TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

EXPRESIONES BOLEANA Y SIMPLIFICACION

PRESENTA:

IRIS MAYRA SANTIAGO FERIA
ARTURO BETSABE CRUZ CRUZ
JANELY ARLETH MORALES PACHECO

MATEMATICAS DISCRETAS

ASESOR: JOSE ALFREDO ROMAN CRUZ

CUARTA UNIDAD

1ER SEMESTRE
GRUPO:1AS

CARRERA: INGENIERIA EN SISTEMAS COMPUTACIONALES

Tlaxiaco, Oax., septiembre de 2025.
(24/09/2025)
"Educación, ciencia y tecnología, progreso día con día"®

INDICE

Contenido

INTRODUCCION

El ámbito de aplicación de las expresiones booleanas se extiende desde los circuitos integrados más básicos hasta los sistemas computacionales más complejos, sirviendo como lenguaje fundamental para el diseño de hardware digital, la programación de software, la formulación de consultas en bases de datos y el desarrollo de sistemas inteligentes. Cada expresión booleana, independientemente de su complejidad, puede ser representada mediante tablas de verdad que enumeran exhaustivamente todas las combinaciones posibles de valores de entrada y sus correspondientes salidas, proporcionando una representación completa y no ambigua del comportamiento lógico modelado. La evaluación de estas expresiones sigue reglas precisas de precedencia de operadores y puede realizarse mediante sustitución directa o mediante métodos algorítmicos sistemáticos.

La simplificación de expresiones booleanas emerge como una necesidad práctica crítica en ingeniería y computación, ya que las expresiones complejas resultan en implementaciones físicas costosas, ineficientes y propensas a errores. Los métodos de simplificación buscan transformar expresiones booleanas en formas equivalentes, pero más simples, reduciendo el número de operaciones lógicas requeridas y, consecuentemente, el número de componentes físicos necesarios para su implementación en hardware.

Una expresión booleana es básicamente una combinación de variables binarias relacionadas mediante operadores lógicos como AND, OR y NOT, que siempre evalúan a un resultado binario.

OBJETIVOS

las expresiones booleanas buscan establecer un lenguaje formal que describa el comportamiento de circuitos electrónicos, especificando cómo las combinaciones de señales de entrada producen determinadas señales de salida. Este formalismo permite a los ingenieros predecir el comportamiento de sistemas digitales antes de su construcción física, reduciendo costos y tiempos de desarrollo.

las expresiones booleanas es servir como base para la síntesis de circuitos lógicos, transformando especificaciones funcionales en implementaciones prácticas utilizando compuertas lógicas básicas. Al proporcionar una representación no ambigua de funciones lógicas, permiten el análisis sistemático de propiedades como la completitud funcional, la equivalencia entre circuitos y la detección de condiciones redundantes. En el campo de la programación, las expresiones booleanas constituyen el fundamento de las estructuras de control de flujo, permitiendo la implementación de decisiones complejas mediante condiciones lógicas combinadas. Además, en bases de datos y sistemas de información, facilitan la construcción de consultas complejas mediante operadores lógicos que filtran y combinan criterios de búsqueda.

La simplificación de expresiones booleanas persigue como objetivo principal optimizar las implementaciones físicas de circuitos digitales, reduciendo el número de compuertas lógicas requeridas, minimizando el consumo de energía, mejorando la velocidad de operación y aumentando la confiabilidad del sistema. Al eliminar términos redundantes y literales innecesarios, la simplificación conduce a diseños más económicos que requieren menos componentes físicos, menos espacio en circuitos integrados y menos conexiones entre elementos.

MATERIALES

- ✓ Computadora
- ✓ Cuaderno
- ✓ Lápiz
- ✓ Goma
- ✓ Lapiceros
- ✓ Internet

Educación
Secretaría de Educación Pública

TECNOLÓGICO NACIONAL DE MÉXICO®

TABLA DE ILUSTRACIONES

EXPRESIONES BOLEANAS

Las expresiones booleanas representan el fundamento algebraico de la lógica digital moderna, constituyendo un sistema matemático formal que opera exclusivamente con dos valores discretos: verdadero y falso, representados convencionalmente como 1 y 0 respectivamente. Este sistema debe su nombre al matemático inglés George Boole, quien en su obra "The Laws of Thought" (1854) estableció los principios de lo que inicialmente se denominó "álgebra de la lógica". Las expresiones booleanas son combinaciones estructuradas de variables booleanas, constantes lógicas y operadores que siguen reglas algebraicas específicas y siempre evalúan a un valor binario. En esencia, una expresión booleana es una formulación simbólica que representa una función lógica donde cada variable constituye una proposición que puede ser verdadera o falsa, y los operadores definen relaciones lógicas entre estas proposiciones. La importancia fundamental de estas expresiones radica en su capacidad para modelar decisiones lógicas, circuitos digitales y procesos de computación mediante un formalismo matemático riguroso y consistentemente aplicable.

