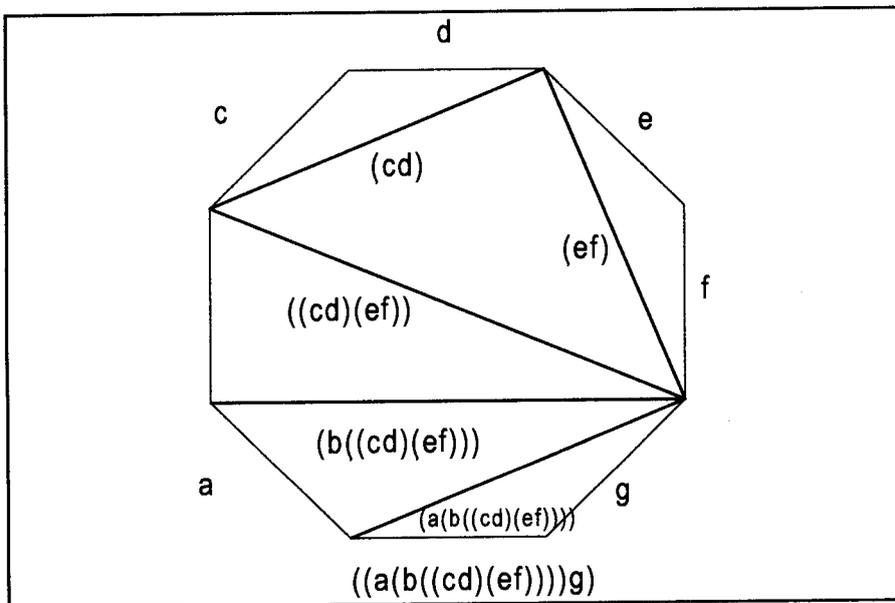


Figura 41. Correspondencia entre triangulaciones de un octágono y las asociaciones de una lista

V.4.4. Enumeración de árboles trivalentes

Un árbol trivalente, es un árbol binario completo, es decir; todos los nodos (excepto la raíz y las hojas) siempre tienen grado 2.

Como se explicó en el capítulo III un árbol puede ser definido como una lista de listas; la manera de asociar ésta lista de forma que siempre tenga un grado 2 en los vértices diferentes de la raíz y de las hojas se analizará a partir de las siguientes observaciones:

El caso base se presenta la figura 42, en ella se muestra como un árbol trivalente de tres vértices puede ser ordenado en dos formas diferentes.

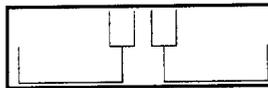


Figura 42. *Árbol trivalente con tres vértices*

Un árbol trivalente de 4 vértices puede ser representado de 5 formas diferentes como lo muestra la figura 43.

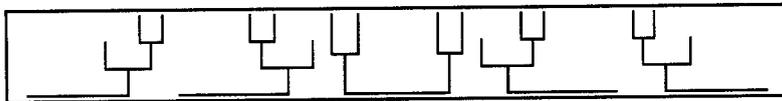


Figura 43. *Árbol trivalente con cuatro vértices*

La figura 44 muestra como con 5 vértices un árbol trivalente puede representarse de 14 formas distintas.

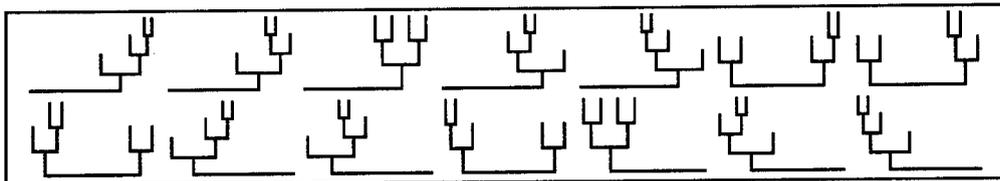


Figura 44. *Árbol trivalente con cinco vértices*

De igual forma un árbol trivalente de 6 vértices puede ser representado por 42 formas diferentes y uno de siete vértices se representa en 132 formas como lo muestra la figura 45 y 46 respectivamente.

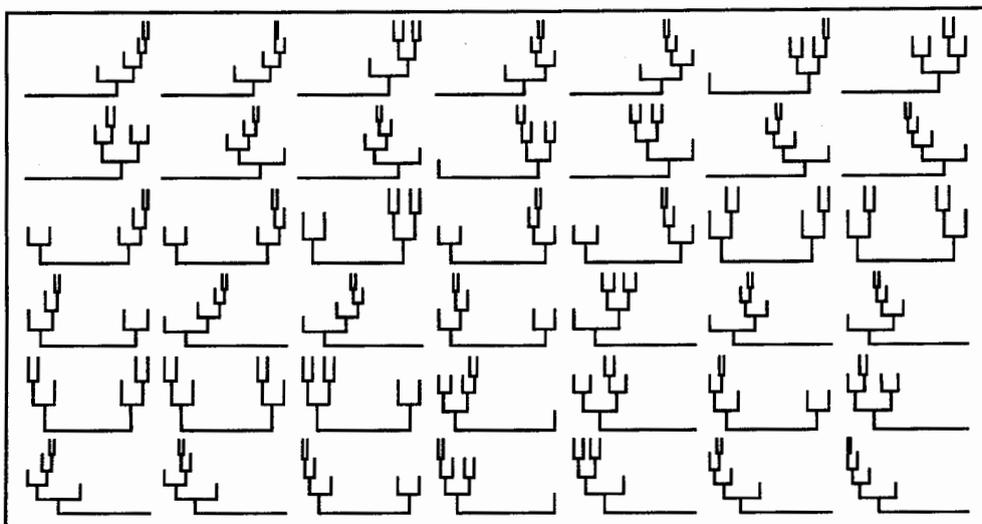


Figura 45. Árboles trivalentes de seis vértices

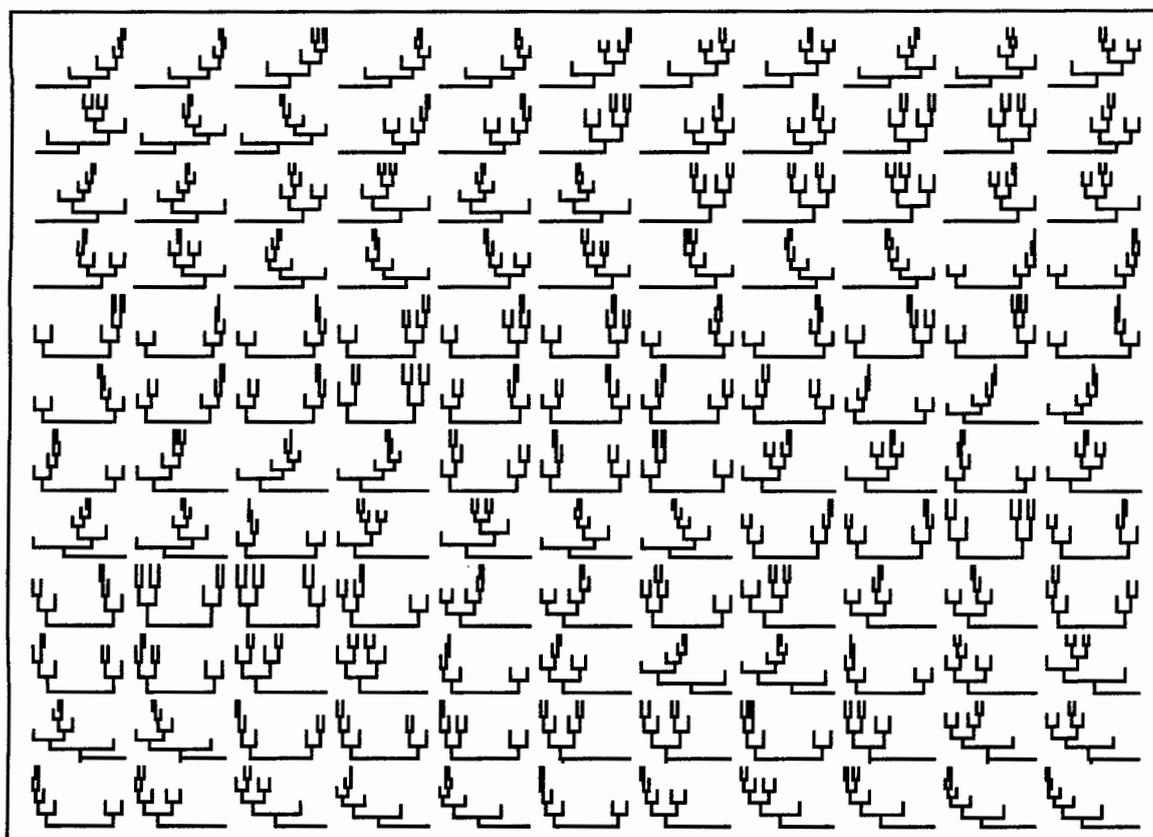


Figura 46. Árboles trivalentes con siete vértices

Un árbol trivalente con ocho vértices un árbol puede representarse de 429 formas distintas. Ver figura 47.

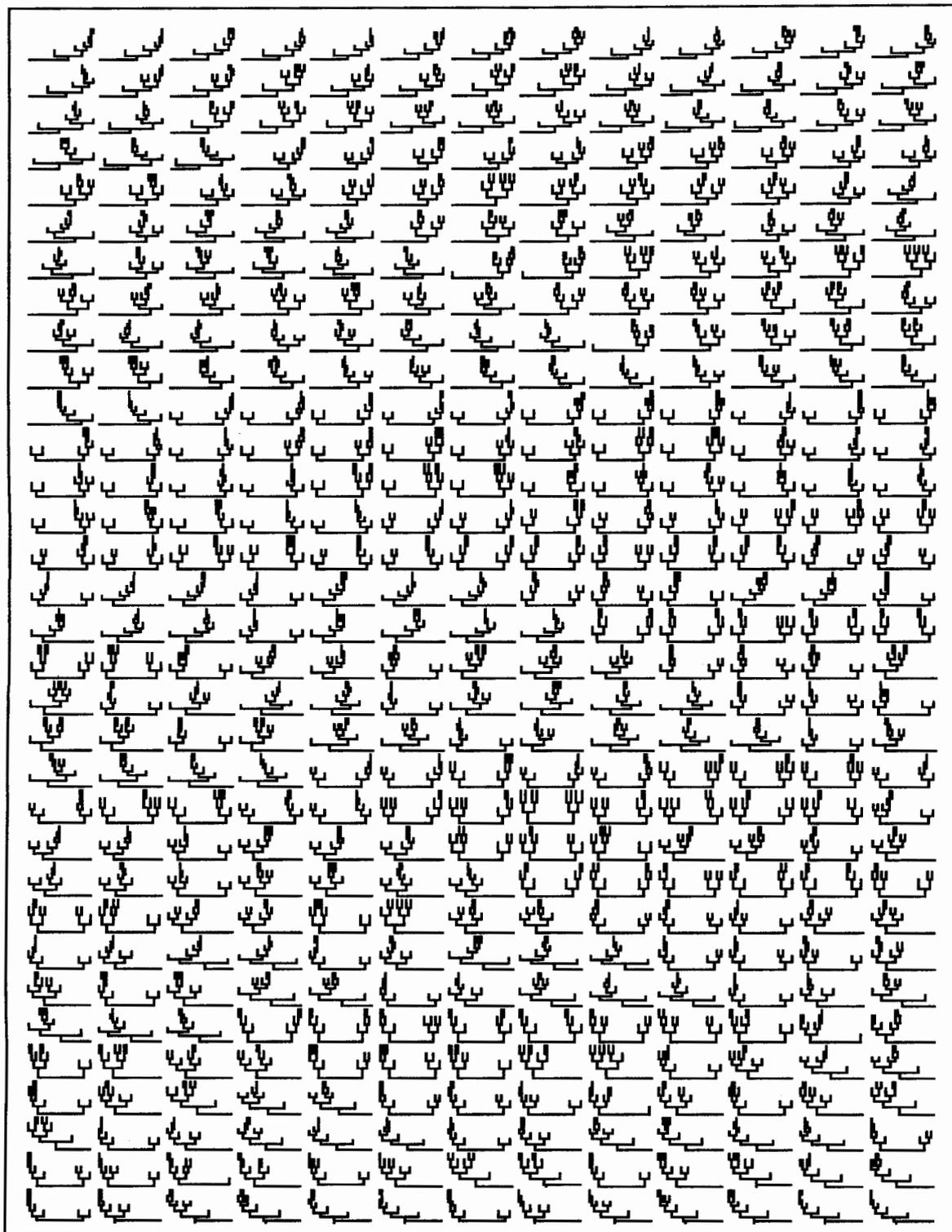


Figura 47. Árboles trivalente con ocho vértices

V.4.5. Relación entre triangulaciones, árboles trivalentes y números binarios.

El conocimiento de la triangulación de los polígonos no solo se aplica a la enumeración de árboles trivalentes y a la asociación de factores también se aplica en un método para generar números binarios.

La figura 48 muestra la relación de un triángulo, un árbol trivalente y la asociación de 2 elementos y su número binario correspondiente. Obsérvese que para la construcción de números binarios, el paréntesis izquierdo "(" se sustituye por 1, los números y los paréntesis derechos ")" por ceros.

