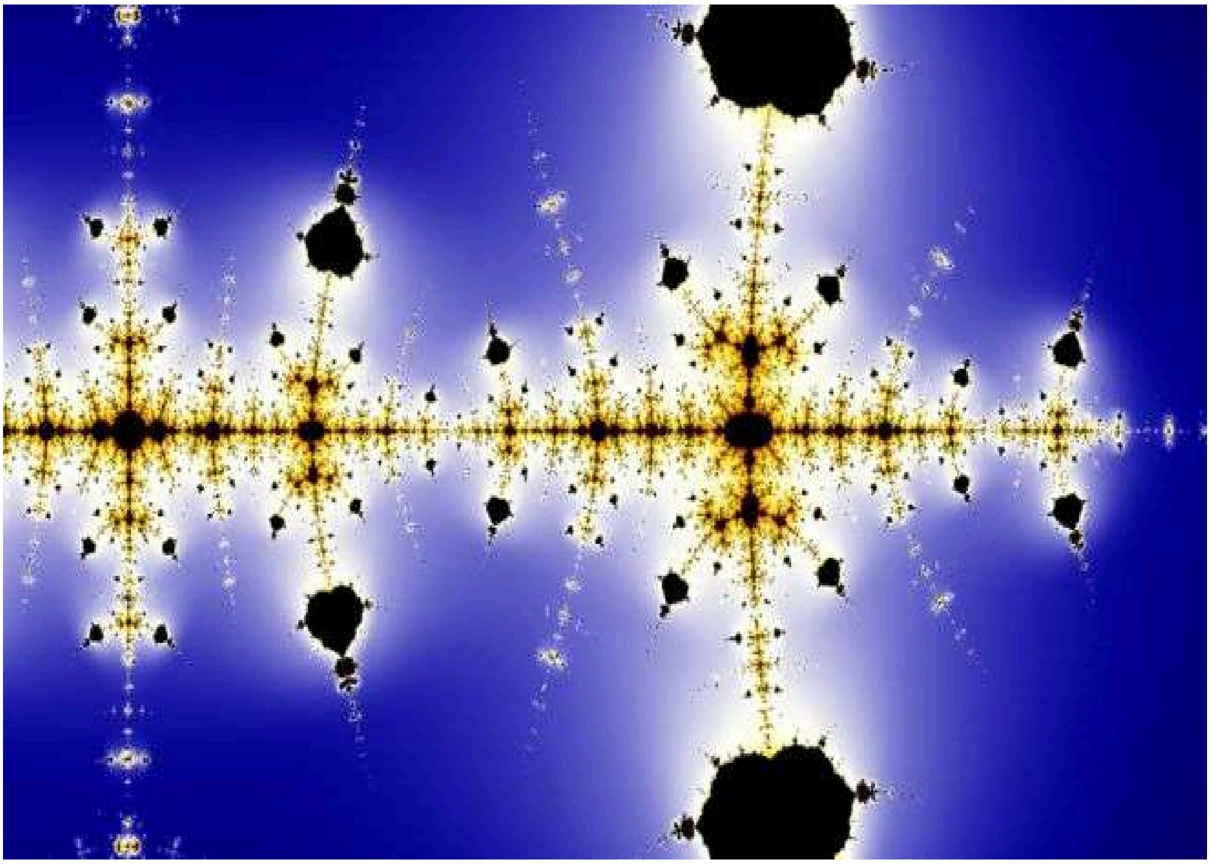


NOTAS DE CLASE

MATEMÁTICAS DISCRETAS

{Con ejemplos de programación en Python, Java, PHP, C++, C#, Javascript, PSeint y pseudocódigo}

/* ****** **Jaime E. Montoya M.** ****** */

NOTAS DE CLASE
MATEMÁTICAS DISCRETAS

{Con ejemplos de programación en Python, Java, PHP, C++, C#, Javascript, PSeint y pseudocódigo}

```
/**
* Versión 4.0
* Año: 2026
* Licencia software: GNU GPL
* Licencia doc: GNU Free Document License (GNU FDL)
*/


public class Author {

    String name = “Jaime E. Montoya M.”;

    String profession = “Ingeniero Informático”;

    String employment = “Docente y desarrollador”;

    String city = “Medellín - Antioquia - Colombia”;

    int year = 2026;

}
```

Tabla de contenido

Introducción 10
Lista de abreviaturas 10
Prefijos y unidades de almacenamiento 12
  Prefijos 12
  Unidades de almacenamiento 12
Capítulo 0. Preliminares: aritmética y álgebra 14
  Los conjuntos numéricos 14
    Símbolos comunes en notación de conjuntos 14
    El conjunto de los números naturales N 14
    El conjunto de los números enteros Z 14
    El conjunto de los números racionales Q 14
    El conjunto de los números irracionales Q’ 15
    El conjunto de los números reales R 15
    El conjunto de los números complejos C 15
  Propiedades fundamentales de los números reales 15
    Propiedad asociativa 16
    Propiedad conmutativa 16
    Módulo de la suma y el producto. Elementos neutros 16
    Inversos 16
    Propiedad distributiva 17
    Resta y división 17
  Algoritmo 17
  Orden y Valor absoluto 17
    Orden 17
    Valor absoluto 18
      Propiedades (teoremas y corolarios) del valor absoluto 18
  Números pares e impares 19
    Propiedades de las operaciones entre números pares e impares 20
  Números primos 20
      Tablero virtual 20
  Sumatoria, Productoria y Factorial 21
    Sumatoria 21
    Productoria 22
    Factorial 23
  Potencias, exponentes y radicales 23
    Potencia 23
    Exponentes 24
      Exponentes enteros 24
      Exponentes fraccionarios 24
    Radicales 24
      Propiedades de los radicales 25

Tablero virtual 25
Operadores básicos en matemáticas y computación 26
Operadores aritméticos 26
Tablero virtual 27
Operación de módulo 28
Tablero virtual 28
Operadores relacionales o de comparación 29
Tablero virtual 29
Operadores lógicos o booleanos 29
Notación algorítmica 30
Preguntas 31
Ejercicios 31
Capítulo 1. Sistemas numéricos 33
Sistemas numéricos 33
Sistemas no posicionales 33
El sistema de numeración egipcio 33
El sistema de numeración azteca 34
El sistema de numeración griego 34
Tablero virtual 34
Sistemas semi-posicionales 34
El sistema de numeración romano 35
Sistemas posicionales o ponderados 35
El sistema de numeración babilónico 35
El sistema de numeración chino clásico 36
El sistema de numeración indoarábigo 36
El sistema de numeración maya 36
El sistema decimal moderno (Notación decimal) 36
Sistemas numéricos computacionales 36
Teorema Fundamental de la Numeración 38
Sistema Binario 38
Sistema Octal 38
Sistema Decimal 39
Sistema Hexadecimal 39
Conversión de bases 39
Conversión de un número en base 10 a una base cualquiera 39
Conversión de base 10 a base 2 40
Conversión de base 10 a base 8 40
Conversión de base 10 a base 16 40
Conversión de un número en una base cualquiera a base 10 40
Conversión de base 2 a base 10 41
Conversión de base 8 a base 10 41
Conversión de base 16 a base 10 41
Tabla de conversión entre decimal, binario, octal y hexadecimal 42
Convertir números en base 2 a base 8 43

Convertir números en base 8 a base 2 44
Convertir números en base 2 a base 16 44
Convertir números en base 16 a base 2 45
Tablero virtual 45
Operaciones con números binarios 46
Suma de números binarios 47
Tablero virtual 47
Resta de números binarios 48
Tablero virtual 49
Producto de números binarios 49
Tablero virtual 50
Preguntas 50
Ejercicios 50
Capítulo 2. Lógica proposicional 52
Técnicas de demostración matemática 52
Terminología usada en la demostración matemática 52
Proposición 52
Axioma 52
Postulado 52
Demostración 52
Teorema 53
Hipótesis 53
Tesis o conclusión 53
Técnicas de demostración matemática 53
Técnica de demostración: Demostración directa 53
Técnica de demostración: Reducción al absurdo (Reductio ad absurdum) 53
Técnica de demostración: Inducción matemática 53
Técnica de demostración: Contraejemplo 54
Paradojas matemáticas 54
Proposiciones 55
Valores de verdad, constantes lógicas o booleanas 55
Operadores, conectivos y símbolos lógicos 56
Operaciones lógicas y tablas de verdad 56
Número de combinaciones 56
Construir la tabla de verdad 57
Negación 57
Conjunción 57
Disyunción 57
Disyunción exclusiva 58
Condicional o implicación 58
Bicondicional o equivalencia 58
Tablero virtual 59
Tautología 60
Contradicción 60

Indeterminación 61
Leyes de la lógica proposicional 61
Leyes de idempotencia para la conjunción y la disyunción 61
Leyes de identidad para la conjunción y la disyunción 61
Leyes conmutativas para la conjunción y la disyunción 61
Leyes asociativas para la conjunción y la disyunción 62
Leyes distributivas para la conjunción y la disyunción 62
Ley de la doble negación 62
Ley del tercero excluido o del inverso de la disyunción 62
Ley de contradicción o del inverso de la conjunción 62
Leyes de De Morgan 62
Ley del Modus Ponendo Ponens 62
Ley del silogismo 62
Caracterización de la implicación (forma alternativa del condicional) 62
Caracterización de la equivalencia (forma alternativa del bicondicional) 62
Simetría del bicondicional 63
Contrarrecíproco 63
Leyes de absorción 63
Preguntas 63
Ejercicios 64
Capítulo 3. Conjuntos 65
Conjunto 65
Representación de conjuntos 65
Conjunto vacío 66
Cardinalidad de un conjunto 67
Símbolos comunes en notación de conjuntos 67
Variables en conjuntos 67
Conjunto Referencial o Universal 68
Subconjuntos 68
Propiedades de los subconjuntos 69
Igualdad de conjuntos 69
Operaciones entre conjuntos 69
Unión 69
Intersección 69
Diferencia - 69
Diferencia simétrica 70
Complemento de un conjunto 70
Conjunto potencia o conjunto de partes P 70
Producto cartesiano 71
Intervalos 71
Intervalo abierto 72
Intervalo cerrado 72
Intervalo semiabierto o semicerrado 72
Aplicaciones con intervalos 72

Álgebra de conjuntos 73
Leyes de idempotencia para la unión y la intersección 73
Leyes de identidad para la unión y la intersección 73
Leyes conmutativas para la unión y la intersección 73
Leyes asociativas para la unión y la intersección 74
Leyes distributivas para la unión y la intersección 74
Leyes de absorción para la unión y la intersección 74
Leyes inversas para la unión y la intersección 74
Ley de doble negación 74
Leyes de De Morgan 74
Leyes adicionales 74
Preguntas 75
Ejercicios 75
Capítulo 4. Relaciones y funciones 78
Relaciones 78
Pareja ordenada 78
Relación 78
Dominio y rango de una relación 79
Representación gráfica de las relaciones y grafos dirigidos o digrafos 80
Algunos tipos de relaciones 81
Relación complemento 81
Relación unión 81
Relación intersección 81
Relación inversa 82
Relación compuesta 82
Propiedades de las relaciones 83
Relaciones reflexivas 83
Relaciones simétricas 83
Relaciones antisimétricas 83
Relaciones transitivas 84
Relaciones de equivalencia 84
Funciones 84
Funciones conocidas 86
Función sobreyectiva 91
Función inyectiva o uno a uno 92
Función biyectiva 93
Función inversa 94
Función compuesta 94
Preguntas 94
Ejercicios 95
Capítulo 5. Grafos 96
Definiciones fundamentales 96
Vértice 96
Arista 96

- Grafo 97
- Subgrafo 97
- Grafo dirigido 98
- Camino en un grafo 99
- Longitud de un camino 99
- Camino simple 99
- Ciclo simple 99
- Ciclo hamiltoniano 100
- Grafos no dirigidos 100
- Estructuras de datos en la representación de grafos 100
- Estructura de lista 101
- Estructuras matriciales 101
- Operaciones básicas con grafos en computación 101
- Capítulo 6. Las compuertas lógicas 105
- ¿Qué es una compuerta lógica? 105
- Compuertas lógicas 107
- Compuerta NOT 107
- Compuerta AND 107
- Compuerta OR 108
- Compuerta NAND 109
- Compuerta NOR 109
- Compuerta XOR 110
- Compuerta XNOR 111
- Compuerta IF 111
- Preguntas 112
- Ejercicios 112
- Capítulo 7. Álgebra booleana y mapas de Karnaugh 115
- Funciones booleanas 115
- Identidades del álgebra booleana 116
- Principio de dualidad booleana 117
- Representación de funciones booleanas 117
- Formas canónica y estándar de las expresiones booleanas 118
- Minitérmino (minterms) (mi) 119
- Maxitérmino (maxterms) (Mi) 119
- Suma de productos SOP 119
- Fórmula canónica disyuntiva o de minitérminos: suma de minitérminos (SOP) 120
- Producto de sumas POS 120
- Fórmula canónica conjuntiva o de maxitérminos: producto de maxitérminos (POS) 121
- Mapa de Karnaugh (K-Map) 123
- Minimización de una SOP 126
- Minimización de una POS 129
- Preguntas 131
- Ejercicios 132

Fuentes y referencias adicionales 133
- Textos impresos 133
- Referencias en Internet 133

Apéndice 136
- Apéndice A. Aplicaciones con aritmética y álgebra 136
  - Raíces de la ecuación cuadrática 136
  - Cálculo del factorial 136
  - Determinar números primos 137
  - Encontrar números perfectos 137
  - Determinar la paridad de un número 137
- Apéndice B. Aplicaciones con sistemas numéricos 138
  - Tabla de caracteres ASCII 138
  - Convertir un número en base 10 a una base cualquiera 140
  - Convertir un número en una base cualquiera a base 10 140
- Apéndice C. Aplicaciones con conjuntos 140
  - Operaciones entre conjuntos 140
- Apéndice D. Aplicaciones con compuertas lógicas 142
- Apéndice E. Aplicaciones con mapas de Karnaugh 142

Introducción

Este documento es un complemento a las clases presenciales y virtuales, y está basado en la bibliografía del curso, así como de otras fuentes adicionales que se indican a lo largo del texto, además de la experiencia del autor en su función docente en las áreas de ciencias básicas. No se pretende reemplazar los textos guías con este manual, sino servir de ayuda didáctica y apoyo académico a los estudiantes.

La guía incluye, además de los conceptos teóricos, ejemplos, gráficas, desarrollos en clase, y al final de cada capítulo, unas preguntas y ejercicios que permitan reforzar los conceptos y promover la práctica y el estudio de los conceptos vistos.

Al final de este manual, se indican fuentes y referencias adicionales que el estudiante puede consultar. Las notas al pie de página contienen enlaces a lecturas complementarias.

El apéndice de este texto presenta distintas aplicaciones de las matemáticas discretas en programación; los desarrollos son presentados en distintos lenguajes de programación tales como C/C++, PHP, Python, Java, Javascript y C#, entre otros, así como en pseudocódigo y PSeInt.

Lista de abreviaturas

ASCII: American Standard Code for Information Interchange

BD: Base de Datos

CF: Contador de Frecuencias

DyV: Divide y Vencerás

ED: Estructura(s) de Datos

IDE: Integrated Development Environment (Entorno de Desarrollo Integrado)

LSL: Lista Simplemente Ligada

LSLC: Lista Simplemente Ligada Circular

LDL: Lista Doblemente Ligada

LDLC: Lista Doblemente Ligada Circular

MCD: Máximo Común Divisor

Matemáticas discretas

NS: Nassi-Schneiderman (diagrama)

PHP: PHP Hypertext Preprocessor

POO: Programación Orientada a Objetos

SI: Sistema de Información

SO: Sistema Operativo

VSC: Visual Studio Code

Prefijos y unidades de almacenamiento

Prefijos

Prefijo	Nombre/Significado	Equivalencia
f	Femto	10^{-15}
p	Pico	10^{-12}
n	Nano	10^{-9}
µ	Micro	10^{-6}
m	Mili	10^{-3}
c	Centi	10^{-2}
d	Deci	10^{-1}
-	Uno/unidad	10^{0}
da	Deca	10^{1}
h	Hecto	10^{2}
K	Kilo	10^{3}
M	Mega	10^{6}
G	Giga	10^{9}
T	Tera	10^{12}
P	Peta	10^{15}

Unidades de almacenamiento

Unidad	Símbolo	Equivalencia
Bit	b	0, 1
Byte	B	2^{3}b = 8b
Kilobyte	KB	1024B = 2^{10}B = 8192b
Megabyte	MB	1024KB = 1048576B

Gigabyte	GB	1024MB
Terabyte	TB	1024GB
Peta	PB	1024TB

Capítulo 0. Preliminares: aritmética y álgebra

Los conjuntos numéricos

La idea de conjunto se emplea bastante en matemáticas, por ser un concepto básico, no tiene una definición formal. Más adelante, se hablará un poco más acerca de los conjuntos y las operaciones que podemos realizar con ellos, por ahora veremos los conjuntos de números más conocidos.

Símbolos comunes en notación de conjuntos

Existen varios símbolos muy utilizados particularmente en el tratamiento de conjuntos.

$\in$: $Pertenece.\ Ejemplo:\ a \in A.\ Se\ lee:\ 'a'\ pertenece\ (es\ elemento\ de)\ a\ 'A'$
$\notin$: $No\ pertenece.\ Ejemplo:\ a \notin A.\ Se\ lee:\ 'a'\ no\ es\ elemento\ 'A'$
$\forall_x$: $Cuantificador\ universal.\ Ejemplo: \forall_x\ x > 9.\ Se\ lee:\ Para\ todo\ x\ se\ cumple\ que\ x\ es\ mayor\ que\ 9$
$\exists_x$: $Cuantificador\ existencial.\ Ejemplo:\ \exists_x\ x > 9.\ Se\ lee:\ Existe\ al\ menos\ un\ x\ tal\ que\ x\ es\ mayor\ que\ 9$
$\exists!_x$: $Cuantificador\ existencial$ único. $Se\ lee:\ Existe\ exactamente\ un\ x$
$\subset$: $Contenido\ en,\ o\ es\ subconjunto\ de.\ Ejemplo:\ A \subset B.\ Se\ lee:\ A\ es\ subconjunto\ de\ B$
$\subseteq$: $Es\ subconjunto\ de\ o\ igual\ al\ conjunto.\ Ejemplo:\ A \subseteq B.\ Se\ lee:\ A\ es\ subconjunto\ o\ igual\ a\ B$

El conjunto de los números naturales **N**

Los números naturales surgen de la necesidad del hombre por contar. Es un conjunto infinito y lo simbolizamos así:
$\aleph\ =\ \{1\ , 2, 3, 4, 5,...\}$

El conjunto de los números enteros **Z**

Los números enteros incluyen a los números naturales, esto es a los números positivos, a sus respectivos negativos y al cero. Es un conjunto infinito y lo simbolizamos así:

$Z\ =\ \{..., -\ 5, -\ 4, -\ 3, -\ 2, -\ 1,\ 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ ...\}$

Podemos decir entonces que $N \subseteq Z$

El conjunto de los números racionales **Q**

Este conjunto incluye a los números enteros, las fracciones como $\frac{3}{4}$, $-\frac{9}{5}$, los números decimales conmensurables como *2.33, -5.99* y los números decimales inconmensurables periódicos como $0.333333...$, $6.778877887788...$ Es un conjunto infinito y lo simbolizamos así:

$Q = \{\frac{p}{q} \mid p \in Z \wedge q \in Z;\ q \neq 0\}$

Por tanto $N \subseteq Z \subseteq Q$

El conjunto de los números irracionales **Q'**

Los números que no son racionales, se denominan irracionales, y son decimales inconmensurables que no son periódicos. Ejemplos de número irracionales, son $\pi = 3.141592...$, $\sqrt{2}$, $e = 2.718281...$, $\sqrt[3]{7}$., etc. Es un conjunto infinito y lo simbolizamos así:

$Q' = \{x \in R \mid x \notin Q\}$

El conjunto de los números reales **R**

El conjunto de los números reales está conformado por los números racionales y los irracionales. Ejemplos de número reales son todos los que hemos visto anteriormente. Es un conjunto infinito y lo simbolizamos así:

$\Re = Q \cup Q'$

El conjunto de los números complejos **C**

El conjunto de los números complejos está conformado por números de la forma

$z = a + bi;\ donde\ a,\ b \in R;\ i = \sqrt{-1}\ es\ la\ unidad\ imaginaria$

Ejemplos de número complejos son: $-54, 45 + 3i, 9 - 2i$. Decimos que los reales están estrictamente contenidos en los complejos:

$R \subset C$

<u>Observación</u>
En este curso nos limitaremos al trabajo dentro del conjunto de los números reales **R**.

Propiedades fundamentales de los números reales

El estudio de las propiedades nos permite entender para qué fines sirven, reconocer sus implicaciones y poder derivar o concluir otras cosas de ellas, en otras palabras, cómo trabajar y usarlas en distintas situaciones. Vemos cuáles son:

Sean a, b, c número reales y vamos a aplicar las operaciones de suma y multiplicación.

Observación
La suma $+$ y la multiplicación $\times$ son operaciones fundamentales, mientras que la resta $-$ y la división $\div$ son derivaciones de éstas. Estas operaciones son *binarias*, en el sentido que involucra dos *operandos*.

Propiedad asociativa

La suma y la multiplicación son asociativas, esto es, los operandos se pueden agrupar de cualquier forma.

$a + (b + c) = (a + b) + c = a + b + c$
$a.(b.c) = (a.b).c = a.b.c$

Propiedad conmutativa

La suma y la multiplicación son conmutativas, es decir, el orden de los sumandos o los factores no altera ni la suma ni el producto, respectivamente.

$a + b = b + a$
$a.b = b.a$

Módulo de la suma y el producto. Elementos neutros

El módulo o elemento neutro, es un valor que al ser computado, no altera el resultado. El módulo o elemento neutro en la suma es el cero (0), ya que al sumar cualquier número por éste, obtenemos el mismo número. Similarmente sucede en la multiplicación, donde el uno (1) es el elemento neutro o módulo de esta operación. Los números 0 y 1 también son llamados *elementos identidad* para la suma y la multiplicación, respectivamente.

$a + 0 = 0 + a = a$
$a \times 1 = 1 \times a = a$

Inversos

Todo número real **a** tiene su *inverso aditivo* **-a** (llamado también el negativo de **a**) y satisface:

$a + (-a) = (-a) + a = 0$

Similarmente, todo número **a** distinto de cero tiene un único *inverso multiplicativo* $\mathbf{a^{-1}}$ que satisface:

$a.(a^{-1}) = (a^{-1}).a = 1, \ si\ a \neq 0$

Cabe recordar que $a^{-1} = \frac{1}{a}$

Propiedad distributiva

$a.(b + c) = a.b + a.c$
$(b + c).a = b.a + c.a$

Resta y división

La suma y la multiplicación son operaciones básicas, mientras que la resta y la división derivan de éstas. La resta es la suma de un inverso aditivo y la división es la multiplicación por un recíproco. Tenemos entonces.

$a - b = a + (-b)$
$\frac{a}{b} = a \div b = a.b^{-1} = a.\frac{1}{b}$

Algoritmo

Aunque es un término bastante empleado tanto en matemáticas como en las ciencias computacionales, es común encontrar variantes en la definición. Sin embargo, el consenso general en ciencias, permite definir un ***algoritmo*** como *un conjunto de pasos finitos y ordenados que buscan la solución de un problema*. Este nombre al parecer tuvo influencia en el matemático persa Al-Juarismi, que en latín antiguo se conocía como *Algorithmi*.

En la antigüedad hubo desarrollos de procesos algorítmicos para resolver problemas, entre ellos se encuentra uno de los más famosos conocido como *Algoritmo de Euclides* para hallar el *Máximo Común Divisor (MCD*) de dos enteros.

Orden y Valor absoluto

Orden

En términos *geométricos*, decimos que un número **a** es menor que otro número **b**, si **a** se encuentra a la izquierda de **b** en la recta numérica. Recordemos que todo número real, excepto el 0, es negativo o positivo; entonces, si tenemos estos números, decimos que:

$a < b \ si\ b - a > 0 \ , esto\ es, b - a\ es\ positivo$

Ejemplo 0.1

1. $-3 < -2 \; ya \; que \; -2-(-3) = -2+3 = 1 > 0 \; positivo$
   $Observe \; además, \; que \; -2 \; está \; a \; la \; derecha \; de \; -3 \; en \; la \; recta \; numérica$
2. $5 < 11 \; ya \; que \; 11 - 5 = 6 > 0 \; positivo \; (5 \; está \; a \; la \; izquierda \; de \; 11 \; en \; la \; recta \; numérica)$

Valor absoluto

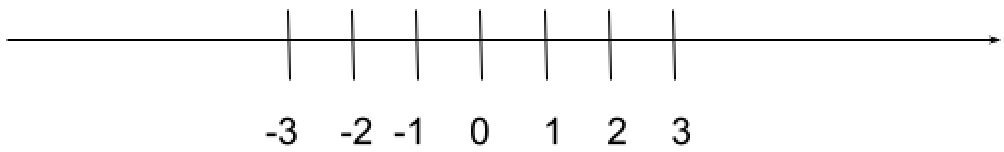
En términos *geométricos*, se define el **valor absoluto** de un número como la distancia que hay desde el cero (0) hasta dicho número en la *recta numérica*; en términos prácticos, la función valor absoluto tiene como fin convertir un número a positivo. Se simboliza encerrando el número entre dos barras: **| a |** donde ***a*** es cualquier número real.

Estrictamente, la función valor absoluto de un número se define así:

$|a| = a \; si \; a \geq 0 \; ó \; |a| = -a, \; si \; a < 0$

<u>Ejemplo 0.2</u>
Observando la recta numérica, podemos ver que la distancia desde el 0 hasta el 3 y desde el 0 hasta el -3 es la misma, esto es, 3 unidades; aplicando la definición de valor absoluto, obtenemos de igual manera 3:

$|3| = 3$
$|-3| = -(-3) = 3$



Propiedades (teoremas y corolarios) del valor absoluto

Sean $a, b \in R$ Se cumplen las siguientes propiedades para el valor absoluto.

1. $||a|| = |a|$
2. $|-a| = |a|$
3. $|a.b| = |a|.|b|$
4. $\left|\frac{a}{b}\right| = \frac{|a|}{|b|} \; si \; b \neq 0$
5. $|a|^2 = a^2$
6. $|x| < a \; si \; y \; solo \; sí \; -a < x < a$
7. $|x| \leq a \; si \; y \; solo \; sí \; -a \leq x \leq a$
8. $|x| > a \; si \; y \; solo \; sí \; x > a \; o \; x < -a$
9. $|x| \geq a \; si \; y \; solo \; sí \; x \geq a \; o \; x \leq -a$
10. $|a+b| \leq |a| + |b| \; (desigualdad \; triangular)$

Ejemplo 0.3
Encuentre el valor absoluto de las siguientes cantidades

1. $|5| = 5$
2. $|-9| = -(-9) = 9$
3. $|-254| = -(-254) = 254$
4. $|0| = 0$
5. $|-1| = 1$

Números pares e impares

Un número ***par*** es un *número entero,* tal que puede escribirse como:

$2k,\ con\ k \in Z$

Decimos que los números pares son exactamente divisibles por *2* y también son múltiplos de *2*.

El conjunto de los números pares es infinito:

$P = \{..., -6, -4, -2, 0, 2, 4, 6, ...\}$

Ejemplo 0.4

1. $0 = 2 \times 0,\ 0 \in Z$
2. $2 = 2 \times 1,\ 1 \in Z$
3. $4 = 2 \times 2,\ 2 \in Z$
4. $6 = 2 \times 3,\ 3 \in Z$
5. $-8 = 2 \times (-4),\ -4 \in Z$

Un número ***impar*** es un *número entero,* tal que puede escribirse como:

$2k + 1,\ con\ k \in Z$

El conjunto de los números impares también es infinito:

$I = \{..., -7, -5, -3, -1, 1, 3, 5, 7, ...\}$

Ejemplo 0.5

1. $1 = 2 \times 0 + 1,\ 0 \in Z$
2. $3 = 2 \times 1 + 1,\ 1 \in Z$
3. $5 = 2 \times 2 + 1,\ 2 \in Z$
4. $7 = 2 \times 3,\ 3 \in Z$
5. $-9 = 2 \times (-5) + 1,\ -5 \in Z$

Propiedades de las operaciones entre números pares e impares

Sean p_1, p_2 dos números pares e i_1, i_2 dos números impares. Se cumplen las siguientes operaciones entre ellos:

1. $p_1 + p_2 = 2n$
2. $p_1 . p_2 = 2n$
3. $p_1 + i_1 = 2n + 1$
4. $p_1 . i_1 = 2n$
5. $i_1 + i_2 = 2n$
6. $i_1 \times i_2 = 2n + 1$

<u>Demostración</u>

1. $p_1 + p_2 = 2a + 2b = 2(a + b) = 2c = 2n$
2. $p_1 . p_2 = 2a \times 2b = 2(a.2b) = 2c = 2n$
3. $p_1 + i_1 = 2a + 2b + 1 = 2(a + b) + 1 = 2c + 1 = 2n + 1$
4. $p_1 . i_1 = 2a(2b + 1) = 2a \times 2b + 2a = 2(a.2b) + 2a = 2(c + a) = 2n$
5. $i_1 + i_2 = 2a + 1 + 2b + 1 = 2(a + b + 1) = 2c = 2n$
6. $i_1 \times i_2 = (2a + 1)(2b + 1) = 2a \times 2b + 2a + 2b + 1 = 2(a.2b) + 2a + 2b + 1$
   $= 2(2ab + a + b) + 1 = 2c + 1 = 2n + 1$

<u>Nota: Paridad del cero</u>
El cero (0) es un número par que cumple con las propiedades comentadas arriba.

Números primos

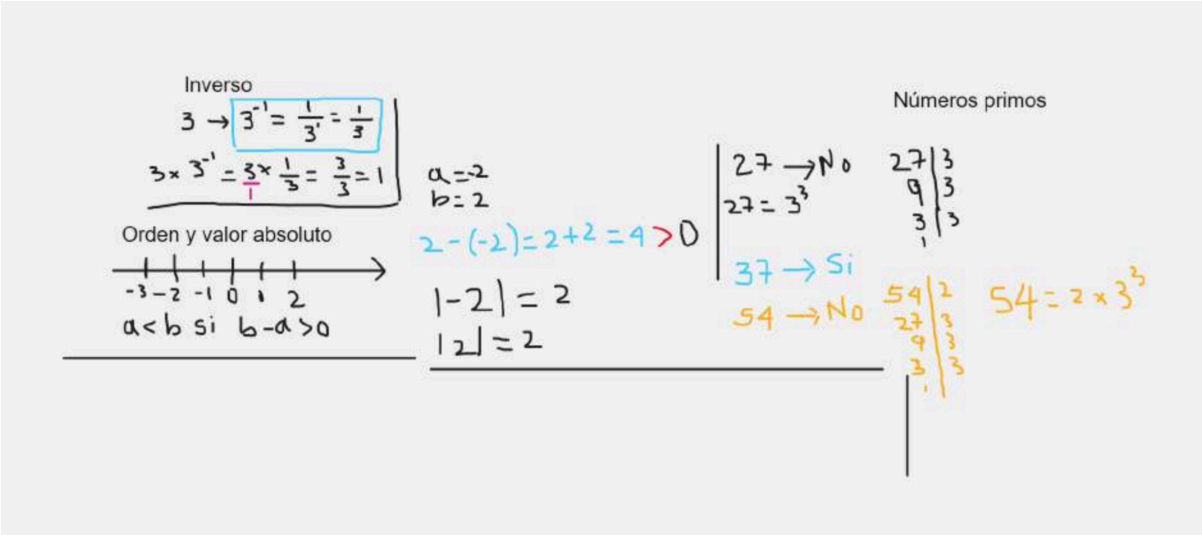
Un número primo es un número entero positivo que tiene solo dos divisores exactos distintos: el mismo número y la unidad.

<u>Ejemplo 0.6</u>
Los siguientes son algunos números primos

2, 3, 5, 7, 11, 13, 17, 23, 29, 31,...

Tablero virtual

Ilustración en clase virtual del valor absoluto y los números primos



Sumatoria, Productoria y Factorial

Sumatoria

Es una notación matemática utilizada para representar la suma de varios términos, o incluso infinitos, como es común encontrar en el cálculo, lo que simplifica la escritura de sumas grandes.
La sumatoria se representa con la letra griega Σ (*sigma* mayúscula).

Nota
La sumatoria es una suma, por tanto podemos utilizar ambos términos sin problema en este contexto.

Notación

$\sum_{i=m}^{n} a_i = a_m + a_{m+1} + a_{m+2} + ... + a_n$

Esto se lee: Sumatoria de *i*, desde *i* igual a *m* hasta *n*, de *a* sub *i*

Se debe cumplir, además, que:
$m \leq n$

Si $m = n$, entonces $\sum_{i=m}^{m} a_i = a_m$

Por definición, si $m > n$, entonces $\sum_{i=m}^{n} a_i = 0$

El número de términos a sumar es, por tanto: $n - m + 1$

<u>Ejemplo 0.7</u>

1. $\sum_{i=1}^{5} i = 1 + 2 + 3 + 4 + 5 = 15.\ Términos\ a\ sumar:\ 5 - 1 + 1 = 5$
2. $\sum_{i=3}^{8} i = 3 + 4 + 5 + 6 + 7 + 8 = 33.\ Términos\ a\ sumar:\ 8 - 3 + 1 = 6$
3. $\sum_{i=1}^{3} i^2 = 1^2 + 2^2 + 3^2 = 1 + 4 + 9 = 14$
4. $\sum_{i=1}^{\infty} i = 1 + 2 + 3 + 4 + 5 + ... = \infty.\ Términos\ a\ sumar:\ infinitos$

Productoria

Es una notación matemática utilizada para representar el producto de varios términos, o incluso infinitos, simplificando la escritura de grandes multiplicaciones.
La sumatoria se representa con la letra griega Π (*pi* mayúscula).

<u>Nota</u>
La productoria es una multiplicación, por tanto podemos utilizar ambos términos sin problema en este contexto.