CARACTERISTICAS

1. OPERADORES LÓGICOS FUNDAMENTALES

- OPERADOR AND (CONJUNCIÓN LÓGICA): Representado simbólicamente como ·, ∧. Desde una perspectiva semántica, A AND B es verdadero únicamente cuando ambas proposiciones A y B son verdaderas simultáneamente
- OPERADOR OR (DISYUNCIÓN LÓGICA): Denotado mediante los símbolos +, ∨, este operador implementa la disyunción inclusiva. La expresión A OR B evalúa a verdadero cuando al menos una de las variables A o B es verdadera.
- OPERADOR NOT (NEGACIÓN LÓGICA): Representado como ', ¬, o mediante una barra superior, el operador NOT implementa la negación o complementación. Este operador unario invierte el valor de la variable a la que se aplica: si A=0 entonces A'=1, y si A=1 entonces A'=0.
- OPERADOR NAND: Definido como la negación de la conjunción, A NAND B = (A·B)'. Este operador es notable porque constituye un operador universal, meaning que cualquier función booleana puede implementarse utilizando exclusivamente operadores NAND.
- OPERADOR NOR: Definido como la negación de la disyunción, A NOR B = (A+B)'. Al igual que NAND, el operador NOR es universal.
- OPERADOR XOR (OR EXCLUSIVO): Representado como ⊕, este operador implementa la disyunción exclusiva, donde A XOR B es verdadero cuando A y B tienen valores diferentes.
- OPERADOR XNOR (EQUIVALENCIA): Como la negación de XOR, A XNOR B es verdadero cuando A y B tienen el mismo valor.

2. PROPIEDADES ALGEBRAICAS ESTRUCTURALES

- PROPIEDAD CONMUTATIVA: significa que el orden de los operandos no afecta el resultado: A+B = B+A y A·B = B·A.
- PROPIEDAD ASOCIATIVA: Los operadores AND y OR son asociativos, permitiendo agrupar términos arbitrariamente sin afectar el resultado: A+(B+C) = (A+B) +C y A·(B·C) = (A·B) ·C

- PROPIEDAD DISTRIBUTIVA: AND distribuye sobre OR [$A\cdot(B+C) = A\cdot B + A\cdot C$] y OR distribuye sobre AND [$A+(B\cdot C) = (A+B)\cdot(A+C)$].
- PROPIEDADES DE IDENTIDAD: Existen elementos identidad para cada operador: 0 para OR ($A+0=A$) y 1 para AND ($A\cdot 1=A$). Estos elementos constituyen los límites del sistema y proporcionan bases para simplificaciones elementales.
- PROPIEDADES DE COMPLEMENTO: Cada elemento tiene un complemento único tal que $A+A'=1$ y $A\cdot A'=0$. Estas propiedades encapsulan los principios lógicos del tercero excluido y de no contradicción, respectivamente.
- PROPIEDADES DE IDEMPOTENCIA: Las operaciones AND y OR son idempotentes: $A+A=A$ y $A\cdot A=A$.
- PROPIEDADES DE ABSORCIÓN: Las leyes de absorción establecen que $A+(A\cdot B)=A$ y $A\cdot(A+B)=A$. Estas propiedades permiten eliminar términos redundantes en expresiones complejas y son particularmente útiles en procedimientos de minimización.
- PROPIEDADES DE INVOLUCIÓN: La negación es una operación involutiva, meaning que aplicar dos veces la negación restaura el valor original: $(A')' = A$. Esta propiedad justifica la eliminación de dobles negaciones en expresiones simplificadas.
- LEYES DE MORGAN: Estas leyes fundamentales establecen relaciones sistemáticas entre conjunción, disyunción y negación: $(A+B)' = A'\cdot B'$ y $(A\cdot B)' = A'+B'$. Las leyes de Morgan permiten "distribuir" la negación sobre paréntesis y transformar entre formas conjuntivas y disyuntivas, siendo herramientas esenciales para la manipulación de expresiones booleanas.
- PRINCIPIO DE DUALIDAD: El álgebra booleana exhibe una dualidad profunda donde cualquier identidad válida permanece válida cuando se intercambian los operadores AND y OR y los valores 0 y 1. Este principio permite derivar nuevas identidades a partir de conocidas y proporciona una perspectiva unificadora sobre la estructura del sistema.