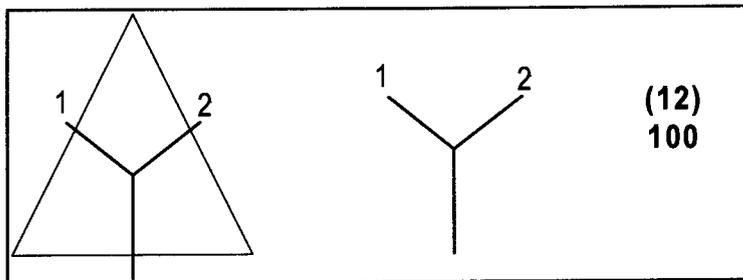


Figura 48. Asociación de un triángulo, un árbol trivalente y la generación de un número binario

De la misma manera, la figura 49 muestra la relación de la triangulación de un cuadrado, 3 vértices de un árbol ternario y la asociación de tres elementos.

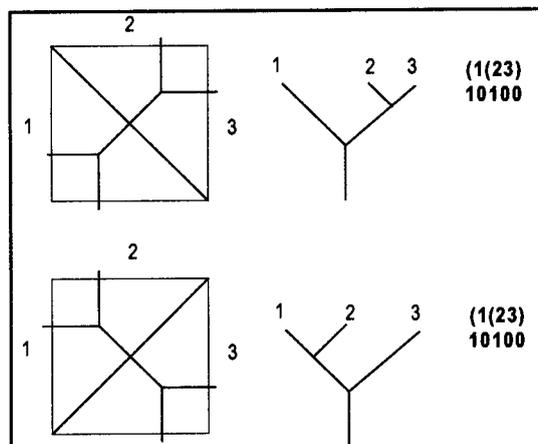


Figura 49. Triangulación de un cuadrado, un árbol ternario y la generación de números binarios

La figura 50 muestra la relación de un pentágono y sus formas de triangulación, la asociación de factores y la generación de cinco números binarios.

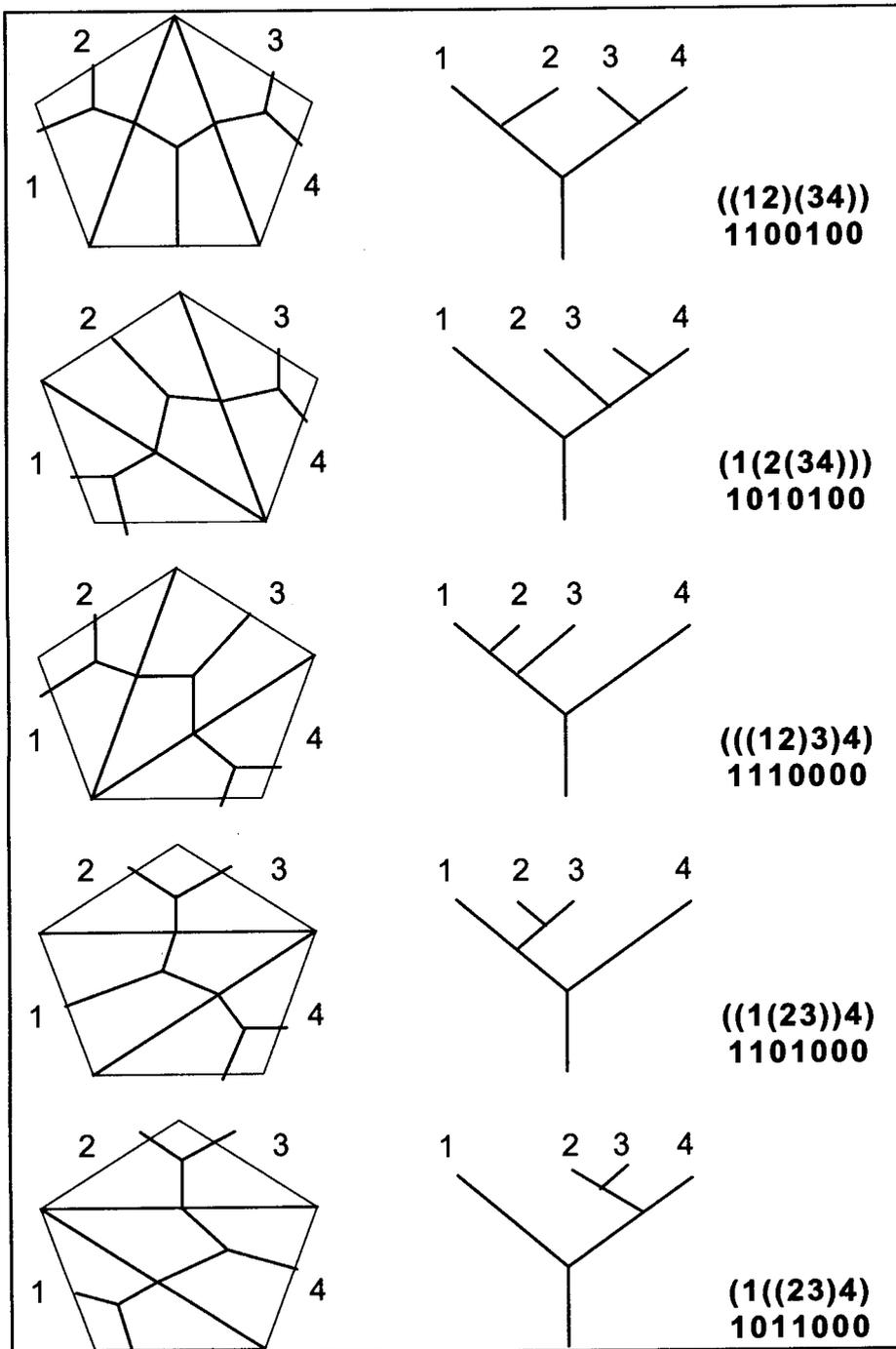


Figura 50. Relación entre un pentágono, su triangulación, árboles ternarios y cinco números binarios.

V.5 Enumeración de árboles no isomorfos

En las secciones anteriores se enumeraron los árboles que se podían generar a partir de n vértices; ahora el interés de ésta sección es analizar cuantos árboles diferentes o no isomorfos se pueden trazar a partir de n vértices.

Para enumerar tales estructuras, se recuerda que el universo de Relaciones se representa por 2^{n^2} [Rangel,González 1998]; las relaciones que nos interesan son las no reflexivas y simétricas, representadas por $2^{\frac{n(n-1)}{2}}$, también sabemos que las relaciones que cumplan con la propiedades de los árboles son $(n-1)$; por tanto el número de árboles no isomorfos se genera por :

$$\binom{\frac{n(n-1)}{2}}{n-1}$$

La tabla 34, muestra la función para determinar que son 20 el número de árboles no isomorfos que se pueden generar a partir de 4 vértices.

Código	$n = 4;$ 2^{n^2} $2^{\frac{n(n-1)}{2}}$ Binomial $\left[\frac{n(n-1)}{2}, n-1 \right]$
Ejemplo	65536 64 20

Tabla 34. Función para determinar el número de árboles no isomorfos.

La tabla 35 muestra una función para obtener los árboles representativos de árboles no isomorfos de cuatro vértices.

Código

```

upperDiag[vertices_] := Module[{m =  $\frac{\text{vertices}(\text{vertices} - 1)}{2}$ , d, aristas = vertices - 1},
  d = Map[IntegerDigits[#, 2, m]&, Range[2m - 1]];
  Select[d, (Plus @@ # == aristas)&]]

MatrixSim[d_] := Module[{n, m, i, j, k = 1},
  n =  $\frac{1 + \sqrt{1 + 8 \text{Length}[d]}}{2}$ ;
  m = Table[0, {n}, {n}];
  Do[
    Do[
      m[[i, j]] = m[[j, i]] = d[[k]]; k = k + 1, {j, i - 1}, {i, 2, n}];
    m]

allNonRefSym[n_] := MatrixSim /@ upperDiag[n]

someZeroRowQ[m_] := Position[(Plus @@ #)& /@ m, 0] != {}

MatrixForm /@ (u = Select[allNonRefSym[4], Not[someZeroRowQ[#]&]])

```

Ejemplo

$$\left\{ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \right.$$

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \left. \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \right\}$$

Tabla 35. Función para representar árboles no isomórfos por matriz

De la función anterior pueden deducirse los árboles por medio de una notación corta, como lo muestra la tabla 36.

<p>Código</p> <pre>v = (fromMatrix[{1, #}]& /@ u); shortNotation /@ v</pre>
<p>Ejemplo</p> <pre>{"1(4(2,3))", "1(4(3(2)))", "1(4(2(3)))", "1(3(4(2)))", "1(3,4(2))", "1(3(2,4))", "1(3(2(4)))", "1(3(2),4)", "1(2(4(3)))", "1(2,4(3))", "1(2(3(4)))", "1(2(3,4))", "1(2(3),4)", "1(2,3(4))", "1(2(4),3)", "1(2,3,4)" }</pre>

Tabla 36. *Arboles no isomorfos de cuatro vértices en notación corta*

VI. LÍNEAS DE INVESTIGACIÓN

Los árboles son frecuentemente utilizados en análisis sintáctico, en redes de comunicación, construcción de compiladores, árboles de comunicación telefónica, ordenamiento de correo, estudio de estructuras de datos, clasificación, teoría de codificación, en ciertos problemas de optimización y en relación de datos en base de datos jerárquicos.

Actualmente las líneas modernas de investigación que Microsoft explora son:

1. Aprendizaje.- El objetivo de los algoritmos de aprendizaje es determinar un modelo gráfico a partir de un conjunto de datos 'en bruto' u observaciones realizadas sobre el comportamiento del sistema.
2. Fusión de redes.- Es posible que para un mismo problema, se encuentren modelos parciales o totales del mismo en fuentes distintas (bases de datos). La reunión de estos modelos en un grafo que modelice el sistema en su globalidad es un problema muy interesante, pero que ha recibido muy poca atención.
3. Modelos en la Toma Decisiones.- En la mayoría de los problemas prácticos no interesa únicamente el conocer el valor de una probabilidad sino el determinar qué opción entre un conjunto de decisiones posibles es la que maximiza la utilidad esperada. Los diagramas de influencia son modelos gráficos que permiten la modelización de este tipo de situaciones mediante la inclusión de nodos de decisión que representan las posibles opciones y nodos de utilidad que valoran los resultados.

En las líneas modernas de investigación, se presenta un método de análisis de variables sociológicas, basadas en árboles. Es un caso para el modelo de la elección racional en combate a la pobreza

VI.1 Un método de análisis de variables sociológicas, basadas en árboles.

Caso. El modelo de la elección racional en combate a la pobreza

- Un Enfoque algorítmico y computacional -

Los científicos sociales se sirven de herramientas para poder sustituir una realidad con un modelo matemático, por el fenómeno mismo; tal es el caso de la Elección Racional, la cual trata de definir un hecho, como una secuencia de acontecimientos. La Elección racional, por su complejidad, puede modelar sus procesos en la Teoría de grafos; en este caso, los árboles exploran posibles escenarios de un problema, integrando un sistema de análisis en la toma de decisiones.

El modelo que se presenta a continuación, es un prototipo en la toma de decisiones racionales para salir de la pobreza, integrando el punto de vista de las familias como sujeto que tiene deseos y necesidades, interactuando en un ambiente que ofrece oportunidades. El ambiente de desarrollo y experimentación es teórico. Las variables pertenecen al sector rural y se desdoblan en forma simple para su comprensión.

Este capítulo, muestra la utilidad de modelos matemáticos para la comprensión de fenómenos sociales, concretando la aplicación de las llamadas estructuras arborescentes en la solución de una problemática social contemporánea como lo es la pobreza en el sector rural de nuestro país.

Con este ejercicio, queremos ilustrar sobre potencialidades que dispone el uso de modelos usados en las matemáticas, con un enfoque computacional para soluciones sociales.

A continuación se presenta una síntesis de la Teoría de Elección Racional, que nos ayudará a comprender la lógica de pensamiento de la creación de los árboles de decisiones y sus resultados.

VI.1.1 Síntesis de la Teoría de elección racional

La Elección Racional [Elster 1995] puede ser definida como un ordenamiento de preferencias basadas en una función de utilidad.

Una preferencia trata de alcanzar una meta en forma óptima, definida por la acción racional. Así mismo la acción racional es el resultado de tres decisiones óptimas: deseos, creencias y pruebas. La figura 51 muestra el proceso ideal de una acción racional.

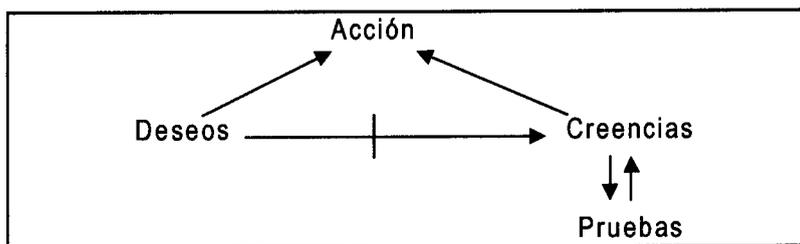


Figura 51. Proceso de acción racional. La elección racional es un ordenamiento de preferencias, esas preferencias son el resultado de acciones racionales y cada acción racional es el conjunto de tres decisiones óptimas.

