0. ALGEBRA DI BOOLE

Nel lavoro di programmazione capita spesso di dovere ricorrere ai principi della logica degli enunciati ed occorre conoscere almeno alcuni concetti base dell'**algebra degli enunciati (o delle proposizioni)** detta anche **algebra booleana** o di **Boole** dal matematico inglese George Boole (1815-1864).

Gli oggetti che fanno parte dell'algebra di Boole sono gli **enunciati.**

DEF: Si definisce **enunciato** una **proposizione** che può essere soltanto vera oppure soltanto falsa e chr non può mai essere contemporaneamente vera e falsa oppure indeterminata.

DEF: Con il termine **valore di verità di un enunciato** si intende la sua verità oppure la sua falsità ossia il suo essere vero oppure falso.

Nell'Algebra di Boole un enunciato può essere essere **semplice** oppure **composto.**

DEF: Un. **enunciato composto** è un enunciato formato da due o più enunciati semplici (chiamati sottoenunciati) collegati tra loro attraverso appositi **connettivi logici.**

La **proprietà fondamentale** di un **enunciato composto** è che il suo valore di verità viene completamente determinato dai valori di verità degli enunciati semplici (sottoenunciati) che ne fanno parte e dal/i connettivo/i logico/i utilizzato/i.

Nell'Algebra di Boole i **connettivi logici** (detti anche operatori booleani) si dividono in **connettivi logivi FONDAMENTALI e connettivi logici DERIVATI.**

I CONNETTIVI LOGICI FONDAMENTALI

A) CONGIUNZIONE LOGICA (AND oppure e oppure $\wedge$ oppure et oppure &)

Il connettivo logico **AND** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	q	p AND q
V	V	V
V	F	F
F	V	F
F	F	F

In altre parole l'**enunciato composto p AND q** risulta **VERO** solo nel caso in cui entrambi gli **enunciati semplici p** e **q** sono **VERI,** mentre risulta **FALSO** in tutti gli altri casi.

B) DISGIUNZIONE LOGICA (OR oppure o oppure $\vee$ oppure vel oppure |)

Il connettivo logico **OR** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	q	p OR q
V	V	V
V	F	V
F	V	V
F	F	F

In altre parole l'**enunciato composto p OR q** risulta **FALSO** solo nel caso in cui entrambi gli **enunciati semplici p** e **q** sono **FALSI** mentre risulta **VERO** in tutti gli altri casi (ossia quando almeno uno degli enunciati semplici è VERO).

C) NEGAZIONE LOGICA (NOT oppure **non)**
Il connettivo logico **NOT** è un operatore **unario** (ossia agisce su un solo enunciato per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	**NOT p**
V	**F**
F	**V**

In altre parole l'**enunciato composto NOT p** (detto anche **"negazione di p")** risulta **VERO** se l'**enunciato semplice p** è **FALSO** mentre risulta **FALSO** se l'**enunciato semplice p** è **VERO ossia** assume **valore di verità opposto** rispetto al valore di verità posseduto dall'enunciato semplice su cui viene applicato

I tre connettivi logici sopra definiti - **AND**, **OR** e **NOT** – si definiscono "**fondamentali"** in quanto essi cosrtituiscono un insieme **"funzionalmente completo"** di operatori nell'algebra di Boole.

Infatti tutti gli altri connettivi logici possibili sono chiamati connettivi **"derivati"** in quanto possono essere espressi mediante un'opportuna combinazione (espressione) di uno o più connettivi logici fondamentali (ossia **AND**, **OR** e **NOT**).

<u>I CONNETTIVI LOGICI DERIVATI</u>

Esempio di **connettivi logici "derivati"** sono **XOR**, **NAND**, **NOR** e **XNOR**

D) OPERATORE LOGICO NAND
Il connettivo logico **NAND** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	**q**	**p NAND q**
V	**V**	**F**
V	**F**	**V**
F	**V**	**V**
F	**F**	**V**

In altre parole l'**enunciato composto p NAND q** risulta **FALSO** quando entrambi gli enunciati semplici **p** e **q** sono **VERI** mentre risulta **VERO** in tutti gli altri casi

N.B. Il NAND è un connettivo derivato in quanto si potrà dimostrare utilizzando il concetto di equivalenza logica illustrato più avanti che

$$\textbf{p NAND q} \equiv \textbf{NOT (p AND q)}$$

E) OPERATORE LOGICO NOR
Il connettivo logico **NOR** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	q	p NOR q
V	V	F
V	F	F
F	V	F
F	F	V

