Clase 7

Proposiciones compuestas, leyes lógicas y puertas lógicas

7.1 Introducción

La lógica es el estudio de los métodos y los principios indispensables para distinguir el razonamiento correcto del razonamiento incorrecto. Los métodos y las técnicas de la lógica han sido desarrollados esencialmente con el propósito de aclarar esta distinción.

Además, se erige como la herramienta formal de razonamiento de la mayor parte de las experiencias educativas de licenciaturas relacionadas con la computación, sobre todo de las que están más concernidas con las matemáticas y la programación (Fernández *et al*., 2007). Sin embargo, la lógica matemática no solo funciona para el cálculo de valores de verdad o la demostración de teoremas, sino también para tener consciencia de los procesos internos del pensamiento y con ello organizar ideas, sistematizar procesos, incluso para explicar algo con claridad.

Por tal motivo, es conveniente que los estudiar los contenidos relacionados con áreas de la computación valoren la práctica de la lógica como una aplicación en el mundo de la tecnología de las comunicaciones y que comprendan que el uso de un razonamiento ordenado puede ser de gran utilidad en muchos aspectos de la vida.

7.2 Clasificación de las proposiciones compuestas

Las tablas de valores de verdad permiten clasificar a las estructuras proposicionales
en tres grandes grupos: tautologías, contradicciones y contingencias.

7.2.1 Tautología

Se llama tautología a toda proposición compuesta que resulta verdadera en todos los casos independientemente de los valores de verdad que correspondan a las proposiciones simples intervinientes.

p	q	$p \vee q$	$p \Rightarrow p \vee q$
V	V	V	V
V	F	V	V
F	V	V	V
F	F	F	V

7.2.2 Contradicción

Se llama contradicción a toda proposición compuesta que resulta falsa en todos los casos, independientemente de los valores de verdad que correspondan a las proposiciones simples intervinientes.

Ejemplo

Demostrar que la proposición compuesta $(p \wedge q) \wedge \sim p$ es una contradicción.

p	q	~p	p ∧ q	(p ∧ q) ∧ ~p
V	V	F	V	F
V	F	F	F	F
F	V	V	F	F
F	F	V	F	F

7.2.3 Contingencia

Se llama contingencia a toda proposición compuesta cuya tabla de 4 verdad tiene por lo menos un valor V y un valor F.

Ejemplo

Demostrar que la proposición $\sim p \Rightarrow q$ es una contingencia.

p	q	~p	~p ⇒ q
V	V	F	V
V	F	F	V
F	V	V	V
F	F	V	F

7.3 Equivalencia Lógica

Dos proposiciones son **lógicamente equivalentes** o simplemente **equivalentes** si tienen las mismas tablas de verdad y por lo tanto la doble implicación entre ambas será una tautología.

Ejemplo
Demostrar que las proposiciones $\sim(p \wedge q)$ y $(\sim p \vee \sim q)$ son equivalentes.

p	q	p ∧ q	~(p∧q)	~p	~q	~p ∨ ~q	~(p∧q) ⇔ ~p ∨ ~q
V	V	V	F	F	F	F	V
V	F	F	V	F	V	V	V
F	V	F	V	V	F	V	V
F	F	F	V	V	V	V	V

7.4 Leyes Lógicas

En el cálculo proposicional se utilizan las siguientes leyes lógicas cuya demostración se reduce a la confección de la correspondiente tabla de valores de verdad.

Algunas de las leyes lógicas son las siguientes:

- **Involución:** $\sim(\sim p) \Leftrightarrow p$
- **Idempotencia:**

  a) $(p \wedge p) \Leftrightarrow p$

  b) $(p \vee p) \Leftrightarrow p$
- **Conmutatividad:**

  a) de la disyunción: $p \vee q \Leftrightarrow q \vee p$

  b) de la conjunción: $p \wedge q \Leftrightarrow q \wedge p$
- **Asociatividad:**

  a) de la disyunción: $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

  b) de la conjunción: $(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
- **Distributividad:**

  a) de la disyunción respecto de la conjunción:

  $p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$

  b) de la conjunción respecto de la disyunción:

  $p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$
- **Leyes de De Morgan:**

  a) La negación de una disyunción es equivalente a la conjunción de las negaciones: $\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$

  b) La negación de una conjunción es equivalente a la disyunción de las negaciones: $\sim(p \wedge q) \Leftrightarrow \sim p \vee \sim q$
- **Negación de una implicación**

La negación de una implicación es equivalente a la conjunción del antecedente con la negación del consecuente: $\sim(p \Rightarrow q) \Leftrightarrow (p \wedge \sim q)$

Ejemplo

Simbolizar, negar y retraducir las siguientes proposiciones:

a) La empresa "SaluDiet" almacena frutas secas o vegetales.

b) Cuando el gerente estudia el monto de ventas diarias, toma decisiones acertadas.

c) Los empleados recibieron cheques o pagos en efectivo

d) Durante la última semana la Bolsa de Valores no aumentó y los inversores estuvieron atentos

Desarrollo

a) p: La empresa "SaluDiet" almacena frutas secas
q: La empresa "SaluDiet" almacena vegetales

$p \vee q$: La empresa "SaluDiet" almacena frutas secas o vegetales.