El objetivo de la elección racional, es optimizar la probabilidad de hallar el mejor medio para fines dados. Para lograrlo; trata de determinar que hará una persona bajo ciertas circunstancias y verifica si lo que hizo la persona, es algo que la teoría predijo que haría.

En general, las teorías pueden tener dos limitaciones:

La indeterminación, es el caso en que la teoría no puede dar predicciones regulares y la impropiedad que es el caso en donde fallan las predicciones.

Los límites en el caso de la Elección Racional son tres:

Primero. La Teoría es indeterminada ya que en la construcción de caminos para alcanzar metas del mejor modo, no brinda recetas inequívocas.

Segundo. Puede ser inapropiada, cuando el individuo actúa "irracionalmente".

Tercero. La impropiedad, que es el caso en donde fallan las predicciones. Para evitar ésta situación, los científicos observan las siguientes premisas:

- a. En todo momento, el sujeto puede comparar dos opciones (deseos u oportunidades)
- b. El sujeto debe ser congruente con sus decisiones
- c. Compara valores entre sí para definir cuando sigue con sus preferencias y cuando no.

Aún con las limitaciones mencionadas, el modelo puede ser una herramienta útil en el análisis de decisión, ya que brindar un marco de certidumbre en la elección, y por otro lado proporciona un nuevo sistema de pruebas y críticas.

VI.2.1.1 Planteamiento del problema

Construcción modelo (prototipo) de elección racional, para el combate a la pobreza, en el sector rural.

VI.2.1.2 Justificación Teórica

La importancia de tratar el problema de la pobreza, radica en la historia misma del país, ya que en México existen 40 millones de personas en condiciones de pobreza extrema, a pesar de que a lo largo de historia moderna en México, se han aprobado muchos programas de bienestar social; tales como PROGRESA, PROCAMPO, etc. En la actualidad, los presupuestos del gobierno estatal y/o federal, dependen de la aprobación de sus respectivas legislaturas; dicha aprobación se liga al análisis de la distribución del gasto, especialmente a los rubros dedicados a combatir la pobreza. La ciudad de Santiago de Querétaro, no es la excepción, por ejemplo; el 28 de enero del 2000 se da a conocer la distribución del gasto para la lucha frontal contra la pobreza (Diario Oficial de la ciudad de Santiago de Querétaro, *La Sombra de Arteaga*, Acuerdo para la distribución de los recursos del ramo 33 combate a la pobreza extrema); lo que demuestra como un hecho la importancia y la actualidad del tema.

La importancia teórica resalta, ante la ausencia de modelos matemáticos que sirvan para la explicación de fenómenos sociales.

VI.2.1.3 Objetivos

- Proponer una metodología de análisis matemático, de un fenómeno social.
- Identificar las variables sociológicas más importantes para combatir la pobreza
- Indicar las opciones que racionalmente tenga una comunidad para salir de la pobreza.

VI.2.1.4 Antecedentes

Por una parte, los estudios sobre la pobreza, generalmente son dirigidos desde el nivel gubernamental hacia los sujetos (individuos, comunidades, municipios o estados). Por otra parte, los análisis tradicionales de la pobreza, se basan básicamente en la observación de un fenómeno, construcción de un marco teórico, y análisis del problema mediante entrevistas, encuestas y estudios de conteo, tales como los Censos.

VI.2.1.5 Definición del problema.

A través de un modelo matemático, analizar la pobreza desde el punto de vista de una familia o una comunidad, cuya meta sería salir de la pobreza.

VI.2.1.6 Hipótesis

Dado que las estructuras arborescentes sirven como modelos abstractos para explicar una gran variedad de relaciones, es posible construir un modelo de elección racional de combate a la pobreza, mediante árboles de decisión.

VI.2.1.7 Metodología

- Definición de pobreza
- Establecer el modelo
 - Definir las variables que intervienen en el problema
 - Delimitación del problema
 - Construcción del modelo
 - Proponer clases de familias
 - Análisis de hipótesis.
 - Construcción de hipótesis y calificación

- Pruebas.
 - Determinar el comportamiento de las variables por sistema de hipótesis.
- Resultados
- Conclusiones

VI.2.1.7.1 Definición de pobreza

Existen varias definiciones de pobreza, la idea central es, que son condiciones socio-económicas debajo de un nivel considerado como mínimo indispensable para una vida digna. Para nuestra aplicación, la pobreza es un rezago de las cuatro necesidades de un i-ésimo municipio respecto al rezago estatal en esa misma necesidad (Manual de Operación del programa PROGRESA Programa de Educación y salud para la gente en pobreza extrema).

VI.2.1.7.2 Establecer el modelo

Dada la definición anterior, y a fin de establecer el modelo, se seleccionaron como documentos jueces, los siguientes documentos:

D1. Diario Oficial de la ciudad de Santiago de Querétaro, *La Sombra de Arteaga*. PODER EJECUTIVO 28 de enero del 2000. Acuerdo para la distribución de los recursos del ramo 33 (combate a la pobreza extrema).

D2. Manual de Operación del programa PROGRESA (Programa de Educación y salud para la gente en pobreza extrema).

Es importante señalar que los documentos jueces son documentos generados por expertos y considerados como una base para el análisis para observar cuales son las variables más importantes que influyen en la pobreza.

VI.2.1.7.2.1 Definir las variables que intervienen en el problema

Del documento juez D1, que ofrece una fórmula para la asignación de recursos del ramo 33 (Fondo de aportaciones a la pobreza) y de las variables que socioeconómicas que registra el programa PROGRESA, se observó que las variables de mayor impacto en la pobreza eran las siguientes:

1. El ingreso. Dinero obtenido según la actividad económica de cada miembro.
2. Educación. Medida principalmente como la población que sabe leer y escribir,
3. Condiciones de la vivienda. Que al menos exista el servicio de drenaje y electricidad.
4. Medio ambiente. Existencia de servicios de educación, servicio de salud y abasto.

VI.2.1.7.2.2 Delimitación del problema

A fin de limitar nuestro modelo las variables seleccionadas, de acuerdo a la fórmula para calcular la distribución de los gastos son las siguientes:

1. Ingreso
2. No. De miembros
3. Oficio
4. Nivel de estudios

VI.2.1.7.2.3 Construcción del modelo

Un modelo de elección racional permite estudiar un problema de forma multilateral, es decir; analiza un acontecimiento como un conjunto de hechos, para nuestro estudio el modelo de la pobreza, sería modelado junto con deseos y oportunidades, tal y como lo muestra la figura 52.

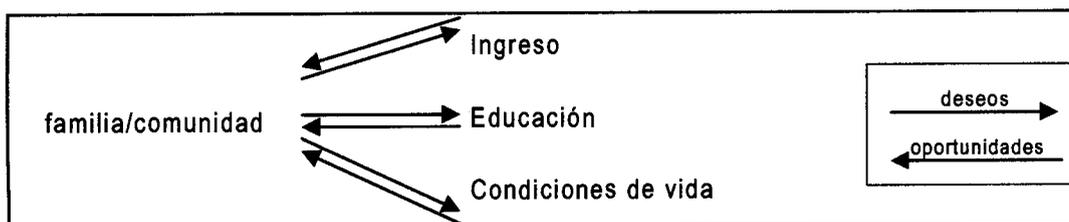


Figura 52. Iteración de la pobreza con deseos y oportunidades

Cabe mencionar que un modelo para analizar problemas en las ciencias sociales son árboles de decisiones, pero también se pueden modelar por sistemas complejos no estocásticos. Por ejemplo; la dinámica poblacional y economía caótica [Monroy 1997]. Prigogine muestra múltiples ejemplos sociales resueltos

bajo éste enfoque. Dadas las características del problema, se modelará por un árbol de decisiones.

La importancia del modelo social basado en la elección racional, es optimizar la probabilidad de hallar el mejor medio para fines dados. Para lograrlo; trata de determinar que hará un sujeto en ciertas circunstancias y verifica si lo que hizo la sujeto, es algo que la teoría predijo que haría [Elster1991].

Ahora bien, de acuerdo a la definición, al modelo de Iteración de la pobreza y a las variables seleccionadas, el modelo se vería como la figura 53. El modelo de árbol de decisión marca la historia, la dinámica y el orden temporal que en ese momento es importante para la creación del objeto "sujeto"

Cada variable, representa un actor social que se ve modificado de a cuerdo en función de sus propios criterios de desdoblamiento

El desdoblamiento de las variables se basa en las variables utilizadas por el programa PROGRESA, para definir la pobreza.

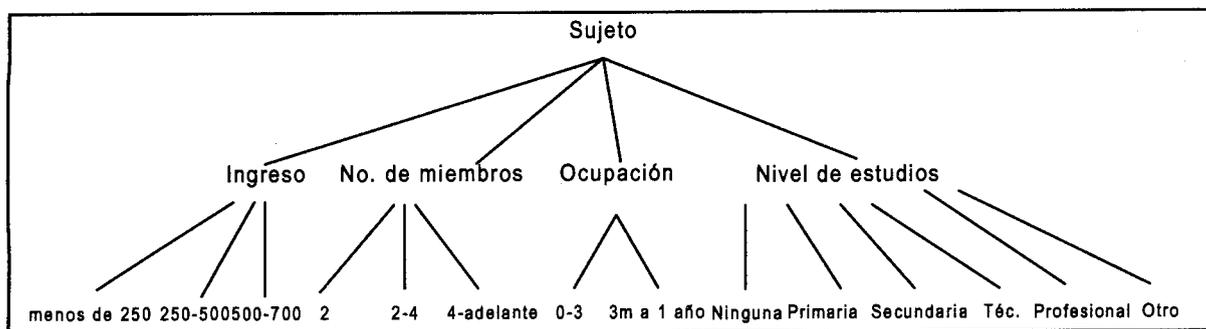


Figura 53. Modelo de pobreza.

VI.2.1.7.2.3.1 Proponer clases de familias

Al analizar el árbol anterior observamos que el número de familias es: (por regla del producto) $6 \cdot 2 \cdot 3 \cdot 3 = 108$ clases de familia que serían las que se muestran a continuación en las siguientes 3 tablas de tipologías.

Tipologías de Familias 1					
Clase	Ingreso	No. De miembros	Ocupación	Nivel de estudios	Calificación
1	Menos de 250	2	1 a 3 meses	Ninguno	
2	Menos de 250	2	1 a 3 meses	Primaria	
3	Menos de 250	2	1 a 3 meses	Secundaria	
4	Menos de 250	2	1 a 3 meses	Técnico	
5	Menos de 250	2	1 a 3 meses	Profesional	
6	Menos de 250	2	1 a 3 meses	Otro	
7	Menos de 250	2	3 a 1 año	Ninguno	
8	Menos de 250	2	3 a 1 año	Primaria	
9	Menos de 250	2	3 a 1 año	Secundaria	
10	Menos de 250	2	3 a 1 año	Técnico	
11	Menos de 250	2	3 a 1 año	Profesional	
12	Menos de 250	2	3 a 1 año	Otro	
13	Menos de 250	2-4	1 a 3 meses	Ninguno	
14	Menos de 250	2-4	1 a 3 meses	Primaria	
15	Menos de 250	2-4	1 a 3 meses	Secundaria	
16	Menos de 250	2-4	1 a 3 meses	Técnico	
17	Menos de 250	2-4	1 a 3 meses	Profesional	
18	Menos de 250	2-4	1 a 3 meses	Otro	
19	Menos de 250	2-4	3 a 1 año	Ninguno	
20	Menos de 250	2-4	3 a 1 año	Primaria	
21	Menos de 250	2-4	3 a 1 año	Secundaria	
22	Menos de 250	2-4	3 a 1 año	Técnico	
23	Menos de 250	2-4	3 a 1 año	Profesional	
24	Menos de 250	2-4	3 a 1 año	Otro	
25	Menos de 250	4-7	1 a 3 meses	Ninguno	
26	Menos de 250	4-7	1 a 3 meses	Primaria	
27	Menos de 250	4-7	1 a 3 meses	Secundaria	
28	Menos de 250	4-7	1 a 3 meses	Técnico	
29	Menos de 250	4-7	1 a 3 meses	Profesional	
30	Menos de 250	4-7	1 a 3 meses	Otro	
31	Menos de 250	4-7	3 a 1 año	Ninguno	
32	Menos de 250	4-7	3 a 1 año	Primaria	
33	Menos de 250	4-7	3 a 1 año	Secundaria	
34	Menos de 250	4-7	3 a 1 año	Técnico	
35	Menos de 250	4-7	3 a 1 año	Profesional	
36	Menos de 250	4-7	3 a 1 año	Otro	