<u>Notación</u>

$$\prod_{k=m}^{n} a_k = a_m \times a_{m+1} \times a_{m+2} \times ... \times a_n$$

Esto se lee: Productoria de *k*, desde *k* igual a *m* hasta *n*, de *a* sub *k*

Se debe cumplir, además, que:
$m \leq n$

Si $m = n$, entonces $\prod_{k=m}^{n} a_k = a_m$

Por definición, si $m > n$, entonces $\prod_{k=m}^{n} a_k = 1$

El número de términos a multiplicar es, por tanto: $n - m + 1$

<u>Ejemplo 0.8</u>

1. $\prod_{k=1}^{5} k = 1 \times 2 \times 3 \times 4 \times 5 = 120.\ Términos\ a\ multiplicar:\ 5 - 1 + 1 = 5$
2. $\prod_{k=5}^{8} k = 5 \times 6 \times 7 \times 8 = 1680.\ Términos\ a\ multiplicar:\ 8 - 5 + 1 = 4$

3. $\prod_{k=1}^{3} k^2 = 1^2 \times 2^2 \times 3^2 = 1 \times 4 \times 9 = 36$
4. $\prod_{k=1}^{\infty} k = 1 \times 2 \times 3 \times 4 \times 5 \times ... = \infty.\ Términos\ a\ multiplicar: infinitos$

Factorial

El *factorial* de un número entero positivo, se define como el producto de los números desde uno hasta dicho número. Se simboliza acompañando al número del signo de cierre de exclamación (***!***).

$n! = 1 \times 2 \times 3 \times ... \times (n-2) \times (n-1) \times n$
Y se lee: *n* factorial o factorial de *n*.

Dado que conocemos la notación de productoria, podemos escribir:

$n! = \prod_{k=1}^{n} k$

El factorial de *n* ($n \in Z_+$ $-\ enteros\ positivos\ -$) también se define de manera *recursiva*:

$n! = 1,\ si\ n = 0\ o\ n = 1$
$n! = n(n-1)!,\ si\ n > 1$

Esto se conoce como un proceso *recurrente* (*recursivo*), ya que se llama a sí mismo hasta llegar a un estado básico, el cual determina cuándo debe dejar de ser recurrente el proceso para no caer en un ciclo infinito; para el caso del factorial, sus estados básicos son 0 y 1.

Observe que por definición tenemos que:

$0! = 1$

Ejemplo 0.9

1. $5! = 5 \times 4! = 5 \times 4 \times 3! = 5 \times 4 \times 3 \times 2! = 5 \times 4 \times 3 \times 2 \times 1! = 5 \times 4 \times 3 \times 2 \times 1 = 120$
2. $4! = 4 \times 3 \times 2 \times 1 = 24$
3. $7! = 5040$

Potencias, exponentes y radicales

Potencia

Potencia de una expresión es la misma expresión o el resultado de tomarla como factor dos o más veces. La potencia es una operación basada en multiplicaciones sucesivas, donde se toma un número llamado *base* y se multiplica tantas veces indique otro número llamado *exponente*.

$a^{b} = c,\ donde$
a: $base$
b: $exponente$
c: $potencia$

Exponentes

Los exponentes están ligados a unas reglas que nos facilitan el trabajo de cálculo con potencias. Aunque las reglas se aplican en general a todos los números, los exponentes fraccionarios tienen una interpretación que veremos más adelante.

Exponentes enteros

Veamos las principales reglas para trabajar con exponentes enteros.

1. $a^{m}.a^{n} = a^{m+n}$
2. $(a^{m})^{n} = a^{m.n}$
3. $a^{0} = 1\ si\ a \neq 0,\ o\ lo\ que\ es\ equivalente\ a\ decir\text{:}\ a^{-n} = \frac{1}{a^{n}},\ si\ a \neq 0$
4. $\frac{a^{m}}{a^{n}} = a^{m-n},\ si\ a \neq 0$
5. $(a.b)^{n} = a^{n}.b^{n}$
6. $(\frac{a}{b})^{n} = \frac{a^{n}}{b^{n}},\ si\ b \neq 0$

Exponentes fraccionarios

Los exponentes fraccionarios tienen una interpretación propia, tal y como se enuncia en el texto Álgebra Elemental "*Toda cantidad elevada a un exponente fraccionario equivale a una raíz cuyo índice es el denominador del exponente y la cantidad subradical la misma cantidad elevada a la potencia que indica el numerador del exponente*"[1]. Esto es:

$a^{\frac{m}{n}} = \sqrt[n]{a^{m}}$

Radicales

Recordemos las partes de un radical:
$\sqrt[n]{a^{m}} = b\ ,\ donde$
$\sqrt{\ }\ símbolo\ \ o\ símbolo\ radical$
n: $índice\ del\ radical\ (si\ no\ aparece,\ se\ sobreentiende\ que\ es\ 2\ o\ raíz\ cuadrada)$
a^{m}: $cantidad\ subradical$

[1] Álgebra de Elemental. Aurelio Baldor. Interpretación del exponente fraccionario. Página 402

b: $raíz$

Propiedades de los radicales

$\sqrt[n]{a.b.c} = \sqrt[n]{a}.\sqrt[n]{b}.\sqrt[n]{c}$

$\sqrt[n]{\frac{a}{b}} = \frac{\sqrt[n]{a}}{\sqrt[n]{b}}\ si\ b \neq 0$

Del álgebra sabemos que el símbolo $\sqrt{a},\ a \geq 0$, se define como el único *x no negativo* tal que $x^2 = a$

<u>Ejemplo 0.10</u>

$\sqrt{4} = 2; \sqrt{0} = 0; \sqrt{\frac{9}{25}} = \frac{3}{5}$

<u>Nota</u>

$\sqrt{4} \neq -2,\ aun\ cuando\ (-2)^2 = 4,\ ya\ que\ \sqrt{4}\ denota\ \text{ú}nicamente\ la\ ra\text{í}z\ positiva\ de\ 4$

De la definición de $\sqrt{a}$ se deduce que:

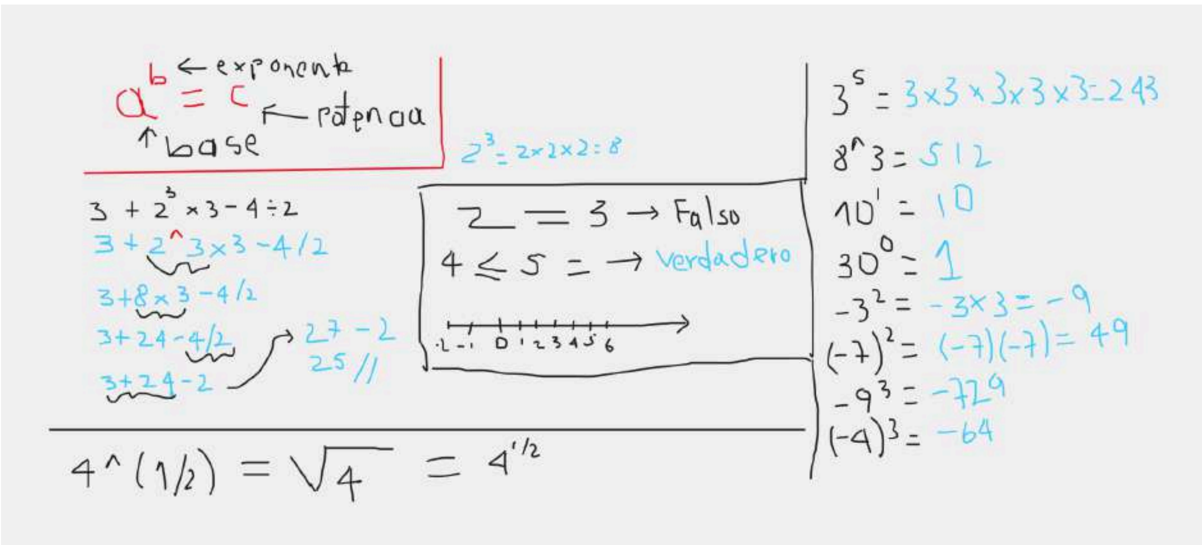
$\sqrt{x^2} = |x|$

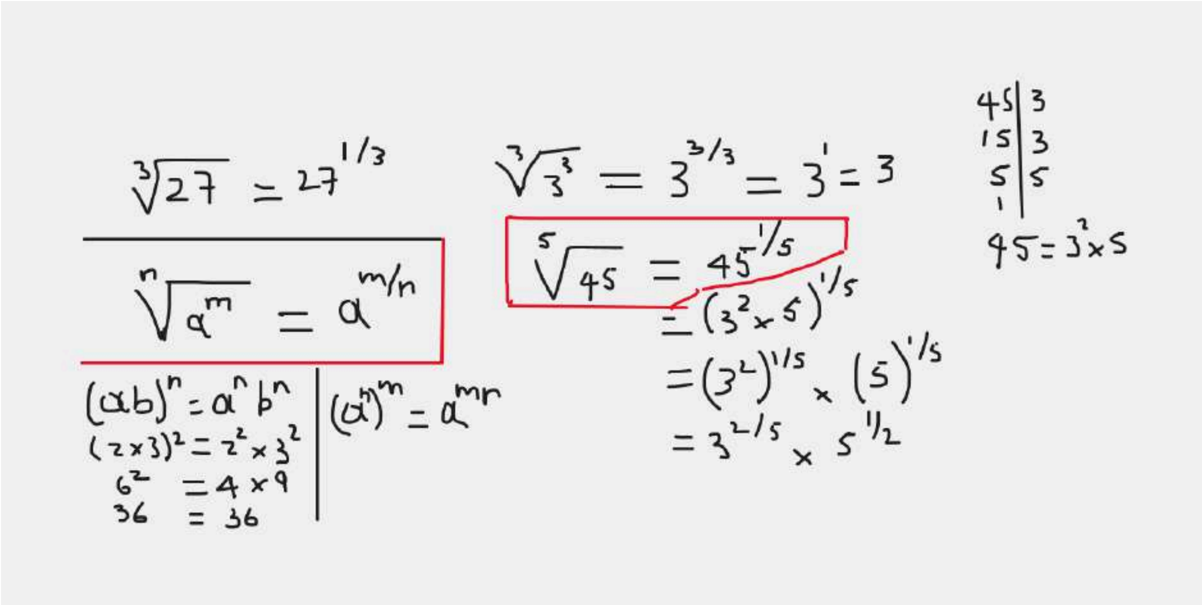
<u>Ejemplo 0.11</u>

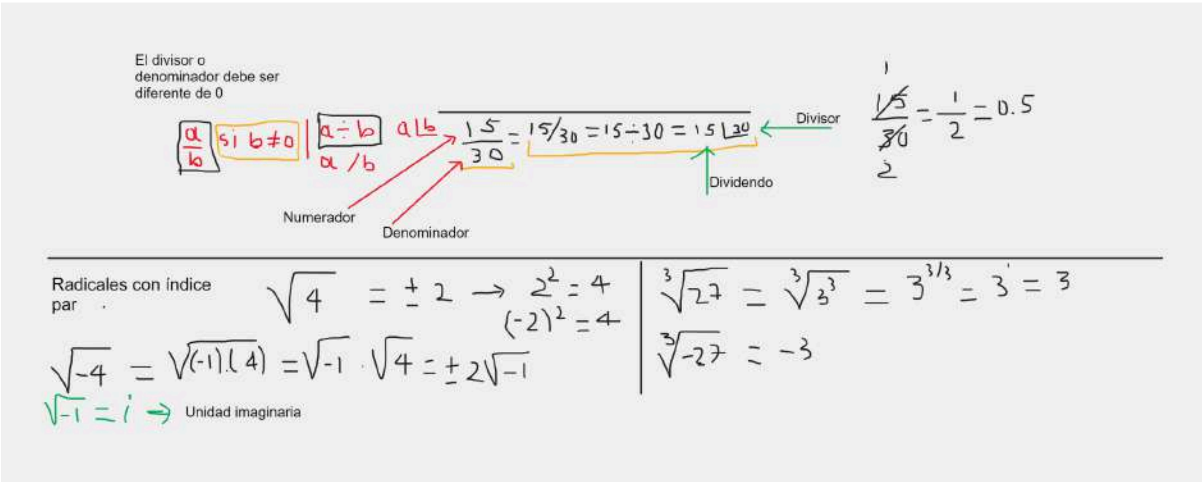
$\sqrt{5^2} = |5| = 5;\ \sqrt{(-3)^2} = |-3| = 3$

Tablero virtual

Ilustración en clase virtual de operaciones con exponentes, radicales y conceptos teóricos.







Operadores básicos en matemáticas y computación

Un *operador* es un símbolo usado en matemáticas para representar una operación a realizar, la cual puede ser unaria (con un *operando*) o binaria (con dos *operandos*). En la aritmética y en el álgebra se cuenta, entre otros, con varios operadores elementales; cada operador tiene una *prioridad* asignada, lo cual significa que los de mayor prioridad, se ejecutarán primero. Se dividen en tres grupos, de los cuales se muestra su representación matemática, así como algorítmica.

Operadores aritméticos

Utilizados para realizar cálculos (operaciones) matemáticos

Nombre Operador	Símbolo matemático y algorítmico (computacional)	Prioridad

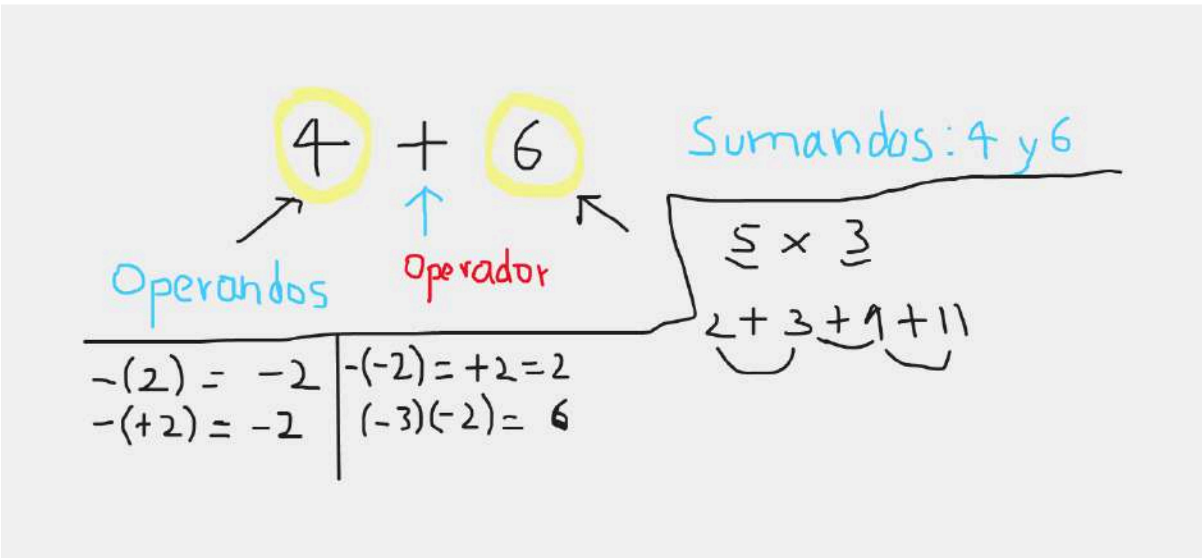
Negación aritmética unaria	$-$	Alta
Potencia	x^y $\wedge$ $**$	Alta - media
Raíz cuadrada	$\sqrt{x}$ |/ $raiz2(x)$ $sqrt(x)$	
Multiplicación	$\times$ $*$. $()()$	Media
División	$\div$ $/$ $\frac{a}{b}$ a|div	
Módulo	$\%$ mod	
División entera	\ div $//$	
Suma	$+$	Baja
Resta	$-$	

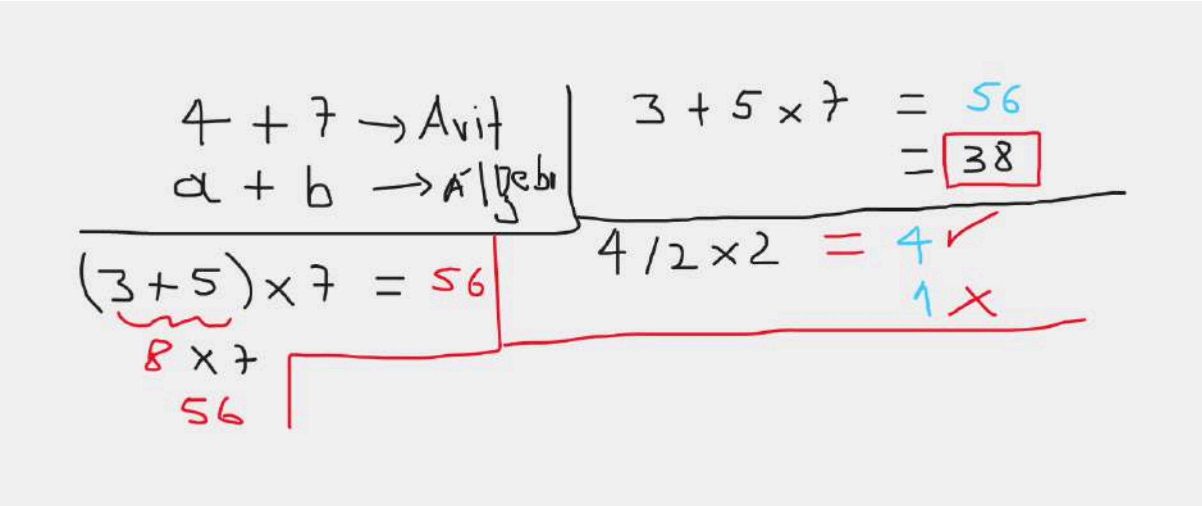
Notas

- La negación aritmética es una operación *unaria* que consiste en negar el símbolo del número (operando). Ejemplo: $-(+2)$, $-(8)$, $-(-5)$, -94
- La prioridad se refiere al orden en que los operadores se efectúan en una expresión aritmética: los de mayor prioridad se efectúan primero.
- Las operaciones encerradas entre paréntesis se efectúan primero, por lo que tienen mayor prioridad. Los paréntesis modifican la prioridad de los operadores en una expresión.
- Si hay dos operadores de igual prioridad, se ejecuta primero el que se encuentre más a la izquierda, esto es, se sigue el orden de izquierda a derecha.

Tablero virtual

Ilustración en clase virtual acerca de las partes que intervienen en una expresión matemática y prioridad de los operadores.





Operación de módulo

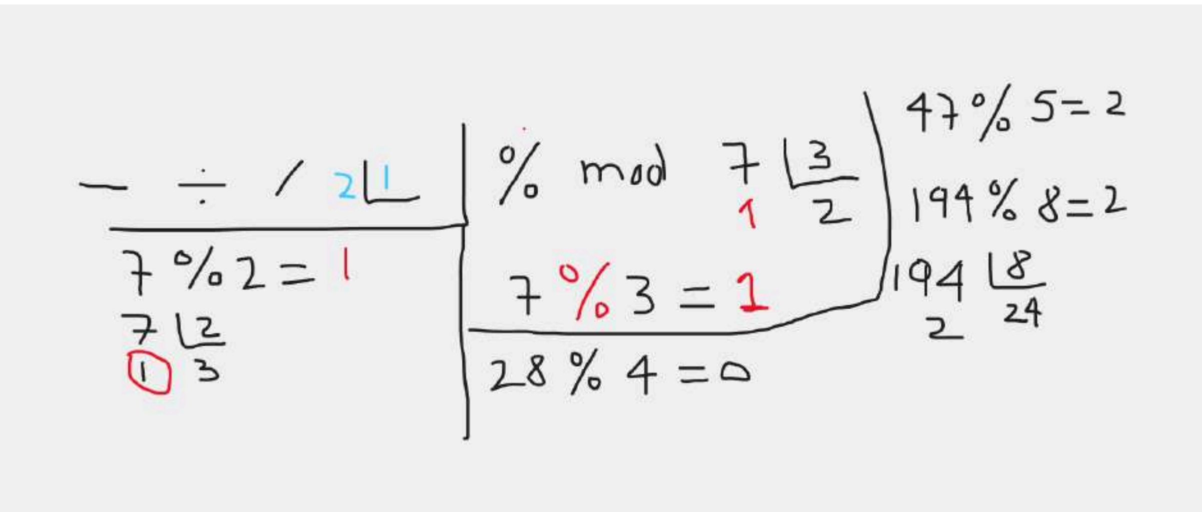
Es una división entera que devuelve el residuo de ésta. Se representa con el símbolo ***%*** ó la palabra ***mod***, entre otros usados.

Ejemplo 0.12

1. *7 % 5 = 2*
2. *17 % 2 = 1*
3. *48 mod 4 = 0*
4. *57 % 6 = 3*
5. *49 mod 5 = 4*
6. *9 mod 20 = 9*
7. *55 % 11 = 0*

Tablero virtual

Ilustración en clase virtual del uso del operador de módulo.



Operadores relacionales o de comparación

Permiten realizar comparaciones entre dos expresiones; el resultado de una comparación es un valor booleano (verdadero (V) o falso (F); 1 ó 0, respectivamente); esto son:

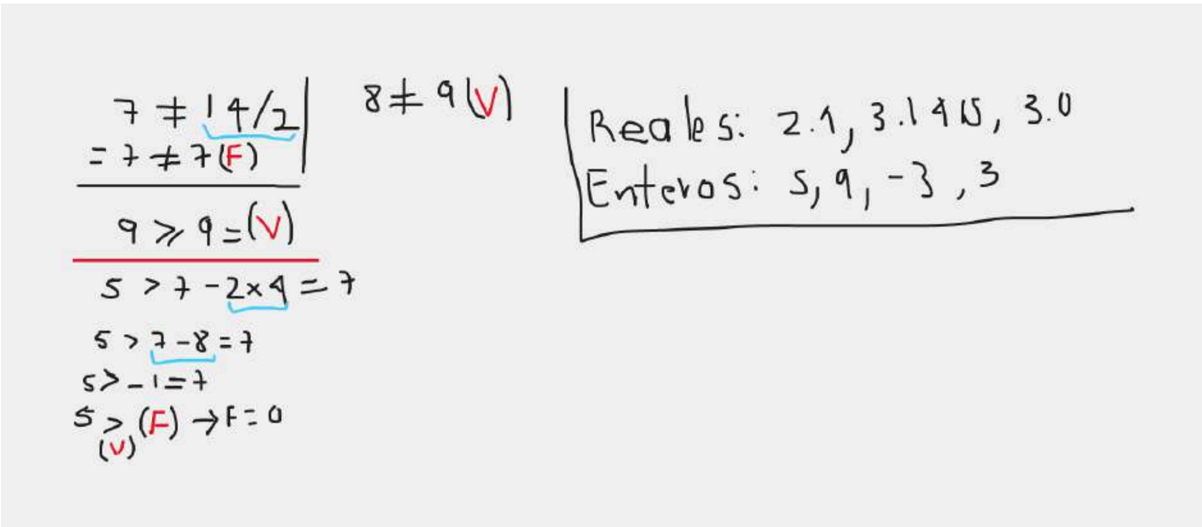
Nombre Operador	**Símbolo**	**Prioridad**
Igual	$=$ $==$	Alta
Diferente	$\neq$ $<>$ $!=$	
Mayor que	$>$	Media
Menor que	$<$	
Mayor o igual que	$\geq$ $>=$	Baja
Menor o igual que	$\leq$ $<=$	

<u>Ejemplo 0.13</u>

1. $8 \neq 9 \rightarrow (V)$
2. $9 \geq 9 \rightarrow (V)$
3. $7 \neq 14 \div 2 \rightarrow (F)$
4. $9 \times 2 \leq 50 \div 10 \rightarrow (F)$
5. $-8 = 8 \rightarrow (F)$

Tablero virtual

Ilustración en clase virtual de operaciones usando operadores de comparación.



Operadores lógicos o booleanos

Permiten conectar expresiones de comparación y realizar operaciones lógicas. El valor devuelto (verdadero o falso) depende del conectivo lógico utilizado, según las leyes del álgebra proposicional y booleana; estos son algunos:

Nombre Operador	**Símbolo**	**Prioridad**

Negación lógica unaria	$\neg$ $\sim$ $-$! no not	Alta ↓ (mayor a menor)
Conjunción	$\wedge$ && • y and	
Disyunción	$\vee$ || + o or	
Disyunción exclusiva	$\veebar$ $\oplus$ W $'o\ bien'$ xor eor	

Al llegar a la sección de *lógica matemática* realizaremos operaciones con estos operadores y se ampliará más este tema.

Notación algorítmica

Al trabajar con computadores y lenguajes de programación, algunos símbolos matemáticos son difíciles de obtener desde los caracteres estándar del teclado. Es por ello que las expresiones matemáticas deben ser reescritas cuando las llevamos a un lenguaje de programación utilizando para ello la notación algorítmica típica de la informática.

Para ello, nos basaremos en los operadores vistos anteriormente y su prioridad, así como en las propiedades del álgebra para los números reales y observando cuáles de estos operadores pueden ser usados en un lenguaje determinado. En lógica de programación, el tema de los operadores puede flexibilizarse, pero siempre manteniendo la notación algorítmica. Veamos algunos ejemplos.

Ejemplo 0.14
Escribir en notación algorítmica las siguientes expresiones matemáticas

1. $ab + 3ac^3$
2. $\sqrt[3]{b^2} + \frac{a}{3}$
3. $\frac{a-2b+3c}{\sqrt{2}}$
4. $a \geq 0 \wedge b \neq (4 + 2ab^3) \vee [\neg(a + 2 < b) \wedge (-9 = c)]$

Solución
1. $a * b + 3 * a * c \wedge 3$
2. $b \wedge (2/3) + a / 3$
3. Veamos varias formas de escribir esta expresión
    a. $(a - 2 * b + 3 * c) / 2 \wedge (1/2)$
    b. $(a - 2 * b + 3 * c) / 2 \wedge 0.5$
    c. $(a - 2 * b + 3 * c) / raizc(2);$ $donde\ raizc()\ es\ una\ función$
4. Veamos cómo escribir esta expresión que incluye todos los operadores. Para la conjunción podemos usar: **y**, **and**, ó **&&**, que son admitidos en lógica de programación y algunos lenguajes; análogamente para la disyunción podemos usar: o, or ó ||. Por último, podemos usar para la negación: **no**, **not** ó **!**. Recordemos que los operadores lógicos trabajan como conectivos.

    $a >= 0 \,\&\&\, b <> (4 + 2 * a * (b \wedge 3)) \,||\, (!\,(a + 2 < b) \,\&\&\, (-9 = c))$

Nota

Observe que por la prioridad de los operadores, no es necesario usar paréntesis en algunas expresiones, a no ser que se quiera modificar ésta.

Preguntas

1. Describa los operadores más comunes utilizados en matemáticas e informática
2. ¿Qué es un operador matemático?
3. ¿Cuál es la operación inversa de la suma?
4. ¿Cuál es la operación inversa de la multiplicación?
5. ¿Cuál es la operación inversa de la potencia?
6. ¿Qué es un número primo?
7. ¿Cómo se interpreta el orden y el valor absoluto?
8. ¿Qué devuelve una operación de comparación?
9. ¿Cuáles son los operadores fundamentales?
10. ¿Cuáles son los valores de verdad de las constantes booleanas?
11. ¿A qué hace referencia la prioridad y cómo puede alterarse?
12. ¿Todos los números primos son impares?
13. ¿Cuáles son las propiedades de los números reales?
14. ¿Cuáles son las partes de una potencia y una raíz?
15. ¿Cuáles son los módulos de la suma y el producto?
16. ¿Cuántos números primos hay entre 1 y 25, entre 26 y 50, 51 y 75, 76 y 100? ¿Cómo se comporta el patrón, disminuyen o aumentan en cada intervalo?

Ejercicios

1. Realice los siguientes cálculos para encontrar el valor numérico de cada expresión si $a = 3,\ b = 4,\ c = -1,\ d = -5,\ e = 2$
    a. $ab^2 + 3c - \sqrt{b}$
    b. $5e - 2bcd + 4(d^3 - 2a + c) + a\,\%\,e$
    c. $(\frac{b}{e} + \frac{e}{c}) - 6(\frac{c^2}{a})$
    d. $\sqrt[3]{d^6} + \frac{a+b+c}{e} + e - b\,\%\,2$
    e. $\sqrt{ab - a} + 5d \div c - a^4be$
2. Reescriba cada expresión del punto 1) en notación algorítmica o informática
3. Teniendo en cuenta los resultados encontrados en el punto 1), determine el valor lógico de las siguientes comparaciones (las letras corresponden a resultados encontrados en los literales del punto 1), no a los valores numéricos dados allí)
    a. `a > b`
    b. `ab = cd / e`
    c. `d ^ 2 <= 5ae - b/2`
    d. `d <> 2ea`
    e. `3/c >= 6be + 4a`
4. Determine el valor absoluto para cada resultado encontrado en el punto 1)
5. Determine si los siguientes números son primos o no

a. 154
b. 96
c. 63
d. 37
e. -13

6. Calcule: $\sum_{i=1}^{7} (i + 2)^2$; $\prod_{j=3}^{6} \sqrt{(j - 1)}$; $8!$; $\sum_{i=1}^{4} i!$
7. Demuestre que las siguientes expresiones en valor absoluto son válidas:
   a. |abcd| = |a|.|b|.|c|.|d|
   b. |abc| = |ab|.|c|
   c. |ab| / |cd| = |a|.|b| / (|c|.|d|)
   d. ||a|| = |a|
   e. |ab| / |c| = |a||b| / |c| = |a| / |c| * |b|
8. Sean p_1, p_2 dos números pares e i_1, i_2 dos números impares. Demuestre que:
   a. $2p_1$ *es par*
   b. $2i_1$ *es par*
   c. $(i_1)^3$ *es impar*
   d. $(p_1 + i_1)(p_2 + i_2)$ *es impar*
   e. $p_1(i_1 + i_2) + (i_2)^2$ *es impar*

Capítulo 1. Sistemas numéricos

Sistemas numéricos

Un sistema de numeración es un conjunto de símbolos y de normas a través del cual pueden expresarse la cantidad de objetos en un conjunto, es decir, a través del cual pueden representarse todos los números válidos. Esto quiere decir que todo sistema de numeración contiene un conjunto determinado y finito de símbolos, además de un conjunto determinado y finito de reglas mediante las cuales combinarlos.

Los sistemas de numeración fueron una de las principales invenciones humanas en la antigüedad, y cada una de las civilizaciones de antaño tuvo su propio sistema, relacionado con su modo de ver el mundo, o sea, con su cultura.

Los sistemas de numeración pueden clasificarse en tres grandes tipos distintos, que se describen a continuación.

Sistemas no posicionales

Son aquellos en los que a cada símbolo le corresponde un valor fijo, sin importar la posición que ocupe dentro de la cifra (si aparece primero, a un lado o después).

Los sistemas de numeración no posicionales fueron los primeros en existir y tuvieron las bases más primitivas: los dedos de las manos, nudos en una cuerda u otros métodos de registro para coordinar conjuntos numéricos. Por ejemplo, si se cuenta con los dedos de una mano, luego se podrá contar en manos enteras.

En estos sistemas los dígitos tienen un valor propio, independientemente de su ubicación en la cadena de símbolos, y para formar nuevos símbolos, deben sumarse los valores de los símbolos (por eso se les conoce también como sistemas aditivos). Estos sistemas eran sencillos, fáciles de aprender, pero requerían de numerosos símbolos para expresar grandes cantidades, de modo que no eran del todo eficientes. Son ejemplos de este tipo de sistemas los siguientes.

El sistema de numeración egipcio

Surgido alrededor del III milenio a. C., tenía como base la decena (10) y empleaba jeroglíficos diferentes para cada orden de unidades: uno para la unidad, uno para la decena, uno para la centena y así sucesivamente hasta el millón.

El sistema de numeración azteca

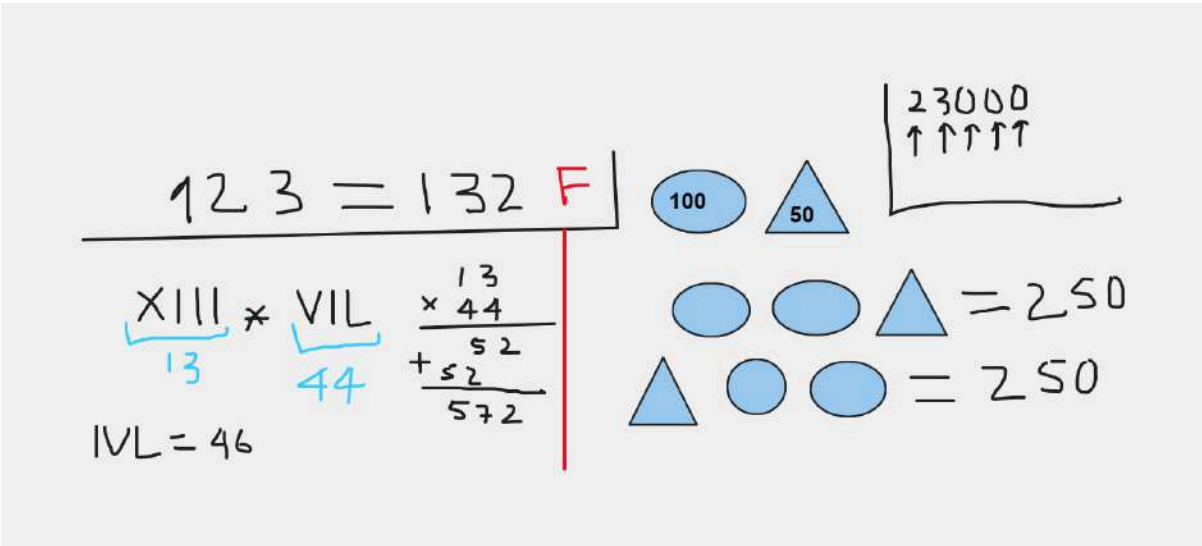
Propio del imperio mexica, tenía la veintena como base (20) y empleaba objetos concretos como símbolos: una bandera equivalía a 20 unidades, una pluma o unos cabellos equivalían a 400, una bolsa o costal a 8000, entre otros.

El sistema de numeración griego

Específicamente el jónico, fue inventado y difundido en el Mediterráneo oriental a partir del siglo IV a. C., en sustitución del sistema preexistente. Era un sistema alfabético que empleaba letras para significar números, haciendo coincidir la letra con su lugar cardinal en el alfabeto (`A = 1, B = 2`). Así, se asignaba a cada cifra del 1 al 9 una letra, a cada decena otra letra específica, a cada centena otra más, hasta emplear 27 letras: las 24 del alfabeto griego y tres caracteres especiales.