METODOS DE SIMPLIFICACION

- TÉCNICA DE FACTORIZACIÓN: La factorización consiste en identificar factores comunes en múltiples términos y extraerlos aplicando la ley distributiva inversa.
- TÉCNICA DE EXPANSIÓN: La expansión, opuesta a la factorización, aplica la ley distributiva en su dirección directa para revelar estructuras ocultas que permitan posteriores simplificaciones.
- TÉCNICA DE ABSORCIÓN DIRECTA: Las leyes de absorción permiten eliminar términos redundantes inmediatamente.
- TÉCNICA DE CONSENSO: El teorema del consenso establece que $A\cdot B + A'\cdot C + B\cdot C = A\cdot B + A'\cdot C$. Este teorema permite eliminar el término redundante $B\cdot C$ cuando están presentes los otros dos términos. La forma dual es $(A+B)\ (A'+C)\ (B+C) = (A+B)\ (A'+C$
- TÉCNICA DE COMPLEMENTACIÓN E IDEMPOTENCIA: La aplicación estratégica de las propiedades de complemento ($A+A'=1$) e idempotencia ($A+A=A$) permite simplificaciones significativas la idempotencia permite eliminar términos duplicados, mientras que la introducción de términos de la forma $A+A'$ (que siempre es 1) puede facilitar factorizaciones.
- APLICACIÓN SISTEMÁTICA DE MORGAN: Las leyes de Morgan permiten transformar entre formas de producto de sumas y suma de productos, lo que puede revelar oportunidades de simplificación no evidentes en la forma original.

EJERCICIOS

EJERCICIO 1:

Expresión original:

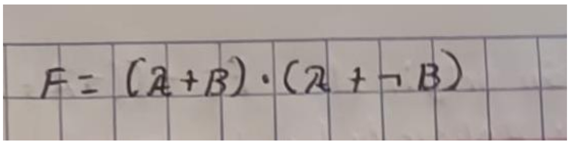


Ilustración 1 expresión original

Paso 1: aplicar la ley distributiva para simplificar A:

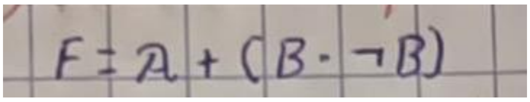


Ilustración 2 simplificación de A

Paso 2: aplicar la ley de complemento a $(B \cdot \neg B)$:

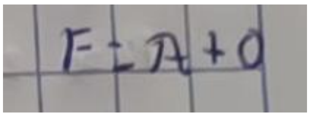


Ilustración 3 complemento del producto

Paso 3: aplicar la ley de identidad:

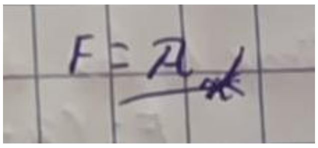


Ilustración 4 simplificación final

EJERCICIO 2:

Expresión original:

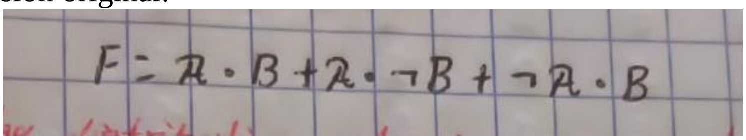


Ilustración 5 expresión original

Paso 1: aplicar la ley distributiva a los dos primeros términos:

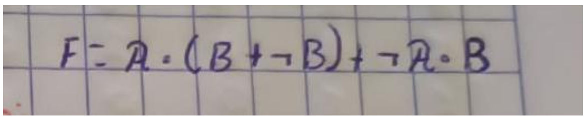


Ilustración 6 ley distributiva

Paso 2: aplicar la ley de complemento:

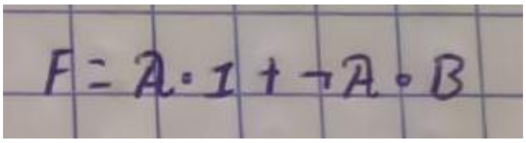


Ilustración 7 ley del complemento

Paso 3: aplicar la ley de identidad:

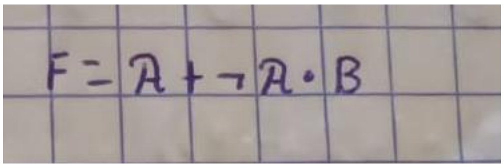


Ilustración 8 ley de identidad

Paso 4: aplicar la ley de absorción:

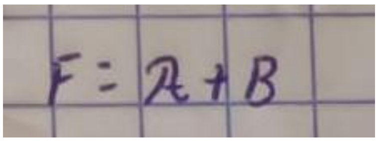


Ilustración 9 simplificación final

TABLAS DE VERDAD

Ejercicio 1: $F = (A + B) \cdot (A + \neg B)$

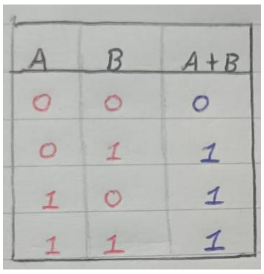
A	B	A+B	$A + \neg B$	$(A + B) \cdot (A + \neg B)$	A
0	0	0	1	0	0
0	1	1	0	0	0
1	0	1	1	1	1
1	1	1	1	1	1