Tabla 37. Primera tipologías de familias

Tipologías de Familias 2

Clase	Ingreso	No. De miembros	Ocupación	Nivel de estudios	de Calificación
37	250 a 500 pesos	2	1 a 3 meses	Ninguno	
38	250 a 500 pesos	2	1 a 3 meses	Primaria	
39	250 a 500 pesos	2	1 a 3 meses	Secundaria	
40	250 a 500 pesos	2	1 a 3 meses	Técnico	
41	250 a 500 pesos	2	1 a 3 meses	Profesional	
42	250 a 500 pesos	2	1 a 3 meses	Otro	
43	250 a 500 pesos	2	3 a 1 año	Ninguno	
44	250 a 500 pesos	2	3 a 1 año	Primaria	
45	250 a 500 pesos	2	3 a 1 año	Secundaria	
46	250 a 500 pesos	2	3 a 1 año	Técnico	
46	250 a 500 pesos	2	3 a 1 año	Profesional	
48	250 a 500 pesos	2	3 a 1 año	Otro	
49	250 a 500 pesos	2-4	1 a 3 meses	Ninguno	
50	250 a 500 pesos	2-4	1 a 3 meses	Primaria	
51	250 a 500 pesos	2-4	1 a 3 meses	Secundaria	
52	250 a 500 pesos	2-4	1 a 3 meses	Técnico	
53	250 a 500 pesos	2-4	1 a 3 meses	Profesional	
54	250 a 500 pesos	2-4	1 a 3 meses	Otro	
55	250 a 500 pesos	2-4	3 a 1 año	Ninguno	
56	250 a 500 pesos	2-4	3 a 1 año	Primaria	
57	250 a 500 pesos	2-4	3 a 1 año	Secundaria	
58	250 a 500 pesos	2-4	3 a 1 año	Técnico	
59	250 a 500 pesos	2-4	3 a 1 año	Profesional	
60	250 a 500 pesos	2-4	3 a 1 año	Otro	
61	250 a 500 pesos	4-7	1 a 3 meses	Ninguno	
62	250 a 500 pesos	4-7	1 a 3 meses	Primaria	
63	250 a 500 pesos	4-7	1 a 3 meses	Secundaria	
64	250 a 500 pesos	4-7	1 a 3 meses	Técnico	
65	250 a 500 pesos	4-7	1 a 3 meses	Profesional	
66	250 a 500 pesos	4-7	1 a 3 meses	Otro	
67	250 a 500 pesos	4-7	3 a 1 año	Ninguno	
68	250 a 500 pesos	4-7	3 a 1 año	Primaria	
69	250 a 500 pesos	4-7	3 a 1 año	Secundaria	
70	250 a 500 pesos	4-7	3 a 1 año	Técnico	
71	250 a 500 pesos	4-7	3 a 1 año	Profesional	
72	250 a 500 pesos	4-7	3 a 1 año	Otro	

Tabla 38. Segunda tipologías de familias

Tipologías de Familias 3

Clase	Ingreso	No. De miembros	Ocupación	Nivel de estudios	de	Calificación
73	500 a 700 pesos	2	1 a 3 meses	Ninguno		
74	500 a 700 pesos	2	1 a 3 meses	Primaria		
75	500 a 700 pesos	2	1 a 3 meses	Secundaria		
76	500 a 700 pesos	2	1 a 3 meses	Técnico		
77	500 a 700 pesos	2	1 a 3 meses	Profesional		
78	500 a 700 pesos	2	1 a 3 meses	Otro		
79	500 a 700 pesos	2	3 a 1 año	Ninguno		
80	500 a 700 pesos	2	3 a 1 año	Primaria		
81	500 a 700 pesos	2	3 a 1 año	Secundaria		
82	500 a 700 pesos	2	3 a 1 año	Técnico		
83	500 a 700 pesos	2	3 a 1 año	Profesional		
84	500 a 700 pesos	2	3 a 1 año	Otro		
85	500 a 700 pesos	2-4	1 a 3 meses	Ninguno		
86	500 a 700 pesos	2-4	1 a 3 meses	Primaria		
87	500 a 700 pesos	2-4	1 a 3 meses	Secundaria		
88	500 a 700 pesos	2-4	1 a 3 meses	Técnico		
89	500 a 700 pesos	2-4	1 a 3 meses	Profesional		
90	500 a 700 pesos	2-4	1 a 3 meses	Otro		
91	500 a 700 pesos	2-4	3 a 1 año	Ninguno		
92	500 a 700 pesos	2-4	3 a 1 año	Primaria		
93	500 a 700 pesos	2-4	3 a 1 año	Secundaria		
94	500 a 700 pesos	2-4	3 a 1 año	Técnico		
95	500 a 700 pesos	2-4	3 a 1 año	Profesional		
96	500 a 700 pesos	2-4	3 a 1 año	Otro		
97	500 a 700 pesos	4-7	1 a 3 meses	Ninguno		
98	500 a 700 pesos	4-7	1 a 3 meses	Primaria		
99	500 a 700 pesos	4-7	1 a 3 meses	Secundaria		
100	500 a 700 pesos	4-7	1 a 3 meses	Técnico		
101	500 a 700 pesos	4-7	1 a 3 meses	Profesional		
102	500 a 700 pesos	4-7	1 a 3 meses	Otro		
103	500 a 700 pesos	4-7	3 a 1 año	Ninguno		
104	500 a 700 pesos	4-7	3 a 1 año	Primaria		
105	500 a 700 pesos	4-7	3 a 1 año	Secundaria		
106	500 a 700 pesos	4-7	3 a 1 año	Técnico		
107	500 a 700 pesos	4-7	3 a 1 año	Profesional		
108	500 a 700 pesos	4-7	3 a 1 año	Otro		

Tabla 39. Tercera tipología de familias

VI.2.1.7.2.3.2 Análisis de hipótesis.

Para trabajar con precisión debe de reducirse este número a un número representativo de familias. Por lo que se debe construir un sistema de hipótesis y calificarlas.

a) La calificación de variables se observa en la siguiente tabla según las oportunidades.

Tabla de calificación de variables

Óptimas	1.
Medianas	.75
Bajas	.50
Nulas	.25

Tabla 40. Escala de calificación de variables

Construcción de hipótesis y calificación

El problema de construir un sistema de hipótesis, consiste en generar un mínimo de hipótesis que prueben o desapruében el modelo, al menos se deben generar 70, pero entre mayor es el número de hipótesis, el modelo es más estable.

En una primera instancia se pensó en la combinación de todas las variables para generar el sistema de hipótesis, pero en el anexo 3, se muestra el resultado y conclusión por que no es viable este método.

La metodología que se propone para analizar y general el sistema de hipótesis es la asociación de términos, de la enumeración de árboles trivalentes ver capítulo de enumeración, árboles trivalentes, relación de triangulación de figuras y generación de números binarios.

A continuación se presenta las siguientes 7 tablas, donde se construye el sistema de 287 hipótesis (Todas las hipótesis, por ejemplo; se leen: a menos de \$250 de ingreso y con una familia de 4 integrantes, sus posibilidades de salir de la pobreza son.....)

Tabla de Hipótesis 1

		Análisis de 1 variable		
Ingreso	Posibilidades			
Menos de 250	.25			
250 a 500	.75			
500 a 700	1.			
No. De miembros				
2	1.			
2-4	.50			
4-7	.25			
Ocupación				
1 a 3 meses	.25			
3 a 1 año	1.			
Educación				
Ninguno	.25			
Primaria	.50			
Secundaria	.50			
Técnico	.75			
Profesional	.75			
Otro (capacitación)	1.			
		2 variables Ingreso /# de integrantes		
Menos de 250	2	1.25		
Menos de 250	2-4	.75		
Menos de 250	4-7	.50		
250 a 500	2	1.75		
250 a 500	2-4	1.25		
250 a 500	4-7	1.0		
500 a 700	2	2.0		
500 a 700	2-4	1.50		
500 a 700	4-7	1.25		
		2 variables Ingreso / ocupación		
Menos de 250	1 a 3 meses	.50		
Menos de 250	3 a 1 año	1.25		
250 a 500	1 a 3 meses			
250 a 500	3 a 1 año			
500 a 700	1 a 3 meses			
500 a 700	3 a 1 año			

Tabla 41. Tabla de hipótesis uno sobre tipologías de familias

Tabla de Hipótesis 2				
		2 variables Ingreso / ocupación		
Menos de 250	Ninguno			
Menos de 250	Primaria			
Menos de 250	Secundaria			
Menos de 250	Técnico			
Menos de 250	Profesional			
Menos de 250	Otro (capacitación)			
250 a 500	Ninguno			
250 a 500	Primaria			
250 a 500	Secundaria			
250 a 500	Técnico			
250 a 500	Profesional			
250 a 500	Otro (capacitación)			
500 a 700	Ninguno			
500 a 700	Primaria			
500 a 700	Secundaria			
500 a 700	Técnico			
500 a 700	Profesional			
500 a 700	Otro (capacitación)			
		3 variables Ingreso, no. de miembros, empleo		
Menos de 250	2	1 a 3 meses		
Menos de 250	2	3 a 1 año		
Menos de 250	2-4	1 a 3 meses		
Menos de 250	2-4	3 a 1 año		
Menos de 250	4-7	1 a 3 meses		
Menos de 250	4-7	3 a 1 año		
250 a 500	2	1 a 3 meses		
250 a 500	2	3 a 1 año		
250 a 500	2-4	1 a 3 meses		
250 a 500	2-4	3 a 1 año		
250 a 500	4-7	1 a 3 meses		
250 a 500	4-7	3 a 1 año		
500 a 700	2	1 a 3 meses		
500 a 700	2	3 a 1 año		
500 a 700	2-4	1 a 3 meses		
500 a 700	2-4	3 a 1 año		
500 a 700	4-7	1 a 3 meses		
500 a 700	4-7	3 a 1 año		

Tabla 42. Tabla de hipótesis dos sobre tipologías de familias

Tabla de Hipótesis 3			
		3 variables Ingreso, # de miembros, educación	
Menos de 250	2	Ninguno	
Menos de 250	2	Primaria	
Menos de 250	2	Secundaria	
Menos de 250	2	Técnico	
Menos de 250	2	Profesional	
Menos de 250	2	Otro (capacitación)	
Menos de 250	2-4	Ninguno	
Menos de 250	2-4	Primaria	
Menos de 250	2-4	Secundaria	
Menos de 250	2-4	Técnico	
Menos de 250	2-4	Profesional	
Menos de 250	2-4	Otro (capacitación)	
Menos de 250	4-7	Ninguno	
Menos de 250	4-7	Primaria	
Menos de 250	4-7	Secundaria	
Menos de 250	4-7	Técnico	
Menos de 250	4-7	Profesional	
Menos de 250	4-7	Otro (capacitación)	
250 a 500	2	Ninguno	
250 a 500	2	Primaria	
250 a 500	2	Secundaria	
250 a 500	2	Técnico	
250 a 500	2	Profesional	
250 a 500	2	Otro (capacitación)	
250 a 500	2-4	Ninguno	
250 a 500	2-4	Primaria	
250 a 500	2-4	Secundaria	
250 a 500	2-4	Técnico	
250 a 500	2-4	Profesional	
250 a 500	2-4	Otro (capacitación)	
250 a 500	4-7	Ninguno	
250 a 500	4-7	Primaria	
250 a 500	4-7	Secundaria	
250 a 500	4-7	Técnico	
250 a 500	4-7	Profesional	
250 a 500	4-7	Otro (capacitación)	
500 a 700	2	Ninguno	
500 a 700	2	Primaria	
500 a 700	2	Secundaria	
500 a 700	2	Técnico	
500 a 700	2	Profesional	
500 a 700	2	Otro (capacitación)	
500 a 700	2-4	Ninguno	
500 a 700	2-4	Primaria	
500 a 700	2-4	Secundaria	
500 a 700	2-4	Técnico	
500 a 700	2-4	Profesional	
500 a 700	2-4	Otro (capacitación)	
500 a 700	4-7	Ninguno	
500 a 700	4-7	Primaria	
500 a 700	4-7	Secundaria	
500 a 700	4-7	Técnico	
500 a 700	4-7	Profesional	
500 a 700	4-7	Otro (capacitación)	

Tabla 43. *Tabla de hipótesis tres sobre tipologías de familias*

Tabla de Hipótesis 4

		3 variables Ingreso, empleo, educación		
Menos de 250	1 a 3 meses	Ninguno		
Menos de 250	1 a 3 meses	Primaria		
Menos de 250	1 a 3 meses	Secundaria		
Menos de 250	1 a 3 meses	Técnico		
Menos de 250	1 a 3 meses	Profesional		
Menos de 250	1 a 3 meses	Otro (capacitación)		
Menos de 250	3 a 1 año	Ninguno		
Menos de 250	3 a 1 año	Primaria		
Menos de 250	3 a 1 año	Secundaria		
Menos de 250	3 a 1 año	Técnico		
Menos de 250	3 a 1 año	Profesional		
Menos de 250	3 a 1 año	Otro (capacitación)		
250 a 500	1 a 3 meses	Ninguno		
250 a 500	1 a 3 meses	Primaria		
250 a 500	1 a 3 meses	Secundaria		
250 a 500	1 a 3 meses	Técnico		
250 a 500	1 a 3 meses	Profesional		
250 a 500	1 a 3 meses	Otro (capacitación)		
250 a 500	3 a 1 año	Ninguno		
250 a 500	3 a 1 año	Primaria		
250 a 500	3 a 1 año	Secundaria		
250 a 500	3 a 1 año	Técnico		
250 a 500	3 a 1 año	Profesional		
250 a 500	3 a 1 año	Otro (capacitación)		
500 a 700	1 a 3 meses	Ninguno		
500 a 700	1 a 3 meses	Primaria		
500 a 700	1 a 3 meses	Secundaria		
500 a 700	1 a 3 meses	Técnico		
500 a 700	1 a 3 meses	Profesional		
500 a 700	1 a 3 meses	Otro (capacitación)		
500 a 700	3 a 1 año	Ninguno		
500 a 700	3 a 1 año	Primaria		
500 a 700	3 a 1 año	Secundaria		
500 a 700	3 a 1 año	Técnico		
500 a 700	3 a 1 año	Profesional		
500 a 700	3 a 1 año	Otro (capacitación)		