Tablero virtual

Ilustración en clase virtual de cómo podría ser un sistema numérico no posicional con dos figuras que tienen valores asignados y de cómo se puede utilizar.



Sistemas semi-posicionales

Son aquellos en los que el valor de un símbolo tiende a ser fijo, pero se puede modificar en situaciones particulares de aparición (aunque suelen constituir más bien excepciones). Se entiende como un sistema intermedio entre el posicional y el no posicional.

Los sistemas de numeración semi-posicionales combinan la noción del valor fijo de cada símbolo con ciertas normas de posicionamiento, por lo que pueden entenderse como un sistema híbrido o mixto entre posicionales y no posicionales. Gozan de facilidades para representar cifras grandes, manejando el orden de los números y procedimientos formales como la multiplicación, de modo que representan un paso adelante en complejidad respecto de los sistemas no posicionales.

En buena medida, el surgimiento de los sistemas semi-posicionales puede entenderse como el tránsito hacia un modelo más eficiente de numeración que pudiera satisfacer las necesidades más complejas de una economía más desarrollada, como la de los grandes imperios de la antigüedad clásica. El sistema numérico semi posicional más conocido es el romano.

El sistema de numeración romano

Creado en la antigüedad romana, sobrevive hasta nuestros días. En este sistema se construían las cifras usando ciertas letras mayúsculas del alfabeto latino (`I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000`, etc.), cuyo valor era fijo y operaba en base a la adición y la sustracción, dependiendo del lugar de aparición del símbolo. Si el símbolo se hallaba a la izquierda de un símbolo de igual o menor valor (como en `II = 2` o en `XI = 11`), se debían sumar los valores totales; mientras que si el símbolo estaba a la izquierda de un símbolo de mayor valor (como en `IX = 9`, o `IV = 4`), debían restarse.

Sistemas posicionales o ponderados

Son aquellos en los que el valor de un símbolo está determinado tanto por su propia expresión, como por el lugar que ocupe dentro de la cifra, pudiendo valer más o menos o expresar distintos valores dependiendo de dónde se encuentre.

Los sistemas de numeración posicionales son los más complejos y eficientes de los tres tipos que existen. La combinación del valor propio de los símbolos y el valor asignado por su posición les permite construir con muy pocos caracteres cifras muy altas, sumando y/o multiplicando el valor de cada uno, lo cual los hace sistemas más versátiles y prácticos.

Generalmente, los sistemas posicionales emplean un conjunto fijo de símbolos y a través de su combinatoria se produce el resto de las cifras posibles, hasta el infinito, sin necesidad de crear nuevos signos, sino inaugurando nuevas columnas de símbolos. Desde luego, esto implica que un error en la cadena altera también el valor total de la cifra.

Los primeros ejemplos de sistemas de este tipo surgieron en el seno de los grandes imperios o las culturas antiguas más exigentes en materia cultural y comercial, como el Imperio babilónico del milenio II a. C. Algunos ejemplos de este tipo de sistema de numeración se comentan a continuación.

El sistema de numeración babilónico

Inventado por los antiguos pueblos mesopotámicos, es considerado el primer sistema posicional y que tuvo una fuerte influencia en el sistema actual de numeración sexagesimal, utilizado en distintas aplicaciones, como para indicar el tiempo, por ejemplo, en horas, minutos y segundos. Utilizaban un instrumento de sección triangular que cuando se ponía sobre arcilla, dejaba marcas en forma de cuña que podían orientarse de distintas maneras. Esto es conocido como escritura cuneiforme (figura de cuña). No disponía de un símbolo para el cero.

El sistema de numeración chino clásico

Sus orígenes se remontan aproximadamente al 1500 a. C. y es un sistema muy estricto de representación vertical de los números a través de símbolos propios, combinando dos sistemas distintos: uno para la escritura coloquial y cotidiana, y otro para los registros comerciales o financieros. Era un sistema decimal que disponía de nueve signos diferentes que podían ubicarse uno junto al otro para sumar sus valores, a veces intercalando un signo especial o alternando la ubicación de los signos para indicar una operación específica.

El sistema de numeración indoarábigo

Inventado por los antiguos sabios de la India y heredado luego por los árabes musulmanes, llegó a Occidente a través del Al-Ándalus y acabó reemplazando a los números romanos tradicionales. En este sistema, similar al decimal moderno, se representan las unidades del 0 al 9 mediante glifos específicos, que representaban mediante líneas y ángulos el valor de cada uno. El sistema de funcionamiento de este sistema es básicamente el mismo que el sistema decimal moderno occidental.

El sistema de numeración maya

Fue creado para medir el tiempo, en lugar de usarlo para transacciones comerciales; su base era vigesimal (20) y sus símbolos se corresponden con el calendario propio de esta civilización precolombina. Las cifras, agrupadas de 20 en 20, se representan con signos básicos (rayas, puntos y caracoles o conchas); y para pasar a la siguiente veintena, se añade un punto en el siguiente nivel de escritura. Además, los mayas fueron de las primeras culturas en utilizar el número cero, necesario para su sistema posicional.

El sistema decimal moderno (Notación decimal)

Con apenas los dígitos (símbolos) del 0 al 9 permite construir cualquier cifra posible, añadiendo columnas cuyo valor se suma conforme se avanza hacia la derecha, teniendo como base la decena (10). Así, añadiendo símbolos a 1 podemos construir 10, 195, 1958 o 19589. Es importante aclarar que los símbolos que emplea provienen de la *numeración indoarábiga*.

Sistemas numéricos computacionales

También es posible clasificar los sistemas de numeración basados en la cifra que utilizan de base para sus cálculos. Así, por ejemplo, el sistema occidental actual es decimal (pues su base es 10, utilizando diez símbolos para escribir cualquier número), mientras que el sistema de numeración sumerio era sexagesimal (su base era 60).

En informática se utilizan varios sistemas de numeración, además del decimal:

- Binario: base 2
- Decimal: base 10 (es el que utilizamos en la cotidianidad)
- Octal (base 8)
- Hexadecimal (base 16)

La razón de utilizar el sistema en base 2, es que todos los computadores utilizan el sistema de numeración binaria para representar la información mediante señales eléctricas. Los sistemas de numeración actuales son posicionales, es decir, un símbolo puede tener un significado diferente en función de su posición.

Leemos los números de izquierda a derecha, teniendo presente que el número a la izquierda tiene mayor valor que otro que esté a su derecha.

La base indica cuántos dígitos (símbolos) hacen parte de ésta. Los dígitos de la base serán siempre menores a ésta:

$0 \leq d < b$ $(donde\text{: } d\ es\ cualquier\ dígito\ del\ sistema\ numérico;\ b\ es\ la\ base)$

Así, la base 2 posee dos símbolos; la base 8, ocho símbolos; la base 10, diez símbolos; y la base 16, dieciséis símbolos, respectivamente.

Ejemplo 1.1
Sea el número 1234 escrito en notación decimal.

Aquí la cifra *1,* a pesar de ser menor que la cifra *4*, pero por estar a la izquierda de todo el conjunto, es la de mayor valor, ya que su posición representa un millar, mientras que el 4, al estar en el último lugar, solo representa 4 unidades.

Ahora vamos a posicionar cada dígito comenzando desde la posición cero (0) de acuerdo a su *peso* (valor) en el número, lo cual significa que el dígito más a la derecha tendrá la posición 0:

$1\ \ 2\ \ 3\ \ 4\ \leftarrow Número$
$\uparrow\ \uparrow\ \uparrow\ \uparrow$
$3\ \ 2\ \ 1\ \ 0\ \leftarrow Posición\ de\ cada\ dígito\ en\ el\ número$

Así, el número 1234 puede escribirse de la siguiente manera usando la notación decimal:
$1234 = 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$
$1234 = 1 \times 1000 + 2 \times 100 + 3 \times 10 + 4 \times 1$
$1234 = 1000 + 200 + 30 + 4 = 1234$

Esto nos lleva a enunciar el siguiente teorema, conocido como el "*Teorema Fundamental de la Numeración"*.

Teorema Fundamental de la Numeración

Este teorema establece la forma general de construir números en base decimal partiendo de números en cualquier sistema de numeración posicional, incluyendo la misma base decimal. El teorema permite relacionar una cantidad expresada en cualquier sistema de numeración posicional con la misma cantidad expresada en el sistema decimal.

$$N = \sum_{i=-k}^{n-1} d_i b^i = d_{n-1} b^{n-1} + \ldots + d_1 b^1 + d_0 b^0 + d_{-1} b^{-1} + \ldots + d_{-k} b^{-k}$$

Donde:
N: número válido en el sistema de numeración decimal
n: total de dígitos del número en una base dada
d_i: dígito i-ésimo del número; es uno de los símbolos permitidos en el sistema numérico
b: base del sistema de numeración
k: número de dígitos de la parte decimal en la base dada

Para el caso particular de números enteros (sin parte decimal), la expresión es:

$$N = \sum_{i=0}^{n-1} d_i b^i = d_{n-1} b^{n-1} + \ldots + d_1 b^1 + d_0 b^0$$

Sistema Binario

Su base es 2. Es el código que entienden los dispositivos electrónicos, entre ellos los computadores; se representa utilizando solamente los símbolos (dígitos) cero (`0`) y uno (`1`); esto, dada la naturaleza misma de estos sistemas que trabajan con dos niveles de estado: *Apagado (Off)* o *Encendido (On)*. Los números representados en esta base deben tener como subíndice el número 2.

Ejemplo 1.2
Los siguientes números están escritos en binario (base 2)

11_2, 1000011_2, 10101110_2, 10000_2, 11111_2

Sistema Octal

Su base es 8, una potencia exacta de 2, lo cual facilita la conversión entre binario y octal. Este sistema numérico se basa en 8 dígitos: `0, 1, 2, 3, 4, 5, 6, 7`. Este sistema ha sido utilizado en informática para indicar direcciones de memoria y longitudes de palabra entre otros aspectos. Los números representados en esta base deben tener como subíndice el número 8.

Ejemplo 1.3
Los siguientes números están escritos en octal (base 8)

452_8, 3422_8, 156000_8, 10000_8, 7450_8

Sistema Decimal

Su base es 10. El sistema decimal es el que comúnmente conocemos; este sistema de números está basado en el diez con los dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Los números representados en esta base no requieren tener como subíndice el número 10, ya que se sobreentiende que están en base 10, aunque podría ponerse.

Ejemplo 1.4
Los siguientes números están escritos en decimal (base 10)

1, 1000_{10}, 130000, 987, 0

Sistema Hexadecimal

Su base es 16, una potencia exacta de 2, lo cual facilita la conversión de binario a hexadecimal y viceversa. El sistema hexadecimal está conformado por los símbolos (dígitos) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Donde cada letra equivale numéricamente al valor en decimal:
A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Ejemplo 1.5
Los siguientes números están escritos en hexadecimal (base 16 o base H)

110_{16}, ABC_{16}, $D55_{16}$, 10000_H, $F91EB_{16}$

Conversión de bases

Un número escrito en una base puede ser convertido a otra base cualquiera mediante una serie de operaciones matemáticas que incluyen sumas, multiplicaciones, divisiones y potencias. Tenemos fundamentalmente, dos tipos de conversión: de base 10 a cualquier base y viceversa.

Conversión de un número en base 10 a una base cualquiera

El proceso consiste en realizar divisiones enteras sucesivas por la base hasta encontrar un cociente de cero (0). Todos los residuos se toman desde el último encontrado hasta el primero formando una cadena que nos debe indicar el número buscado en dicha base.

Conversión de base 10 a base 2

Ejemplo 1.6
Convertir 11 → base 2

$11 \div 2 = 5\ y\ sobra\ 1\ (operación\ 1)$
$5 \div 2 = 2\ y\ sobra\ 1\ (operación\ 2)$
$2 \div 2 = 1\ y\ sobra\ 0\ (operación\ 3)$
$1 \div 2 = 0\ y\ sobra\ 1\ (operación\ 4)$

Tomamos los residuos del último al primero (desde la última operación a la primera), para obtener la cadena: 1011.
Esto es $11 = 1011_2$.

Conversión de base 10 a base 8

Ejemplo 1.7
Convertir 11 → base 8

$11 \div 8 = 1\ y\ sobra\ 3\ (operación\ 1)$
$1 \div 8 = 0\ y\ sobra\ 1\ (operación\ 2)$

Tomamos los residuos del último al primero, para obtener la cadena: 13.
Esto es $11 = 13_8$.

Conversión de base 10 a base 16

Ejemplo 1.8
Convertir 11 → base 16

$11 \div 16 = 0\ y\ sobra\ 11\ (operación\ 1)$

Tomamos los residuos del último al primero, para obtener la cadena: 11 = B.
Esto es $11 = B_{16}$.

Conversión de un número en una base cualquiera a base 10

Para esto se tendrá en cuenta lo mencionado arriba, en el sentido de que los sistemas numéricos modernos son posicionales y a cada dígito del número se le asigna o le corresponde una posición numérica, iniciando en cero (0) para el dígito de menor valor, esto es, el que se encuentre más a la derecha (último), y asignando posiciones consecutivas a los siguientes (es decir, 1 al penúltimo, 2 al antepenúltimo, etc.); además de tener presente de igual forma el valor numérico de cada cifra del número dado.

Ya indicado lo anterior, procedemos de la siguiente forma: sumamos los productos correspondientes al dígito de una posición por la base elevada a dicha posición. En otras palabras, aplicamos el teorema fundamental de la numeración:

$$N = \sum_{i=-k}^{n-1} d_i b^i$$

Conversión de base 2 a base 10

Ejemplo 1.9

Convertir $1011_2 \rightarrow$ base 10

$$N = \sum_{i=-k}^{n-1} d_i b^i = \sum_{i=0}^{3} d_i (2)^i$$

$= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= (1 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1)$

$= 8 + 0 + 2 + 1$

$= 11$

Conversión de base 8 a base 10

Ejemplo 1.10

Convertir $13_8 \rightarrow$ base 10

$(1 \times 8^1) + (3 \times 8^0)$

$= (1 \times 8) + (3 \times 1)$

$= 8 + 3$

$= 11$

Conversión de base 16 a base 10

Ejemplo 1.11

Convertir $B_{16} \rightarrow$ base 10

$(B \times 16^0)$

$= (B \times 1)$ *sustituyendo el valor de* B

$= 11 \times 1$

$= 11$

Tabla de conversión entre decimal, binario, octal y hexadecimal

Es una herramienta útil para realizar conversiones entre bases arbitrarias sin necesidad de pasar por operaciones intermedias. Por ejemplo, con la tabla podemos convertir rápidamente un número en base 2 a base 8 sin tener que hacer las operaciones correspondientes para pasar primero el número en base 2 a base 10, y luego en esta base convertirlo a base 8, esto es, hacer estas conversiones: $base\ 2 \rightarrow base\ 10 \rightarrow\ base\ 8$. Obviamente, debemos tener a la mano esta tabla para poder sacar provecho de los cálculos de conversión rápidos y seguir unas cuantas reglas.

Para esto, tendremos presente que un dígito octal se representa con tres dígitos binarios y un dígito hexadecimal se representa con cuatro dígitos binarios. Esto, gracias a que la base 8 y la base 16 son potencias exactas de 2, la base binaria, en el sistema decimal: $2^3 = 8$ y $2^4 = 16$.

<u>Nota</u>
A pesar de que el método usado para convertir números con la tabla de conversión es muy simple de utilizar manualmente, vale aclarar que algorítmicamente es más complejo de implementar que si se recurre a la conversión intermedia de llevar primero el número a base 10.

Decimal	**Binario**	**Octal**	**Hexadecimal**
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8

9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Nota
Observe que la columna de los números binarios se puede formar como una *tabla de verdad* para n = 4 con ceros y unos. Más adelante al tratar los temas de lógica proposicional y compuertas lógicas, ampliaremos este aspecto.

Convertir números en base 2 a base 8

Tomamos el número en base 2 y lo dividimos tomando de a tres dígitos, comenzando desde la derecha y completando con ceros los dígitos a la izquierda para formar la terna de bits. Luego observamos cada terna y la buscamos en la columna *binario* y extraemos su equivalente en octal. El número en octal se forma uniendo las cadenas obtenidas.

Ejemplo 1.12
Convertir a octal los siguientes números binarios

1. $101101_2 \rightarrow base\ 8$
2. $1110_2 \rightarrow base\ 8$
3. $1111011010011_2 \rightarrow base\ 8$

Solución
1. Tomamos el número y dividimos éste de a tres dígitos: $101 \mid 101$. Observando la tabla, encontramos que $101_2 = 5_8$ Por tanto, $101101_2 = 55_8$
2. Procedemos de manera similar: $001 \mid 110$. Observamos que $001_2 = 1_8$ y $110_2 = 6_8$. Por tanto, $1110_2 = 16_8$
3. $001 \mid 111 \mid 011 \mid 010 \mid 011$. Observamos que $001_2 = 1_8$, $111_2 = 7_8$, $011_2 = 3_8$, $010_2 = 2_8$, $011_2 = 3_8$. Por tanto, $1111011010011_2 = 17323_8$

Convertir números en base 8 a base 2

El número en base 8 se obtiene formando tripletas de bits a partir de un número binario, por tanto, podemos tomar cada dígito octal y mirar su equivalente en binario, cogiendo solo los tres últimos dígitos; al unir toda la cadena de ternas de bits, obtendremos el número binario buscado.

Ejemplo 1.13
Convertir a binario los siguientes números octales

1. $55_8 \rightarrow base\ 2$
2. $16_8 \rightarrow base\ 2$
3. $17323_8 \rightarrow base\ 2$

Solución
1. Tomamos el número y buscamos en la tabla cada dígito octal y extraemos su equivalente binario con los tres últimos dígitos: $5_8 = 101_2$, $5_8 = 101_2$. Por tanto, $55_8 = 101101_2$
2. Procedemos de manera similar: $001 \mid 110$. Observamos que $1_8 = 001_2$ y $6_8 = 110_2$. Por tanto, $16_8 = 1110_2$
3. Observamos que $1_8 = 001_2$, $7_8 = 111_2$, $3_8 = 011_2$, $2_8 = 010_2$, $3_8 = 011_2$. Por tanto, uniendo las cadenas de bits, obtenemos que $17323_8 = 1111011010011_2$

Convertir números en base 2 a base 16

Tomamos el número en base 2 y lo dividimos tomando de a cuatro dígitos, comenzando desde la derecha y completando con ceros los dígitos a la izquierda para formar 'cuartetos' de bits. Luego observamos cada cuarteto y lo buscamos en la columna *binario* y extraemos su equivalente en hexadecimal. El número en hexadecimal se forma uniendo las cadenas obtenidas.

Ejemplo 1.14
Convertir a hexadecimal los siguientes números binarios

1. $101101_2 \rightarrow base\ 16$
2. $1110_2 \rightarrow base\ 16$
3. $1111011010011_2 \rightarrow base\ 16$

Solución
1. Tomamos el número y dividimos éste de a cuatro dígitos y completamos con ceros: $0010 \mid 1101$. Observando la tabla, encontramos que $0010_2 = 2_{16}$, $1101_2 = D_{16}$. Por tanto, $101101_2 = 2D_{16}$

2. Este número no requiere completar dígitos a la izquierda, por tanto, $1110_2 = E_{16}$
3. Dividimos el número binario en grupos de cuatro bits: $0001 \mid 1110 \mid 1101 \mid 0011$. Observamos que $0001_2 = 1_{16}$, $1110_2 = E_{16}$, $1101_2 = D_{16}$, $0011_2 = 3_{16}$. Por tanto, $1111011010011_2 = 1ED3_{16}$

Convertir números en base 16 a base 2

El número en base 16 se obtiene formando grupos de cuatro de bits a partir de un número binario, por tanto, podemos tomar cada dígito hexadecimal y mirar su equivalente en binario, cogiendo los cuatro dígitos; al unir toda las cadenas de cuatro bits, obtendremos el número binario buscado.

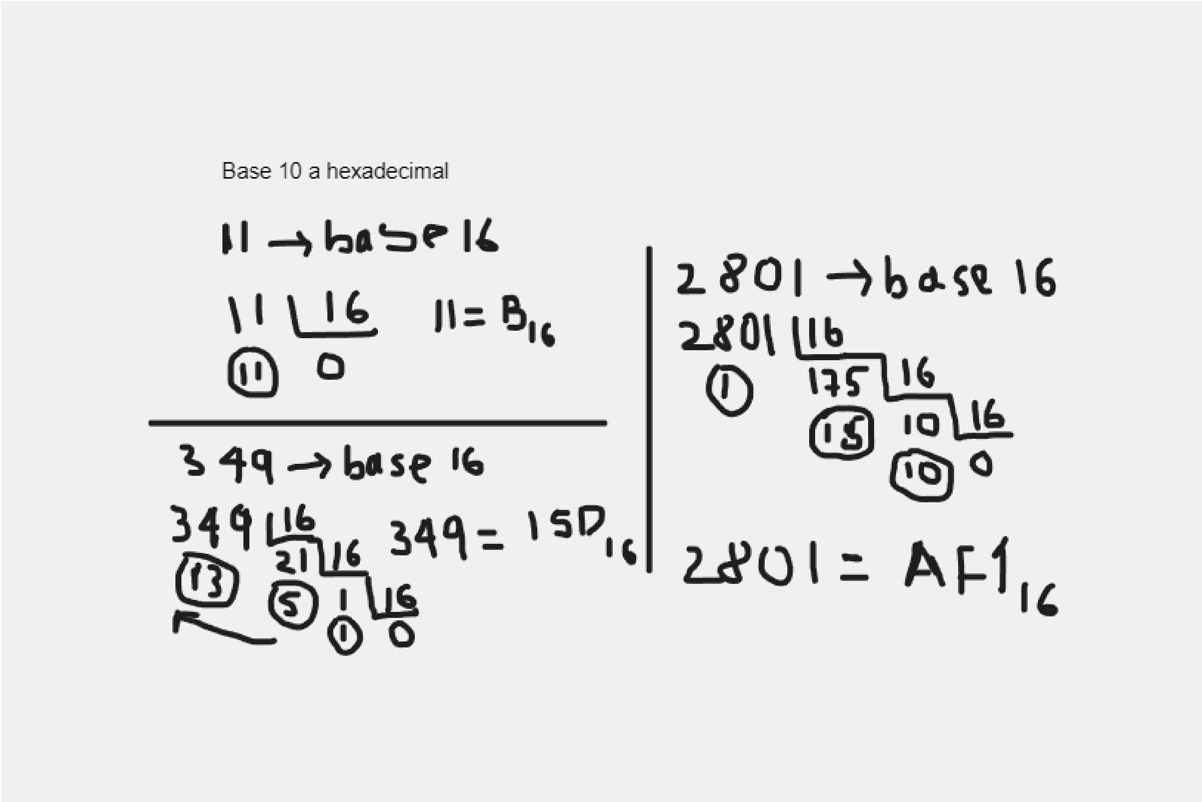
Ejemplo 1.15
Convertir a binario los siguientes números hexadecimales

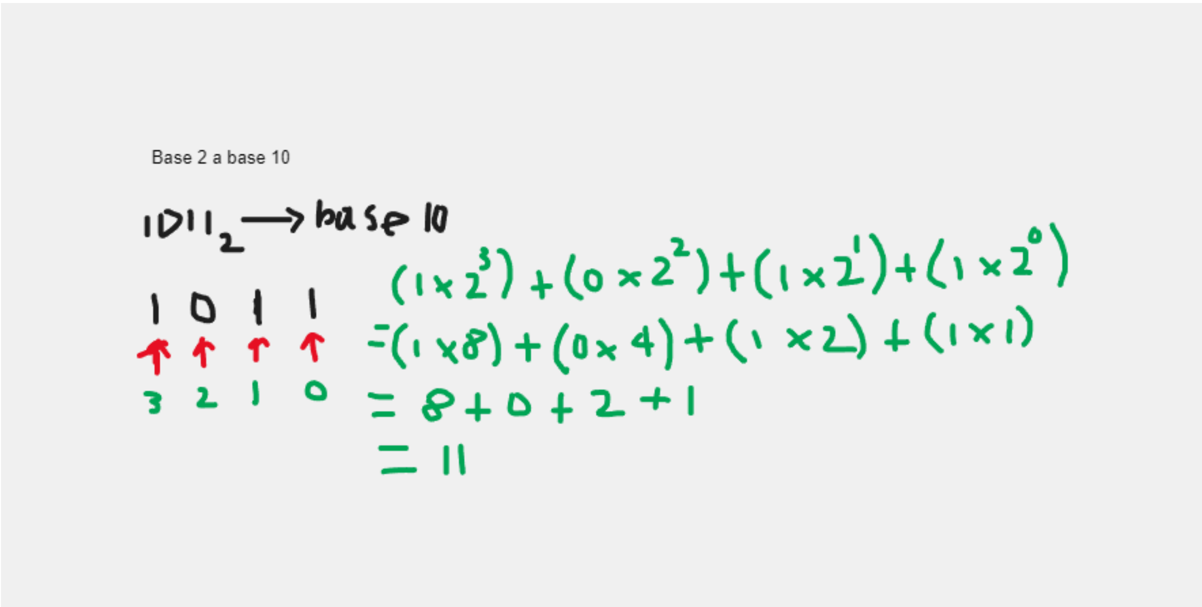
1. $2D_{16} \rightarrow base\ 2$
2. $E_{16} \rightarrow base\ 2$
3. $1ED3_{16} \rightarrow base\ 2$

Solución
1. Tomamos el número y buscamos en la tabla cada dígito hexadecimal y extraemos su equivalente binario con los cuatro dígitos: $2_{16} = 0010_2$, $D_{16} = 1101_2$. Por tanto, $2D_{16} = 101101_2$
2. $E_{16} = 1110_2$
3. Observamos que $1_{16} = 0001_2$, $E_{16} = 1110_2$, $D_{16} = 1101_2$, $3_{16} = 0011_2$. Por tanto, uniendo las cadenas de bits, obtenemos que $1ED3_{16} = 1111011010011_2$

Tablero virtual

Ilustración en clase virtual de conversiones entre bases





Operaciones con números binarios

En informática se hace útil conocer cómo se realizan distintas operaciones entre números binarios. Veamos las operaciones más comunes.

En esta sección vamos a omitir la base del número, ya que se sobreentiende que estamos trabajando con números en binario.

Suma de números binarios

Las combinaciones al sumar dos bits (sumandos) son:

1. `0 + 0 = 0`
2. `0 + 1 = 1`
3. `1 + 0 = 1`
4. `1 + 1 = 10` (en sumas grandes: se pone `0` y va `1` -**arrastre**-)

Ejemplo 1.16
Sumar los siguientes números binarios: `110` y `11`

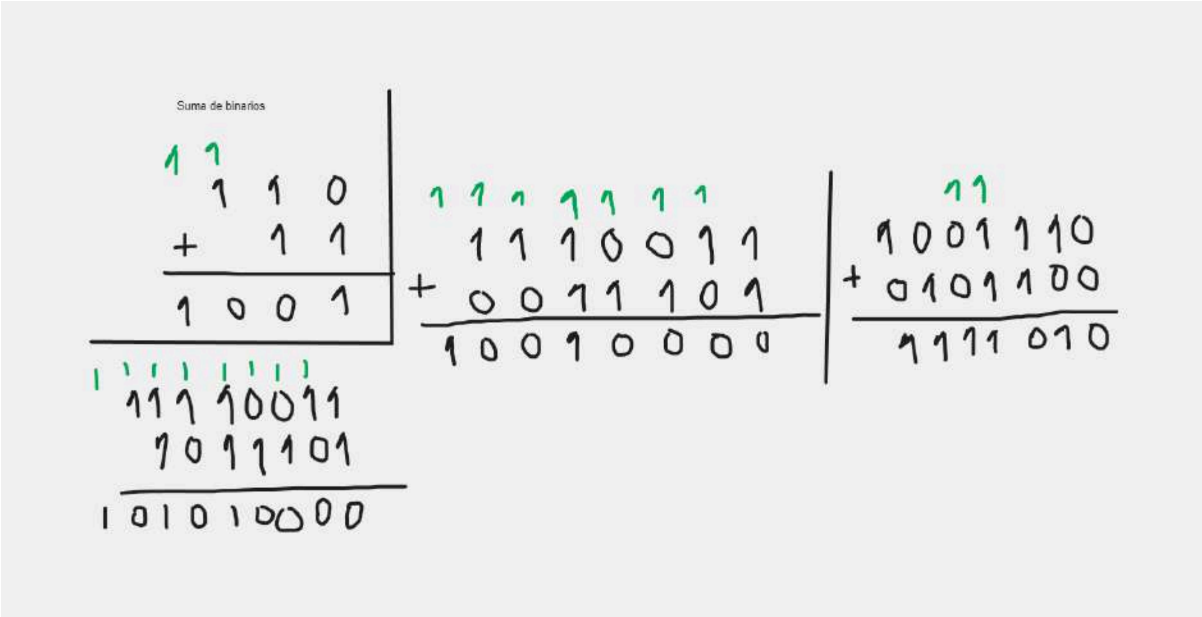
Igual que en la suma de números decimales, los disponemos en columnas

$$\begin{array}{r} 110 \\ +\ 11 \\ \hline 1001 \end{array}$$

Operamos como en el sistema decimal: comenzamos a sumar desde la derecha, en nuestro ejemplo, `1 + 1 = 10`, entonces escribimos `0` en la fila del resultado y *llevamos* `1` (este "1" se llama *arrastre*). A continuación se suma el arrastre a la siguiente columna, y seguimos hasta terminar todas las columnas (exactamente como en el sistema decimal).

Tablero virtual

Ilustración de la suma de números binarios.



Resta de números binarios

El algoritmo de la resta en binario es el mismo que en el sistema decimal. Pero conviene repasar la operación de restar en decimal para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman *minuendo*, *sustraendo* y *diferencia*.

Las restas básicas `0 - 0, 1 - 0 y 1 - 1` son evidentes:

5. `0 - 0 = 0`
6. `1 - 0 = 1`
7. `1 - 1 = 0`
8. `0 - 1` = no cabe o se pide prestado al próximo (**acarreo**).

La resta `0 - 1` se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: `10 - 1 = 1` y me llevo `1`, lo que equivale a decir en decimal, `2 - 1 = 1`. Esa unidad prestada debe devolverse sumándola a la posición siguiente.

<u>Ejemplo 1.17</u>
Restar los siguientes números binarios

1.

$$\begin{array}{r} 110 \\ -\ 11 \\ \hline 011 \end{array}$$

2.

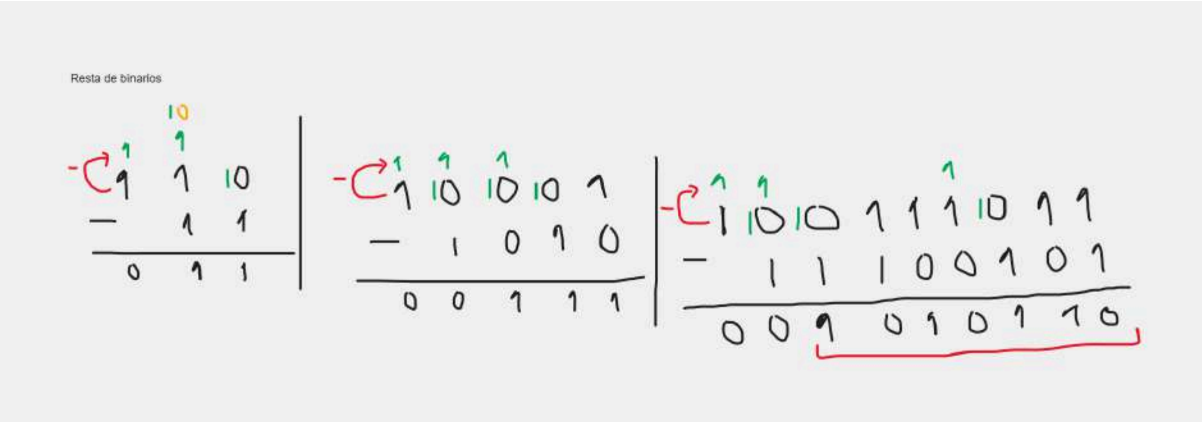
$$
\begin{array}{r}
10001 \\
-\ 01010 \\
\hline
00111
\end{array}
$$

3.

$$
\begin{array}{r}
100111011 \\
-\ 011100101 \\
\hline
001010110
\end{array}
$$

Tablero virtual

Ilustración de la resta de números binarios.



Producto de números binarios

El algoritmo del producto de números binarios es igual al usado para los números en decimal, solo que es es aún más sencillo, ya que solo involucra ceros y unos, y cualquier número multiplicado por cero, da como resultado cero.

Las combinaciones al multiplicar dos bits (factores) son:

9. `0 x 0 = 0`
10. `0 x 1 = 0`
11. `1 x 0 = 0`
12. `1 x 1 = 1`

Ejemplo 1.18

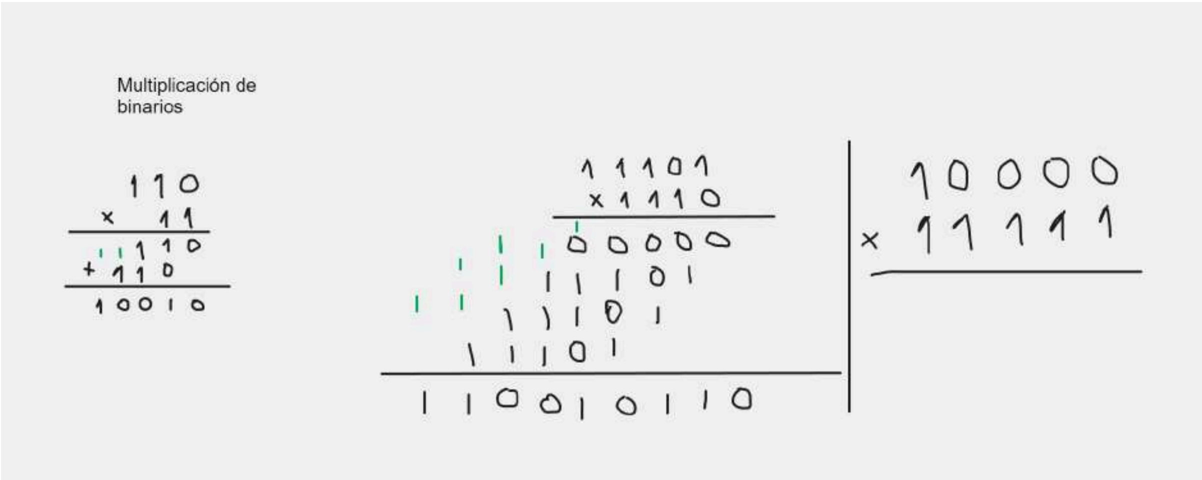
Multiplicar los números binarios `110` y `11`

$$110$$

$$\begin{array}{r} \times\ 11 \\ \hline 110 \\ +\ 110 \\ \hline 10010 \end{array}$$

Tablero virtual

Ilustración de la multiplicación de números binarios.