Ejercicio 2: $F = A \cdot B + A \cdot \neg B + \neg A \cdot B$

A	B	$A \cdot B$	$A \cdot \neg B$	$\neg A \cdot B$	$A \cdot B + A \cdot \neg B + \neg A \cdot B$	A+B
0	0	0	0	0	0	0
0	1	0	0	1	1	1
0	0	0	0	0	0	0
1	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	1	0	0	1	1

DISEÑO EN CROCODILE

Ejercicio 1: Tabla de simplificación:



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

Ilustración 10 tabla de simplificación ejer.1

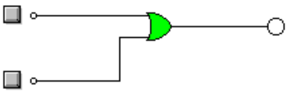


Ilustración 11 A=0, B=0

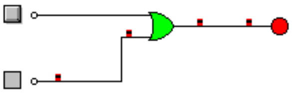


Ilustración 12 A=0, B=1

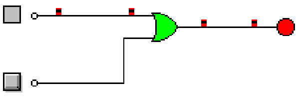


Ilustración 13 A=1, B=0

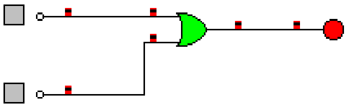
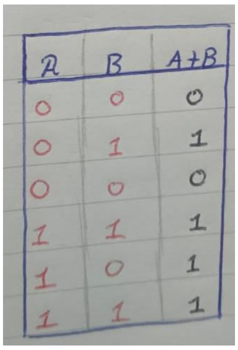


Ilustración 14 A=1, B=1

Ejercicio 2: tabla de simplificación:



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

Ilustración 15 tabla de simplificación ejer.2

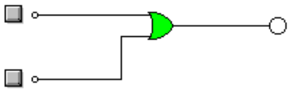


Ilustración 16 A=0, B=0

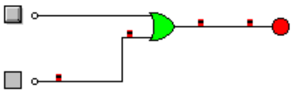


Ilustración 17 A=0, B=1

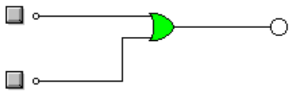


Ilustración 18 A=0, B=0

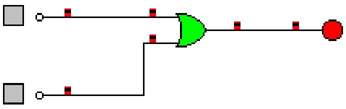


Ilustración 19 A=1, B=1

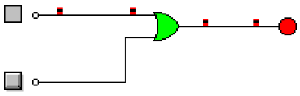


Ilustración 20 A=1, B=0

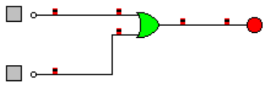


Ilustración 21 A=1, B=1

Educación
Secretaría de Educación Pública
TECNOLÓGICO NACIONAL DE MÉXICO

LISTA DE RESULTADOS

Ejercicio 1:

Expresión inicial:

$$F = (A + B) \cdot (A + \neg B)$$

Simplificación:

$$F = A$$

Ejercicio 2:

Expresión inicial:

$$F = A \cdot B + A \cdot \neg B + \neg A \cdot B$$

Simplificación:

$$F = A + B$$

CONCLUSION

El estudio de las expresiones booleanas y su simplificación constituye un pilar fundamental en el ámbito de la computación, ingeniería de sistemas y diseño digital. Tras un análisis exhaustivo, se evidencia que el dominio de estas técnicas representa una competencia esencial para cualquier profesional del área tecnológica.

Las expresiones booleanas forman la base estructural de la lógica computacional moderna. Desde los circuitos integrados más básicos hasta los algoritmos de inteligencia artificial más complejos, la lógica booleana provee el marco teórico-práctico para representar y manipular decisiones binarias. Su correcta implementación impacta directamente en la eficiencia, confiabilidad y rendimiento de los sistemas digitales.

La simplificación booleana demuestra beneficios cuantificables mediante la aplicación sistemática de leyes algebraicas. Las leyes De Morgan, distributivas, de absorción y del complemento permiten reducir expresiones complejas en formas mínimas equivalentes.

El dominio de las expresiones booleanas y sus técnicas de simplificación representa una competencia crítica que diferencia a los profesionales técnicos excepcionales. Su estudio sistemático y aplicación rigurosa constituyen inversiones con alto retorno en calidad, eficiencia y capacidad de innovación en el desarrollo tecnológico contemporáneo.

El dominio del álgebra booleana representa un diferenciador competitivo en la formación de ingenieros y desarrolladores. No se trata simplemente de un conocimiento teórico, sino de una habilidad práctica que permite optimizar recursos computacionales, reducir costos operativos y mejorar la eficiencia energética en sistemas digitales. Los profesionales que internalizan estos conceptos demuestran capacidades superiores de abstracción y resolución de problemas complejos.