Tabla 44. Tabla de hipótesis cuatro sobre tipologías de familias

Tabla de Hipótesis 5

3 variables: Ingreso, número de miembros v educación			
Menos de 250	2	Ninguno	
Menos de 250	2	Primaria	
Menos de 250	2	Secundaria	
Menos de 250	2	Técnico	
Menos de 250	2	Profesional	
Menos de 250	2	Otro (capacitación)	
Menos de 250	2-4	Ninguno	
Menos de 250	2-4	Primaria	
Menos de 250	2-4	Secundaria	
Menos de 250	2-4	Técnico	
Menos de 250	2-4	Profesional	
Menos de 250	2-4	Otro (capacitación)	
Menos de 250	4-7	Ninguno	
Menos de 250	4-7	Primaria	
Menos de 250	4-7	Secundaria	
Menos de 250	4-7	Técnico	
Menos de 250	4-7	Profesional	
Menos de 250	4-7	Otro (capacitación)	
250 a 500	2	Ninguno	
250 a 500	2	Primaria	
250 a 500	2	Secundaria	
250 a 500	2	Técnico	
250 a 500	2	Profesional	
250 a 500	2	Otro (capacitación)	
250 a 500	2-4	Ninguno	
250 a 500	2-4	Primaria	
250 a 500	2-4	Secundaria	
250 a 500	2-4	Técnico	
250 a 500	2-4	Profesional	
250 a 500	2-4	Otro (capacitación)	
250 a 500	4-7	Ninguno	
250 a 500	4-7	Primaria	
250 a 500	4-7	Secundaria	
250 a 500	4-7	Técnico	
250 a 500	4-7	Profesional	
250 a 500	4-7	Otro (capacitación)	
500 a 700	2	Ninguno	
500 a 700	2	Primaria	
500 a 700	2	Secundaria	
500 a 700	2	Técnico	
500 a 700	2	Profesional	
500 a 700	2	Otro (capacitación)	
500 a 700	2-4	Ninguno	
500 a 700	2-4	Primaria	
500 a 700	2-4	Secundaria	
500 a 700	2-4	Técnico	
500 a 700	2-4	Profesional	
500 a 700	2-4	Otro (capacitación)	
500 a 700	4-7	Ninguno	
500 a 700	4-7	Primaria	
500 a 700	4-7	Secundaria	
500 a 700	4-7	Técnico	
500 a 700	4-7	Profesional	
500 a 700	4-7	Otro (capacitación)	

Tabla 45. Tabla de hipótesis cinco sobre tipologías de familias

Tabla de Hipótesis 6				
4 variables ingreso, No. Integrantes, empleo y educación				
Menos de 250	2	1 a 3 meses	Ninguno	1.75
Menos de 250	2	1 a 3 meses	Primaria	2
Menos de 250	2	1 a 3 meses	Secundaria	2
Menos de 250	2	1 a 3 meses	Técnico	2.25
Menos de 250	2	1 a 3 meses	Profesional	2.25
Menos de 250	2	1 a 3 meses	Otro (capacitación)	2.5
Menos de 250	2	3 a 1 año	Ninguno	2.5
Menos de 250	2	3 a 1 año	Primaria	2.75
Menos de 250	2	3 a 1 año	Secundaria	2.75
Menos de 250	2	3 a 1 año	Técnico	3
Menos de 250	2	3 a 1 año	Profesional	3
Menos de 250	2	3 a 1 año	Otro (capacitación)	3.25
Menos de 250	2-4	1 a 3 meses	Ninguno	1.25
Menos de 250	2-4	1 a 3 meses	Primaria	1.5
Menos de 250	2-4	1 a 3 meses	Secundaria	1.5
Menos de 250	2-4	1 a 3 meses	Técnico	1.75
Menos de 250	2-4	1 a 3 meses	Profesional	1.75
Menos de 250	2-4	1 a 3 meses	Otro (capacitación)	2
Menos de 250	2-4	3 a 1 año	Ninguno	2
Menos de 250	2-4	3 a 1 año	Primaria	2.25
Menos de 250	2-4	3 a 1 año	Secundaria	2.25
Menos de 250	2-4	3 a 1 año	Técnico	2.5
Menos de 250	2-4	3 a 1 año	Profesional	2.5
Menos de 250	2-4	3 a 1 año	Otro (capacitación)	2.75
Menos de 250	4-7	1 a 3 meses	Ninguno	1
Menos de 250	4-7	1 a 3 meses	Primaria	1.25
Menos de 250	4-7	1 a 3 meses	Secundaria	1.25
Menos de 250	4-7	1 a 3 meses	Técnico	1.5
Menos de 250	4-7	1 a 3 meses	Profesional	1.5
Menos de 250	4-7	1 a 3 meses	Otro (capacitación)	1.75
Menos de 250	4-7	3 a 1 año	Ninguno	1.75
Menos de 250	4-7	3 a 1 año	Primaria	2
Menos de 250	4-7	3 a 1 año	Secundaria	2
Menos de 250	4-7	3 a 1 año	Técnico	2.25
Menos de 250	4-7	3 a 1 año	Profesional	2.25
Menos de 250	4-7	3 a 1 año	Otro (capacitación)	2.5

Tabla 46. *Tabla de hipótesis seis sobre tipologías de familias*

Tabla de Hipótesis 7				
4 variables ingreso, No. Integrantes, empleo y educación				
250 a 500	2	1 a 3 meses	Ninguno	2.25
250 a 500	2	1 a 3 meses	Primaria	2.5
250 a 500	2	1 a 3 meses	Secundaria	2.5
250 a 500	2	1 a 3 meses	Técnico	2.75
250 a 500	2	1 a 3 meses	Profesional	2.75
250 a 500	2	1 a 3 meses	Otro	3
250 a 500	2	3 a 1 año	Ninguno	3
250 a 500	2	3 a 1 año	Primaria	3.25
250 a 500	2	3 a 1 año	Secundaria	3.25
250 a 500	2	3 a 1 año	Técnico	3.5
250 a 500	2	3 a 1 año	Profesional	3.5
250 a 500	2	3 a 1 año	Otro	3.75
250 a 500	2-4	1 a 3 meses	Ninguno	1.75
250 a 500	2-4	1 a 3 meses	Primaria	2
250 a 500	2-4	1 a 3 meses	Secundaria	2
250 a 500	2-4	1 a 3 meses	Técnico	2.25
250 a 500	2-4	1 a 3 meses	Profesional	2.25
250 a 500	2-4	1 a 3 meses	Otro	2.5
250 a 500	2-4	3 a 1 año	Ninguno	2.5
250 a 500	2-4	3 a 1 año	Primaria	2.75
250 a 500	2-4	3 a 1 año	Secundaria	2.75
250 a 500	2-4	3 a 1 año	Técnico	3
250 a 500	2-4	3 a 1 año	Profesional	3
250 a 500	2-4	3 a 1 año	Otro	3.25
250 a 500	4-7	1 a 3 meses	Ninguno	1.5
250 a 500	4-7	1 a 3 meses	Primaria	1.75
250 a 500	4-7	1 a 3 meses	Secundaria	1.75
250 a 500	4-7	1 a 3 meses	Técnico	2
250 a 500	4-7	1 a 3 meses	Profesional	2
250 a 500	4-7	1 a 3 meses	Otro	2.25
250 a 500	4-7	3 a 1 año	Ninguno	2.25
250 a 500	4-7	3 a 1 año	Primaria	2.5
250 a 500	4-7	3 a 1 año	Secundaria	2.5
250 a 500	4-7	3 a 1 año	Técnico	2.75
250 a 500	4-7	3 a 1 año	Profesional	2.75
250 a 500	4-7	3 a 1 año	Otro	3

Tabla 47. *Tabla de hipótesis siete sobre tipologías de familias*

Tabla de Hipótesis 8

4 variables ingreso, No. Integrantes, empleo y educación				
500 a 700	2	1 a 3 meses	Ninguno	2.5
500 a 700	2	1 a 3 meses	Primaria	2.75
500 a 700	2	1 a 3 meses	Secundaria	2.75
500 a 700	2	1 a 3 meses	Técnico	3
500 a 700	2	1 a 3 meses	Profesional	3
500 a 700	2	1 a 3 meses	Otro (capacitación)	3.25
500 a 700	2	3 a 1 año	Ninguno	3.25
500 a 700	2	3 a 1 año	Primaria	3.5
500 a 700	2	3 a 1 año	Secundaria	3.5
500 a 700	2	3 a 1 año	Técnico	3.75
500 a 700	2	3 a 1 año	Profesional	3.75
500 a 700	2	3 a 1 año	Otro (capacitación)	4
500 a 700	2-4	1 a 3 meses	Ninguno	2
500 a 700	2-4	1 a 3 meses	Primaria	2.25
500 a 700	2-4	1 a 3 meses	Secundaria	2.25
500 a 700	2-4	1 a 3 meses	Técnico	2.5
500 a 700	2-4	1 a 3 meses	Profesional	2.5
500 a 700	2-4	1 a 3 meses	Otro (capacitación)	2.75
500 a 700	2-4	3 a 1 año	Ninguno	2.75
500 a 700	2-4	3 a 1 año	Primaria	3
500 a 700	2-4	3 a 1 año	Secundaria	3
500 a 700	2-4	3 a 1 año	Técnico	3.25
500 a 700	2-4	3 a 1 año	Profesional	3.25
500 a 700	2-4	3 a 1 año	Otro (capacitación)	3.5
500 a 700	4-7	1 a 3 meses	Ninguno	1.75
500 a 700	4-7	1 a 3 meses	Primaria	2
500 a 700	4-7	1 a 3 meses	Secundaria	2
500 a 700	4-7	1 a 3 meses	Técnico	2.25
500 a 700	4-7	1 a 3 meses	Profesional	2.25
500 a 700	4-7	1 a 3 meses	Otro (capacitación)	2.5
500 a 700	4-7	3 a 1 año	Ninguno	2.5
500 a 700	4-7	3 a 1 año	Primaria	2.75
500 a 700	4-7	3 a 1 año	Secundaria	2.75
500 a 700	4-7	3 a 1 año	Técnico	3
500 a 700	4-7	3 a 1 año	Profesional	3
500 a 700	4-7	3 a 1 año	Otro (capacitación)	3.25

Tabla 48. *Tabla de hipótesis ocho sobre tipologías de familias*

VI.2.1.7.2.4 Pruebas.

El primer paso en ésta experimentación es por el sistema de hipótesis anterior, se gráfica el comportamiento de las variables con la función para graficar:

Código	<pre> rango=Join[iA,iB,iC] ingreso=Table[rango]; ListPlot[ingreso, PlotRange -> {{0,5},{0,5}}, AspectRatio -> 1, PlotStyle -> {PointSize[0.5]}; </pre>
	<pre> {{1., 1.75}, {1., 2.}, {1., 2.}, {1., 2.25}, {1., 2.25}, {1., 2.5}, {1., 2.5}, {1., 2.75}, {1., 3.}, {1., 0.3}, {1., 3.25}, {1., 1.25}, {1., 1.5}, {1., 1.5}, {1., 1.75}, {1., 1.75}, {1., 2.}, {1., 2.}, {1., 2.25}, {1., 2.25}, {1., 2.5}, {1., 2.5}, {1., 2.75}, {1., 0.95}, {1., 1.25}, {1., 1.25}, {1., 1.5}, {1., 1.5}, {1., 1.75}, {1., 1.75}, {1., 2.}, {1., 1.25}, {1., 1.25}, {1., 2.5}, {2., 2.25}, {2., 2.5}, {2., 2.5}, {2., 2.75}, {2., 2.75}, {2., 3.}, {2., 3.}, {2., 3.2}, {2., 3.2}, {2., 3.5}, {2., 3.5}, {2., 3.75}, {2., 1.75}, {2., 2.}, {2., 2.}, {2., 2.25}, {2., 2.25}, {2., 2.5}, {2., 2.5}, {2., 2.75}, {2., 2.75}, {2., 3.}, {2., 3.}, {2., 3.25}, {2., 1.5}, {2., 1.75}, {2., 1.75}, {2., 2.}, {2., 2.}, {2., 2.25}, {2., 2.25}, {2., 2.5}, {2., 2.5}, {2., 2.75}, {2., 2.75}, {2., 3.}, {3., 2.5}, {3., 2.75}, {3., 2.75}, {3., 3.}, {3., 3.}, {3., 3.25}, {3., 3.25}, {3., 3.5}, {3., 3.5}, {3., 3.75}, {3., 3.75}, {3., 4.}, {3., 2.}, {3., 2.2}, {3., 2.5}, {3., 2.5}, {3., 2.75}, {3., 2.75}, {3., 3.}, {3., 3.}, {3., 3.25}, {3., 3.25}, {3., 3.5}, {3., 1.75}, {3., 2.}, {3., 2.25}, {3., 2.25}, {3., 2.5}, {3., 2.5}, {3., 2.75}, {3., 2.75}, {3., 3.}, {3., 3.}, {3., 2.25}} </pre>

Tabla 49. Función para graficar

La salida del programa anterior representa al modelo como una sucesión de puntos, como lo muestra la figura 54.