Preguntas

1. Describa varios sistemas de numeración antiguos y compare su practicidad
2. ¿Los sistemas de numeración antiguos hubieran permitido la evolución de las matemáticas?
3. ¿Qué es un sistema de numeración?
4. ¿Cuáles son los sistemas numéricos usados en informática?
5. ¿Cuál es el sistema numérico predominante en los dispositivos electrónicos, entre ellos el computador?
6. ¿Cuáles son los pasos para convertir de una base a otra?
7. ¿Cuál es el sistema de numeración más importante?
8. ¿Qué resultado entrega la siguiente expresión? $11101111_2 - 10011110_2 <= 7A_{16}$
9. ¿Cómo se aplican los sistemas numéricos en las ciencias computacionales?
10. ¿Qué importancia tiene la lógica matemática en la informática? ¿Cómo se aplica ésta en computación?

Ejercicios

1. Convertir de base 10 a base 2, 8 y 16 los siguientes números y luego realice el proceso contrario con los números encontrados de pasarlos nuevamente a base 10

   a. 119
   b. 40
   c. 66
   d. 364
   e. 77
2. Convertir $14C_{16} \rightarrow$ base 8
3. Convertir $65_8 \rightarrow$ base 2
4. Convertir $1100011111_2 \rightarrow$ base 16
5. Convertir $F98A_{16} \rightarrow$ base 2
6. Teniendo presente el código ASCII[2], escriba (cifre) su nombre en binario, octal, decimal y hexadecimal combinando mayúsculas y minúsculas. Especifique una manera adecuada de separar las cadenas numéricas
7. Si $m = 11101_2$, $n = 10011_2$, $p = 1111_2$, son números binarios, encuentre:
   a. m + n
   b. n + p
   c. m + n + p
   d. m * n
   e. m * n * p
   f. $m^2 - n^2$
   g. m - n
   h. (m + p) * (n - p)
   i. m - n - p
   j. (m + n) - (m + p)
8. Realice las siguientes operaciones con los números binarios
   1100011111 + 1111101000
   1111101000 - 1100011111
   1100011111 x 1111101000
   1100011111 + 1111101000 + 1011101010 + 11100000
   Escriba estos números en octal, decimal y hexadecimal

[2] Ver por ejemplo: [ASCII - Wikipedia, la enciclopedia libre]

Capítulo 2. Lógica proposicional

La lógica proposicional es producto del intento de la humanidad por sistematizar el conocimiento. Esta línea de la lógica matemática ha permitido fortalecer el razonamiento lógico y ha sido aplicada en la comunicación humana.

Técnicas de demostración matemática

Antes de entrar en detalle con la lógica proposicional, veamos las técnicas de demostración existentes en matemáticas, las cuales son una herramienta fundamental no solo en esta disciplina, sino también en las ciencias naturales y aplicadas.

Terminología usada en la demostración matemática

Proposición

Es un enunciado del cual se puede afirmar con exactitud si es falso o verdadero.

Axioma

Es una proposición que por lo evidente, no requiere demostración. También se puede decir acerca del axioma, que es toda proposición que no es deducida de otras.

Postulado

Es una proposición no evidente por sí misma ni demostrada, pero que se acepta como verdadera, ya que no existe otro principio al que pueda ser referida. En otras palabras, es una proposición cuya verdad se admite sin pruebas y que es necesaria como base en razonamientos ulteriores.

Demostración

Una demostración o prueba en matemáticas, y que es una herramienta fundamental en esta disciplina así como en las ciencias naturales y la ingeniería, es un argumento deductivo para asegurar la verdad de una proposición. En la argumentación se pueden usar otras afirmaciones previamente establecidas, tales como teoremas, definiciones, las afirmaciones iniciales (hipótesis) y los axiomas y postulados aceptados.

Existen básicamente cuatro formas de demostración en matemáticas, a saber: directa, reducción al absurdo, inducción matemática y el contraejemplo. Veamos primero algunos conceptos utilizados en la jerga matemática para luego ver en general cada tipo de demostración. En distintas partes del texto se ilustran estos métodos con ejemplos.

Teorema

Es una proposición verdadera que ha sido demostrada previamente por medio de axiomas, postulados y otros teoremas ya demostrados.

Hipótesis

Son las suposiciones iniciales o premisas que se consideran verdaderas. Es el punto de partida que se considera válido cuando se va a demostrar un teorema.

Tesis o conclusión

Es la proposición que se quiere demostrar como verdadera.

Nota
La demostración no solo usa lo que entrega la hipótesis, en los pasos lógicos y reglas de inferencia que se apliquen, es posible utilizar proposiciones que ya han sido demostradas (teoremas)[3], así como los axiomas y postulados aceptados. No sobra decir que una demostración no debería utilizar resultados de otros teoremas que aún no han sido demostrados.

Técnicas de demostración matemática

Técnica de demostración: Demostración directa

Se trata, a partir de una hipótesis y los conocimientos de los axiomas, postulados, teoremas, leyes, etc. existentes, poder probar la veracidad de la tesis propuesta, utilizando una secuencia de pasos lógicos y reglas de inferencia válidas.

Técnica de demostración: Reducción al absurdo (*Reductio ad absurdum*)

También conocida como demostración por **contradicción**, se asume como verdad la negación o falsedad de la proposición que se quiere demostrar, y siguiendo reglas y argumentos lógicos, se busca llegar a una contradicción de lo que se asumió como verdad, lo cual permite inferir que la proposición original era verdadera.

Técnica de demostración: Inducción matemática

El método de inducción matemática es una técnica para demostrar proposiciones que involucran a los números naturales. Consiste en dos pasos: demostrar que la proposición es verdadera para el primer número natural (generalmente 1) y luego demostrar que si la proposición es verdadera para un número natural cualquiera, también lo es para el siguiente número natural.

---

[3] Un teorema es una proposición que se ha demostrado que es verdadera, mientras que una proposición es una afirmación que puede ser verdadera o falsa.

Pasos de la demostración por inducción:

1. Base de la inducción: se verifica que la proposición es verdadera para el primer número natural (generalmente $n = 1$).
2. Hipótesis inductiva: se asume que la proposición es verdadera para un número natural arbitrario k (donde $k \geq 1$). Es decir, se asume que $P(k)$ es verdadera.
3. Paso inductivo: se demuestra que si la proposición es verdadera para k, entonces también es verdadera para $k + 1$. Esto significa probar que $P(k)$ implica $P(k+1)$.

Si se cumplen estos tres pasos, se puede concluir que la proposición es verdadera para todos los números naturales.

Cada una de las condiciones 1 y 2 tiene su propio significado especial. La condición 1 proporciona la base para la inducción; la condición 2 proporciona la justificación para generalizar a partir de esta base, o sea, la justificación para pasar de un caso especial hacia el siguiente, de n hacia $n + 1$. Si no se satisface la condición 1, entonces no existe base para aplicar el método de la inducción matemática, aún si se satisface la condición 2. Por otra parte, cuando sólo se satisface la primera condición y no la segunda, aunque se proporcione una base para la inducción, no existe justificación para hacer la generalización.

La inducción tiene amplias aplicaciones en las matemáticas, pero debe usarse con cuidado o puede conducir a conclusiones erróneas. En los siguientes ejemplos se ilustra cómo usar esta técnica de demostración.

Ejemplo

Demostrar que la suma de los primeros n números naturales es igual a $1 + 2 + 3 + ... + n = n * (n + 1) / 2$.

Demostración

Probemos inicialmente que la proposición es igual para $n = 1$: La suma de los números de 1 a 1 es 1 y con la fórmula se verifica fácilmente que: $1 * (2 + 1) / 2 = 1 * 2 / 2 = 2 / 2 = 1$.

Ahora supongamos que la proposición se cumple para $n = k$: $1 + 2 + 3 + ... + k = k * (k + 1) / 2$.

Demostremos que la suma de los $k + 1$ es $(k + 1) * (k + 1 + 1) / 2 = (k + 1) * (k + 2) / 2 = (k + 1) * (k / 2 + 1)$, con lo que se demuestra que si la proposición es verdadera para $n = k$ lo será también para $n = k + 1$.

Se tiene que:

$1 + 2 + 3 + ... + k + (k + 1) = k * (k + 1) / 2 + (k + 1)$

Factorizando el miembro derecho de la expresión anterior, obtenemos

$(k + 1) * (k / 2 + 1)$

Con lo cual se demuestra que la proposición es verdadera para cualquier número natural.

Ejemplo
La sucesión de los números impares naturales está dada por $1, 3, 5, 7, 9, ...$

Encontrar la fórmula que exprese el número impar u_n en términos del índice n y la validez de dicha expresión en general.

Demostración
Tomemos el primer número de esta sucesión como u_1, el segundo como u_2, etc. así:
$u_1 = 1, u_2 = 3, u_3 = 5, u_4 = 7, u_5 = 9, ...$

$u_1 = 2 * 1 - 1 = 1$
$u_2 = 2 * 2 - 1 = 3$
$u_3 = 2 * 3 - 1 = 5$
$u_4 = 2 * 4 - 1 = 7$
$u_5 = 2 * 5 - 1 = 9$
…
$u_n = 2 * n - 1$

Veamos que la expresión $u_n = 2n - 1, n \in \mathbf{N}$ para encontrar cualquier número impar natural es válida en general.

Ya vimos que la fórmula se cumple para $n = 1, u_1 = 2 * 1 - 1 = 1$

Supongamos que la expresión se cumple para $n = k, u_k = 2 * k - 1, k \in \mathbf{N}$

Probemos que la expresión se cumple también para $n = k + 1$: $u_{k+1} = 2 * (k + 1) - 1 = 2 * k + 2 - 1 = 2 * k + 1$

Para obtener el $(k + 1)$-ésimo número impar basta con sumar 2 al k-ésimo número impar, el cual por hipótesis está dado por $u_k = 2 * k - 1$. Por tanto:
$u_{k+1} = (2 * k - 1) + 2 = 2 * k + 1$ y se concluye que $u_n = 2 * n - 1$

La inducción matemática puede aplicarse a los números enteros. Aunque comúnmente se asocia con los números naturales (enteros positivos), el principio de inducción matemática puede extenderse para probar afirmaciones sobre cualquier conjunto bien ordenado, incluyendo los enteros. La clave es adaptar el paso base y el paso inductivo para que sean válidos dentro del conjunto de enteros.

El principio de inducción matemática se basa en el concepto de que si una propiedad es verdadera para un caso base y si, asumiendo que es verdadera para un valor cualquiera, también es verdadera para el siguiente valor, entonces la propiedad es verdadera para todos los valores en el conjunto.

Ejemplo
Si n es un número entero, entonces $2n + 1$ es un número impar.

Demostración
Tomemos como caso base $n = 0$: $2(0) + 1 = 1$, que es un número impar
Supongamos que para $n = k$ se tiene que $2k + 1$ es un número impar
Probemos que para $n = k + 1$ se cumple que $2(k + 1) + 1$ es un número impar
Resolviendo
$2(k + 1) + 1 = 2k + 2 + 1 = 2k + 3$
Por hipótesis sabemos que $2k + 1$ es impar, por tanto, si sumamos 2 a este número, obtenemos el siguiente número impar: $2k + 1 + 2 = 2k + 3$
De esto concluimos que $2n + 1$ es un número impar

Ejemplo
La suma de los primeros n naturales impares es igual a n^2.
Por ejemplo, si $n = 4$: $1 + 3 + 5 + 7 = 16 = 4^2$.

Demostración
Para $n = 1$ se tiene que $1^2 = 1$.
Supongamos que si $n = k$ se cumple que la suma de los primeros k naturales es k^2.
Probemos que si $n = k + 1$ se cumple que la suma de los primeros $k + 1$ naturales es $(k + 1)^2$. Hasta el k-ésimo elemento la suma es k^2; basta con sumar 2 a este número para obtener el (k + 1)-ésimo elemento y sumarlo, pero $u_k = 2k - 1$ como se vio anteriormente, por tanto $k^2 + (2k - 1 + 2) = k^2 + 2k + 1 = (k + 1)^2$. Esto demuestra que la suma de los primeros n naturales impares es igual a n^2.

Técnica de demostración: Contraejemplo

Se trata de mostrar con un caso particular que la generalidad de una afirmación no es cierta, esto es, es falsa.

Ejemplo 2.
Afirmación: "Todos los números primos son impares"

Esta afirmación se puede contradecir fácilmente mostrando que el 2 es un número primo que no es impar. Este contraejemplo derrumba la afirmación anterior, por lo que la convierte en una afirmación falsa.

Paradojas matemáticas[4]

Se llaman paradojas matemáticas a ciertos que conllevan a resultados falsos y que parecen deducirse de razonamientos rigurosos, pero durante los cuales se efectuaron operaciones sin sentido o erróneo.

Ejemplo 0.5
Demostrar que $1 = 2$.

Demostración

Sean $a,\ b$ dos números tales que $a = b$. Ahora, multipliquemos ambos miembros de esta ecuación por a:

$a^2 = ab$

Restemos en ambos miembros de esta nueva ecuación b^2:

$a^2 - b^2 = ab - b^2$

Factorizando:

$(a + b)(a - b) = b(a - b)$

Ahora, dividamos ambos miembros de esta ecuación por $(a - b)$, para lo cual obtenemos:

$(a + b) = b$. Como $a = b$, podemos sustituir: $(b + b) = b \Rightarrow 2b = b$.

Y dividiendo por b: $2 = 1$

Todo parece muy bien, pero dentro de las operaciones realizadas hubo un error lógico que pasó inadvertido, el cual se presenta cuando divide ambos miembros de la ecuación por $(a - b)$, operación que da 0 pues $a = b$, y sabemos que la división por 0 carece de sentido.

Este razonamiento incorrecto nos lleva por tanto a un resultado **absurdo**, y que nos dice acerca del cuidado riguroso que se debe tener al validar un enunciado, teorema, teoría o ley científica.

Proposiciones

Una *proposición* es un enunciado del cual se puede afirmar si es *falso* (**f**) o *verdadero* (**v**). Esto significa que es una afirmación que tiene un *valor de verdad* que puede ser **f** ó **v.**

Valores de verdad, constantes lógicas o booleanas

Un valor de verdad indica en qué medida una afirmación es una verdad o una falsedad. En lógica bivalente, como la lógica proposicional o booleana, un valor de verdad representa el estado de una proposición o variable, y qué está dado por una constante lógica (booleana) que tiene uno de dos valores: *Verdadero* (**V** o **1**) o *Falso* (**F** o **0**).

[4] Tomado de: Malba Tahan. El Hombre que Calculaba. Romance. Aventuras de un singular calculista persa. Segunda edición ampliada. Ed. Camacho Roldán. Bogotá, Colombia

<u>Ejemplo 2.1</u>
Indique cuáles de los siguientes enunciados son proposiciones, si lo son, diga cual es su valor de verdad.

1. (?) Hoy vamos a ir a estudiar
2. (F) 9 es un número primo
3. (V) El tablero es blanco
4. (V) 8 = 4 * 2
5. (?) ¿Qué día es hoy?
6. (?) El viernes vamos a cine
7. (V) 4 < 10
8. (?) 10 / 5
9. (?) Tengo hambre
10. (V) 7 es un número primo y es impar

Las proposiciones se representan generalmente con letras minúsculas del alfabeto latino para simplificar las operaciones. Se acostumbra usar las letras ***p, q, r, s, …***; así, podemos representar las proposiciones del ejemplo anterior de esta forma:

1. p: 9 es un número primo
2. q: El tablero es blanco
3. r: 8 = 4 * 2
4. s: 4 < 10
5. t: 7 es un número primo y es impar

Las proposiciones pueden ser *simples* o *compuestas*. Una proposición compuesta está formada por dos o más proposiciones simples unidas por *conectivos lógicos*.

En el caso del ejemplo anterior, las proposiciones ***p, q, r, s*** son proposiciones simples, mientras que la proposición ***t*** es compuesta, ya que está formada por dos proposiciones simples unidas por el conectivo ***y***.

Operadores, conectivos y símbolos lógicos

Son símbolos usados en la lógica proposicional que permiten *unir* o *conectar* proposiciones simples. Todos los conectivos lógicos (*operadores lógicos*) son *binarios*, esto es, requieren de dos proposiciones (*operandos*) para ser usados, a excepción de la negación, la cual es una operación *unaria*. Los conectivos lógicos son usados frecuentemente en informática y se les conoce como *operadores lógicos o booleanos* (como se vio anteriormente); se muestran además símbolos de la lógica proposicional usados regularmente:

Nombre Operador	**Símbolo**
Negación lógica unaria	$\neg$ $\sim$ $-$! ' no not
Conjunción	$\wedge$ && $\times$ $\bullet$ y and
Disyunción	$\vee$ || $+$ o or
Disyunción exclusiva	$\underline{\vee}$ $\oplus$ W $'o\ bien'$ xor eor

Condicional	$\rightarrow$	$\Rightarrow$	$\supset$		
Bicondicional	$\leftrightarrow$	$\Leftrightarrow$	$\equiv$		
Tautología	$\top$	T	1	*Tautología*	
Contradicción	$\perp$	F	0	*Contradicción*	
Valor de verdad verdadero	v	t	1	*Verdadero*	*True*
Valor de verdad falso	f	f	0	*Falso*	*False*

Operaciones lógicas y tablas de verdad

Una ***operación lógica*** es una expresión que involucra proposiciones y operadores entre éstos y que devuelve un valor de verdad para cada combinación dada de valores de verdad de cada proposición. Las ***tablas de verdad***, también llamadas *tablas de valores de verdad*, muestran las posibles combinaciones de los valores de verdad para las proposiciones compuestas de una expresión lógica.

Una proposición tiene dos posibles valores de verdad: **f** o **v**; así, combinamos los posibles casos para varias proposiciones al usar los conectivos lógicos en una tabla de verdad siguiendo las reglas que se definen para cada uno.

Sean $p,\ q$ dos proposiciones. Veamos las tablas de verdad para cada conectivo lógico, esto es, para cada operación lógica que involucre un operador lógico.

Número de combinaciones

Partiendo del número de variables (proposiciones) involucradas en una expresión lógica, podemos saber por teoría combinatoria cuántas combinaciones son posibles de acuerdo a los valores de verdad que pueda tomar cada variable.

Si tenemos ***n*** variables, el número combinaciones es 2^n. Así, determinamos fácilmente el número de filas que tendrá la tabla de verdad. El número de columnas estará dado por el número de variables, más la operación (u operaciones), es decir, la tabla tendrá *n + 1* columnas para el caso de evaluar un operador.

Construir la tabla de verdad

Teniendo en cuenta el número de combinaciones, dividimos entre dos dicho número para la primera columna; el resultado lo dividimos entre dos para la segunda columna; procedemos igualmente para la tercera columna dividiendo entre dos el resultado anterior, y así sucesivamente con las demás. En la primera división ubicamos los valores de verdad “verdaderos”, y en la segunda división ubicamos los valores de verdad “falsos”, aplicados a cada columna.

Negación

$\neg p$ es la negación de *p* y se lee “*no p*”. La negación es verdadera si la proposición es falsa, y es falsa si la proposición es verdadera.

p	$\neg p$

v	f
f	v

Conjunción

$p \wedge q$ es la conjunción entre *p, q* y se lee "*p y q*". La conjunción es verdadera solo si ambas proposiciones son verdaderas

p	q	$p \wedge q$
v	v	v
v	f	f
f	v	f
f	f	f

Disyunción

$p \vee q$ es la disyunción entre *p, q* y se lee "*p ó q*". La disyunción es verdadera si alguna de las proposiciones es verdadera

p	q	$p \vee q$
v	v	v
v	f	v
f	v	v
f	f	f

Disyunción exclusiva

$p \oplus q$ es la disyunción exclusiva entre *p, q* y se lee "*p o bien q*". La disyunción exclusiva es verdadera sólo si una de las proposiciones es verdadera; en otras palabras, es falsa si ambas proposiciones tienen el mismo valor de verdad.

p	q	$p \oplus q$
v	v	f
v	f	v

f	v	v
f	f	f

Condicional o implicación

$p \rightarrow q$ es el condicional o implicación entre *p, q* y se lee "*si p entonces q*". En el condicional, **p** se conoce como *hipótesis* y **q** como la *conclusión*. El condicional es falso sólo si la hipótesis es verdadera y la conclusión es falsa.

p	q	$p \rightarrow q$
v	v	v
v	f	f
f	v	v
f	f	v

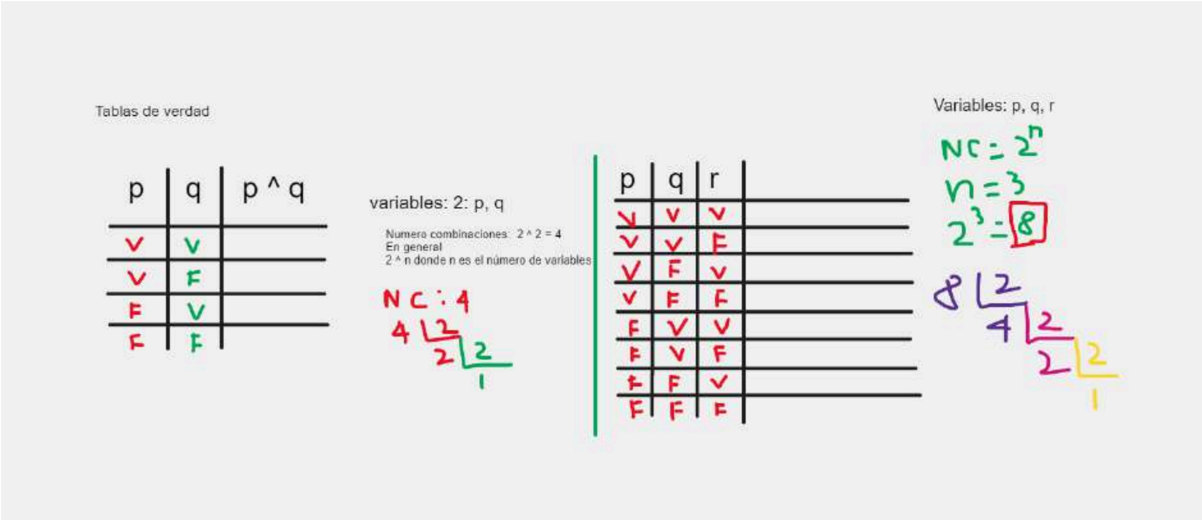
Bicondicional o equivalencia

$p \leftrightarrow q$ es el bicondicional o equivalencia entre *p, q* y se lee "*p sí y sólo si q*". El bicondicional es verdadero si ambas proposiciones tienen el mismo valor de verdad.

p	q	$p \leftrightarrow q$
v	v	v
v	f	f
f	v	f
f	f	v

Tablero virtual

Ilustración de cómo construir la tabla de verdad.



Ejemplo 2.2
Crear la tabla de verdad para las siguientes expresiones

1. Demostrar usando tablas de verdad que el bicondicional es un doble condicional: $p \leftrightarrow q = (p \rightarrow q) \wedge (q \rightarrow p)$
2. $p \rightarrow (q \vee \neg r) \wedge r$

Solución

1.
Número de variables n: 2
Número de combinaciones: $2^2 = 4$

p	q	$(p \rightarrow q)$	$\wedge$	$(q \rightarrow p)$
v	v	v	v	v
v	f	f	f	v
f	v	v	f	f
f	f	v	v	v

2.
Número de variables n: 3
Número de combinaciones: $2^3 = 8$

$p \rightarrow (q \vee \neg r) \wedge r$

p	q	r	$\rightarrow$	$\vee$	$\neg r$	$\wedge$
v	v	v	v	v	f	v
v	v	f	f	v	v	f
v	f	v	f	f	f	f

v	f	f	f	v	v	f
f	v	v	v	v	f	v
f	v	f	v	v	v	f
f	f	v	v	f	f	f
f	f	f	v	v	v	f

Tautología

Es una proposición compuesta que siempre es ***verdadera***, sin importar el valor de verdad que tengan las proposiciones simples que la componen.

Las tautologías tienen un papel preponderante en las matemáticas, ya que constituyen la mayoría de las leyes de la lógica.

Ejemplo 2.3
Comprobar que $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$ es una tautología

p	q	r	$[(p \rightarrow q)$	$\wedge$	$(q \rightarrow r)]$	$\rightarrow$	$(p \rightarrow r)$
v	v	v	v	v	v	v	v
v	v	f	v	f	f	v	f
v	f	v	f	f	v	v	v
v	f	f	f	f	v	v	f
f	v	v	v	v	v	v	v
f	v	f	v	f	f	v	v
f	f	v	v	v	v	v	v
f	f	f	v	v	v	v	v

Contradicción

Es una proposición compuesta que siempre es ***falsa***, sin importar el valor de verdad que tengan las proposiciones simples que la componen.

Ejemplo 2.4
Comprobar que $[p \wedge (p \rightarrow q)] \wedge \neg q$ es una contradicción

p	q	$\wedge$	$(p \rightarrow q)$	$\wedge$	$\neg q$
v	v	v	v	f	f

v	f	f	f	f	v
f	v	f	v	f	f
f	f	f	v	f	v

Indeterminación

Es una proposición compuesta cuya tabla de verdad está formada por resultados tanto verdaderos como falsos. También se conoce como ***incertidumbre***.

Leyes de la lógica proposicional

Son un conjunto de tautologías que nos permiten simplificar y demostrar expresiones lógicas. También se conocen como *leyes del álgebra proposicional*.

Para comprobar que una proposición compuesta es una tautología, se utilizan varios métodos, entre ellos:

- Tablas de verdad
- Álgebra de proposiciones (leyes de la lógica proposicional -*Tautologías Notables*-)
- Método directo
- Método indirecto: exploración, contradicción, contrarrecíproco

Veamos las leyes de la lógica proposicional para las proposiciones lógicas ***p, q, r***

Leyes de idempotencia para la conjunción y la disyunción

1. $p \wedge p \Leftrightarrow p$
2. $p \vee p \Leftrightarrow p$

Leyes de identidad para la conjunción y la disyunción

1. $p \wedge (v) \Leftrightarrow p$
2. $p \wedge (f) \Leftrightarrow f$
3. $p \vee (v) \Leftrightarrow v$
4. $p \vee (f) \Leftrightarrow p$

Leyes conmutativas para la conjunción y la disyunción

1. $(p \wedge q) \Leftrightarrow (q \wedge p)$
2. $(p \vee q) \Leftrightarrow (q \vee p)$

Leyes asociativas para la conjunción y la disyunción

1. $p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$
2. $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$

Leyes distributivas para la conjunción y la disyunción

1. $[p \wedge (q \vee r)] \Leftrightarrow [(p \wedge q) \vee (p \wedge r)]$
2. $[p \vee (q \wedge r)] \Leftrightarrow [(p \vee q) \wedge (p \vee r)]$

Ley de la doble negación

1. $\neg(\neg p) \Leftrightarrow p$

Ley del tercero excluido o del inverso de la disyunción

1. $p \vee \neg p \Leftrightarrow (v)$

Ley de contradicción o del inverso de la conjunción

1. $p \wedge \neg p \Leftrightarrow (f)$

Leyes de De Morgan

1. $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$
2. $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$

Ley del Modus Ponendo Ponens

1. $p \wedge (p \rightarrow q) \rightarrow q \; es \; una \; tautología$

Ley del silogismo

$[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$

Caracterización de la implicación (forma alternativa del condicional)

1. $p \rightarrow q \Leftrightarrow \neg p \vee q$

Caracterización de la equivalencia (forma alternativa del bicondicional)

1. $p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$

Simetría del bicondicional

1. $p \leftrightarrow q \Leftrightarrow q \leftrightarrow p$

Contrarrecíproco

1. $p \to q \Leftrightarrow \neg q \to \neg p$

Leyes de absorción

1. $p \wedge (p \vee q) \Leftrightarrow p$
2. $p \vee (p \wedge q) \Leftrightarrow q$

Ejemplo 2.5
Aplicar las leyes de la lógica para probar los siguientes enunciados

Sean p, q y r proposiciones; demostrar:

1. $\neg(p \to q) \Leftrightarrow (p \wedge \neg q)$
2. $(p \wedge q) \to p\ es\ una\ tautología$
3. $[(p \to q) \wedge \neg q] \to \neg p\ es\ una\ tautología$
4. $\neg[\neg(\neg p \wedge q) \vee (\neg p \vee q)\ es\ una\ contradicción$

Preguntas

5. ¿Cuál es el valor de verdad de esta expresión? $(3 - 5^2 \times 2 > 7 + 4 \div 2) \vee (9 \div 3 \times 4^3 < 7 \times 8 - 20 \div 5)$
6. ¿En lógica proposicional, qué es una tautología, una contradicción y una indeterminación?
7. ¿Qué métodos de demostración existen en lógica proposicional para saber si una proposición es una tautología, una contradicción y una indeterminación?
8. Indicar si la siguiente expresión es una tautología, contradicción o incertidumbre: $(v)\ Y\ (No\ (f)\ O\ (f))$
9. Indicar si la siguiente expresión es verdadera, falsa o no tiene sentido: $(f)\ O\ No\ (No\ (''a'' > ''b''))\ Y\ (5\ {*}{-}\ 8 < 0)$
10. Determine el valor de verdad de la siguiente expresión si: $a = (f),\ b = (v),\ c = (v);\ No(a)\ Y\ No(b\ O\ (c\ \oplus\ No(a)))$
11. Falso o verdadero; justifique. Sabiendo que $p \vee (q \to r)$ es falsa, entonces el valor de verdad de $\neg[p \wedge (q \to \neg r)]$ es verdadero
12. Falso o verdadero; justifique. Sabiendo que r es falsa y $\neg p \wedge (q \to r)$ es verdadera, entonces el valor de verdad de $r \to (\neg q \vee p)$ es falso
13. Falso o verdadero; justifique. Si la proposición $\neg q \to p$ es falsa, entonces la proposición $p \to \neg q$ es verdadera
14. Falso o verdadero; justifique. La contrarrecíproca de la proposición $\neg m \to r$ es $r \to \neg m$

Ejercicios

1. Cree la tabla de verdad para las siguientes expresiones lógicas si *p*, *q*, *r* y *s* son proposiciones
   $[p \wedge (q \rightarrow r)] \leftrightarrow (\neg q \vee p)$
   $[(p \vee \neg s) \wedge (q \rightarrow p)] \rightarrow (\neg r \leftrightarrow s)$
   $(\neg p \oplus r) \oplus (\neg q \leftrightarrow p)$
   $q \rightarrow \neg(r \rightarrow p)$
   $[(p \wedge r) \leftrightarrow (q \otimes p)] \vee \neg(p \rightarrow \neg q)$
2. Demuestre las siguientes equivalencias usando tablas de verdad para las proposiciones *p*, *q* y *r*
   $\neg(p \leftrightarrow q) \Leftrightarrow \neg p \leftrightarrow \neg q$
   $(p \oplus q) \wedge r \Leftrightarrow (p \wedge r) \oplus (q \wedge r)$
3. Escriba las siguientes expresiones lógicas de tal manera que no contengan condicionales ni bicondicionales y cree la respectiva tabla de verdad para verificar los resultados para las proposiciones *p*, *q* y *r*
   $(p \rightarrow q) \wedge (q \leftrightarrow r)$
   $\neg p \rightarrow [q \vee (p \wedge r)]$
   $(p \wedge q) \leftrightarrow (p \vee q)$
   $(p \rightarrow q) \wedge (p \rightarrow r)$
   $(p \leftrightarrow q) \wedge (q \leftrightarrow r)$
4. Aplique las leyes del álgebra de proposiciones para probar las siguientes expresiones si *p*, *q* y *r* son proposiciones:
   $\neg(p \rightarrow q) \Leftrightarrow (p \wedge \neg q)$
   $(p \wedge q) \rightarrow p\ es\ una\ tautología$
   $[(p \rightarrow q) \wedge \neg q] \rightarrow \neg p\ es\ una\ tautología$
   $\neg[\neg(\neg p \wedge q) \vee (\neg p \vee q)]\ es\ una\ contradicción$
   $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)\ es\ una\ tautología$
   $[p \wedge (p \rightarrow q)] \wedge \neg q\ es\ una\ contradicción$
5. Simplifique las siguientes expresiones aplicando las leyes del álgebra proposicional todo lo que sea posible:
   $[\neg q \wedge (p \rightarrow q)] \rightarrow \neg p$
   $[(p \rightarrow q) \wedge \neg q] \rightarrow \neg p$
   $[p \rightarrow (q \rightarrow r)] \rightarrow [(p \wedge q) \rightarrow r]$
   $[\neg p \wedge (p \vee q)] \rightarrow q$
   $p \rightarrow (p \vee q)$
   $\neg(p \rightarrow \neg q) \wedge (\neg p \wedge q)$
   $p \rightarrow (p \wedge \neg q)$
   $\neg m \wedge (\neg m \rightarrow \neg n)$
   $\neg[t \rightarrow (m \wedge t)]$
   $\neg[(p \wedge q) \rightarrow (\neg p \rightarrow q)]$
6. Cree las tablas de verdad para las expresiones de los puntos anteriores y verifique que los resultados sean consistentes con lo encontrado allí.

Capítulo 3. Conjuntos

Antes de entrar en materia con las definiciones de conjuntos, veamos primero qué es un axioma y su importancia en las matemáticas.

Definición: Axioma
Es una proposición que se considera, por lo evidente, verdadera, y que no requiere demostración.

Los axiomas son fundamentales en las matemáticas, y en la ciencia en general, ya que son el punto de partida para poder desarrollar modelos científicos. A partir de ellos se construyen teoremas, los cuales agrupados en un conjunto, permiten formular teorías. Los axiomas son similares a las reglas de un juego, se deben aceptar. Muchos axiomas en ciencias no necesariamente son evidentes, pero se eligen por conveniencia.