Negación: $\sim(p \vee q) \Leftrightarrow \sim p \wedge \sim q$: No es cierto que la empresa "SaluDiet"

Almacena frutas secas o vegetales; o bien: La empresa "SaluDiet" no almacena frutas secas ni vegetales.

b) p: El gerente estudia el monto de ventas diarias
q: El gerente toma decisiones acertadas

$p \Rightarrow q$ **:** Cuando el gerente estudia el monto de ventas diarias, toma decisiones acertadas.

Negación: $\sim(p \Rightarrow q) \Leftrightarrow p \wedge \sim q$: El gerente estudian el monto de ventas diarias y no toma decisiones acertadas.

c) p: Los empleados recibieron cheques
q: Los empleados recibieron pagos en efectivo

$p \vee q$: Los empleados recibieron cheques o pagos en efectivo

Negación: $\sim(p \vee q) \Leftrightarrow (\sim p) \wedge (\sim q)$: Los empleados no recibieron cheques, ni pagos en efectivos.

d) p: Durante la última semana la Bolsa de Valores aumentó
q: Durante la última semana los inversores estuvieron atentos

$\sim p \wedge q$: Durante la última semana la Bolsa de Valores no aumentó y los inversores estuvieron atentos.

Negación: $\sim(\sim p \wedge q) \Leftrightarrow \sim(\sim p) \vee \sim q \Leftrightarrow p \vee \sim q$: Durante la última semana la Bolsa de Valores aumentó o los inversores no estuvieron atentos.

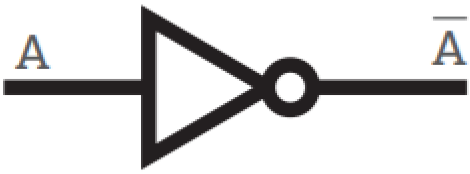
7.5 Puertas Lógicas

Las puertas lógicas son dispositivos que operan con estados lógicos y funcionan igual que una calculadora, de un lado ingresan los datos, esta realiza una operación, y finalmente, se muestra el resultado.

Cada una de las puertas lógicas se las representa mediante un símbolo, y la operación que realiza (operación lógica) se corresponde con una tabla, llamada Tabla de Verdad.

7.5.1 Puerta lógica NOT

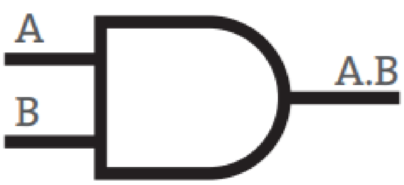
Se trata de un inversor o complemento, es decir, invierte el dato de entrada, por ejemplo; si se coloca su entrada a 1 (nivel alto) se obtendrá en su salida un 0 (o nivel bajo), y viceversa. Esta puerta dispone de una sola entrada. Su opetacion lógica es igual a NEGACION



'A'	NOT 'A'
0	1
1	0

7.5.2 Puerta lógica AND

La puerta lógica AND genera una salida a nivel ALTO solo cuando todas las entradas esta a nivel ALTO. Cuando cualquiera de la entrada esta a nivel bajo, la salida se pone a nivel BAJO. Por tanto, el propósito básico de una puerta AND es determinar cuando ciertas condiciones de entrada son simultáneamente verdaderas, como indican todas sus entradas estando a nivel ALTO, y producir una salida a nivel ALTO, para indicar que esas condiciones son verdaderas.

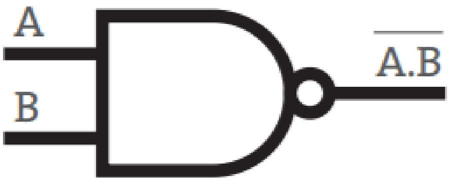


'A'	'B'	'A' AND 'B'
0	0	0
0	1	0
1	0	0
1	1	1

7.5.3 Puerta lógica NAND

La puerta NAND es un elemento lógico popular, debido a que se puede utilizar como una puerta universal, es decir, las puertas NAND se pueden combinar para implementar las operaciones de las compuertas AND, OR y NOT. El termino NAND es una contracción de NOT-AND, e implica una función AND con la salida complementada (negada).