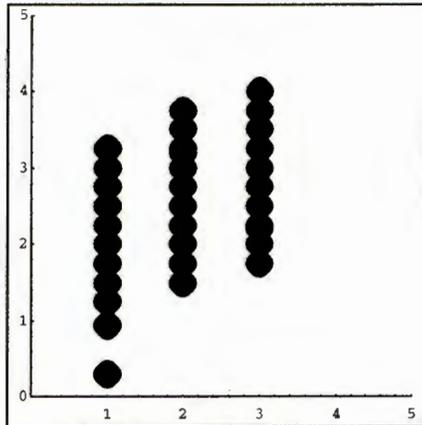


Figura 54. Gráfica de clases de familias

La función *ListPlot3D[rango]* representa una gráfica como una superficie en el espacio de las variables controladas, para observar el comportamiento de las variables de éste modelo en forma tridimensional se muestra la gráfica 55.

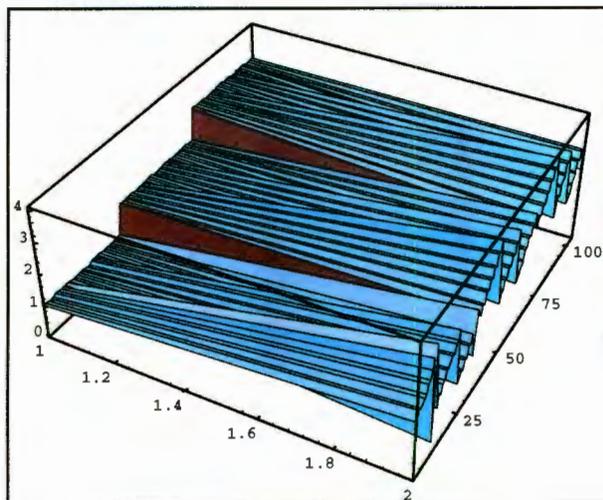


Figura 55. Comportamiento del rango en tercera dimensión, de la asignación de clases de familias.

VI.2.1.7.2.4.1 Determinar el comportamiento de las variables, por sistema de hipótesis.

Del análisis anterior se hace la clasificación ó clases [García, Garzo, 1998] de familias, como lo muestra la tabla siguiente:

Clase de familia	Rango de posibilidades
A	.50 a 1.57
B	1.58 a 2.59
C	2.60 a 3.75

Tabla 50. Tabla de posibilidades

Una vez definidos los datos, la representación del modelo como árbol de decisiones El análisis sobre la eficiencia de éstos algoritmos, se presentan en el capítulo de Enumeración, puede ser especificada, a continuación, en la figura 52.

VI.2.1.7.2.5 Resultados

La figura 56 muestra los resultados, la cual genera un árbol de decisiones, del modelo racional en el combate a la pobreza.

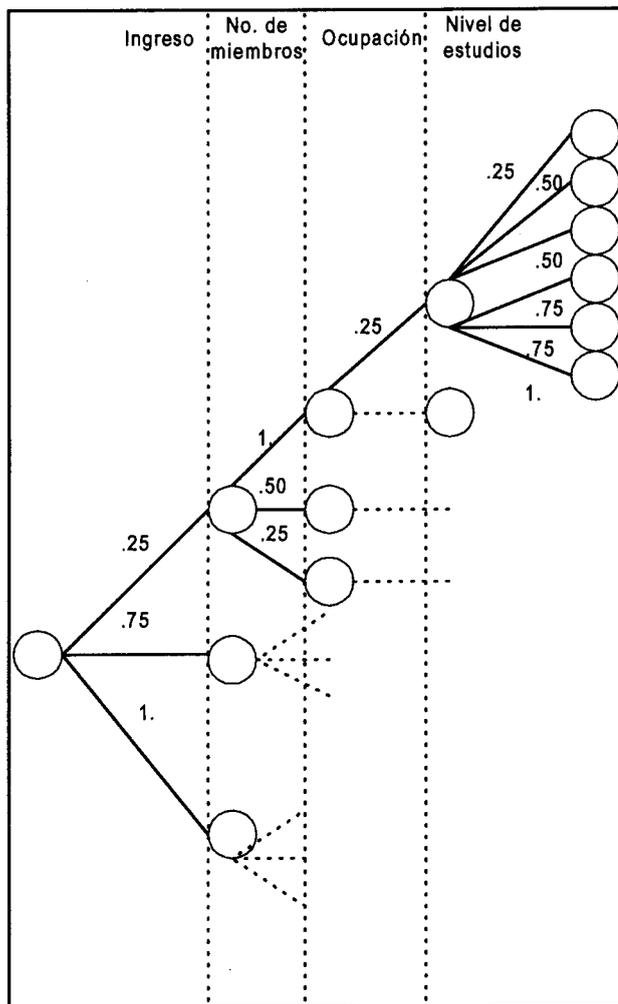


Figura 56. *Árbol de decisiones del modelo racional en el combate a la pobreza.*

Listo el modelo y las variables de puede realizar por ejemplo las siguientes preguntas:

Ejemplo. Caso 1.

Sea una familia, de 3 personas, con un ingreso de entre 250 pesos y 500 pesos, ¿Qué decisiones pueden tomar para salir de la pobreza?

Solución.

La figura 57, representa las decisiones que pueden ser elegidas en base al árbol de decisiones, mostradas en la figura anterior.

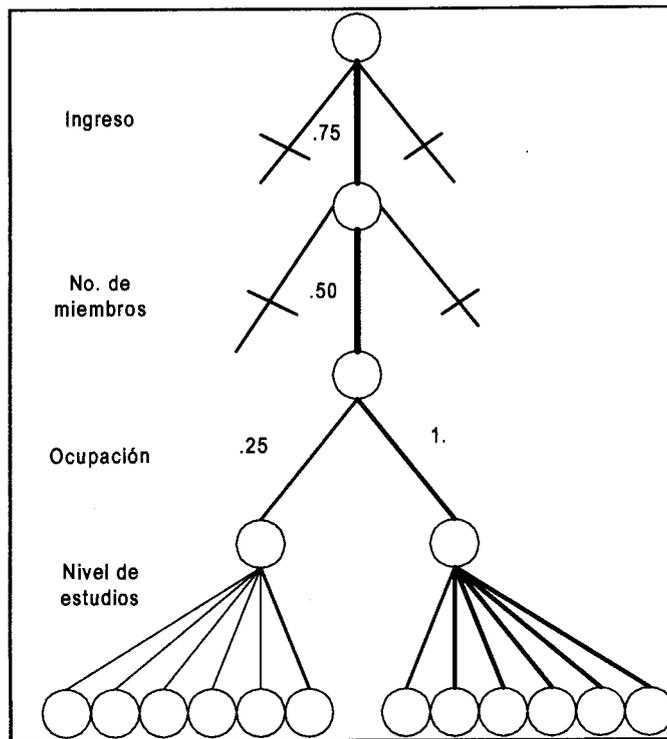


Figura 57. Árbol de decisión para caso 1

Las líneas transversales en las ramas del árbol son podas o ramas que no son evaluadas dadas las características del problema.

Inicialmente, estas personas se ubican en la clase A. Dado que los deseos son una cuestión social, la familia escogerá entre las siguientes opciones para llegar a la clase C, con alguna de las siguientes estrategias:

1. Tratar de mantener su ingreso y el número de hijos, además de conseguir un empleo formal y cursar la primaria/secundaria.
2. Tratar de mantener su ingreso y el número de hijos, además de conseguir un empleo formal y cursar alguna carrera técnica.

3. Tratar de mantener su ingreso y el número de hijos, además de conseguir un empleo formal, y cursar la capacitación.

Dada las anteriores opciones, se considera que una buena política de ayuda a ésta familia, sería que se le apoyara mientras, mejore sus probabilidades de salir de la pobreza, siguiendo la estrategia planteada por su cultura.

ANALISIS DE OPORTUNIDADES.

Para analizar las oportunidades en términos más reales, debemos tomar en consideración los siguientes puntos:

1. Las oportunidades por lo general se traducen en términos monetarios o económicos.
2. La solución a los problemas económicos, consiste en encontrar una ganancia máxima o una pérdida mínima.
3. El árbol de Kuhn es un modelo extensivo de estrategias [Zamora 1978], Prigogine 1996], de información perfecta, ya que el actor, conoce toda la información necesaria para tomar una decisión.

Para ejemplificar el análisis de oportunidades se muestra a continuación un caso.

Caso 1.

Una familia desea salir de la pobreza y cuenta con 700 pesos para invertirlos en algún concepto, la familia puede invertir en la educación de sus hijos ó en arreglar la vivienda.

Por otro lado, se sabe que el gobierno puede apoyarlos con un programa de asistencia social o no.

Las estrategias anteriores, se muestran en el siguiente árbol:

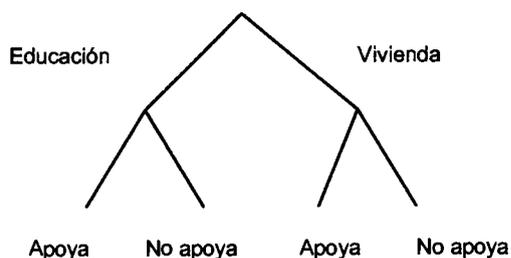


Figura 58. *Árbol de Kuhn que describe las estrategias contra la pobreza*

Las oportunidades referidas en la figura 58, pueden ser representadas en la matriz de la figura 59.

		Estrategias del gobierno	
		Apoya	No apoya
Estrategias de la familia	Educación	2100	700
	Vivienda	1400	1400

Figura 59. *Matriz de oportunidades*

La estrategia de máxima utilidad sería invertir en educación y lograr que el gobierno apoye con un programa social.

En el caso de que conociéramos el deseo del gobierno (.4) para ayudarnos el escenario sería el siguiente:

$$2100 (.4) + 700(.6)=840 + 420 =1260$$

$$1400(.4) + 1400(.6)= 560 +840=1400$$

La mejor oportunidad sería en vivienda.

VI.3.1.8 Conclusiones

El modelo muestra una explicación científica de un problema, dando como resultado un nuevo sistema de pruebas

De la experiencia de trabajar la elección racional, con un modelo matemático, se desprenden las siguientes ventajas y desventajas.

Ventajas

- Jorgen Aasness y Aanond Hylland [Elster 1989] afirman que la incertidumbre de la teoría es mejor que la devastadora ignorancia total, ya que se considera que cuando uno considera un escenario como cierto puedes eliminar otras opciones para tener una mejor solución.
- Con las ventajas técnicas, los resultados de la investigación, son dinámicas; es decir; con pocas modificaciones en el programa, se puede obtener diferentes escenarios y compararlos o cambiar valores, teniendo resultados inmediatos. Por supuesto esto da como consecuencia una memoria histórica y permite la verificación de resultados, con diferentes escenarios.

Desventajas

- Limitaciones técnicas.
- En cuanto a la Teoría de la elección racional, puede fallar en tres niveles: Hallar la mejor acción por creencias y deseos dados, Formar una creencia mejor fundada para cada prueba y Acumular la cantidad de pruebas y deseos para una creencias dada

VII CONCLUSIONES

La presente tesis analiza las estructuras arborescentes mediante su enumeración para conocer el número, tipo y el comportamiento de estas estructuras para incursionar en otras disciplinas que se benefician del enfoque informático.

En el segundo capítulo, se analizó el origen de los árboles para identificar con precisión los problemas que se pueden modelar por medio de grafos, digrafos round robin, haciendo un énfasis mayor en los problemas que pueden ser analizados por árboles; por otra parte se analiza la representación matemática para entender los elementos que componen las relaciones de un problema modelado por medio de árboles. Se concluyó que el modelo de grafos en general puede representar situaciones de la vida real, en particular, los árboles ayudan a analizar problemas organizacionales de tipo jerárquico o los cuales se desea obtener una solución por medio de la optimización de utilidades.

El análisis de paradigmas trata de ubicar si los modelos de pensamiento corresponden a los de programación; en general, el enfoque de la realidad debe corresponder a la lógica del pensamiento del problema, por tanto, la teoría que se aplica en otras ciencias deberá ser congruente con modelos de programación para dar así un resultado óptimo. En esta tesis se propone un enfoque funcional, por que se apega más al objetivo del trabajo, además; porque reduce el tamaño del código y permite la compatibilidad con aplicaciones.

En el tercer capítulo se muestran los árboles con vistas funcionales, es decir, un estudio de los principales componentes de los árboles acompañados de un algoritmo computacional para su mejor comprensión. El objetivo es tener una librería de funciones que permitirá analizar subproblemas de relación del modelo propuesto; por ejemplo servirá para analizar relaciones y hacer pruebas del modelo.

Los árboles con raíces son importantes en las aplicaciones sociales, cuando su análisis y su solución requieren de un cierto orden. El objetivo de este capítulo es hacer una introducción con los árboles n-arios de los principales árboles con raíz y sus aplicaciones más importantes.