Si un axioma conduce a deducir otro, significa que no es axioma; un axioma tampoco puede llevar a contradicción en la teoría que se desarrolla.

Conjunto

El concepto de ***conjunto*** es muy utilizado en matemáticas, pero *no se define*, así como no se definen otros conceptos como ***punto***, ***recta*** o ***plano***, por ejemplo. Se suelen usar sinónimos: *agrupación, clase, reunión, colección*.

Los conceptos matemáticos que no se definen se conocen como *primitivos*, y entre ellos también se encuentra ***elemento*** y ***pertenencia***.

Un *conjunto* puede o no contener *elementos*, y éstos no se repiten, esto es, todos son distintos; se les suele nombrar con letras minúsculas, mientras que las letras mayúsculas se usan para los nombres de los conjuntos.

El orden de los elementos en el conjunto no es relevante en general.

Un conjunto puede ser *finito* si se conoce el número de elementos que contiene, en caso contrario se dice que es un conjunto *infinito*.

Representación de conjuntos

Hay dos formas de representar un conjunto:

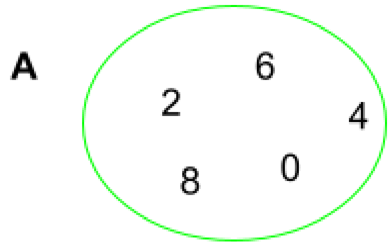
- Diagramas de Venn
- Elementos encerrados entre llaves

Ejemplo 3.1
Sea el conjunto **A** formado por los números pares del 0 al 8.

El conjunto es finito y es fácil de representar en cualquier forma:

Representación con llaves
$A = \{0, 2, 4, 6, 8\}$

Diagrama de Venn



Cuando el número de elementos es muy grande, así sea un número finito, el diagrama de Venn deja de ser práctico y conviene utilizar la notación de llaves, poniendo puntos suspensivos.

Ejemplo 3.2
Sea **B** el conjunto de los números naturales del 1 al 100

$B = \{1, 2, 3, 4, 5,..., 100\}$

Si el conjunto es infinito, se puede utilizar esta forma escribiendo un número significativo de elementos de tal forma que permitan observar la regla que cumplen éstos.

Ejemplo 3.3
Sea C el conjunto formado por los positivos múltiplos de 3

$C = \{3, 6, 9, 12, 15, 18, 21,...\}$

Esta forma de escribir los conjuntos, se conoce por ***extensión***. La otra forma de escribir un conjunto es por ***comprensión***, que veremos más adelante.

Conjunto vacío

En un conjunto que carece de elementos. Se simboliza de cualquiera de las siguientes formas:

- $\varnothing$
- $\{\}$

Cardinalidad de un conjunto | |

Es el número de elementos de un conjunto.

Ejemplo 3.4
$Sea\ A = \{a,\ b, c, d, e\}$
$|A| = 5$

Símbolos comunes en notación de conjuntos

Como vimos en la Unidad 0, existen varios símbolos muy utilizados en el tratamiento de conjuntos. Veamos de nuevo cuales son:

$\in$: $Pertenece.\ Ejemplo:\ a \in A.\ Se\ lee:\ 'a'\ pertenece\ (es\ elemento\ de)\ a\ 'A'$
$\notin$: $No\ pertenece.\ Ejemplo:\ a \notin A.\ Se\ lee:\ 'a'\ no\ es\ elemento\ 'A'$
$\forall_x$: $Cuantificador\ universal. Ejemplo: \forall_x\ x > 9.\ Se\ lee:\ Para\ todo\ x\ se\ cumple\ que\ x\ es\ mayor\ que\ 9$
$\exists_x$: $Cuantificador\ existencial.\ Ejemplo:\ \exists_x x > 9.\ Se\ lee:\ Existe\ al\ menos\ un\ x\ tal\ que\ x\ es\ mayor\ que\ 9$
$\exists!_x$: $Cuantificador\ existencial$ único. $Se\ lee:\ Existe\ exactamente\ un\ x$
$\subset$: $Contenido\ en,\ o\ es\ subconjunto\ de.\ Ejemplo:\ A \subset B.\ Se\ lee:\ A\ es\ subconjunto\ de\ B$
$\subseteq$: $Es\ subconjunto\ de\ o\ igual\ al\ conjunto.\ Ejemplo:\ A \subseteq B.\ Se\ lee:\ A\ es\ subconjunto\ o\ igual\ a\ B$

Variables en conjuntos

Es un símbolo (x, y, z, etc.) que representa un elemento no definido o especificado de un conjunto dado.

Las variables son útiles para escribir los conjuntos por comprensión, expresado mediante alguna regla que deben cumplir éstas.

Ejemplo 3.5
$Sea\ x > 9$

Esta expresión no es una proposición, ya que al no conocer el valor de la *variable* **x**, no es posible determinar un valor de verdad para ésta, a no ser que diéramos un valor específico a la variable x. Esta expresión nos dice que x es cualquier número (no sabemos cuál) mayor que 9.

Podemos definir el conjunto **Q** que cumplan la regla que todos los elementos son números mayores a 9, lo cual puede escribirse así:

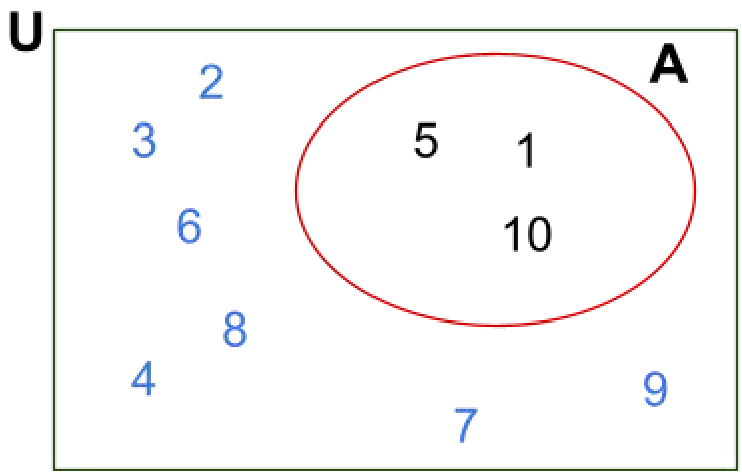
$Q = \{x\ /\ x > 9,\ x \in \aleph\}$

Esta forma de escribir un conjunto se conoce por ***comprensión***, y se lee: “**Q** es el conjunto compuesto cualquier número **x**, tal que ***x*** es mayor que nueve y **x** está en los naturales”.

Conjunto Referencial o Universal

Es el conjunto formado por todos los elementos de estudio en un contexto dado. Se representa por **U**.

<u>Ejemplo 3.6</u>
Sea el conjunto $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ y el conjunto $A = \{1, 5, 10\}$

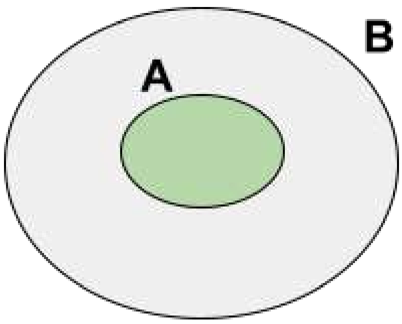


El conjunto A puede ser escrito así:
$A = \{x\,/\,x = 1 \lor x = 5 \lor x = 10,\ x \in U\}$

Subconjuntos

A es subconjunto de B sí y sólo sí todos los elementos de A son elementos de B. Se simboliza: $A \subset B$

<u>Ejemplo 3.7</u>
Representación gráfica de un subconjunto: $A \subset B$



Propiedades de los subconjuntos

1. El conjunto vacío es subconjunto de todo conjunto, esto es, si A es un conjunto, entonces $\{\} \subset A$
2. Todo conjunto es subconjunto de sí mismo; es decir, si A es un conjunto, entonces $A \subset A$
3. Si A es subconjunto de B, no necesariamente B es subconjunto de A
4. Transitividad: Si A es subconjunto de B y es subconjunto de C, entonces A es subconjunto de C

Igualdad de conjuntos

Dos conjuntos A y B son iguales sí y sólo sí todo elemento de A es elemento B. Se escribe: $A = B$

Operaciones entre conjuntos

A través de varias operaciones con conjuntos, es posible crear nuevos conjuntos. Veamos cuáles son.

Unión ∪

La ***unión*** de dos conjuntos A y B, es el conjunto formado por los elementos comunes y no comunes de A y B:

$A \cup B = \{x \,/\, x \in A \lor x \in B\}$

Intersección ∩

La ***intersección*** de dos conjuntos A y B, es el conjunto formado por los elementos comunes entre A y B:

$A \cap B = \{x \,/\, x \in A \land x \in B\}$

Diferencia −

La ***diferencia*** entre el conjunto A y el conjunto B, es el conjunto formado por los elementos comunes de A que no están en B:

$A - B = \{x \,/\, x \in A \land x \notin B\}$

Diferencia simétrica Δ

La ***diferencia simétrica*** entre el conjunto A y el conjunto B, es el conjunto formado por los elementos que pertenecen al conjunto A o al conjunto B y que no sean comunes; se puede expresar de las siguientes formas:

$A \, \Delta \, B = \{x \, / \, x \in (A \cup B) \wedge x \notin (A \cap B)\}$

$A \, \Delta \, B = (A \cup B) - (A \cap B)$
$A \, \Delta \, B = (A - B) \cup (B - A)$

Complemento de un conjunto

Sea A un subconjunto de U. El ***complemento*** de A, simbolizado A', es el conjunto formado por los elementos de U que no están en A.

$A' = \{x \, / \, x \in U \wedge x \notin A\}$
$A' = U - A$

Ejemplo 3.8
Realice las siguientes operaciones sobre los conjuntos `U = {1, 2, 3, a, b, c, d}`, `A = {1, 2, 3}`, `B = {a, b}`: unión, intersección, diferencia, diferencia simétrica, complemento de A

$A \cup B = \{1, 2, 3, a, b\}$
$A \cap B = \{\,\}$
$A - B = \{1, 2, 3\}$
$A \, \Delta \, B = \{1, 2, 3, a, b\}$
$A' = \{a, b, c, d\}$

Conjunto potencia o conjunto de partes P

Sea A un conjunto. El conjunto ***potencia*** de A está formado por todos los subconjuntos del conjunto A.

$P(A) = \{X \, / \, X \subset A\}, \, donde \, X \, es \, cualquier \, subconjunto \, de \, A$

El número de subconjuntos (elementos) del conjunto potencia está dado por 2^n, donde **n** es el número de elementos del conjunto; de ahí que reciba este nombre.

Ejemplo 3.9
$Sea \, A = \{a, b \, c\}$
$El \, número \, de \, subconjuntos \, de \, A \, es$: $|P(A)| = 2^3 = 8$
$P(A) = \{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c,\}, \{b, c\}, \{a, b, c\}\}$

Producto cartesiano ×

El ***producto cartesiano*** entre los conjuntos A y B es otro conjunto formado por los ***pares ordenados*** entre A y B.

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

Un par ordenado (a, b) está formado por dos elementos donde importa el orden, esto quiere decir que $(a, b) \neq (b, a)$

El ***plano cartesiano*** es un claro ejemplo que ilustra pares ordenados, donde la primera coordenada o parámetro indica el eje **x** y la segunda el eje **y**.

Ejemplo 3.10

$Sea\ la\ pareja\ ordenada\ del\ plano\ cartesiano\ (x, y) = (5, -\ 4).\ Punto\ ubicado\ en\ el\ cuarto\ cuadrante$

$Ahora\text{:}\ (y,\ x) = (-\ 4, 5).\ \ Punto\ ubicado\ en\ el\ segundo\ cuadrante$

El número de elementos de este conjunto está dado por el producto entre el número de elementos de A y el número de elementos de B.

$|A \times B| = |A| * |B|$

Ejemplo 3.11

$Sea\ A = \{a, b\},\ B = \{1\ ,\ 2,\ 3\}$

$Cardinalidad\ de\ |A \times B| = |2| \times |3| = 2 \times 3 = 6\ elementos$

$A \times B = \{\{(a, 1)\}, \{(a, 2)\}, \{(a, 3)\}, \{(b, 1)\}, \{(b, 2)\}, \{(b, 3)\}\}$

$B \times A = \{\{(1, a)\}, \{(2, a)\}, \{(a, 2)\}, \{(1, b)\}, \{(2, b)\}, \{(3, b)\}\}$

Observe que $A \times B \neq B \times A$

Intervalos

Un intervalo es un conjunto de números reales que se encuentra comprendido entre dos extremos, a y b. Es un subconjunto de la recta real.

Por ejemplo, los números que satisfagan una condición 0 ≤ x ≤ 15 implican un intervalo que va desde el 0 hasta el 15, incluyendo a ambos.

Existen 3 tipos de intervalos matemáticos, estos son: abierto, cerrado y semiabierto o semicerrado.

Intervalo abierto

Un intervalo abierto es aquel que no incluye los extremos entre los cuales está comprendido, pero sí todos los valores ubicados entre estos. Se representa mediante la desigualdad $a < x < b$, ó también de esta forma: (a, b).

Intervalo cerrado

Un intervalo cerrado es aquel que incluye los extremos entre los cuales está comprendido, y todos los valores ubicados entre estos. Se representa mediante una expresión del tipo $a \leq x \leq b$ ó ó, también de esta forma: $[a, b]$.

Intervalo semiabierto o semicerrado

Un intervalo semiabierto (o semicerrado) es aquel que incluye tan solo uno de los extremos de los valores que están entre ellos, de modo que el otro extremo queda excluido. Pueden estar incluidos o excluidos tanto el extremo derecho como el izquierdo. Se tienen los siguientes casos: $a < x \leq b$ ó ó, también de esta forma: $(a, b]$ (abierto por la izquierda y cerrado por la derecha). $a \leq x < b$ ó ó, también de esta forma: $[a, b)$ (cerrado por la izquierda y abierto por la derecha)

Si uno de los extremos va hasta $+\infty$ ó $-\infty$ puede usarse el símbolo con la condición que dicho extremo debe ser abierto: $(-\infty, 0)$; $[5, +\infty)$.

Observe que los números reales pueden expresarse como el intervalo $\mathbf{R} = (-\infty, +\infty)$.

Aplicaciones con intervalos

Los problemas relacionados con intervalos involucran la solución de inecuaciones, ecuaciones e inecuaciones con valor absoluto, además de ser una de las bases requeridas para la comprensión del cálculo.

Ejemplo 3.12

Resolver las siguientes inecuaciones

a) $2x - 1 <= 7$

   $2x \leq 7 + 1$

   $2x \leq 8$

   $x \leq 4$

   Por tanto la solución es el intervalo donde x es válida: $x \in (-\infty, 4]$.

b) $x^2 - 6x + 8 > 0$

   Factoricemos el miembro izquierdo:

   $(x - 4)(x - 2) > 0$

   De las propiedades de signos, sabemos que:

```
ab > 0 si a > 0 ^ b > 0 v a < 0 ^ b < 0

Entonces:
x - 4 > 0 ^ x - 2 > 0
          v
x - 4 < 0 ^ x - 2 < 0

x > 4 ^ x > 2
      v
x < 4 ^ x < 2
```

`La conjunción nos indica intersección, mientras que la disyunción representa unión en desigualdades que representan intervalos (conjuntos):`

```
(4, +∞) ∩ (2, +∞)
        ∪
(-∞, 4) ∩ (-∞, 2)

(-∞, 2) ∪ (4, +∞)
```

Por tanto la solución está dada donde x es válida: $x \in (-\infty, 2) \cup (4, +\infty)$.

Álgebra de conjuntos

En el desarrollo de la teoría de conjuntos, los procedimientos permiten transformar expresiones en las que hay conjuntos relacionados por uniones, intersecciones o complementos. El álgebra de conjuntos se apoya en varias identidades, que se comentan a continuación.

Veamos las leyes del álgebra de conjuntos considerando los conjuntos ***A***, ***B***, ***C***, ∅ y ***U***

Leyes de idempotencia para la unión y la intersección

1. $A \cup A = A$
2. $A \cap A = A$

Leyes de identidad para la unión y la intersección

1. $A \cup \emptyset = A$
2. $A \cap U = A$

Leyes conmutativas para la unión y la intersección

1. $A \cup B = B \cup A$
2. $A \cap B = B \cap A$

Leyes asociativas para la unión y la intersección

1. $A \cup (B \cup C) = (A \cup B) \cup C$
2. $A \cap (B \cap C) = (A \cap B) \cap C$

Leyes distributivas para la unión y la intersección

1. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
2. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Leyes de absorción para la unión y la intersección

1. $A \cup (A \cap B) = A$
2. $A \cap (A \cup B) = A$

Leyes inversas para la unión y la intersección

1. $A \cup A' = U$
2. $A \cap A' = \emptyset$

Ley de doble negación

1. $(A')' = A$

Leyes de De Morgan

1. $(A \cup B)' = A' \cap B'$
2. $(A \cap B)' = A' \cup B'$

Leyes adicionales

1. Unión del universo: $A \cup U = U$
2. Intersección vacía: $A \cap \emptyset = \emptyset$
3. Unicidad de la identidad: $Si\, X \cap P = P, \forall P \subseteq U, \rightarrow X = U$
4. Complemento del vacío: $\emptyset' = U$
5. Complemento del universo $U' = \emptyset$

Ejemplo 3.13
Demostrar que $A \cap (A \cap B) = A \cap B$

Demostración
$(A \cap A) \cap B\ (Asociativa)$
$A \cap B\ (Idempotencia)$

Ejemplo 3.14
Demostrar que $(A \cup B) \cap (A' \cap B') = \emptyset$

Demostración
$(A \cup B) \cap (A \cup B)'$ $(De\ Morgan)$
$\{\}$ $(Inversa\ para\ la\ intersección)$

Ejemplo 3.15
Demostrar el ejemplo anterior sin emplear De Morgan. Se deja como ejercicio. Sugerencia: aplicar la propiedad distributiva en el primer o segundo paso.

Ejemplo 3.16
Demostrar que $(A \cup B) \cup (A' \cap B') = U$

Demostración
$(A \cup B) \cup (A \cup B)'$ $(De\ Morgan)$
U $(Inversa\ para\ la\ unión)$

Ejemplo 3.17
Demostrar el ejemplo anterior sin emplear De Morgan. Se deja como ejercicio. Sugerencia: aplicar la propiedad distributiva en el primer o segundo paso.

Ejemplo 3.18
Demostrar la intersección vacía: $A \cap \phi = \phi$

Demostración
$\emptyset = A \cap A'$ $(Inversa)$
$(A \cap A') \cap A$ $(Sustitución\ de\ \phi)$
$(A \cap A) \cap A'$ $(Asociativa)$
$A \cap A'$ $(Idempotencia)$
ϕ $(Inversa)$

Preguntas

1. Con sus palabras, describa qué entiende por conjunto
2. Describa las formas de representación de conjuntos
3. ¿Qué se obtiene al realizar una operación entre conjuntos?
4. ¿Los operadores de conjuntos son binarios?
5. ¿En qué contextos de la informática se aplican los conjuntos?

Ejercicios

1. Sean los conjuntos
   $U = \{x/ -5 \leq x \leq 10,\ x \in Z\}$
   $A = \{x/0 \leq x \leq 3,\ x \in Z\}$
   $B = \{2,\ 3,\ 8\}$
   $C = \{x/ -2 < x < 5,\ x \in Z\}$
   Hallar:

   a. $(B - A)' \cup C'$
   b. $A \times (B' - C)$
   c. $P(A);\ P(B);\ P(C)$
   d. $A \Delta (C \Delta B)$
   e. $[A \cap (B - C)'] - (A \cup B)$
2. Sean los conjuntos
   $U = \{a,\ b, c,..., x, y, z, 0, 1, 2, 3... 100\}$
   $A = \{a, b, c,..., z\}$
   $V = \{a, e, i, o, u\}$
   $C = \{0, 1, 2, 3,..., 100\}$
   $P = \{0, 2, 4, 6,..., 100\}$
   $I = \{1, 3, 5, 7,..., 99\}$
   $O = \{1, 2,..., 10,\ a, b, e, u, z\}$
   Hallar:
   a. $(O \cap C) \cup (I - C)$
   b. $A \Delta V$
   c. $P(V)$
   d. $(P \cup I) - O'$
   e. $A'; V'; P'; I'; O'$
3. Sean los conjuntos
   $U = \{a,\ b, c, d, e, f, g, h, i, j, k, l, m, n, o, 1, 2, 3... 15\}$
   $A = \{a, b, c, e, i, 1, 2, 6, 8, 15\}$
   $V = \{a, e, i, o, u\}$
   $B = \{1, 2, 3,..., 10\}$
   Utilice diagramas de Venn para hallar:
   a. $A \cap V$
   b. $A \Delta B$
   c. $A \cup B$
   d. $V - A$
   e. $(A \cap B)'$
   f. A'
   g. B'
   h. V'
   i. $(A \cup B) - B$
   j. $V \cap B$
4. Utilice diagramas de Venn para comprobar
   a. Las leyes distributivas
   b. Leyes de De Morgan
   c. $A - B = A \cap B'$
5. Sabiendo que A, B y C son tres conjuntos cualesquiera, sombree la parte correspondiente a los siguientes conjuntos:
   a. $(B \cup A') - C$
   b. $(A - C) \cap B'$
   c. $(A' \cup B) \cap (B - C)$
6. Utilice las leyes del álgebra de conjuntos para demostrar las siguientes propiedades/expresiones

a. Intersección vacía: $\phi \cap P = \phi$
b. Unicidad de la identidad: $Si\, X \cap P = P, \forall P \subseteq U, \rightarrow X = U$
c. Unicidad del universo: $Si\, X \cap P = \phi \wedge X \cup P = U \rightarrow X = P'$
d. $Si\, X \cup P = P, \forall P \subseteq U \rightarrow X = \phi$
e. $P \cap P = P, \forall P \subseteq U$
f. $P \cap (P \cap Q) = P \cap Q$
g. $(P \cup Q) \cup (P' \cap Q') = U$
h. $(P \cup Q) \cap (P' \cap Q') = \phi$
i. $(P \cap Q) \cup (P' \cup Q') = U$
j. $(P \cap Q) \cap (P' \cup Q') = \phi$
k. $P \cup U = U$ (Sugerencia: revise la intersección vacía)
l. $P \cup P = P$ (Sugerencia: escriba la segunda P como $P \cap U$)
m. $P \cup (P \cup Q) = P \cup Q$ (Sugerencia: utilice ley asociativa y el literal anterior)

Capítulo 4. Relaciones y funciones

En esta sección el concepto de **relación** se explica desde el contexto de la teoría de conjuntos y sirve como base para su estudio posterior en el cálculo.

Relaciones

Es común en la cotidianidad escuchar frases como: "Ana estudia en el TdeA", "Juan es primo de María", "Isabel juega en el equipo de voleibol", etc. En referencia a estas frases, se acostumbra decir que "Ana tiene una *relación* académica con el TdeA", que "Juan tiene una *relación* familiar con María" y que "Isabel tiene una *relación* deportiva con un equipo de voleibol"; en cada caso hay una expresión que vincula o relaciona los elementos de dos conjuntos. Veamos el caso de la estudiante.

El conjunto de habitantes de una ciudad (Medellín, por ejemplo) es el primer conjunto y el segundo son las universidades que tienen sede en Medellín. Podemos entonces establecer la relación "*estudia en*".

Si A es el conjunto de los habitantes de Medellín y B el conjunto de las universidades de Medellín, podemos escribir:

a estudia en b
Donde $a \in A \wedge b \in B$.

Una relación se usa en unión con par de objetos considerados en un orden bien definido; la relación se refiere a la existencia o no de un vínculo entre ciertos pares ordenados; además, hay criterios claros para distinguir un par ordenado de otro. Veamos que es una pareja ordenada.

Pareja ordenada

Es una entidad formada por dos objetos en un orden determinado. Se determina por las siguientes propiedades:

1. Dado cualquier par de objetos a y b, hay un objeto llamado **pareja ordenada** de a y b denotado con el símbolo (a, b), y determinada de manera única por a y b.
2. Si (a, b) y (c, d) son parejas ordenadas, entonces $(a, b) = (c, d)$ *si y sólo si* $a = c$ y $b = d$.

Relación

<u>Definición</u>
Una **Relación** es un **conjunto** cuyos elementos son **parejas ordenadas**.

Así, si R es una relación, escribimos:

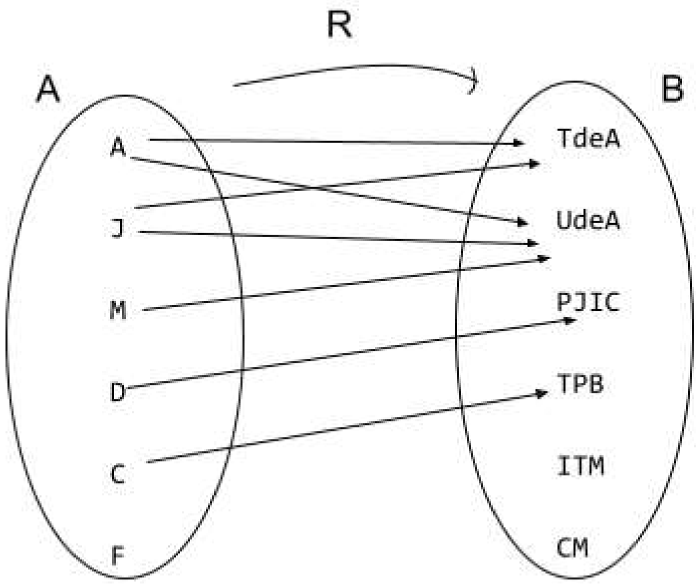
(x, y) ∈ R ó x R y

Para indicar que x está relacionado con y por la relación R.

Ejemplo 4.1
Representemos gráficamente la relación "estudia en", si:
R = {(A,TdeA), (A, UdeA), (J, TdeA), (J, UdeA), (M, UdeA), (D, PJIC), (C, TPB)}

Cada inicial en la primera coordenada de las parejas ordenadas representa el nombre de una persona de una ciudad (por ejemplo, Medellín); la segunda coordenada indica una institución educativa de la ciudad. Aquí puede verse claramente que:

A = {A, J, M, D, C, F}
B = {TdeA, UdeA, PJIC, TPB, ITM, CM}



Nota
Observe que un estudiante puede estudiar en *varias* universidades y a su vez en éstas hay *muchos* estudiantes. Este tema se aplica en la teoría de bases de datos, particularmente en la definición de relaciones entre entidades.

Dominio y rango de una relación

Veamos el conjunto del ejemplo anterior para definir un par de conceptos muy importantes. El conjunto de la izquierda, conocido como el **conjunto de partida**, se conoce como el **Dominio** de R, a veces denotado por D_R. El conjunto de la derecha, conocido como **conjunto de llegada** es llamado **Rango**, **Recorrido** o **Codominio** de R y se denota por R_R.

$D_R = \{x \ / \ \exists \ y, (x, y) \in R\}$

$R_R = \{y \ / \ \exists \ x, \ (x, y) \in R\}$

Esto es, el dominio está conformado por los elementos que son primeras coordenadas en la relación y el rango por la segunda, respectivamente.

Uno de los tipos de relación más simple es el conjunto de parejas ordenadas (a, b), $a \in A$, $b \in B$. Esta relación ya fue estudiada y es conocida como el **producto cartesiano** $A \times B$.

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

Nota
De esto se puede ver fácilmente que una relación R es un subconjunto de un producto cartesiano $A \times B$, tal que $D_R \subseteq A$ y $R_R \subseteq B$. Si R es una relación y $R \subseteq A \times B$, entonces R es una **relación de** A **en** B.

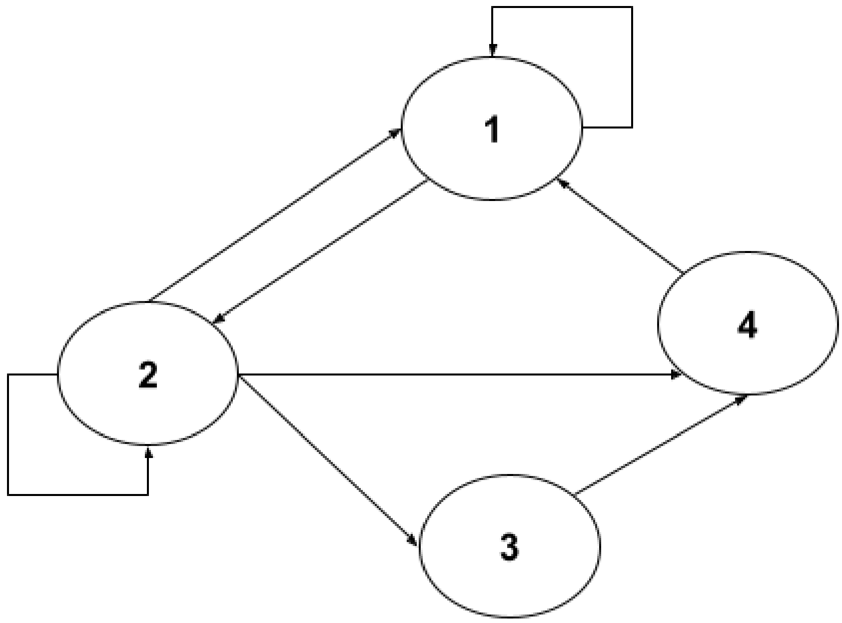
Representación gráfica de las relaciones y grafos dirigidos o digrafos

Ya vimos la representación gráfica mediante diagramas de Venn de una relación trazando flechas entre los elementos de dos conjuntos finitos A y B que se relacionan de alguna manera.

Ahora, si A es un conjunto finito y R una relación de A en A, la relación puede representarse gráficamente dibujando pequeños círculos conocidos como **vértices** (o **nodos**) para cada elemento de A. Luego se traza una recta conocida como **arista** (o **arco**) entre el vértice a_i y el vértice a_j, sí y sólo si $a_i \, R \, a_j$. La gráfica resultante es un **grafo dirigido** o **digrafo**. En el siguiente capítulo se hablará más acerca de los grafos.

Ejemplo 4.2
Sea A un conjunto y R una relación de A en A:
$A = \{1, 2, 3, 4\}$
$R = \{(1, 1), (1, 2), (2, 1), (2, 2), (2, 3), (2, 4), (3, 4), (4, 1)\}$

El grafo dirigido de R es el siguiente:



Algunos tipos de relaciones

Sean R y S relaciones de un conjunto A en un conjunto B. Se tienen los siguientes tipos de relaciones.

Relación complemento

La **relación complemento** de A en B está definida por:
(a, b) ∈ R' sí y sólo si (a, b) ∉ R

Relación unión

Dado que la unión R ∪ S es un subconjunto de A x B, se tiene que a(R ∪ S)b sí y sólo si aRb ó aSb.

Relación intersección

Dado que la unión R ∩ S es un subconjunto de A x B, se tiene que a(R ∩ S)b sí y sólo si aRb y aSb.

<u>**Ejemplo 4.3**</u>
Sean A = {a, b, c, d}, B = {1, 2, 3}.
Sean R y S relaciones de A en B:
R = {(a, 1), (b, 2), (c, 2), (c, 3), (d, 1)}
S = {(a, 2), (a, 1), (c, 3), (d, 2), (c, 1), (d, 3)}
Encontrar R', S', R ∪ S, R ∩ S
Dibuje todas las relaciones (se deja como ejercicio)

<u>Solución</u>

Primero determinemos A x B para encontrar las relaciones complemento

A x B = {(a, 1), (b, 1), (c, 1), (d, 1), (a, 2), (b, 2), (c, 2), (d, 2), (a, 3), (b, 3), (c, 3), (d, 3)}

R’ = {(b, 1), (c, 1), (a, 2), (d, 2), (a, 3), (b, 3), (d, 3)}

S’ = {(b, 1), (d, 1), (b, 2), (c, 2), (a, 3), (b, 3)}

R ∪ S = {(a, 1), (b, 2), (c, 2), (c, 3), (d, 1), (a, 2), (d, 2), (c, 1), (d, 3)}

R ∩ S = {(a, 1), (c, 3)}

Relación inversa

La **relación inversa** de R, denotada por R^{-1}, es una relación de B en A y se define por:

$bR^{-1}a$ sí y solo si aRb

La relación R^{-1} está formada por todas las parejas ordenadas de R escritas en orden inverso, por tanto, se tiene que:
Dominio de R^{-1} = Rango de R
Rango de R^{-1} = Dominio de R
$(R^{-1})^{-1} = R$

Ejemplo 4.4
Sean A = {a, b, c, d}, B = {1, 2, 3}. En el ejemplo anterior teníamos las relaciones R y S de A en B definidas como:

R = {(a, 1), (b, 2), (c, 2), (c, 3), (d, 1)}
S = {(a, 2), (a, 1), (c, 3), (d, 2), (c, 1), (d, 3)}

Por tanto, sus relaciones inversas están dadas por:

R^{-1} = {(1, a), (2, b), (2, c), (3, c), (1, d)}
S^{-1} = {(2, d), (1, a), (3, d), (2, d), (1, c), (3, d)}

Se deja como ejercicio graficar las relaciones R^{-1} y S^{-1}.

Relación compuesta

Si A, B y C son conjuntos, R una relación de A en B y S una relación de B en C, a partir de R y S es posible construir una relación de A en C, llamada relación compuesta de R y S que se expresa con el símbolo $R \circ S$ y está definida así:

Si a $\in$ A y c $\in$ C, entonces a$(R \circ S)$c sí y sólo si para algún b $\in$ B, se tiene aRb y bSc.

Esto es, para determinar si a y c están relacionados por $R \circ S$ se debe encontrar un elemento b $\in$ B para el cual se cumplen las dos condiciones aRb y bSc.

Ejemplo 4.5
Sean A = {a, b, c, d}, B = {1, 2, 3, 4, 5} y C = {u, v, w, x}. R y S relaciones definidas como:

R = {(a, 1), (b, 3), (c, 5), (d, 5), (c, 1), (d, 1), (c, 4)}
S = {(2, u), (3, v), (4, x), (2, v)}

Por tanto $R \circ S = \{(b, v), (c, x)\}$

Se deja como ejercicio graficar las relación $R \circ S$.