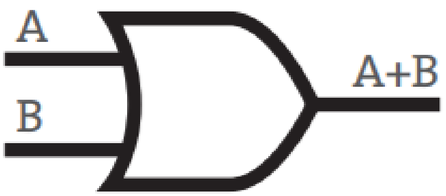
Esta puerta lógica genera una salida a nivel BAJO solo cuando todas las entradas están a nivel ALTO. Cuando cualquiera de las entradas esta a nivel BAJO, la salida se pondrá a nivel ALTO.



'A'	'B'	'A' NAND 'B'
0	0	1
0	1	1
1	0	1
1	1	0

7.5.4 Puerta lógica OR

Una puerta OR genera un nivel ALTO a la salida cuando cualquiera de sus entradas esta a nivel ALTA. La salida se pone a nivel BAJO solo cuando todas las entradas están a nivel BAJO. Por tanto, el propósito de una puerta OR es determinar cuándo una o más de sus entradas están a nivel ALTO y generar una salida a nivel ALTO que indique esta condición.



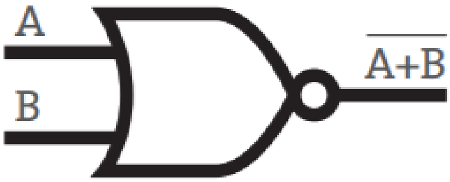
'A'	'B'	'A' OR 'B'
0	0	0
0	1	1
1	0	1
1	1	1

7.4.5 Puerta lógica NOR

La compuerta NOR es un útil elemento lógico porque también se puede emplear como una puerta universal; se pueden usar en combinación para implementar las operaciones AND, OR y del inversor.

El termino NOR en una contradicción de NOT-OR e implica una función OR con la salida invertida (complementada)

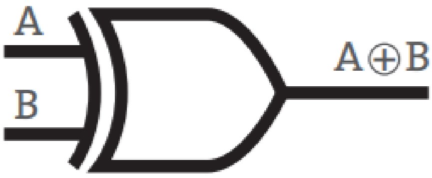
Esta puerta genera una salida a nivel BAJO cuando cualquiera de sus entradas esta a nivel ALTO. Solo cuando todas sus entradas estén a nivel BAJO, la salida se pondrá a nivel ALTO.



‘A’	‘B’	‘A’ NOR ‘B’
0	0	1
0	1	0
1	0	0
1	1	0

7.5.5 Puerta lógica XOR

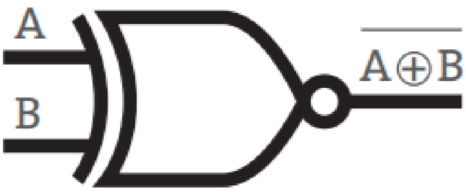
La salida de una puerta OR – exclusiva (XOR) se pone a nivel ALTO solo cuando las dos entradas están a niveles lógicos opuestos.



‘A’	‘B’	‘A’ XOR ‘B’
0	0	0
0	1	1
1	0	1
1	1	0

7.5.6 Puerta lógica XNOR

Al igual que la puerta XOR, la compuerta XNOR solo tiene dos entradas. El circulo en la salida del símbolo de la compuerta XNOR indica que su salida es la opuesta a la de la compuerta XOR. Cuando dos niveles lógicos de entrada son opuestos, la salida de la puerta NOR – exclusiva es nivel BAJO.



‘A’	‘B’	‘A’ XNOR ‘B’
0	0	1
0	1	0
1	0	0
1	1	1

7.5.7 Función lógica

Un conjunto de compuertas construye a la salida una combinación lógica de las entradas que se expresa como una función lógica. Esta función caracteriza al sistema.

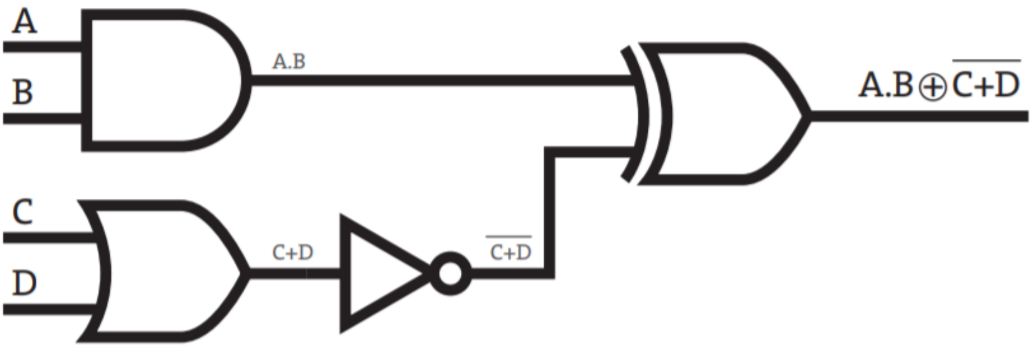


Tabla de verdad de la función lógica

A	B	C	D	Q (Salida)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1