En general, se puede concluir que los árboles libres se usan en los estudios del comportamiento general de una comunidad, El árbol de Kunh es un árbol de decisiones el cual tiene ciertas modificaciones que deben tomarse en cuenta en la programación.

La importancia de la enumeración, analizada en el quinto capítulo es porque ayuda a soportar una gama de decisiones que se toman, al minimizar la red, encontrando los óptimos.

La enumeración nos muestra cómo es posible profundizar en el comportamiento de un algoritmo en general, además de fundamentar la eficiencia de la búsqueda y recuperación de información. Sin duda, el mejor algoritmo de búsqueda son los árboles binarios.

Los árboles no isomorfos, en general proporcionan un conjunto de árboles representativos de un modelo, en otras palabras muestran relaciones representativas de una realidad.

Los árboles trivalentes proporcionaron un sistema de combinación única de variables, que permite analizar sistemáticamente las variables que se usan en las ciencias sociales.

Las aplicaciones más fascinantes, son los árboles libres porque optimiza de redes o de árboles, ya que soportan las decisiones tomadas en un sistema.

El sexto capítulo muestra que la complejidad del mundo y sus problemas obliga al manejo de modelos también complejos pero a la vez comprensibles como lo muestra el Modelo de Elección Racional en Combate a la Pobreza;

Los árboles más adecuados en el tratamiento de las ciencias sociales, son los árboles de decisiones y los de Kuhn en la toma de decisión de estrategias, ya que ayudan a obtener una solución por medio de la optimización de utilidades.

Los modelos matemáticos son herramientas en la búsqueda de soluciones y la planeación de estrategias; pero debe apoyarse en una teoría que respalde la implementación y la interpretación de los resultados (en este caso se empleó la elección racional, para cumplir con estas expectativas).

Es importante señalar que el modelo puede ser aplicado a otros problemas sociales como por ejemplo la inmigración y que refleja una explicación científica

de un problema, dando como resultado un nuevo sistema de pruebas y un análisis sólido en la toma de decisiones.

En general, la aportación de esta tesis es un diseño de un prototipo, para la aplicación estrategias en el combate a la pobreza por elección racional.

VII. Trabajo futuro

El trabajo futuro que debe de desarrollarse a partir de esta tesis es:

- Programar la búsqueda en árboles generales, es decir; representar un árbol posicional finito, en un árbol binario.
- Por otra parte se recomienda analizar bajo el concepto de enumeración, los estructuras de grafos y modelo de torneos.
- Implementar aplicaciones en las ciencias sociales por el modelo de árboles libres, grafos y modelo de torneos.
- Esta tesis tiene los fundamentos para programar el modelo de decisión de estrategias en el combate a la pobreza por elección racional y por tanto se pretende convertir la aplicación en un software de análisis social. La evolución del trabajo presentado comienza por el análisis de trayectorias, para probar la validez de la teoría de elección racional del árbol de decisiones mostrado en el modelo; de esta manera el árbol de posibilidades se convertirá en un árbol de probabilidades. El árbol de probabilidad coadyudara al objetivo de la elección racional, optimizando la probabilidad de hallar el mejor medio para fines dados. Como se había mencionado para lograr éste objetivo; trata de determinar que hará una persona bajo ciertas circunstancias y verifica si lo que hizo la persona, es algo que la teoría predijo que haría. Aún con las limitaciones mencionadas, el modelo puede ser una herramienta útil en el análisis de decisión, ya que brindar un marco de certidumbre en la elección y por otro lado proporciona un nuevo sistema de pruebas y críticas. Obsérvese que la información que se obtuvieran de las pruebas servirían para hacer el historial del fenómeno social, la importancia de tal historial es que proporciona datos para una simulación de escenarios. Por lo que finalmente el software se convertiría en un software de experimentación de fenómenos sociales.

BIBLIOGRAFÍA

- AHO V. A., HOPCROFT J. E., ULLMAN, J. D. 1988. Estructura de datos y algoritmos, Adisson Wesley.
- AHO, V. A., ULLMAN J. 1995. Foundations of computer science, Computer Science Press.
- CHARTRAND, G. 1977. Graph theory, Dover.
- COLOMER, J. M. 1990. El arte de la manipulación política, Anagrama
- ELSTER, J. 1991. Juicios Salomónicos. Las limitaciones de la racionalidad como principio de decisión Sociales, Gedisa.
- ELSTER, J. Tuercas y Tornillos. 1995. Una introducción a los conceptos básicos de las Ciencias Sociales, Gedisa.
- GARCIA P. F., GARZO P. 1998. Estadística, Mc. Graw Hill
- GRASSMANN W. K. 1996. TREMBLAY J.-P., Matemática discreta y lógica. Prentice Hall.
- GRIMALDI, R. P. 1994. Discrete and combinatorial mathematics. Adisson Wesley.
- HOPCROFT, J. E., ULLMAN, J. D. 1997. Introducción a la teoría de autómatas, lenguajes y computación. CECSA.
- HOWAR A, 1988. Introducción al álgebra Lineal.
- JOHNSONBAUGH R. 1997. Discrete mathematics. Prentice Hall.
- KELLEY, D. 1995. Teoría de autómatas y lenguajes formales Prentice Hall.
- KLIEIMAN, A. KLEIMAN E. K. 1985. Matrices, aplicaciones en economía y administración. Limusa.
- KNUTH, D. E. 1973. "The art of computer programming", vol 1, Fundamental algorithms vol 2, Seminumerical algorithms, Adisson Wesley.
- KOLMAN B., BUSBY R. C., ROSS S. 1995. Estructuras de matemáticas discretas para la computación. Prentice Hall.

- MADRON T. W. 1992. Redes de área Local. Grupo Noriega Editores.
- MCLENAN, B. J. 1990. Functional programming, practice and theory, Addison Wesley Publishing Company.
- MONROY, O. C. 1997. Teoría del caos. Computec.
- PFEFFER, J. 1982. Organizaciones y teoría de las organizaciones. Fondo de Cultura Económica.
- PLAFT, J. L. 1977. Computer data structures. Prentice Hall.
- PRAWDA, J. 1989. Métodos y Modelos de Investigación de Operaciones.
- PRIGIONE, I. 1996. El tiempo y el devenir. Coloquio de Cerisy, Editorial Geodisa,
- RANGEL M. J. GONZÁLEZ G. A 1999. Algorithms on trees.
Part I. Basic Algorithms.
Facultad de Informática, publicado por Wolfram Reasearch
<http://www.mathsource.com/content/Applications/ComputerScience/>
- RANGEL M. J. GONZÁLEZ G. A. 1998. Apuntes de matemáticas discretas, un punto de vista algorítmico, (en cinco partes). Universidad Autónoma de Querétaro, Facultad de Informática.
- REINGOLD E. M., NIEVERGELT, DEO N. 1977. Combinatorial algorithms: Theory and practice, Prentice Hall.
- ROBERTS, F. S. 1984. Applied combinatorics, Prentice Hall.
- ROBERTS, F. S. 1986. Graph Theory and the Social Sciences.
- ROSEN K MICHAELS G. J., GROSS J., GROSSMAN J., SHIER D. 1999 Handbook. of discrete and combinatorial mathematics . CRC Press.
- SÁNCHEZ, A. A. 1993. Postulados de los paradigmas de programación. Revista Soluciones avanzadas, año 1, Número 5 septiembre-octubre.
- SHUBIK, M. 1996. Teoría de juegos en las ciencias sociales, Fondo de Cultura Económica.
- WIRTH N. 1987. Algoritmos y estructuras de datos, Prentice Hall.

APÉNDICE A

REVISIÓN HISTÓRICA DE LA TEORÍA DE ÁRBOLES

Algunos de los conceptos matemáticos más importantes en la Ciencia de la Computación son los relacionados a combinatoria, grafos y recursión.

Los grafos son una herramienta fundamental para resolver problemas de combinatoria; en sentido inverso algunas técnicas de combinatoria son utilizadas para resolver problemas de la teoría de grafos [Roberts 1984].

La combinatoria surge en el siglo XVII, gracias a los estudios de los franceses Blaise Pascal (1623-1662) y de Pierre De Fermat (1601-1665) sobre la teoría de juegos de azar. Sus trabajos, formaron los fundamentos de la teoría de la probabilidad, los cuales contenían los principios para determinar el número de combinaciones de elementos de un conjunto finito, así se estableció la tradicional conexión entre combinatoria y probabilidad [Zamora 1978].

En los principios del siglo XVIII, en el año de 1736, el matemático suizo Leonard Euler (1707-1783), se convirtió en el Padre de la teoría de grafos, por el planteamiento y solución del problema de "Los siete Puentes de Königsberg" [Rangel 1998], en la solución usó por primera vez conceptos y métodos de teoría de grafos. También desarrolló la escuela de matemática combinatoria, al publicar sus artículos sobre la partición y descomposición de enteros positivos en sumandos, estableciendo las bases de uno de los métodos fundamentales para el cálculo de configuraciones combinatorias, que es el método de las funciones generadoras.

A finales de siglo XIX, el físico Alemán Gustav Robert Kirchhoff (1824-1887), estudiando redes eléctricas, examina un tipo de grafo muy especial, al cual por primera vez en el año de 1847, se le denomina "árbol".

En Inglaterra Arthur Cayley (1821-1895). Contribuyó a la creación de la teoría de grupos entre otras; introdujo al ámbito de estudios matemáticos la noción de matriz que, posteriormente hallaría una profusa aplicación en estudios tecnológicos como las de las retículas de mallas de los circuitos eléctricos y principios fundamentales de la física, como los postulados de la mecánica cuántica.

En el año de 1857, motivado por el problema de calcular el número de isómeros de hidrocarburos saturados, estudia estos tipos de grafos llamados árboles, haciendo una importante contribución a la teoría de enumeración de grafos, descubriendo la fórmula para la enumeración de árboles libres [Roberts 1984].

Por este tiempo el matemático británico George Boole [Grassmann 1996] (1815-1864) usó métodos de combinatoria en conexión con el desarrollo de la lógica simbólica y junto con las ideas y métodos del matemático francés Jules Henri Poincaré (1854-1912) desarrolló problemas de topología y en 1908, éste último, al estudiar comportamientos caóticos, se convierte en el padre de la Teoría del Caos [Monroy 1997]

A mediados del siglo XIX, se formuló el acertijo teórico más desafiante en la historia de la combinatoria, "El problema de los cuatro colores"; este nos señala que cuatro colores son suficientes para colorear las regiones de un mapa de tal manera que regiones con frontera tengan asignado distinto color. Las soluciones de este acertijo matemático, pasaron a ser fuente de importantes problemas y resultados en teoría de gráficas [Grimaldi 1994].

En el año de 1859, William Rowan Hamilton (1805-1865), matemático irlandés, utilizó ciclos hamiltonianos para componer un rompecabezas que utilizaba las aristas de un dodecaedro regular [Johnsonbaugh 1996]. George Polya (1888-1985) es el pionero de la enumeración de árboles y otras estructuras combinatorias [Grimaldi 1994].

En el siglo XX, se generan estudios concretos de árboles, para diferentes disciplinas; tales como la física, Ciencias computacionales, química, economía, biología y sociología [Kleiman, Kleiman 1985].

APÉNDICE B

LOGARITMOS

Los logaritmos son exponentes en el siguiente sentido: la base- b del logaritmo de x , denotado como $\log_b x$, es el power (poder multiplicación) por la cual la base b puede ser raizada igual a x .