Propiedades de las relaciones

En muchas de las aplicaciones de las relaciones se trabaja con relaciones en un conjunto de A en A o simplemente relaciones en A, ya que satisfacen las siguiente propiedades y puede extenderse a conjuntos infinitos, como el de los **Z** (enteros), **R** (reales), etc.

Relaciones reflexivas

Una relación R en un conjunto A es **reflexiva** si (a, a) ∈ R, para todo a ∈ A; es decir aRa para todo a ∈ A.

Ejemplo 4.6
Sea R = {(x, x) | x ∈ A}
R es reflexiva, ya que (x, x) ∈ R ∀ x ∈ A

Relaciones simétricas

Una relación R en un conjunto A es **simétrica** si cuando (a, b) ∈ R, entonces (b, a) ∈ R. Por el contrario, R no es simétrica si existen a, b ∈ A con (a, b) ∈ R y (b, a) ∉ R.

Ejemplo 4.7
Si A es el conjunto de los números enteros Z, la relación **<** no es simétrica, pues si a < b, no se cumple que b < a. Sea a = 2, b = 4; 2 < 4, pero 4 > 2

Relaciones antisimétricas

Una relación R en un conjunto A es **antisimétrica** si cuando (a, b) ∈ R y (b, a) ∈ R, entonces a = b. Otra forma de expresar esto es diciendo que R es antisimétrica cuando: si a ≠ b, se tiene que (a, b) ∉ R ó (b, a) ∉ R.

<u>Ejemplo 4.8</u>
Si `A = {1, 2, 3, 4}` y `R = {(1, 2), (2, 2), (3, 4), (4, 1)}`
Compruebe que R no es simétrica, pero sí es antisimétrica

R no es simétrica, ya que (1, 2) ∈ R, pero (2, 1) ∉ R
R es antisimétrica, ya que si a ≠ b, (a, b) ∉ R ó (b, a) ∉ R: (1, 2) ∈ R, pero (2, 1) ∉ R, etc.

Relaciones transitivas

Se dice que una relación R en un conjunto A es **transitiva** si cuando (a, b) ∈ R y (b, c) ∈ R, entonces (a, c) ∈ R.

<u>Ejemplo 4.9</u>
Si A es el conjunto de los números enteros Z, la relación **<** es transitiva. Si a < b y b < c, se tiene que a < c. Sea a = 2, b = 4, c = 7; 2 < 4, 4 < 7 y 2 < 7

Relaciones de equivalencia

Una relación R sobre un conjunto A se llama relación de **equivalencia** en A si es reflexiva, simétrica y transitiva.

<u>Ejemplo 4.10</u>
Si `A = {1, 2, 3, 4}` y `R = {(1, 1), (1, 2), (2, 1), (2, 2), (3, 4), (4, 3), (3, 3), (4, 4)}`
Se puede verificar fácilmente que R es una relación de equivalencia (se deja como ejercicio para que el estudiante lo compruebe)

<u>Consulta</u>
Relación de módulo $a \equiv b \, mod \, n$. Por ejemplo el módulo 2: $a \equiv b \, mod \, 2$

Funciones

Una **función** es un tipo de relación especial que goza de gran importancia en las matemáticas y en las ciencias en general apareciendo en distintos contextos. Por ejemplo, hemos escuchado frases como éstas:

- La velocidad es una función del tiempo
- La temperatura en función de la presión o viceversa
- El volumen en función de la temperatura
- El crecimiento poblacional en función del tiempo
- El costo en función de la producción

Es por ello que es uno de los temas más tratados en matemáticas y que es esencial su comprensión.

Una función es una relación entre dos conjuntos en la que a cada elemento del primer conjunto le corresponde un único elemento del segundo conjunto. Veamos una definición formal de *función*:

Función (definición)
Sean A y B conjuntos no vacíos. Una **función** f de A en B es una relación de A en B, tal que:

a) El dominio de f es el conjunto A.
b) Si (a, b) y (a, c) pertenecen a f, entonces $b = c$.

La primera propiedad indica que $\forall a \in A$, a es el primer elemento de una pareja de la relación; la segunda propiedad nos dice que si $(a, b) \in f$ entonces b queda determinada de manera única por a. De esta forma escribimos:

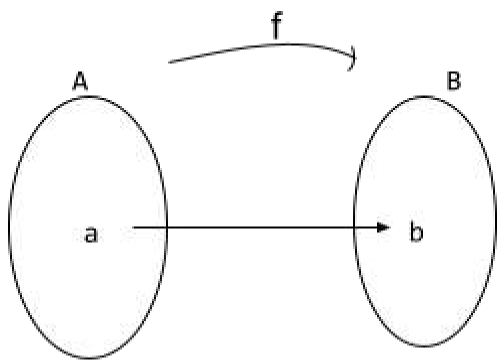
$f(a) = b$

Y podemos escribir f como un conjunto:

$f = \{(a, f(a)) | a \in A\}$

Las funciones asignan o hacen corresponder a cada elemento $a \in A$ el único elemento de $f(a) \in B$.

El símbolo $f: A \rightarrow B$ se utiliza para indicar que f es una función del conjunto A al conjunto B. En la notación $f(a)$, al elemento $a \in A$ se le conoce como argumento de la función f, y para nombrar $f(a) = b$ se utilizan los nombres: **valor** de la función f en a, **imagen** de a, entre otros.



$(a, b) \in f \Leftrightarrow b = f(a)$

Ejemplo 4.11
Sean
$A = \{1, 2, 3, 4\}$
$B = \{a, b, c, d\}$
$f = \{(1, a), (2, a), (3, d), (4, c)\}$ una relación

La relación f es una función:
Su dominio corresponde al conjunto A y ningún elemento de este conjunto aparece como primer elemento de dos o más parejas ordenadas diferentes.

Siendo así, podemos escribir:
$f(1) = a$
$f(2) = a$
$f(3) = d$
$f(4) = c$

De esto podemos ver que el rango de la función f es el conjunto:

$R_f = \{a, d, c\} \subset B$

Observe como $a \in B$ aparece como segundo elemento de dos parejas ordenadas, lo cual no afecta la definición de función. Por tanto, una función puede tomar el mismo valor (imagen) para dos elementos de A.

Ejemplo 4.12
Sean
$A = \{a, b, c, d, e\}$
$B = \{1, 3, 4, 5, 8, 9\}$
$f = \{(a, 1), (b, 4), (c, 5), (a, 3), (d, 8), (e, 9)\}$ una relación
$g = \{(a, 1), (b, 3), (c, 3), (e, 9)\}$ una relación
Determine si f y g son funciones.

Solución
Para f puede decirse fácilmente que no lo es, ya que un elemento de A aparece en dos parejas ordenadas como primer elemento, esto es, tiene dos imágenes: $(a, 1)$, $(a, 3)$.

Para g se tiene que tampoco es una función porque hay elementos de A que no están relacionados con algún elemento de B, esto es, el dominio de g no es igual al conjunto A.

Funciones conocidas

En los números reales existen algunas funciones particulares que son fácilmente reconocibles y también son muy comunes en distintas aplicaciones de la ciencia. Algunas de ellas son:

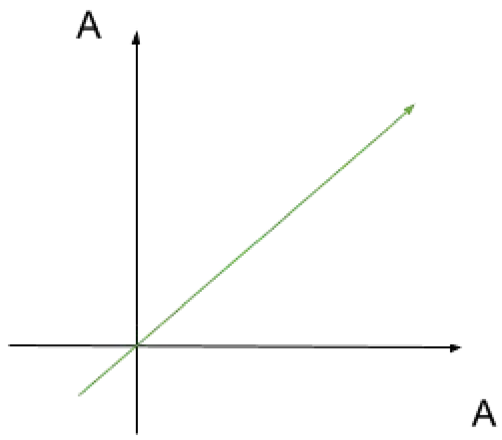
- Función idéntica
- Función constante
- Función lineal
- Función cuadrática
- Función cúbica

Ejemplo 4.13

Si A es cualquier conjunto, la **función idéntica** en A (denotada a veces como $i_A(a)$)

$f(a) = a \ \forall a \in A$

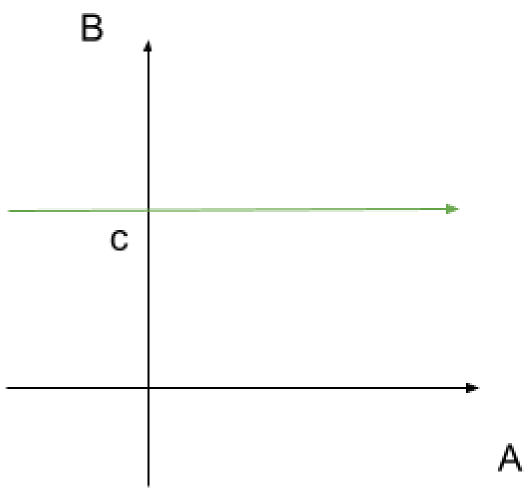
Ó también:
$i_A(a) = a \ \forall a \in A$



Ejemplo 4.14
Si f es una función de un conjunto A en un conjunto B, definida por:

$f(x) = c \ \forall x \in A$

Donde c es un elemento fijo de B; la función f se llama **función constante** en c.



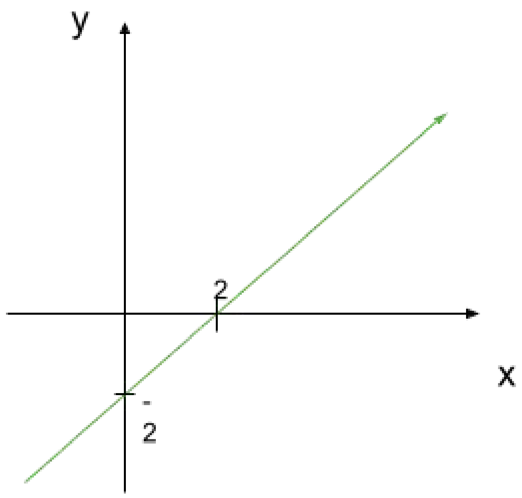
Nota
Muchas de las funciones que encontramos en la práctica son funciones de **R** en **R**, donde A y B son conjuntos de números reales, las cuales se denominan funciones reales de una variable real o simplemente **funciones reales**.

Ejemplo 4.15
Si f es una función de un conjunto A en un conjunto B, definida por:

$f(x) = mx + b, \forall x \in A$, donde m, b son constantes $\in \mathbf{R}$, se llama **función lineal**. La constante m determina la **pendiente** o **inclinación** de la recta, y la constante **b** determina el **punto de corte** de la recta con el eje vertical **y**.

Si $m = 1;\ b = -2$, se tiene: $f(x) = x - 2$

Para $x = 0,\ y = -2$; para $y = 0,\ x = 2$



Ejemplo 4.16
Si $A = B = \mathbf{R}$ y la regla que asocia a cada número real x el número real "el cuadrado de x", se expresa así:

$$\begin{aligned} f:\ &\mathbf{R} \to \mathbf{R} \\ &x \to x^2 \end{aligned}$$

O simplemente la función f definida por

$f(x) = x^2\ \forall x \in \mathbf{R}$

Esta función se conoce como la **función cuadrática**.

Grafique la función y determine el dominio e imagen de ésta.

Solución
El dominio es **R**. (todos los números reales se pueden aplicar a f y el dominio siempre tendrá sentido)

Para hallar la imagen, despejamos y en términos de x:

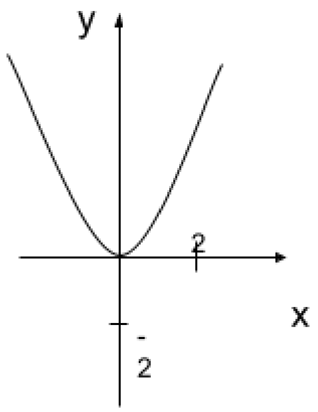
$x = \pm\sqrt{y}$

Por tanto I_f: $\mathbf{R}^+ = [0, +\infty)$

Tabla de valores para la gráfica:

x	-3	-2	-1	0	1	2	3
y	9	4	1	0	1	4	9

Gráfica:



<u>Ejemplo 4.17</u>
Si A = B = **R** y la regla que asocia a cada número real x está dada por:

f: **R** → **R**

$x \to y = \frac{x^2+3x-1}{x^2+1}$

$x \to y = f(x) = \frac{x^2+3x-1}{x^2+1}\ \forall x \in \mathbf{R}$

Determine el dominio de la función.

El dominio está definido donde la función tiene sentido; como el denominador $x^2 + 1 \neq 0$ $\forall x \in \mathbf{R}$, esto es, nunca se hace 0, entonces el dominio de f es **R**.

<u>Ejemplo 4.18</u>
Si A = $\mathbf{R}^+$ B = **R** y la regla que asocia a cada número real x positivo el número real "la raíz cuadrada de x", se expresa así:

f: **R** → **R**

$x \to y = f(x) = \sqrt{x}\ \forall x \in \mathbf{R}^+$

Grafique la función y determine el dominio y rango de ésta.

Nota
El dominio de R está definido para todos los x $\in$ **R** donde f(x) tiene sentido.

Ejemplo 4.19
Si $f(x) = \frac{x}{x^2-4x+3}$ Determine el dominio de la función
Solución

Esta función puede escribirse así:

$$f(x) = \frac{x}{(x-3)(x-1)}$$

$$f\colon R - \{1, 3\} \to R$$
$$x \to y = f(x) = \frac{x}{(x-3)(x-1)}$$

Es decir, el dominio de **R** son todos los reales excepto el 1 y el 3, valores de x donde f(x) deja de tener sentido.

Ejemplo 4.20
Si $f(x) = \frac{x+3}{\sqrt{x-2}}$ Determine el dominio de la función

Solución

f(x) tiene sentido si
$x - 2 > 0\, o\, x > 2$

Por tanto D_f: (2, +∞).

Es decir, el dominio de **R** son todos los reales mayores que 2.

Ejemplo 4.21
Si $f(x) = \sqrt{\frac{x}{1-x}}$
Determine el dominio de la función

Solución

Tenemos que:
$$f\colon A \to R$$
$$x \to y = f(x) = \sqrt{\frac{x}{1-x}}$$

f(x) es válida cuando

$\frac{x}{1-x} \geq 0 \wedge 1 - x > 0$
$\frac{x}{1-x} \geq 0\, si\, x \geq 0 \ \wedge 1 - x > 0 \ \vee x \leq 0 \wedge 1 - x < 0$
$x \geq 0 \ \wedge x < 1 \ \vee x \leq 0 \wedge x > 1$
$[0, 1) \cup \{ \}$

Por tanto, el conjunto $A = \{x \in \mathbf{R} \mid 0 \leq x < 1\}$ es el dominio de la función

Es decir, el dominio de **R** está conformado por los números en el intervalo $[0, 1)$.

Ejemplo 4.22
Si $f\colon R \to R$, dada por $f(x) = \frac{1}{x^2-1}$ Determine el dominio e imagen de la función

Solución

El dominio de $f(x)$ está definido donde $x^2 - 1 \neq 0$.
$x^2 - 1 = (x + 1)(x - 1);\ x + 1 \neq 0 \wedge x - 1 \neq 0;\ x \neq -1 \wedge x \neq 1$
Por tanto D_f: $\mathbf{R} - \{-1, 1\}$

Para encontrar la imagen de f, despejamos x en términos de y.
$y = f(x) = \frac{1}{x^2-1};\ x^2 - 1 = \frac{1}{y};\ x^2 = 1 + \frac{1}{y};\ x = \pm\sqrt{1 + \frac{1}{y}}$
La imagen de f está definida donde
$1 + \frac{1}{y} \geq 0 \wedge y \neq 0;\ \frac{1}{y} \geq -1 \wedge y \neq 0;\ y \geq -1 \wedge y \neq 0;$
Por tanto I_f: $[-1, 0) \cup (0, +\infty) = [-1, +\infty) - \{0\}$

Ejemplo 4.23
Si $A = B = \mathbf{R}$ y $f(x) = 0\, si\, x \in Q \ \vee f(x) = 1\, si\, x \notin Q$. Muestre los puntos:
$f(0), f(-2), f(\pi), f(\sqrt{3}), f(\frac{7}{10})$

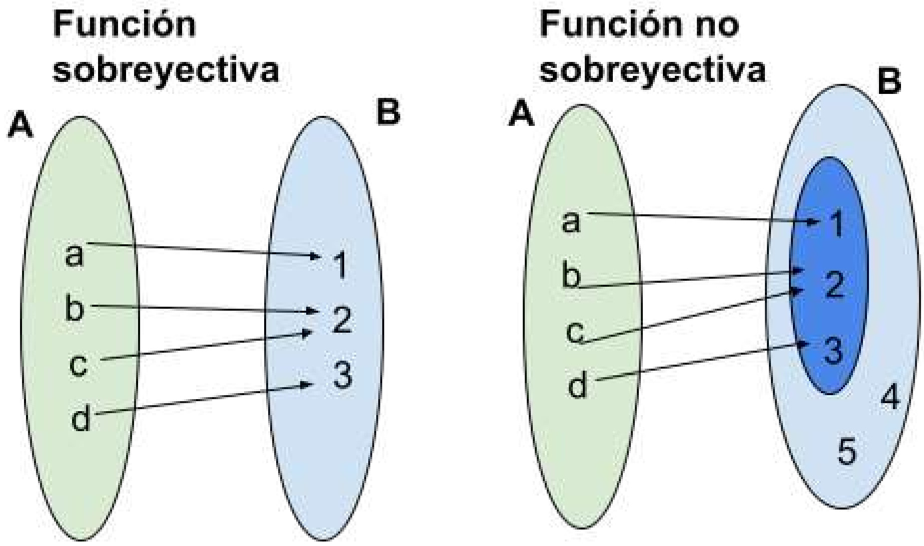
Función sobreyectiva

Si $f\colon A \to B$ es **sobreyectiva** si todo elemento de $b \in B$ es el segundo elemento de algún par ordenado $(a, b) \in f$, o de otra forma, si para cada $b \in B$ se puede encontrar algún elemento $a \in A$, tal que $b = f(a)$, es decir, todo elemento de B es imagen de algún elemento de A, entonces f es *sobreyectiva*.

Definición formal:

$\forall y \in Cod_f\, \exists x\, Dom_f \mid f(x) = y$

Es decir, para cualquier elemento y del codominio, existe otro elemento x del dominio tal que y es la imagen de x por f.



Ejemplo 4.24
Si A = {a, b, c, d}, B ={1, 2, 3}, f = {(a, 2), (b, 1), (c, 3), (d, 1)}. Determinar si f es sobreyectiva.

La función es sobreyectiva porque cada elemento de B (codominio) es imagen de algún elemento de A, o también porque es segunda coordenada de alguna pareja ordenada de f.

Ejemplo 4.25
Determinar si f(x) = 3x + 5 es sobreyectiva.

Esta es una función de **R** en **R**. Es posible encontrar un elemento a ∈ R tal que f(a) = b. En efecto, si f(a) = 3a + 5, entonces: b = 3a + 5. Por tanto: a = (b - 5) / 3. Por tanto la función es sobreyectiva.

Ejemplo 4.26
Determinar si f(x) = 2x - 6 es sobreyectiva.

Esta es una función de **R** en **R**. Aquí el dominio y el codominio es **R**. Es posible encontrar un elemento a ∈ R tal que f(a) = b. En efecto, si f(a) = 2a - 6, entonces: b = 2a - 6. Por tanto: a = (b + 6) / 2. Por tanto la función es sobreyectiva.

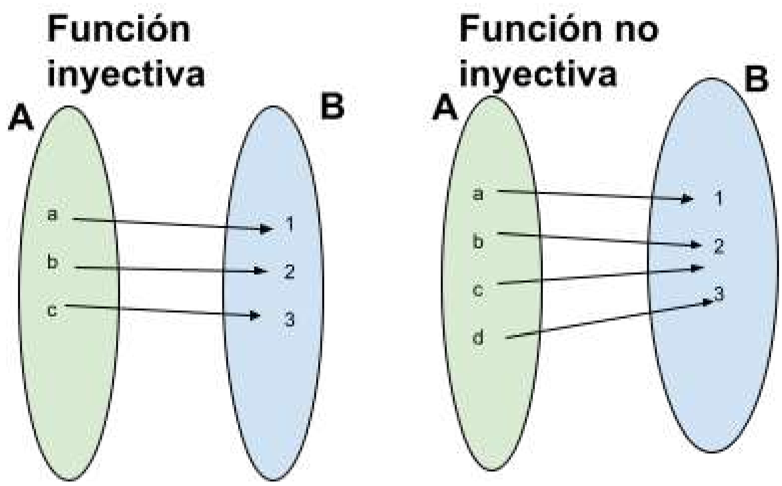
Función inyectiva o uno a uno

Una función f: $A \rightarrow B$ es **inyectiva** o **uno a uno** si para todo a, b ∈ A se tiene que:
Si a ≠ b implica que f(a) ≠ f(b). Es decir, que elementos diferentes de A le corresponden, por f, elementos diferente de B, o, de otra forma: f(a) = f(b) implica que a = b.

Una función es inyectiva cuando no hay dos elementos del dominio que tengan la misma imagen.

Definición formal:

$\forall a,\ b \in Dom_f,\ si\, f(a) = f(b) \Rightarrow a = b$
Es decir, para cualesquiera dos elementos a y b en el dominio de la función, si sus imágenes f(a) y f(b) son iguales, entonces necesariamente a y b también lo son.



Ejemplo 4.27
f(x) = 3x + 5. Determinar si f es inyectiva.

f(a) = f(b); 3a + 5 = 3b + 5, de donde se tiene que a = b, y por tanto f es inyectiva.

Ejemplo 4.28
Si A = {a, b, c, d}, B ={1, 2, 3}, f = {(a, 2), (b, 1), (c, 3), (d, 1)}. Determinar si f es inyectiva.

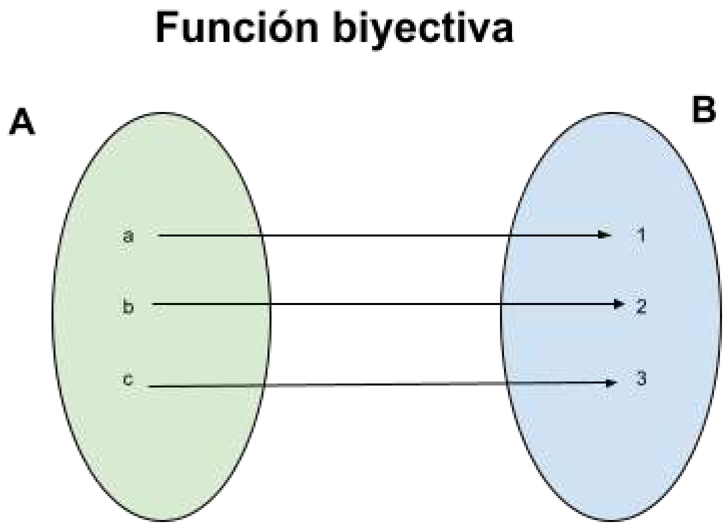
Se tiene que f(b) = 1 = f(d). Por tanto f no es inyectiva porque hay dos elementos del dominio que tienen la misma imagen.

Función biyectiva

Una función que es inyectiva y sobreyectiva a la vez, se dice que es **biyectiva**.

Formalmente se define una función biyectiva como se indica a continuación:
$\forall y \in Cod_f\ \exists!\ x \in Dom_f \mid f(x) = y$

Esto es, para todo elemento y del codominio, existe un único elemento x en el dominio tal que y es la imagen de x por f.



Ejemplo 4.29
Determinar si f(x) = 3x + 5 es biyectiva.

Vimos en los ejemplos anteriores que esta función es tanto inyectiva como sobreyectiva, por consiguiente, es biyectiva.

Función inversa

Una función f: $A \rightarrow B$ es **invertible** si su relación inversa f^{-1}: $B \rightarrow A$ es una función.

Teorema
Sea f: $A \rightarrow B$ una función. f^{-1}: $B \rightarrow A$ es una función si y sólo si f es inyectiva y sobreyectiva, esto es, para que la relación inversa de una función sea también una función, se requiere que ésta sea una biyección.

Función compuesta

Si f: $A \rightarrow B$ y g: $B \rightarrow C$ son funciones, la relación compuesta g o f también es una función de A en C. Se puede ver que (a, c) ∈ g o f si, y sólo si (a, b) ∈ f y (b, c) ∈ g para algún b ∈ B. Entonces b = f(a), c = g(b), c = g(f(a)).

Nota
Al aplicar g o f primero actúa f y luego g.

Preguntas

1. ¿Qué se entiende por codominio y rango en una relación?
2. ¿Qué se entiende por codominio, rango e imagen de una función?
3. ¿Cuál es la diferencia entre relación y función?

Ejercicios

1. Encuentre el dominio e imagen de las siguientes funciones

Capítulo 5. Grafos

En matemáticas y en ciencias de la computación, la teoría de grafos (también llamada teoría de las gráficas) estudia las propiedades de los **grafos** (también llamadas **gráficas**). Un grafo es una estructura discreta compuesta y no vacía, por un conjunto de objetos llamados **vértices** (o **nodos**) y una selección de pares de vértices llamados **aristas** (o **arcos**) (*edges* en inglés) que pueden ser orientados o no. Típicamente, un grafo se representa mediante una serie de puntos (los vértices) conectados por líneas (las aristas).

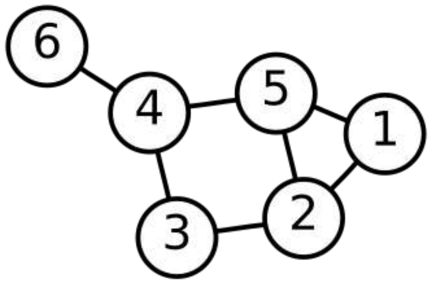


Figura 6.1. Grafo con seis vértices y siete aristas

Definiciones fundamentales

Vértice

Los **vértices** constituyen uno de los dos elementos que forman un grafo. Como ocurre con el resto de las ramas de las matemáticas, a la Teoría de Grafos no le interesa saber qué son los vértices, sin embargo también los podemos llamar **nodos**, que se representan como "**puntos**" en la gráfica.

Diferentes situaciones en las que pueden identificarse objetos y relaciones que satisfagan la definición de grafo pueden verse como grafos y así aplicar la Teoría de Grafos en ellos.

Arista

Una **arista** es una **línea**, orientada o no, que une dos vértices. Corresponde a una **relación** entre dos vértices de un grafo. En un grafo no dirigido, se trata de relaciones simétricas sin dirección, mientras que en un grafo dirigido son relaciones direccionales, también conocidas como **arcos**.

Grafo

Un **grafo** es una **relación** entre dos **conjuntos** `G = (V, A)`, donde `V` es el conjunto de **vértices** y `A` es el conjunto de **aristas**; este último es un conjunto de pares de la forma `(u, v)` tal que `u, v ∈ V`. También se pueden usar otras notaciones equivalentes para indicar una arista: `(a, b) = ab = a → b`; la última forma se aplica cuando el grafo es dirigido; las otras dos pueden usarse en grafos no dirigidos, aunque el contexto también puede indicar lo contrario.

En teoría de grafos, sólo queda lo esencial del dibujo: la forma de las aristas no son relevantes, sólo importa a qué vértices están unidas. La posición de los vértices tampoco importa, y se puede variar para obtener un dibujo más claro.

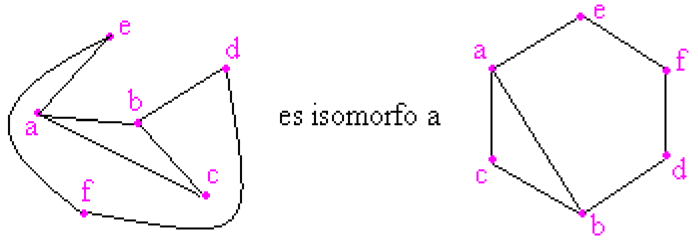


Figura 6.2. Grafos isomorfos[5]. V = { a, b, c, d, e, f }, y A = { ab, ac, ae, bc, bd, df, ef }.

Muchas situaciones pueden ser modeladas con un grafo: redes de uso cotidiano, una red de carreteras que conecta ciudades, una red eléctrica o la red de gas de una ciudad, el control de flujo de un programa (los vértices pueden ser sentencias de control y los arcos la transferencia del flujo), etc.

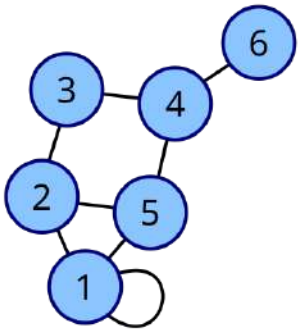


Figura 6.3. Grafo no dirigido

Subgrafo

Un **subgrafo** de un grafo `G` es un grafo cuyos conjuntos de vértices y aristas son subconjuntos de los de `G`. Se dice que un grafo `G` contiene a otro grafo `H` si algún subgrafo de `G` es `H` o es isomorfo a `H` (dependiendo de las necesidades de la situación).

[5] "Isomorfo" se refiere a objetos que tienen la misma forma o estructura, incluso si están compuestos de diferentes materiales o tienen propiedades distintas. En varios contextos, el concepto de "isomorfismo" describe una relación de equivalencia estructural.

El **subgrafo inducido** de G es un subgrafo G' de G tal que contiene todas las aristas adyacentes al subconjunto de vértices de G.

Definición
Sea $G = (V, A)$. $G' = (V', A')$ se dice subgrafo de G si:

3. $V' \subseteq V$
4. $A' \subseteq A$
5. (V', A') es un grafo

Si $G' = (V', A')$ es subgrafo de G, para todo $v \in G$ se cumple $gr(G', v) \leq gr(G, v)$

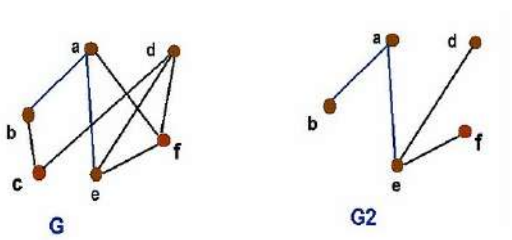


Figura 6.4. G2 es un subgrafo de G.

Grafo dirigido

Un **grafo dirigido** G consiste en un conjunto de vértices V y un conjunto de arcos dirigidos A. Un arco o arista es un **par ordenado** de vértices (v, w); v es la *cola* y w es la *cabeza* de la arista.

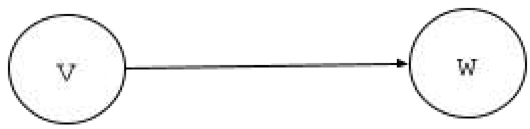


Figura 6.5. Representación gráfica del arco $(v, w) = v \rightarrow w$.

En la figura puede verse como la "cola" está en el vértice v, mientras que la punta de la flecha indica la "cabeza" en el vértice w del par ordenado. Se dice que el arco $v \rightarrow w$ va de v a w, y que w es adyacente a v.

Los vértices y los arcos pueden tener una **etiqueta**, la cual puede representar un nombre o un valor. En la figura 6.5 los nodos están etiquetados como v y w respectivamente.

Ejemplo 5.1
Grafo dirigido con 6 vértices (nodos) y 8 aristas (arcos). Cada vértice y cada arista está etiquetada

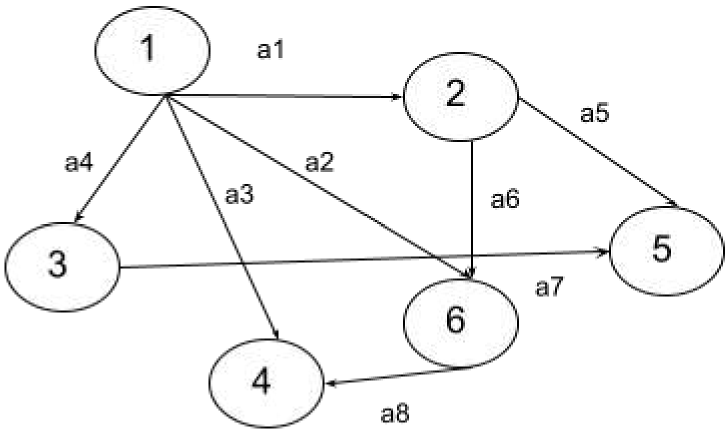


Figura 6.6. Grafo dirigido con 6 vértices (nodos) y 8 aristas (arcos)

V = {1, 2, 3, 4, 5, 6}
A = {a1, a2, a3, a4, a5, a6, a7, a8} = {(1, 2), (1, 6), (14), (1, 3), (2, 5), (2, 6), (3, 5), (6, 4)}

Observe cómo el orden de las parejas ordenadas en el conjunto A de las aristas especifican su dirección.

Camino en un grafo

Un **camino en un grafo** es una secuencia de vértices $v_1, v_2, \ldots, v_n$ tal que $v_1 \rightarrow v_2$, $v_2 \rightarrow v_3, \ldots, v_{n-1} \rightarrow v_n$ son arcos. Este camino va del vértice v_1 al vértice v_n, pasa por los vértices $v_2, v_3, \ldots, v_{n-1}$ y termina en el vértice v_n.

Longitud de un camino

La **longitud de un camino** es el número de arcos en ese camino. Como consecuencia, se tiene que la longitud de un camino $v \rightarrow v$ es 0.

Camino simple

Un camino es simple si todos sus vértices, excepto quizá el primero y el último son distintos.

Ciclo simple

Es un camino simple de longitud por lo menos uno, que empieza y termina en el mismo vértice.

Camino hamiltoniano
En teoría de grafos es una trayectoria que visita cada vértice de un grafo exactamente una vez.

Ciclo hamiltoniano

Si en un camino hamiltoniano, el recorrido comienza y termina en el mismo vértice, se llama **ciclo hamiltoniano** o **circuito hamiltoniano**. Es un ciclo donde se recorren todos los vértices exactamente una vez, excepto quizá el vértice del que parte y al cual llega.

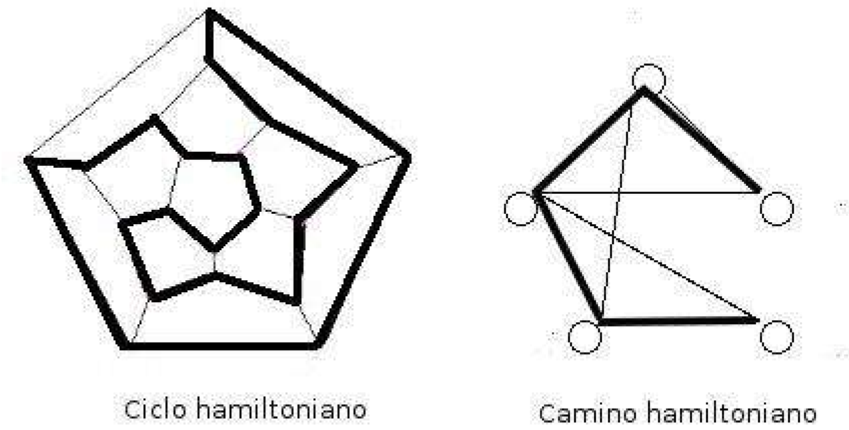


Figura 6.7. Ciclo y camino hamiltoniano[6]

Ejemplo 5.2
En el grafo de la figura 6.6 la secuencia (camino) 1, 2, 6, 4 es tal que 1 → 2, 2 → 6, 6 → 4 son arcos y tiene una longitud de camino igual a 3; la secuencia 1, 3, 5 con los arcos 1 → 3, 3 → 5, tiene una longitud igual a 2. Se observa que no hay ciclos simples en este grafo.

Grafos no dirigidos

Un grafo no dirigido (muchas veces descrito simplemente como grafo) G = (V, A) consta de un conjunto finito de vértices V y de un conjunto de aristas A. Se diferencia de un grafo dirigido en que cada arista en A es un par no ordenado de vértices.

Estructuras de datos en la representación de grafos

Existen diferentes formas de almacenar grafos en un computador. La estructura de datos depende de las características del grafo y el algoritmo usado para manipularlo. Entre las estructuras más sencillas y usadas se encuentran las listas y las matrices, aunque frecuentemente se usa una combinación de ambas. Las listas son preferidas en grafos dispersos porque tienen un eficiente uso de la memoria. Por otro lado, las matrices proveen acceso rápido, pero pueden consumir grandes cantidades de memoria.

[6] Imagen tomada de: https://poliformat.upv.es

Estructura de lista

3. **Lista de incidencia**: las aristas son representadas con un vector de pares (ordenados, si el grafo es dirigido), donde cada par representa una de las aristas.

4. **Lista de adyacencia**: cada vértice tiene una lista de vértices los cuales son adyacentes a él. Esto causa redundancia en un grafo no dirigido (ya que `A` existe en la lista de adyacencia de `B` y viceversa), pero las búsquedas son más rápidas.

En esta estructura de datos la idea es asociar a cada vértice `i` del grafo una lista que contenga todos aquellos vértices `j` que sean adyacentes a él. De esta forma sólo reservará memoria para los arcos adyacentes a `i` y no para todos los posibles arcos que pudieran tener como origen `i`. El grafo, por tanto, se representa por medio de un vector de `n` componentes (`|V| = n`) donde cada componente va a ser una lista de adyacencia correspondiente a cada uno de los vértices del grafo. Cada elemento de la lista consta de un campo indicando el vértice adyacente. En caso de que el grafo sea etiquetado, habrá que añadir un segundo campo para mostrar el valor de la etiqueta.

Estructuras matriciales

5. **Matriz de incidencia**: el grafo está representado por una matriz de `A` (aristas) por `V` (vértices), donde `[arista, vértice]` contiene la información de la arista (1 - conectado, 0 - no conectado). También puede representarse con una matriz de `V` (vértices) por `A` (aristas), donde `[vértice, arista]` contiene la información de la arista (1 - conectado, 0 - no conectado).

6. **Matriz de adyacencia**: el grafo está representado por una matriz cuadrada `M` de tamaño n^2, donde `n` es el número de vértices. Si hay una arista entre un vértice `x` y un vértice `y`, entonces el elemento $m_{x,y}$ es `1`, de lo contrario, es `0`.

Operaciones básicas con grafos en computación

Las operaciones básicas con grafos en computación incluyen la comprobación de la existencia de una arista entre dos vértices, la obtención de vértices adyacentes, la inserción y eliminación de vértices y aristas, y la determinación de la adyacencia de un vértice a otro. Aquí un listado de las más comunes:

3. Inserción de un vértice: agrega un nuevo vértice al grafo.

4. Recorrido del grafo: permite visitar todos los vértices y aristas del grafo, normalmente en un orden determinado.

5. Determinación de la adyacencia: verifica si un vértice es adyacente a otro.

6. Búsqueda de un vértice: permite encontrar un vértice específico en el grafo.

7. Comprobación de la existencia de una arista: determina si existe una conexión (arista) entre dos vértices específicos.

8. Obtención de vértices adyacentes: devuelve la lista de vértices a los que un vértice dado está conectado.

9. Eliminación de un vértice: borra un vértice y todas las aristas que lo conectan a otros vértices.

10. Inserción de una arista: añade una nueva conexión entre dos vértices existentes.

11. Eliminación de una arista: elimina la conexión entre dos vértices.

Ejemplo 5.3
La siguiente figura muestra un grafo no dirigido con 6 vértices y 8 arcos. Encontrar y representar:

9. Matriz de incidencia
10. Matriz de adyacencia
11. Lista de incidencia
12. Lista de adyacencia

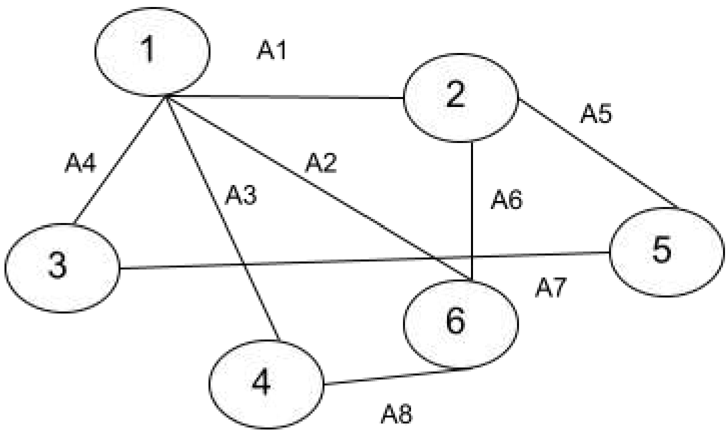


Figura. Grafo no dirigido con 6 vértices (nodos) y 8 aristas (arcos)

Matriz de incidencia
Esta matriz está conformada de `m` vértices y `n` aristas. En un vector columna almacenamos los vértices del grafo y en un vector fila almacenamos sus aristas. Si `matriz[vértice, arista] = 1` es por que el vértice está conectado con la arista, y es igual a cero en caso contrario.

Vertice / Arista	A1	A2	A3	A4	A5	A6	A7	A8
1	1	1	1	1	0	0	0	0
2	1	0	0	0	1	1	0	0

3	0	0	0	1	0	0	1	0
4	0	0	1	0	0	0	0	1
5	0	0	0	0	1	0	1	0
6	0	1	0	0	0	1	0	1

Matriz de adyacencia

Esta matriz está conformada de m filas por m columnas, donde m es el número de vértices del grafo, esto es, es una matriz cuadrada. Si `matriz[i, j] = 1` es por que el vértice `i` está conectado con el vértice `j`, y es igual a cero en caso contrario. Usamos el vector de vértices implementado para la matriz de incidencia y lo utilizamos tanto como vector fila como vector columna.

Vértice	1	2	3	4	5	6
1	0	1	1	1	0	1
2	1	0	0	0	1	1
3	1	0	0	0	1	0
4	1	0	0	0	0	1
5	0	1	1	0	0	0
6	1	1	0	1	0	0

Lista de incidencia

Creamos un vector para almacenar las aristas del grafo, donde cada arista es una pareja (ordenada si el grafo es dirigido) (v_i, v_j), $0 \leq i, j < n$ (n total de vértices), $v_i, v_j \in V$.

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

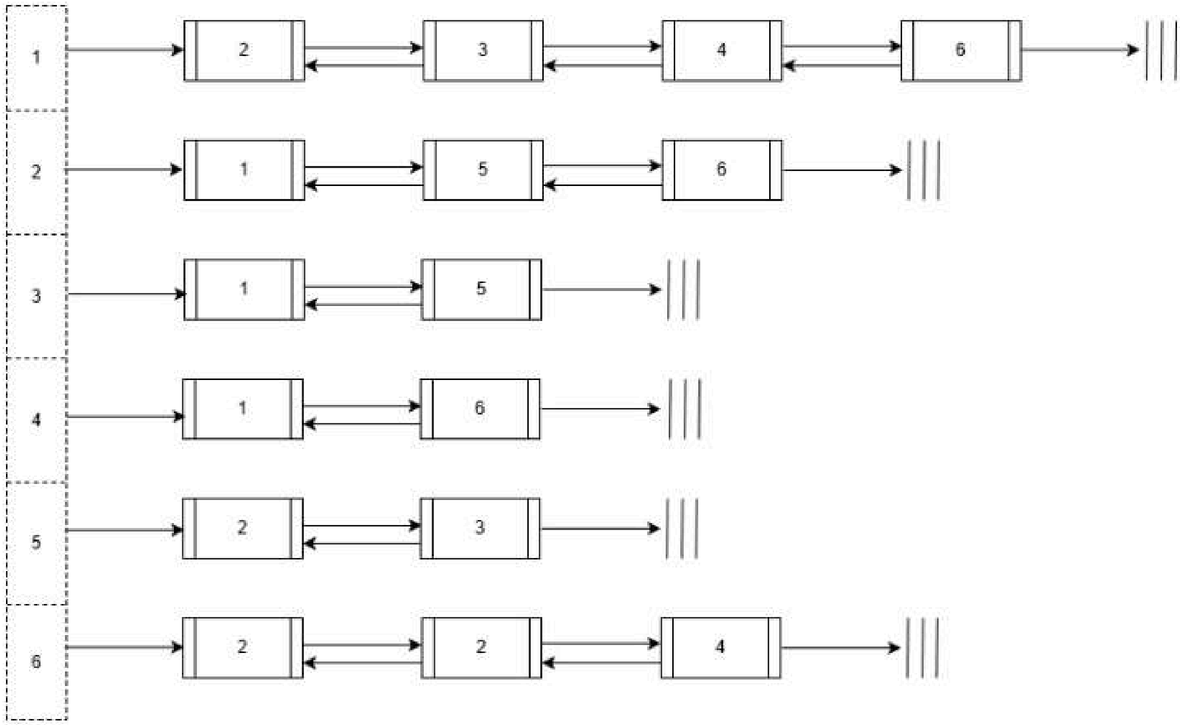
Lista de adyacencia

Creamos un vector (o lista) para almacenar los vértices y de cada uno de éstos se desprende una lista con los vértices adyacentes a éstos.

1	2	3	4	5	6
2	1	1	1	2	1
3	5	5	6	3	2
4	6	0	0	0	4
6	0	0	0	0	0

Esta representación puede verse como un vector que contiene cada vértice (nodo) del grafo, y de cada posición de éste, se desprende otro vector con los vértices adyacentes al vértice correspondiente. Esta representación puede llevarse a cabo con una matriz de n columnas, donde n es el número de nodos, sin embargo, es necesario rellenar con ceros los lugares vacíos donde hay adyacencia con un nodo cualquiera.

La representación con arreglos (vectores) y listas ligadas se puede ver como en la siguiente figura (aquí se usa una LDL, pero se puede usar cualquier tipo de lista ligada):



Ejemplo 5.4

Operaciones sobre grafos. Realizar las principales operaciones sobre grafos dirigidos y no dirigidos

Capítulo 6. Las compuertas lógicas[7]

¿Qué es una compuerta lógica?

En resumen, una ***compuerta lógica*** es la mínima operación digital que se puede realizar a nivel electrónico. Existen al menos 4 operaciones lógicas básicas, a saber:

- Negación: **NOT** (NO)
- Multiplicación: **AND** (Y)
- Suma: **OR** (O)
- Comparación: **XOR** (O BIEN)

El resto de las operaciones se realizan con las anteriores y sus respectivas negaciones. Una compuerta lógica es un conjunto de transistores que son capaces de realizar dichas operaciones. Estos son los bloques básicos con los que están construidos los sistemas digitales actuales.

Las *Compuertas Lógicas* son circuitos electrónicos conformados internamente por transistores que se encuentran con arreglos especiales con los que otorgan señales de voltaje como resultado en una salida de forma booleana que se obtiene por operaciones lógicas binarias. También niegan, afirman, incluyen o excluyen según sus propiedades lógicas. Estas compuertas se pueden aplicar en otras áreas de la ciencia como mecánica, hidráulica o neumática, además de las ciencias computacionales.

Existen diferentes tipos de compuertas y algunas de estas son más complejas, con la posibilidad de ser simuladas por compuertas más sencillas. Todas tienen una *tabla de verdad* asociada que explica los comportamientos en los resultados que otorga, dependiendo del valor booleano que tenga en cada una de sus entradas.

[7] Las imágenes son tomadas de: Las Compuertas Lógicas y sus Operaciones Lógicas

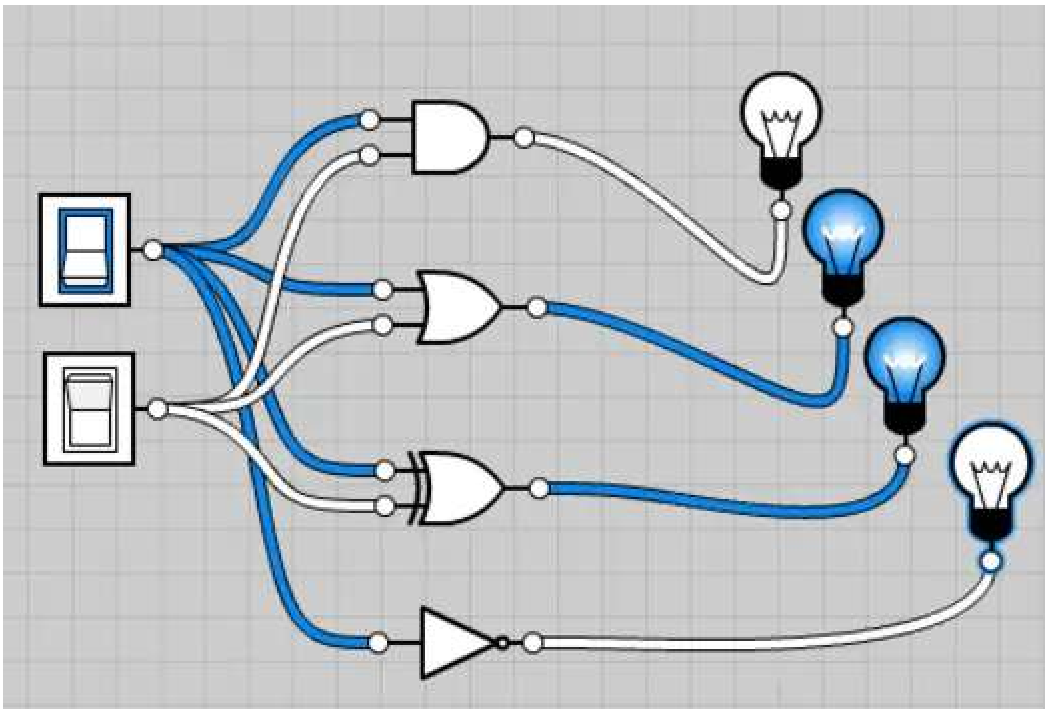


Figura. Ilustración de compuertas lógicas

Trabajan en dos estados, "**1**" o "**0**", (los cuales pueden asemejarse a la lógica proposicional con sus correspondientes valores “verdadero” y “falso” respectivamente). El estado **1** tiene un valor de **5V** como máximo y el estado **0** tiene un valor de **0V**, como mínimo y existiendo un umbral entre estos dos estados donde el resultado puede variar sin saber con exactitud la salida que nos entregará; **V** es la unidad de *voltaje* en el Sistema Internacional de Unidades SI . Las lógicas se explican a continuación:

La ***lógica positiva*** es aquella que con una señal en alto se acciona, representando un 1 binario y con una señal en bajo se desactiva. representado un 0 binario.

La ***lógica negativa*** proporciona los resultados inversamente, una señal en alto se representa con un 0 binario y una señal en bajo se representa con un 1 binario.

<u>Nota</u>

Las *proposiciones lógicas* vistas anteriormente serán análogas a las ***variables booleanas*** de esta sección y las *constantes lógicas* serán tratadas como ***constantes booleanas*** (**bits**) con los valores de verdad **1** para *Verdadero* y **0** para *Falso*, respectivamente. Se acostumbra nombrar las variables booleanas con las primeras letras del alfabeto latino en mayúscula: **A, B, C, …** Las diferentes *expresiones lógicas* serán tratadas como ***ecuaciones booleanas*** y tendrán sus respectivas *tablas de verdad*; así mismo se muestra la *representación gráfica* de las principales *compuertas lógicas*.

Sean A, B dos señales de entrada cualesquiera. Tenemos las siguientes compuertas lógicas:

Compuertas lógicas

Compuerta NOT

Esta compuerta tiene solo una entrada y una salida y ésta actúa como un inversor. Para esta situación en la entrada se pondrá un 1 y la salida otorgará un 0; en el caso contrario, ésta recibirá un 0 y entregará un 1. Por lo cual todo lo que llegue a su entrada, será inverso en su salida.

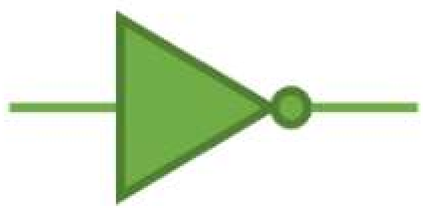
Si A es una señal de entrada, entonces la ecuación de la compuerta **NOT** es:

$A = A'$

Tabla de verdad de la compuerta NOT

A	A'
0	1
1	0

Representación de la compuerta NOT



Compuerta AND

Esta compuerta es representada por una multiplicación en el *Álgebra de Boole*. Indica que es necesario que en todas sus entradas se tenga un estado binario 1 para que la salida otorgue un 1 binario. En caso contrario de que falte alguna de sus entradas con este estado o no tenga siquiera una accionada, la salida no podrá cambiar de estado y permanecerá en 0. Esta puede ser simbolizada por dos o más interruptores en serie de los cuales todos deben estar activos para que esta permita el flujo de la corriente.

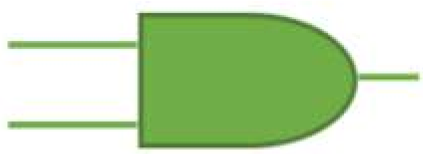
Si A y B son señales de entrada, entonces la ecuación de la compuerta **AND** es:

$C = A \times B$

Tabla de verdad de la compuerta AND

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

Representación de la compuerta AND



Compuerta OR

En el Álgebra de Boole esta es una suma binaria. Esta compuerta permite que con cualquiera de sus entradas en estado binario 1, su salida pasará a un estado 1 también. No es necesario que todas sus entradas estén accionadas para conseguir un estado 1 a la salida, y tampoco causa algún inconveniente. Para lograr un estado 0 a la salida, todas sus entradas deben estar en el mismo valor de 0. Se puede interpretar como dos interruptores en paralelo, que sin importar cual se accione, será posible el paso de la corriente.

Si A y B son señales de entrada, entonces la ecuación de la compuerta **OR** es:
$C = A + B$

Tabla de verdad de la compuerta OR

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Representación de la compuerta OR

Compuerta NAND

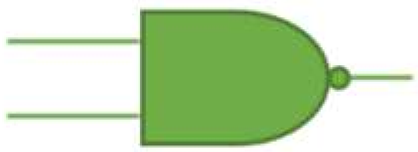
También denominada como ***AND negada***, esta compuerta trabaja al contrario de una AND, ya que al no tener entradas en 1 o solamente alguna de ellas, concede un 1 en su salida, pero si tiene todas sus entradas en 1 la salida se presenta con un 0.

Si A y B son señales de entrada, entonces la ecuación de la compuerta **NAND** es:
$C = (A \times B)'$

Tabla de verdad de la compuerta NAND

A	B	$(A \times B)'$
0	0	1
0	1	1
1	0	1
1	1	0

Representación de la compuerta NAND



Compuerta NOR

Así como vimos anteriormente con la compuerta NAND, la compuerta OR también tiene su versión inversa (***OR negado***). Esta compuerta cuando tiene sus entradas en estado 0 su salida estará en 1, pero si alguna de sus entradas pasa a un estado 1 sin importar en qué posición, su salida será un estado 0.

Si A y B son señales de entrada, entonces la ecuación de la compuerta **NOR** es:
$C = (A + B)'$

Tabla de verdad de la compuerta NOR

A	B	$(A + B)'$
0	0	1
0	1	0
1	0	0
1	1	0

Representación de la compuerta NOR

Compuerta XOR

También llamada ***OR exclusiva***, actúa como una suma binaria de un dígito cada uno y el resultado de dicha suma representa la salida. Otra manera de verlo, es que con valores de entrada igual el estado de salida es 0 y con valores de entrada diferente, la salida será 1.

Si A y B son señales de entrada, entonces la ecuación de la compuerta **XOR** es:
$C = A \oplus B$
$C = (A \times B') + (A' \times B)$

Tabla de verdad de la compuerta XOR

A	B	$A \oplus B$	A'	B'	$A \times B'$	$+$	$A' \times B$
0	0	0	1	1	0	0	0
0	1	1	1	0	0	1	1
1	0	1	0	1	1	1	0
1	1	0	0	0	0	0	0

Representación de la compuerta XOR

Compuerta XNOR

También conocida como ***NOR exclusiva***. Esta es todo lo contrario a la compuerta XOR, ya que cuando las entradas sean iguales se presentará una salida en estado 1 y si son diferentes la salida será un estado 0.

Si A y B son señales de entrada, entonces la ecuación de la compuerta **XNOR** es:
$C = (A \oplus B)'$
$C = (A \times B) + (A' \times B')$

Tabla de verdad de la compuerta XNOR

A	B	$(A \oplus B)'$	A'	B'	$A \times B$	$+$	$A' \times B'$
0	0	1	1	1	0	1	1
0	1	0	1	0	0	0	0
1	0	0	0	1	0	0	0
1	1	1	0	0	1	1	0

Representación de la compuerta XNOR

Nota
Observe que esta compuerta entrega las mismas salidas que el operador bicondicional tratado en el capítulo de Lógica Proposicional

Compuerta IF

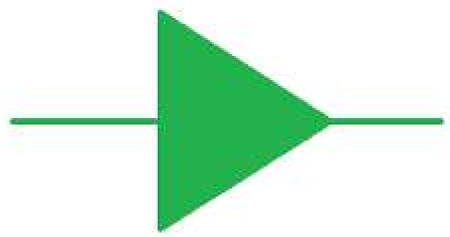
Esta compuerta no es una muy utilizada o reconocida, ya que su funcionamiento en estados lógicos es parecido a si solo hubiera un cable conectado, porque exactamente lo que se le ponga en la entrada, se encontrará en la salida. También es conocida como un ***buffer***, y en la práctica se utiliza como amplificador de corriente o como seguidor de tensión para adaptar impedancias, ya que puede aumentar la corriente suministrada a la salida mientras se retiene el estado lógico.

Si A es una señal de entrada, entonces la ecuación de la compuerta **IF** es:
$C = A$

Tabla de verdad de la compuerta IF o buffer

A	C
0	0
1	1

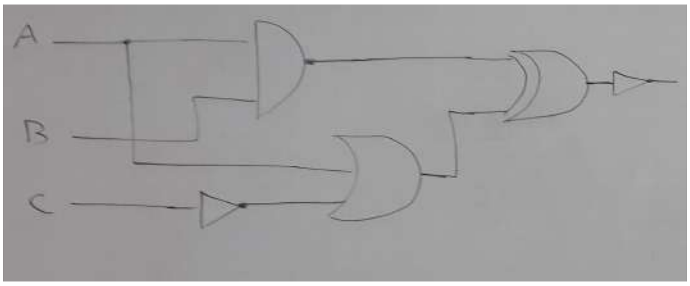
Representación de la compuerta IF



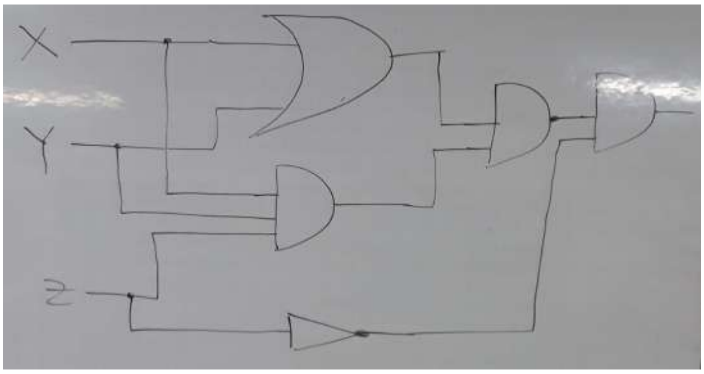
Preguntas

Ejercicios

1. Para las siguientes ecuaciones, diseñe el respectivo circuito lógico con su tabla de verdad
   a. AB + (A + C')(A'C) + B'
   b. (A + B + C)(A + B)' + (B'C')'
2. Dados los siguientes circuitos lógicos, determine su respectiva ecuación de salida y tabla de verdad
   a.



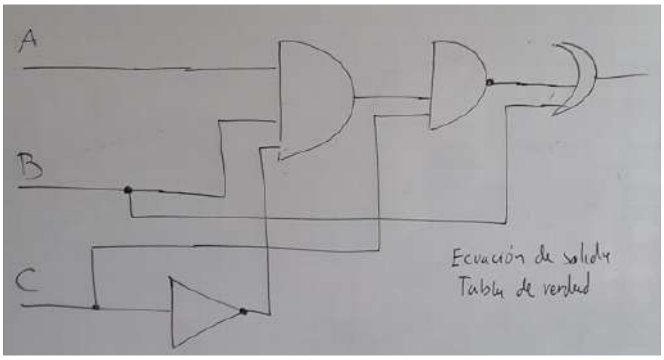
b.



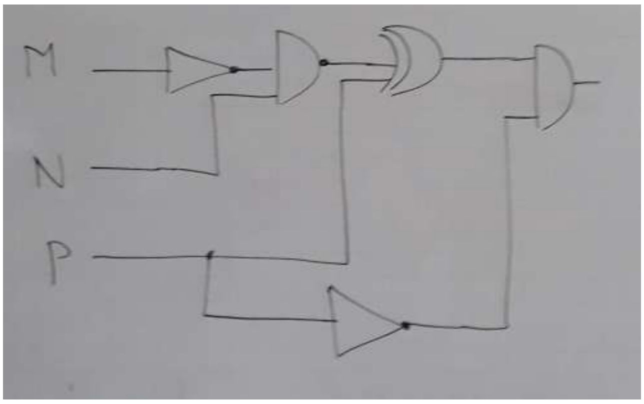
3. Muestre el circuito lógico para la siguiente ecuación de salida e indique cuál es su tabla de verdad

   $(A \oplus B) + (B + C') + ABC$

4. Determine la ecuación de salida y la tabla de verdad del siguiente circuito lógico



5. ¿Cuál es el circuito lógico y la tabla de verdad de la función de salida F = (XY + Z') + (XZ)' + X'?
6. ¿Cuál es la función de salida y la tabla de verdad del siguiente circuito lógico?


M
N
P

Capítulo 7. Álgebra booleana y mapas de Karnaugh

Funciones booleanas

El álgebra de Boole proporciona las reglas suficientes para trabajar con el conjunto $\{0, 1\}$ de números binarios y realizar operaciones sobre éste, permitiendo, entre otros aspectos, el desarrollo aplicado a la tecnología, por ejemplo en la creación de dispositivos electrónicos que pueden estudiarse utilizando dichas reglas, ya que se basan en mecanismos digitales, base de estos componentes.

Las tres operaciones fundamentales, y que permiten la creación de otros operadores, son:

- Negación o complemento: **`NOT, ‘`**
- Multiplicación o producto: **`AND, x`**
- Suma: **`OR, +`**

Sea el conjunto $B = \{0, 1\}$. Tenemos las siguientes definiciones/propiedades:

- Una *variable* es un símbolo que representa un elemento arbitrario del álgebra, ya sea una constante o una fórmula completa. La variable **x** se denomina **variable booleana** si asume únicamente valores del conjunto `B`
- Se llama **literal** a una variable booleana o al complemento de ésta
- Una **constante booleana** es cualquier elemento del conjunto `B`, así, tenemos que las constantes booleanas son **`0`** y **`1`**
- Una **operación booleana** es una expresión que contiene variables, constantes y operadores booleanos
- Una función siempre devuelve un valor; así, una **función booleana** tendrá como objetivo devolver un valor booleano, por tanto: $f(x) = 1 \vee f(x) = 0, \forall x \in B$. Para funciones de n variables también se cumple esta propiedad: $f(x_1, x_2, x_3, ..., x_n) = 1 \vee f(x_1, x_2, x_3, ..., x_n) = 0, \ donde \ x_i \in B, \ 1 \leq i \leq n$

Una **función booleana** se puede representar utilizando expresiones construidas a partir de constantes y variables booleanas y las operaciones fundamentales que pueden realizarse sobre ellas, tal como el complemento (negación) y la suma y multiplicación binarias.

Una **expresión booleana** en las variables $x_1, x_2, ..., x_n$ tiene las siguientes características:

- $0, 1, x_1, x_2, ..., x_n$ son expresiones booleanas
- Si E_1 y E_2 son expresiones booleanas, entonces $E'_1, (E_1 \times E_2) \ y \ (E_1 + E_2)$ también son expresiones booleanas

- Cada expresión booleana representa una **función booleana**. Los valores de esta función se obtienen sustituyendo `0` y `1` en las variables presentes en la expresión
- Dos funciones $f(x)\ y\ g(x)$ son **equivalentes** si $f(x) = g(x),\ x \in B$. Esta misma definición puede extenderse para dos funciones de n variables

Identidades del álgebra booleana

Son proposiciones equivalentes que se pueden demostrar utilizando tablas de verdad. Son particularmente útiles para simplificar el diseño de circuitos. Estas identidades son básicamente las mismas utilizadas en la lógica proposicional dentro del contexto de los circuitos digitales.

Sean $x, y, z\ variables\ booleanas$, se tienen las siguientes identidades del álgebra booleana:

Identidad	**Nombre**
$(x')' = x$	Doble negación, complemento o inversión
$x + x = x$ $x \times x = x$	Idempotencia para la suma y la multiplicación
$x + 0 = x$ $x \times 0 = 0$ $x + 1 = 1$ $x \times 1 = x$	Identidades para la suma y la multiplicación
$x + x' = 1$ $x \times x' = 0$	Inversos de la suma y la multiplicación (tercer excluido y contradicción)
$x + y = y + x$ $x \times y = y \times x$	Conmutativa para la suma y la multiplicación
$x + (y + z) = (x + y) + z$ $x \times (y \times z) = (x \times y) \times z$	Asociativa para la suma y la multiplicación
$x + (y \times z) = (x + y) \times (x + z)$ $x \times (y + z) = (x \times y) + (x \times z)$	Distributiva para la suma y la multiplicación
$(x + y)' = x' \times y'$ $(x \times y)' = x' + y'$ $(x + y + z)' = x' \times y' \times z'$ $(x \times y \times z)' = x' + y' + z'$	De Morgan para la suma y la multiplicación De Morgan para más de dos variables
$x + (x \times y) = x$ $x \times (x + y) = x$	Absorción para la suma y la multiplicación

Principio de dualidad booleana

Cualquier teorema o identidad algebraica deducible de las identidades anteriores, puede transformarse en un segundo teorema o identidad válida sin más que intercambiar `(+)` `por` `(x)` y **`1`** por **`0`**.

Representación de funciones booleanas

La tabla de verdad es la herramienta que permite representar una función booleana para cada combinación de entrada. Si la función está definida para todas las combinaciones se llama **función completa**, en caso contrario **función incompleta**.

Una **fórmula de conmutación** es la expresión booleana de una función lógica.

Veamos la representación de una función de 4 variables mediante la tabla de verdad con sus respectivas fórmulas de conmutación:

X_1	X_2	X_3	X_4	$F(X_1,X_2,X_3,X_4,)$
0	0	0	0	F(0, 0, 0, 0)
0	0	0	1	F(0, 0, 0, 1)
0	0	1	0	F(0, 0, 1, 0)
0	0	1	1	F(0, 0, 1, 1)
0	1	0	0	F(0, 1, 0, 0)
0	1	0	1	F(0, 1, 0, 1)
0	1	1	0	F(0, 1, 1, 0)
0	1	1	1	F(0, 1, 1, 1)
1	0	0	0	F(1, 0, 0, 0)
1	0	0	1	F(1, 0, 0, 1)

1	0	1	0	F(1, 0, 1, 0)
1	0	1	1	F(1, 0, 1, 1)
1	1	0	0	F(1, 1, 0, 0)
1	1	0	1	F(1, 1, 0, 1)
1	1	1	0	F(1, 1, 1, 0)
1	1	1	1	F(1, 1, 1, 1)

Ejemplo 6.1
Simplificar usando las leyes booleanas la siguiente función:

```
F(A, B, C) = (A + B)’ x (B’ + C’) + (A’ x C’)

⇔ (A’ x B’) x (B’ + C’) + (A’ x C’) → DeMorgan
⇔ [(A’ x B’ x B’) + (A’ x B’ x C’)] + (A’ x C’) → Asoc. y Distr.
⇔ [(A’ x B’) + C’ x (A’ x B’)] + (A’ x C’) → Asociativa y conmutativa
⇔ (A’ x B’) x [1 + C’] + (A’ x C’) → Factor común
⇔ (A’ x B’) x (1) + (A’ x C’) → Identidad
⇔ (A’ x B’) + (A’ x C’)] → Identidad
```

Formas canónica y estándar de las expresiones booleanas

Cualquier función booleana se puede expresar como una *suma* de **minitérminos** o como un *producto* de **maxitérminos**; a estas formas se les dice que están en **forma canónica o estándar**.

Para cualquier expresión booleana, existe una forma canónica o estándar **SOP** (***Sum Of Products***) y **POS** (***Product Of Sums***) equivalente. En una forma canónica todas las variables o sus complementos aparecen en cada uno de los términos, esto es, aparecen todos los literales en cada término.