En símbolos diremos $y = \log x$, entonces $b^y = x$

En ese sentido $b^{\log_b x} = x$, es la definición de logaritmo

Multiplicando por una base c $\log_c (b^{\log_b x}) = \log_c X$

Aplicando la igualdad $\log_b (x^n) = n \log_b x$ y substituyendo para quitar el exponente de b dentro de los paréntesis tenemos

$$\log_b x * \log_c b = \log_c x$$

diviendiendo ambos lados por \log_b

$$\log_b x = \frac{\log_c x}{\log_c b} \quad \text{ésta formula sirve para cambiar bases de algoritmos}$$

Para convertir un logaritmo base 2 se substituye $b=2$ y $c=e$

$$\log_2 x = \frac{\log_e x}{\log_e 2}$$

resulta la fórmula para cambiar logaritmos base 2 a logaritmos naturales

$$\log x = \frac{\ln x}{\ln 2}$$

APÉNDICE C

PROGRAMA PROCEDURAL DE UN ÁRBOL

El objetivo del programa es crear un árbol binario y leerlo en inorden, posorden y en orden.

```
/*
Programa...:Arbol binario.c
*/

/*Declaración de
constantes, estructuras y
variables generales*/

#define NULL 0
#define MIN 0
#define MAX 27

/*Definición de la
estructura de árbol*/

typedef struct nodo
{
    char inf;
    struct nodo *izq;
    struct nodo *der;
}celda;

celda *raiz, *arbol, *hoja;
char letras[MAX]= "QWERTYUIOPASDFGHJKLYZXCVBNM";
int l;

/*Declaración de funciones
y procedimientos*/

void inicializa[]
/*Inicializa el apuntador principal*/
{
    arbol=[celda *] malloc[sizeof[celda]];
    arbol->izq=NULL;
    arbol->der=NULL;
}

/**/
void generador[]
{
    hoja=[celda *]malloc[sizeof[celda]];
    hoja->inf=letras[l];
    hoja->izq=NULL,
    hoja->der=NULL;
}
```

```

/**/
void construye[]
{
if [l==0]
{
raiz =hoja;
arbol=hoja;
printf["%d%c\n",l,arbol->inf];
}
else
{
printf["compara: %c%c\n",arbol->inf,hoja->inf];
if [arbol->inf<hoja->inf]
{
if [arbol->der==NULL]
{
arbol->der=hoja;
arbol=arbol->der;
printf["der %d%c\n",l,arbol->inf];
}
else
{
arbol=arbol->der;
construye[];
}
}
else
{
if [arbol->inf>hoja->inf]
{
if [arbol->izq==NULL]
{
arbol->izq=hoja;
arbol=arbol->izq;
printf["izq%d%c\n",l, arbol->inf];
}
else
{
arbol=arbol->izq;
construye[];
}
}
}
}
}
}
}
}
}

```

```

/**/
void imprime[]
{
celda *izq, *der;
printf["\n"];
printf["raiz %c", raiz->inf];
izq=raiz->izq;
printf["izq %c",izq->inf];
der=raiz->der;
printf["der %c", der->inf];
}

```

```

}
/**/
void preorden [celda *ficus]
{
if [ficus!=NULL]
{
printf["%c", ficus->inf];
if [ficus->izq !=NULL] preorden[ficus->izq];
if [ficus->der !=NULL] preorden [ficus->der];
}
}
/**/

void enorden[celda *ficus]
{
if [ficus!=NULL]
{
enorden[ficus->izq];
printf["%c", ficus->inf];
enorden[ficus->der];
}
}
/**/

void postorden [celda *ficus]
{
if [ficus!=NULL]
{
if [ficus->izq !=NULL] postorden[ficus->izq];
if [ficus->der !=NULL] postorden[ficus->der];
printf["%c", ficus->inf];
}
}
/**/

main[] /*Bloque principal*/
{
clrscr[];
printf["ARBOL BINARIO. \n"];
inicializa[];
for[l=MIN; l<=MAX-1; l++]
{

```

```
generador[];  
construye[];  
arbol=raiz;  
}  
imprime[];  
printf["\n Preorden :"];  
preorden[raiz];  
printf["\n Enorden :"];  
enorden[raiz];  
printf["\n Posorden :"];  
postorden[raiz];  
}
```

APÉNDICE D

SÍNTESIS DEL MANUAL DEL PROGRAMA PROGRESA

MUESTRA DE COMBINACIÓN DE VARIABLES

El Consejo Nacional de Población (CONAPO), con el auxilio de otras instituciones tales como: IMSS-SOLIDARIDAD, CONAFE, SECTOR SALUD y SECTOR EDUCATIVO, busca índices de marginación a nivel localidad y los divide en regiones, mediante una ficha de información llamada: CECASEH, (Cédula de **Características Socio-Económicas de los Hogares**).

En ella se concentra información sobre los siguientes puntos:

- a) Características del jefe del hogar o de la persona responsable del hogar.
- b) Tamaño y composición de la familia sexo/edad/analfabetismo/Nivel educativo (de cada miembro)
- c) Actividad económica de cada miembro, origen de los ingresos y su presencia en el hogar.
- d) Existencia de discapacitados en la familia
- e) Características de la vivienda y servicios básicos (agua, drenaje o electricidad).
- f) Posesión de bienes, tierras ó animales.

Además, también registra si existen en la localidad o cerca de ella, Servicios de educación, de salud y/o de Abasto.

Los datos anteriores, se traducen en porcentajes estandarizados a nivel nacional, los cuales muestran el nivel de pobreza. De ésta manera, comparando los datos de las comunidades, se pueden detectar cuales viven en condiciones de extrema pobreza.

Básicamente la evaluación se basa en los siguientes rangos de pobreza:

- Tener más de 4 niños de entre 0 a 11 años,
- Ingreso per cápita no superior a los 200 (la escala de medición en éste estudio llega a 700 pesos).
- población de 15 años o más sin primaria terminada.

Cuando se localiza una familia en éstas condiciones, se toma el nombre de la madre o de la persona encargada del cuidado de la familia, y se les otorga una beca mensual.

La dependencia de Gasto Social y Obra pública de gobierno del estado, controlan la fase terminal del programa; registrando mensualmente el número de beneficiarios, fase de incorporación (1-9) y nombre del titular del número de familia.

APÉNDICE E

Para demostrar la importancia de una metodología para analizar las variables de un problema, se muestra a continuación, una función la cual lista las combinaciones de las variables.

Función 6

que muestra parte, ya que ocupa originalmente 10 hoja como éstas para mostrar las combinaciones posibles, en una lista todas las combinaciones de las variables

Outer[List,{iA,iB,iC},{f1,f2,f3},{e1,e2},{n1,n2,n3,n4,n5,n6}]

```

{((((({0.25, f1, e1, n1}, {0.25, f1, e1, n2}, {0.25, f1, e1, n3}, {0.25, f1, e1, n4}, {0.25, f1, e1, n5}, {0.25, f1, e1, n6}),
  {{{0.25, f1, e2, n1}, {0.25, f1, e2, n2}, {0.25, f1, e2, n3}, {0.25, f1, e2, n4}, {0.25, f1, e2, n5}, {0.25, f1, e2, n6}}}},
  {{{0.25, f2, e1, n1}, {0.25, f2, e1, n2}, {0.25, f2, e1, n3}, {0.25, f2, e1, n4}, {0.25, f2, e1, n5}, {0.25, f2, e1, n6}}}},
  {{{0.25, f2, e2, n1}, {0.25, f2, e2, n2}, {0.25, f2, e2, n3}, {0.25, f2, e2, n4}, {0.25, f2, e2, n5}, {0.25, f2, e2, n6}}}},
  {{{0.25, f3, e1, n1}, {0.25, f3, e1, n2}, {0.25, f3, e1, n3}, {0.25, f3, e1, n4}, {0.25, f3, e1, n5}, {0.25, f3, e1, n6}}}},
  {{{0.25, f3, e2, n1}, {0.25, f3, e2, n2}, {0.25, f3, e2, n3}, {0.25, f3, e2, n4}, {0.25, f3, e2, n5}, {0.25, f3, e2, n6}}}}}),
  {{{({1., f1, e1, n1}, {1., f1, e1, n2}, {1., f1, e1, n3}, {1., f1, e1, n4}, {1., f1, e1, n5}, {1., f1, e1, n6}),
  {{{1., f1, e2, n1}, {1., f1, e2, n2}, {1., f1, e2, n3}, {1., f1, e2, n4}, {1., f1, e2, n5}, {1., f1, e2, n6}}}},
  {{{1., f2, e1, n1}, {1., f2, e1, n2}, {1., f2, e1, n3}, {1., f2, e1, n4}, {1., f2, e1, n5}, {1., f2, e1, n6}}}},
  {{{1., f2, e2, n1}, {1., f2, e2, n2}, {1., f2, e2, n3}, {1., f2, e2, n4}, {1., f2, e2, n5}, {1., f2, e2, n6}}}},
  {{{1., f3, e1, n1}, {1., f3, e1, n2}, {1., f3, e1, n3}, {1., f3, e1, n4}, {1., f3, e1, n5}, {1., f3, e1, n6}}}},
  {{{1., f3, e2, n1}, {1., f3, e2, n2}, {1., f3, e2, n3}, {1., f3, e2, n4}, {1., f3, e2, n5}, {1., f3, e2, n6}}}}}),
  {{{({0.25, f1, e1, n1}, {0.25, f1, e1, n2}, {0.25, f1, e1, n3}, {0.25, f1, e1, n4}, {0.25, f1, e1, n5}, {0.25, f1, e1, n6}),
  {{{0.25, f1, e2, n1}, {0.25, f1, e2, n2}, {0.25, f1, e2, n3}, {0.25, f1, e2, n4}, {0.25, f1, e2, n5}, {0.25, f1, e2, n6}}}},
  {{{0.25, f2, e1, n1}, {0.25, f2, e1, n2}, {0.25, f2, e1, n3}, {0.25, f2, e1, n4}, {0.25, f2, e1, n5}, {0.25, f2, e1, n6}}}},
  {{{0.25, f2, e2, n1}, {0.25, f2, e2, n2}, {0.25, f2, e2, n3}, {0.25, f2, e2, n4}, {0.25, f2, e2, n5}, {0.25, f2, e2, n6}}}},
  {{{0.25, f3, e1, n1}, {0.25, f3, e1, n2}, {0.25, f3, e1, n3}, {0.25, f3, e1, n4}, {0.25, f3, e1, n5}, {0.25, f3, e1, n6}}}},
  {{{0.25, f3, e2, n1}, {0.25, f3, e2, n2}, {0.25, f3, e2, n3}, {0.25, f3, e2, n4}, {0.25, f3, e2, n5}, {0.25, f3, e2, n6}}}}}),
  {{{({0.5, f1, e1, n1}, {0.5, f1, e1, n2}, {0.5, f1, e1, n3}, {0.5, f1, e1, n4}, {0.5, f1, e1, n5}, {0.5, f1, e1, n6}),
  {{{0.5, f1, e2, n1}, {0.5, f1, e2, n2}, {0.5, f1, e2, n3}, {0.5, f1, e2, n4}, {0.5, f1, e2, n5}, {0.5, f1, e2, n6}}}},
  {{{0.5, f2, e1, n1}, {0.5, f2, e1, n2}, {0.5, f2, e1, n3}, {0.5, f2, e1, n4}, {0.5, f2, e1, n5}, {0.5, f2, e1, n6}}}},
  {{{0.5, f2, e2, n1}, {0.5, f2, e2, n2}, {0.5, f2, e2, n3}, {0.5, f2, e2, n4}, {0.5, f2, e2, n5}, {0.5, f2, e2, n6}}}},
  {{{0.5, f3, e1, n1}, {0.5, f3, e1, n2}, {0.5, f3, e1, n3}, {0.5, f3, e1, n4}, {0.5, f3, e1, n5}, {0.5, f3, e1, n6}}}},
  {{{0.5, f3, e2, n1}, {0.5, f3, e2, n2}, {0.5, f3, e2, n3}, {0.5, f3, e2, n4}, {0.5, f3, e2, n5}, {0.5, f3, e2, n6}}}}}),
  {{{({0.25, f1, e1, n1}, {0.25, f1, e1, n2}, {0.25, f1, e1, n3}, {0.25, f1, e1, n4}, {0.25, f1, e1, n5}, {0.25, f1, e1, n6}),
  {{{0.25, f1, e2, n1}, {0.25, f1, e2, n2}, {0.25, f1, e2, n3}, {0.25, f1, e2, n4}, {0.25, f1, e2, n5}, {0.25, f1, e2, n6}}}},
  {{{0.25, f2, e1, n1}, {0.25, f2, e1, n2}, {0.25, f2, e1, n3}, {0.25, f2, e1, n4}, {0.25, f2, e1, n5}, {0.25, f2, e1, n6}}}},
  {{{0.25, f2, e2, n1}, {0.25, f2, e2, n2}, {0.25, f2, e2, n3}, {0.25, f2, e2, n4}, {0.25, f2, e2, n5}, {0.25, f2, e2, n6}}}},
  {{{0.25, f3, e1, n1}, {0.25, f3, e1, n2}, {0.25, f3, e1, n3}, {0.25, f3, e1, n4}, {0.25, f3, e1, n5}, {0.25, f3, e1, n6}}}},
  {{{0.25, f3, e2, n1}, {0.25, f3, e2, n2}, {0.25, f3, e2, n3}, {0.25, f3, e2, n4}, {0.25, f3, e2, n5}, {0.25, f3, e2, n6}}}}}),
  {{{({1.25, f1, e1, n1}, {1.25, f1, e1, n2}, {1.25, f1, e1, n3}, {1.25, f1, e1, n4}, {1.25, f1, e1, n5}, {1.25, f1, e1, n6}),
  {{{1.25, f1, e2, n1}, {1.25, f1, e2, n2}, {1.25, f1, e2, n3}, {1.25, f1, e2, n4}, {1.25, f1, e2, n5}, {1.25, f1, e2, n6}}}},
  {{{1.25, f2, e1, n1}, {1.25, f2, e1, n2}, {1.25, f2, e1, n3}, {1.25, f2, e1, n4}, {1.25, f2, e1, n5}, {1.25, f2, e1, n6}}}},
  {{{1.25, f2, e2, n1}, {1.25, f2, e2, n2}, {1.25, f2, e2, n3}, {1.25, f2, e2, n4}, {1.25, f2, e2, n5}, {1.25, f2, e2, n6}}}},
  {{{1.25, f3, e1, n1}, {1.25, f3, e1, n2}, {1.25, f3, e1, n3}, {1.25, f3, e1, n4}, {1.25, f3, e1, n5}, {1.25, f3, e1, n6}}}},
  {{{1.25, f3, e2, n1}, {1.25, f3, e2, n2}, {1.25, f3, e2, n3}, {1.25, f3, e2, n4}, {1.25, f3, e2, n5}, {1.25, f3, e2, n6}}}}}),

```