Ejemplo 6.2
La función F de cuatro variables que se muestra a continuación contiene todos los literales en cada uno de los términos, por tanto se encuentra en forma canónica.

```
F(A, B, C, D) = A’B’CD’ + AB’CD + ABC’D’
ó
```

```
F(A, B, C, D) = (A’ x B’ x C x D’) + (A x B’ x C x D) + (A x B x C’ x D’)
```

Notas

- Dos funciones booleanas son **equivalentes** sí, y sólo si, sus formas canónicas son idénticas.
- La expresión algebraica en SOP o POS en la que no todos los términos son canónicos, recibe el nombre de **forma normalizada**.

Ejemplo 6.3
Forma normalizada y estándar de la función `F` que se muestra a continuación.

Forma normalizada SOP
$F(A,\ B,\ C)\ =\ (A \times B)\ +\ (A' \times B \times C')$

Forma canónica POS
$F(A,\ B,\ C)\ =\ (A'\ +\ B'\ +\ C) \times (A\ +\ B'\ +\ C) \times (A\ +\ B\ +\ C)$

Minitérmino (*minterms*) (m_i)

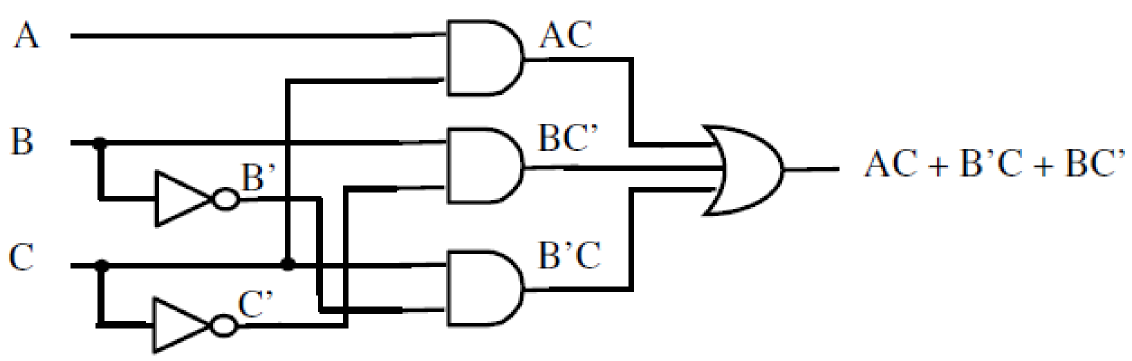
Término producto en el que aparecen todos los literales.

Maxitérmino (*maxterms*) (M_i)

Término suma en el que aparecen todos los literales.

Suma de productos SOP

Es una expresión booleana compuesta por dos o más términos formados por multiplicaciones de literales que son sumadas.



Ejemplo 6.4
Algunas sumas de productos:

```
(A x B) + (C x D)
```

`(A x B) + (B x C’) + (A’ x B x C)`

Fórmula canónica disyuntiva o de minitérminos: suma de minitérminos (SOP)

Dada la lista completa de minitérminos y asignando `1’s` y `0’s` arbitrariamente a las variables, siempre hay uno, y sólo un minitérmino que toma el valor `1`. Un minitérmino es un *término producto* que es `1` exactamente en una línea de la tabla de verdad. La fórmula compuesta por todos los mintérminos será idénticamente `1`. Cada fórmula de conmutación puede expresarse como **suma de minitérminos**, y esa fórmula es única. Un mintérmino se designa por m_i siendo `i` el número decimal correspondiente de la tabla de verdad. Para el producto, el `0` se asocia a la variable complementada y el `1` a la variable sin complementar.

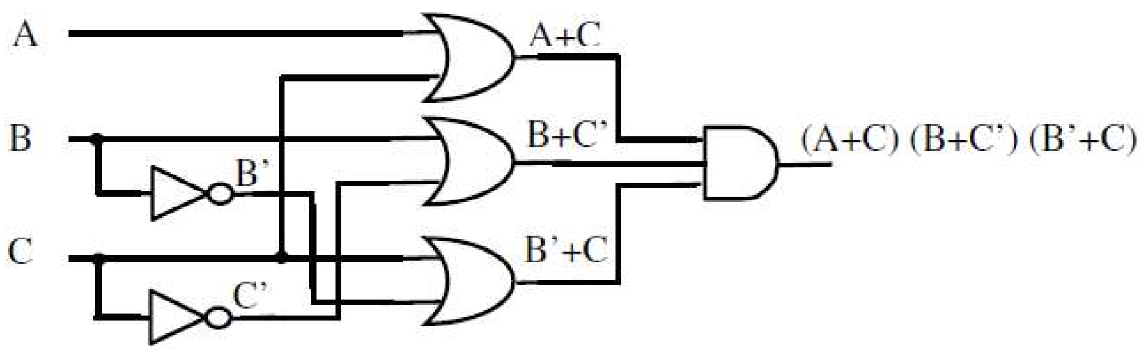
El método para encontrar la expresión equivalente en forma canónica SOP se basa en la propiedad `A + A’ = 1`, para multiplicar por `1`

Para encontrar la forma canónica, seguimos estos pasos:

- Creamos la tabla de verdad del enunciado
- Referenciamos las filas de la solución que contengan una salida de `1`. En dicha fila obtenemos el *minitérmino* con cada variable: si su valor es `1`, se toma igual; si es `0`, se toma negada (complemento) y multiplicamos estos literales
- La suma de cada minitérmino nos da la forma canónica SOP de la expresión original

Producto de sumas POS

Es una expresión booleana compuesta por dos o más términos formados por sumas de literales que son multiplicadas.



<u>Ejemplo 6.5</u>
Algunas sumas de productos:

`(A + B) x (C + D)`
`(A + B) x (B + C’) x (A’+ B + C)`

Fórmula canónica conjuntiva o de maxitérminos: producto de maxitérminos (POS)

Dada la lista completa de maxitérminos y asignando `1's` y `0's` arbitrariamente a las variables, siempre hay uno, y sólo un maxitérmino que toma el valor `0`. Un maxitérmino es un *término suma* que es `0` exactamente en una línea de la tabla de verdad. La fórmula compuesta por todos los maxitérminos será idénticamente 0. Cada fórmula de conmutación puede expresarse como **producto de maxitérminos**, y esa fórmula es única. Un maxitérmino se designa por `Mi` siendo `i` el número decimal correspondiente de la tabla de verdad. En la suma, el `1` se asocia a la variable complementada y el `0` a la variable sin complementar.

El método para encontrar la expresión equivalente en forma canónica POS se basa en la propiedad `A x A' = 0`, para sumar `0`

Para encontrar la forma canónica, seguimos estos pasos:

- Creamos la tabla de verdad del enunciado
- Referenciamos las filas de la solución que contengan una salida de `0`. En dicha fila obtenemos el *maxitérmino* con cada variable: si su valor es `0`, se toma igual; si es `1`, se toma negada (complemento) y sumamos estos literales
- La suma de cada maxitérmino nos da la forma canónica POS de la expresión original

Ejemplo 6.6

Sea `F = (A x B) + (A x B x C')`. Obtener las formas canónicas SOP y POS equivalentes.

Creamos la tabla de verdad para la función `F`:

A	B	C	(A x B)	+	(A x B x C')
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	1	1	0

La forma canónica SOP equivalente es

`(A x B x C') + (A x B x C)`

Podemos verificar que esta expresión es equivalente resolviendo su correspondiente tabla de verdad

A	B	C	(A x B x C’)	+	(A x B x C)
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	1	1	0
1	1	1	0	1	1

La forma canónica POS equivalente es

(A + B + C) x (A + B + C’) x (A + B’ + C) x (A + B’ + C’) x (A’ + B + C) x (A’ + B + C’)

Podemos verificar que esta expresión es equivalente resolviendo su correspondiente tabla de verdad

A	B	C	(A+B+C)	x	(A+B+C’)	x	(A+B’+C)	x	(A+B’+C’)	x	(A’+B+C)	x	(A’+B+C’)
0	0	0	0	0	1	0	1	0	1	0	1	0	1
0	0	1	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	1	1	0	0	0	1	0	1	0	1
0	1	1	1	1	1	1	1	0	0	0	1	0	1
1	0	0	1	1	1	1	1	1	1	0	0	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1

Observe que las 3 tablas de verdad conllevan al mismo resultado, indicando que las 3 funciones correspondientes son equivalentes

Mapa de Karnaugh (K-Map)

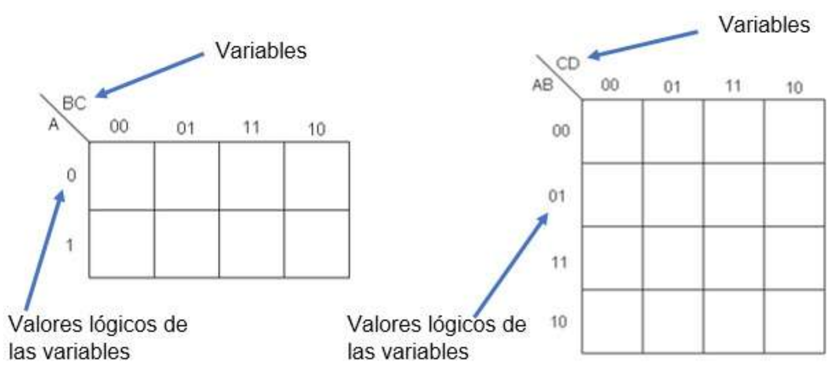
Los **mapas de Karnaugh (K-Maps)** son una herramienta empleada para la simplificación de funciones lógicas. A diferencia de la simplificación por las leyes del álgebra booleana, los mapas son un método gráfico que implica conocer las representaciones canónicas SOP y POS de las funciones.

El Mapa de Karnaugh tiene la característica de que puede ser visto como una representación en dos dimensiones de la tabla de verdad. En la tabla de verdad se ubican las variables en columnas y las combinaciones de tales variables determinan un valor de salida, `0` o `1`, sin embargo, en el mapa las variables se ponen como si se tratara de un plano cartesiano, respetando cada una de las combinaciones que de ellas se generan, y colocando en la intersección de las combinaciones de las variables, el valor de salida.

Una de las ventajas que traen los mapas, es que no es necesario realizar operaciones algebraicas, además de que la función de salida se entrega minimizada.

Los mapas muestran la relación que existe entre las entradas y las salidas de un circuito lógico, si se aplica adecuadamente el resultado será el más simplificado posible. Pueden ser utilizados para cualquier número de variables de entrada, sin embargo se recomienda un máximo de seis variables.

En la siguiente figura vemos dos ejemplos de la representación de los mapas de Karnaugh con 3 y 4 variables.[8]



Observe la forma de ordenar las variables y los valores lógicos que puede tener cada variable o combinación de variables. Esta forma de numerar las filas y columnas se conoce como **código binario reflejado** o **código Gray**, nombrado así en honor del investigador Frank Gray; éste es un sistema de numeración binario en el que dos números consecutivos difieren solamente en uno de sus dígitos (de ahí que la fila y la columna ‘11’ en la figura anterior no sean las últimas, para no cambiar dos bits a la vez).

El código Gray fue diseñado en 1947 en los laboratorios Bell originalmente para prevenir señales ilegales (señales falsas o viciadas en la representación) de los switches electromecánicos, y actualmente es usado para facilitar la corrección de errores en los

[8] Imagen tomada de [Mapas de Karnaugh – Sistemas Digitales]

sistemas de comunicaciones, tales como los sistemas de televisión por cable y la televisión digital terrestre. En los K-Maps se aplica el código Gray en la nomenclatura de las filas y columnas, minimizando así los errores en las simplificaciones de expresiones booleanas al cambiar solo un dígito cuando hay una transición de valores de la variable.

En los sistemas de codificación tradicionales, el cambio de posición de un bit puede provocar errores en la lectura. El código Gray, al cambiar solo un bit a la vez, reduce significativamente la posibilidad de estos errores durante la transición entre valores.

Ejemplo 6.7

Convertir la siguiente tabla de verdad de dos (2) variables que tiene una salida S en mapa de Karnaugh:

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

B	0	1
A		
0	1	0
1	0	1

Ejemplo 6.8

Convertir la siguiente tabla de verdad de tres (3) variables que tiene una salida S en mapa de Karnaugh:

A	B	C	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0

1	1	1	1

BC	0 0	0 1	1 1	1 0
A				
0	0	1	0	0
1	1	1	1	0

Ejemplo 6.9

Convertir la siguiente tabla de verdad de cuatro (4) variables que tiene una salida S en mapa de Karnaugh:

A	B	C	D	S
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

CD	0 0	0 1	1 1	1 0
AB				

0 0	0	0	0	1
0 1	1	0	0	1
1 1	1	0	1	1
1 0	0	0	0	1

Ejemplo 6.10
Mapa de Karnaugh de cinco (5) variables A, B, C, D, E que tiene alguna salida `S`. (Se deja como ejercicio crear la tabla de verdad correspondiente a este K-Map)

CDE / AB	000	001	011	010	110	111	101	100
0 0	0	0	1	1	1	0	0	0
0 1	1	0	1	1	0	0	1	1
1 1	0	0	1	0	0	0	0	1
1 0	1	0	1	1	0	1	0	0

Minimización de una SOP

Para minimizar una expresión dada en *Suma de Productos*, seguimos los siguientes pasos:

- Ponemos un **1** en el mapa de Karnaugh por cada término de la expresión SOP en la celda correspondiente
- Las demás celdas se rellenan con **0**
- Agrupar encerrando todos los 1
- La agrupación de estos "unos" debe ser en forma de **rectángulos** o **cuadrados**, no se puede en triángulos o diagonales
- Mientras más "unos" encerremos en cuadrados, más variables se podrán eliminar
- El mapa se puede "doblar" con el objetivo de encontrar más "unos" que se puedan encerrar
- Para encerrar "unos", se debe cumplir que sea una cantidad igual a 2^n, `1 <= n <= 3`, es decir, `1, 2, 4 y 8` "unos"; otro valor no es válido. Para más variables, tomamos `n = 4, n = 5`, etc., para obtener una cantidad de "unos" igual a `16, 32`, etc.
- Se determina la expresión **SOP mínima** a partir del mapa de Karnaugh
- Tendremos presentes las reglas del álgebra booleana: `A + A = A`, (idempotencia) y `A + A' = 1` (inverso -esto equivale a multiplicar por 1-)

Ejemplo 6.11
Hallar la función mínima de la expresión `A'B'C + A'BC' + ABC' + ABC` usando mapas de Karnaugh.

Tenemos una SOP. Recordemos que si tenemos un 1, la variable se toma normal, y si tenemos un 0 se toma negada.

$A'B'C \ + \ A'BC' \ + \ ABC' \ + \ ABC$
$0\ 0\ 1 \quad 0\ 1\ 0 \quad 1\ 1\ 0 \quad 1\ 1\ 1$

Tenemos tres variables, por tanto nuestra matriz para el mapa de Karnaugh tendrá ocho (8) celdas:

$n = 3;\ 2^{n} = 2^{3} = 8$

BC	0 0	0 1	1 0	1 1
A				
0	0	1	1	0
1	0	0	1	1

Primer rectángulo (rojo)
No hay cambio en las variables, las tomamos de acuerdo a su valor.
$A'B'C$

Segundo rectángulo (Verde)
A cambia de valor (de 0 a 1), por tanto no la tomamos.
BC'

Tercer rectángulo (Azul)
C cambia de valor (de 0 a 1), por tanto no la tomamos.
AB

Por tanto, la función mínima está dada por la suma de los productos encontrados:

$f_{minsop} = A'B'C + BC' + AB$

<u>Nota: justificación de la simplificación</u>
Veamos cómo se obtiene la simplificación de las expresiones a partir del mapa de Karnaugh teniendo presente las leyes del álgebra booleana mencionadas arriba.

Para el segundo rectángulo en verde:
Hacemos una POS para obtener:
$(A' + A) \times (B + B) \times (C' + C') = 1 \times B \times C' = BC'$

Para el tercer rectángulo en azul:
Hacemos una POS para obtener:
$(A + A) \times (B + B) \times (C' + C) = A \times B \times 1 = AB$

<u>Ejemplo 6.12</u>
Hallar la función mínima de la expresión `A’B’C’D + AB’C’D’ + A’B’CD + AB’C’D` usando mapas de Karnaugh.

Tenemos una SOP.

$A'B'C'D + AB'C'D' + A'B'CD + AB'C'D$
0 0 0 1 1 0 0 0 0 0 1 1 1 0 0 1

Tenemos cuatro variables, por tanto nuestra matriz para el mapa de Karnaugh tendrá dieciséis (16) celdas:

$n = 4;\ 2^n = 2^4 = 16$

<u>Observación</u>
Escribimos las filas y columnas en este orden para evitar el problema de adyacencia en cambio de valor de las variables, ya que solo es permitido un cambio de valor. Si ponemos la última fila y columna como (1, 1) tendríamos un doble cambio de variable al tener dos variables, lo cual no es permitido

CD	0 0	0 1	1 1	1 0
AB				
0 0	0	1	1	0
0 1	0	0	0	0
1 1	0	0	0	0
1 0	1	1	0	0

Primer rectángulo (rojo)
Hay un cambio en el valor de la variable C, por tanto la descartamos.
$A'B'D$

Segundo rectángulo (Verde)
Hay un cambio en el valor de la variable D, por tanto la descartamos.
$AB'C'$

Por tanto, la función mínima está dada por la suma de los productos encontrados:

$f_{minsop} = A'B'D + AB'C'$

<u>Justificación</u>
Sumamos dos términos por tener encerrados dos “unos” en cada rectángulo. Si se encierran más “unos”, entonces se incluyen más términos en la suma.

Para el primer rectángulo:
$(A' + A') \times (B' + B') \times (C' + C) \times (D + D) = A' \times B' \times 1 \times D = A' \times B' \times D = A'B'D$

Para el segundo rectángulo:
$(A + A) \times (B' + B') \times (C' + C') \times (D' + D) = A \times B' \times C' \times 1 = A \times B' \times C' = AB'C'$

Minimización de una POS

Para minimizar una expresión dada en *Productos de Sumas*, seguimos los siguientes pasos:

- Ponemos un **0** en el mapa de Karnaugh por cada término de la expresión POS en la celda correspondiente
- Las demás celdas se rellenan con **1**
- Agrupar encerrando todos los 0
- La agrupación de estos "ceros" debe ser en forma de **rectángulos** o **cuadrados**, no se puede en triángulos o diagonales
- Mientras más "ceros" encerremos en cuadrados, más variables se podrán eliminar
- El mapa se puede "doblar" con el objetivo de encontrar más "ceros" que se puedan encerrar
- Para encerrar "ceros", se debe cumplir que sea una cantidad igual a 2^n, `1 <= n <= 3`, es decir, `1, 2, 4` y `8` "ceros"; otro valor no es válido. Para más variables, tomamos `n = 4, n = 5`, etc., para obtener una cantidad de "ceros" igual a `16, 32`, etc.
- Se determina la expresión **POS mínima** a partir del mapa de Karnaugh
- Tendremos presentes las reglas del álgebra booleana: `A x A = A`, (idempotencia) y `A x A' = 0` (inverso -esto equivale a sumar 0-)

Ejemplo 6.13

Hallar la función mínima de la expresión `(A' + B' + C) x (A' + B + C') x (A + B + C') x (A + B + C)` usando mapas de Karnaugh.

Tenemos un POS. Recordemos que si tenemos un 0, la variable se toma normal, y si tenemos un 1 se toma negada.

```
(A' + B' + C) x (A' + B + C') x (A + B + C') x (A + B + C)
 1    1    0     1    0   1      0   0   1      0   0   0
```

Tenemos tres variables, por tanto nuestra matriz para el mapa de Karnaugh tendrá ocho (8) celdas:

$n = 3;\ 2^n = 2^3 = 8$

BC	0 0	0 1	1 1	1 0
A				
0	0	0	1	1
1	1	0	1	0

Primer rectángulo (rojo)
No hay cambio en las variables, las tomamos de acuerdo a su valor.
A’ + B’ + C

Segundo rectángulo (Verde)
A cambia de valor (de 0 a 1), por tanto no la tomamos.
B + C’

Tercer rectángulo (Azul)
C cambia de valor (de 0 a 1), por tanto no la tomamos.
A + B

Por tanto, la función mínima está dada por el producto de sumas encontradas:
F_{minpos} = (A’ + B’ + C) x (B + C’) x (A + B)

<u>Nota: justificación de la simplificación</u>
Veamos cómo se obtiene la simplificación de las expresiones a partir del mapa de Karnaugh teniendo presente las leyes del álgebra booleana mencionadas arriba.

Para el segundo rectángulo en verde:
Hacemos una SOP para obtener:
(A' x A) + (B x B) + (C' x C') = 0 + B + C' = B + C'

Para el tercer rectángulo en azul:
Hacemos una SOP para obtener:
(A x A) + (B x B) + (C' x C) = A + B + 0 = A + B

<u>Ejemplo 6.14</u>
Hallar la función mínima de la expresión (A’ + B’ + C’ + D) x (A + B’ + C’ + D’) x (A’ + B’ + C + D) x (A + B’ + C’ + D) usando mapas de Karnaugh.

Tenemos una POS.

```
(A’ + B’ + C’ + D)x(A + B’ + C’ + D’)x(A’ + B’ + C + D)x(A + B’ + C’ + D)
 1    1    1    0   0    1    1    1    1    1    0   0   0    1    1    0
```

Tenemos cuatro variables, por tanto nuestra matriz para el mapa de Karnaugh tendrá dieciséis (16) celdas:

$n = 4;\ 2^n = 2^4 = 16$

Observación
Escribimos las filas y columnas en este orden para evitar el problema de adyacencia en cambio de valor de las variables, ya que solo es permitido un cambio de valor. Si ponemos la última fila y columna como (1, 1) tendríamos un doble cambio de variable al tener dos variables, lo cual no es permitido.

CD / AB	0 0	0 1	1 1	1 0
0 0	1	1	1	1
0 1	1	1	0	0
1 1	0	1	1	0
1 0	1	1	1	1

Primer rectángulo (rojo)
Hay un cambio en el valor de la variable C, por tanto la descartamos.
A + B’ + C’

Segundo rectángulo (Verde)
Hay un cambio en el valor de la variable D, por tanto la descartamos.
A’ + B’ + D

Por tanto, la función mínima está dada por el producto de las sumas encontradas:
F_{minpos} = (A + B’ + C’) x (A’ + B’ + D)

Justificación
Multiplicamos dos términos por tener encerrados dos “ceros” en cada rectángulo. Si se encierran más “ceros”, entonces se incluyen más términos en el producto.

Para el primer rectángulo:
$(A \times A) + (B' \times B') + (C' \times C') + (D' \times D) = A + B' + C + 0 = A + B' + C$

Para el segundo rectángulo:
$(A' \times A') + (B' \times B') + (C' \times C) + (D \times D) = A' + B' + 0 + D = A' + B' + D$

Preguntas

Ejercicios

1. Dibuje el circuito respectivo para las siguientes expresiones, luego simplifique las ecuaciones aplicando las leyes del álgebra booleana todo lo que sea posible y dibuje el circuito resultante; cree además, la tabla de verdad para el circuito ¿Qué significa la salida?
   a. $[B' \times (A' + B)]' + A'$
   b. $[A' \times (A + B)]' + B$
   c. $P' + (P + Q)$
   d. $(X' + Y')' \times (X' \times Y)$
   e. $M' + (M \times N')$
   f. $M' \times ((M')' + N')$
   g. $[T' + (M \times T)]'$
   h. $[(A \times B)' + ((A')' + B)]'$
   i. $(P' + Q)' = (P \times Q')$
   j. $(A \times B)' + A\ siempre\ es\ 1$
   k. $[(A' + B) \times B']' + A'\ es\ siempre\ 1$
   l. $[(A' \times B)' + (A' + B)]'\ siempre\ es\ 0$
   m. $[X \times (X' + Y)] \times Y'\ es\ siempre\ 0$
2. Describa las formas estándar de las expresiones booleanas para:
   a. Suma de productos (SOP)
   b. Producto de sumas (POS)
   c. Muestre los circuitos lógicos para SOP y POS
   d. Muestre las tablas de verdad de SOP y POS
3. Convertir a la forma estándar
   a. ABC' + BC + A'
   b. A + B'C + A'BC + B + C
   c. (A + C)(B + C)
   d. (A + B + C')(B + C)'
4. Mapas de Karnaugh
   a. Describa qué son los mapas de Karnaugh
   b. Ilustre con ejemplos los mapas de Karnaugh para 2, 3 y 4 variables

Fuentes y referencias adicionales

El presente documento está disponible en Internet en la url:
https://github.com/innovasistemas/textos/blob/master/matematicas-discretas-v2026.pdf

Textos impresos

GRIMALDI, Ralph. Matemáticas Discretas y Combinatoria. Tercera Edición. Addison-Wesley Iberoamerican

JOHSONBAUGH Richard. Matemáticas Discretas. Cuarta Edición. Prentice Hall

Universidad Nacional del Sur – Departamento de Matemáticas. Matemáticas Discretas

Bogart, Kenneth P. Matemáticas Discretas. Primera edición. Limusa S.A. México D.F., 1996.

Uribe Calad, Julio A. Matemática. Una propuesta curricular 10. Bedout Editores S.A. Medellín, 1990.

Uribe Calad, Julio A. Matemática. Una propuesta curricular 11. Bedout Editores S.A. Medellín, 1990.

Pérez Ordoñez, Edgar; Palacio Silva, Emiliano; Villamizar Villamizar, Armando. Matemática MEGA. Tomo 1. Primera edición. Terranova Editores Ltda. Santafé de Bogotá D.C., 1999.

Fleming, Walter; Varberg, Dale. Álgebra y trigonometría con geometría analítica. Tercera edición. Prentice Hall. Madrid, 1991.

Leithold, Louis. El Cálculo con Geometría Analítica. Sexta edición. Harla. México D.F., 1992.

Baldor, Aurelio. Álgebra elemental. México, 1970.

Referencias en Internet

Sistema de numeración
En Internet
Sistema de numeración - Wikipedia, la enciclopedia libre

Sistema de numeración
En Internet
https://concepto.de/sistema-de-numeracion/

Números pares e impares
En Internet
Números pares e impares - Wikipedia, la enciclopedia libre

Operaciones con números binarios
En Internet
Operaciones con números binarios

Lógica proposicional
En Internet
Lógica Proposicional

UNAM. Conjuntos
En Internet (pdf)
Unidad 1 Conjunto Potencia Ejemplo Si X = {a} 1

Diferencia simétrica
En Internet (pdf)
DIFERENCIA SIMETRICA MATERIA, TAREA Y EVALUACION 22/04/2020 Referencia

Funciones Inyectivas, Sobreyectivas y Biyectivas
En Internet
Funciones Inyectivas, Sobreyectivas y Biyectivas

Compuertas lógicas
En Internet
Compuertas Lógicas y tabla de verdad - Guia de Mecatronica

Las compuertas lógicas y sus operaciones lógicas
En Internet
Las Compuertas Lógicas y sus Operaciones Lógicas

Compuertas lógicas
En Internet
Compuertas Lógicas AND OR NOT XOR HETPRO TUTORIALES

Compuertas lógicas
En Internet
Compuertas Lógicas — MecatrónicaLATAM

Compuertas lógicas
En Internet
Funciones booleanas

Álgebra de Boole. Fundamentos y aplicaciones básicas en la electrónica digital moderna
En Internet

Álgebra de Boole. Fundamentos y aplicaciones básicas en la electrónica digital moderna (Presentación PowerPoint)

Reglas básicas del álgebra de Boole
En Internet
Reglas básicas del álgebra de Boole

Álgebra de Boole
En Internet (pdf)
TEMA 3. Álgebra de Boole

Mapas de Karnaug
En Internet
Mapas de Karnaugh – Sistemas Digitales

Código Gray
En Internet
Código Gray - Wikipedia, la enciclopedia libre.

Teoría de grafos. Universidad de Pamplona
En Internet (en pdf)
Teoría de grafos

Apéndice

A continuación se presentan aplicaciones de programación de algunos de los temas tratados usando desarrollos algorítmicos presentados en pseudocódigo y en distintos lenguajes de programación como Python, Java, PHP y C/C++ entre otros.

Apéndice A. Aplicaciones con aritmética y álgebra

Este apéndice contiene desarrollos relacionados con los temas vistos en la Unidad 0.

Raíces de la ecuación cuadrática

Programa Python:

```
import math
print("Raíces de la ecuación cuadrática a * x ** 2 + b*x + c = 0")
a = float(input("a: "))
b = float(input("b: "))
c = float(input("c: "))
if a != 0:
    discrim = b ** 2 - 4 * a * c
    if discrim > 0:
        x1 = (-b + math.sqrt(discrim)) / (2 * a)
        x2 = (-b - math.sqrt(discrim)) / (2 * a)
        print(f"x1: {x1}")
        print(f"x2: {x2}")
    elif discrim == 0:
        x1 = -b / (2 * a)
        print(f"Raíces iguales: x: {x1}")
    else:
        print("Raíces imaginarias")
else:
    print("No es una ecuación cuadrática")
```

Cálculo del factorial

Programa PHP:

```
public function factorial($n)
{
    $f = 1;
    for ($i = 1; $i <= $n; $i++) {
        $f *= $i;
    }
    return $f;
```

```
}
```

Determinar números primos

Programa PHP:

```
public function prime($n)
{
    $i = 2;
    $sw = TRUE;
    while ($i <= $n / 2 && $sw) {
        if ($n % $i == 0) {
            $sw = FALSE;
        } else {
            $i++;
        }
    }
    return $sw;
}
```

Encontrar números perfectos

Programa PHP:

```
public function perfect($n)
{
    $sum = 0;
    for ($i = 1; $i <= $n / 2; $i++) {
        if ($n % $i == 0) {
            $sum += $i;
        }
    }
    $sw = $sum == $n ? TRUE : FALSE;
    return $sw;
}
```

Determinar la paridad de un número

Programa PHP:

```
public function even($n)
{
    $k = 0;
    $sw = FALSE;
    while ($k <= $n && !$sw) {
```

```
        if (2 * $k == $n) {
            $sw = TRUE;
        } else {
            $k++;
        }
    }
    return $sw;
}
```

Apéndice B. Aplicaciones con sistemas numéricos

Este apéndice contiene desarrollos relacionados con los temas vistos en la Unidad 1. También incluye una tabla de caracteres ASCII

Tabla de caracteres ASCII[9]

ASCII (acrónimo inglés de *American Standard Code for Information Interchange* — Código Estándar Estadounidense para el Intercambio de Información—), es un código de caracteres basado en el alfabeto latino. Fue creado en 1963 por el Comité Estadounidense de Estándares (ASA, conocido desde 1969 como el Instituto Estadounidense de Estándares Nacionales, o ANSI) como una evolución de los conjuntos de códigos utilizados en telegrafía.

En el estándar ASCII cada carácter del alfabeto latino es representado numéricamente, ya sea en decimal u otra base numérica.

En decimal, los códigos del 32 al 126 se conocen como caracteres imprimibles, y representan letras, dígitos y algunos símbolos especiales.

La siguiente figura muestra los caracteres imprimibles del código ASCII[10]

[9] Ver más acerca del código ASCII en [ASCII - Wikipedia, la enciclopedia libre]
[10] Imagen tomada de: https://workshops.nuevofoundation.org/es/secret-messages/activity-5/

Caracteres ASCII imprimibles

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

Los ejemplos a continuación son métodos de una clase que a su vez dispone de la propiedad `digitHex`, la cual es un arreglo asociativo, tal y como se muestra a continuación.

Programa PHP:
```
private array $digitHex;

public function __construct()
{
    $this->digitHex = [
        'A' => '10',
        'B' => '11',
        'C' => '12',
        'D' => '13',
        'E' => '14',
        'F' => '15'
    ];
```

```
}
```

Convertir un número en base 10 a una base cualquiera

Programa PHP:

```
public function base10ToN($number, $base)
{
    $div = $number;
    $stringNumber = $number == 0 ? "0" : "";
    while ($div > 0) {
        $res = $div % $base;
        $res = $res > 9 ? array_search($res, $this->digitHex) : $res;
        $div = (int)($div / $base);
        $stringNumber = $res . $stringNumber;
    }
    return $stringNumber;
}
```

Convertir un número en una base cualquiera a base 10

Programa PHP:

```
public function baseNTo10($stringNumber, $base)
{
    $sum = 0;
    $n = strlen($stringNumber);
    for ($i = 0; $i < $n; $i++) {
        $digit = strtoupper(substr($stringNumber, $i, 1));
        $digit = is_numeric($digit) ? $digit : $this->digitHex[$digit];
        $sum += $digit * pow($base, $n - $i - 1);
    }
    return $sum;
}
```

Apéndice C. Aplicaciones con conjuntos

Este apéndice contiene desarrollos relacionados con los temas vistos en la Unidad 2.

Operaciones entre conjuntos

Programa Javascript:

```
class Sets
```

```
{
    union(arrayA, arrayB)
    {
        let arrayResult = arrayA.slice();
        arrayB.forEach ((element) => {
            if (this.findElement(arrayA, element) === -1) {
                arrayResult[arrayResult.length] = element;
            }
        });
        return arrayResult;
    }

    intersection(arrayA, arrayB)
    {
        let arrayResult = [];
        arrayA.forEach ((element) => {
            if (this.findElement(arrayB, element) > -1) {
                arrayResult[arrayResult.length] = element;
            }
        });
        return arrayResult;
    }

    minus(arrayA, arrayB)
    {
        let arrayResult = [];
        arrayA.forEach ((element) => {
            if (this.findElement(arrayB, element) === -1) {
                arrayResult[arrayResult.length] = element;
            }
        });
        return arrayResult;
    }

    cartesianProduct(arrayA, arrayB)
    {
        let i = 0;
        let arrayResult = [];
        arrayA.forEach ((elementA) => {
            arrayB.forEach ((elementB) => {
                arrayResult[i] = `(${elementA}, ${elementB})`;
                i++;
            });
        });
        return arrayResult;
    }
```

```
    findElement(array, element, property = '')
    {
        let i = 0;
        let pos = -1;
        let elemArray;
        while (i < array.length && pos == -1) {
            elemArray = property === '' ? array[i] : array[i][property];
            if (elemArray === element) {
                pos = i;
            } else {
                i++;
            }
        }
        return pos;
    }
}
```

Apéndice D. Aplicaciones con compuertas lógicas

Este apéndice contiene desarrollos relacionados con los temas vistos en la Unidad 3

Apéndice E. Aplicaciones con mapas de Karnaugh

Este apéndice contiene desarrollos relacionados con los temas vistos en la Unidad 4.