In altre parole l'**enunciato composto p NOR q** risulta **VERO** quando entrambi gli enunciati semplici **p** e **q** sono **FALSI** mentre risulta **FALSO** in tutti gli altri casi

N.B. Il NOR è un connettivo derivato in quanto si potrà dimostrare utilizzando il concetto di equivalenza logica illustrato più avanti che

$$\textbf{p NOR q} \equiv \textbf{NOT (p OR q)}$$

F) DISGIUNZIONE ESCLUSIVA (XOR oppure **o esclusivo** oppure **aut)**
Il connettivo logico X**OR** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	q	p XOR q
V	V	F
V	F	V
F	V	V
F	F	F

In altre parole l'**enunciato composto p OR q** risulta **VERO** solo nel caso in cui **al più uno solo** dei due **enunciati semplici p** e **q** risulti **VERO** mentre risulta **FALSO** in tutti gli altri casi

oppure in altri termini

l'**enunciato composto p XOR q** risulta **VERO** solo nel caso in cui i due **enunciati semplici p** e **q** hanno **valori di verità diversi** mentre risulta **FALSO** se i due **enunciati semplici p** e **q** hanno **valori di verità uguali.**

N.B. Lo XOR è un connettivo derivato in quanto si potrà dimostrare utilizzando il concetto di equivalenza logica illustrato più avanti che

$$\textbf{p XOR q} \equiv \textbf{(p AND (NOTq)) OR ((NOTp) AND q)}$$

G) OPERATORE LOGICO XNOR

Il connettivo logico **XNOR** è un operatore **binario** (ossia agisce su due enunciati per crearne un altro) completamente definito dalla seguente **tavola di verità**

p	q	p XNOR q
V	V	V
V	F	F
F	V	F
F	F	V

In altre parole l'**enunciato composto p XNOR q** risulta **VERO** solo nel caso in cui gli **enunciati semplici p** e **q** hanno lo stesso valore di verità mentre risulta **FALSO** in tutti gli altri casi

N.B. Lo XNOR è un connettivo derivato in quanto si potrà dimostrare utilizzando il concetto di equivalenza logica illustrato più avanti che

p XNOR q ≡ NOT (p XOR q)

<u>LE TAVOLE DI VERITÀ</u>

Combinando in vario modo gli enunciati semplici generici del tipo **p**, **q**, **r** ed i connettivi logici **fondamentali e/o derivati** si possono ottenere enunciati molto più complessi.

La **forma enunciativa** è un enunciato composto **F(p, q, r, ..)** da enunciati semplici variabili attraverso i connettivi logici

Il **valore di verità di una forma enunciativa** è noto quando si conoscono i valori di verità delle sue variabili.

Un modo semplice per conoscerlo è quello che prevede la *costruzione della sua tavola di verità.*

Il **numero di colonne** di una tavola, di verità **NON è noto a priori** e dipende dalla complessità della forma enunciativa da calcolare.

Il **numero di righe** di una tavola di verità è noto priori ed è pari al valore dell'espressione

$$\mathbf{2}^{\text{<numero\_enunciati\_semplici>}}$$

Esempio:

Supponiamo di voler conoscere i valori di verità della seguente forma enunciativa

NOT(p AND (NOT q))

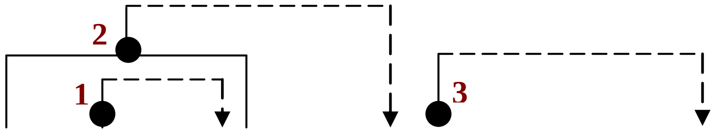
Costruiamo la tavola di verità seguendo i seguenti passi:

- *prima occorre individuare gli enunciati semplici contenuti nella forma enunciativa. Nel nostro caso **p** e **q**;*
- *poi occorre individuare gli enunciati composti contenuti nella forma enunciativa <u>partendo dall'enunciato composto più interno</u> fino all'enunciato composto totale. Nel nostro caso **NOT q**, **(p AND (NOT q))** e **NOT(p AND (NOT q))**;*
- *disegnare una tabella con tante colonne quanti sono gli elementi individuati nei primi due punti;*
- *applicare nell'ordine le tavole di verità dei connettivi logici fondamentali AND, OR, XOR e NOT.*

Legenda:

● *applicazione di una tavola di verità fondamentale e/o derivata (AND, OR, NOT, NAND, NOR, XOR, XNOR)*

– – – – *risultato di una tavola di verità fondamentale e/o derivata (AND, OR, NOT, NAND, NOR, XOR, XNOR)*

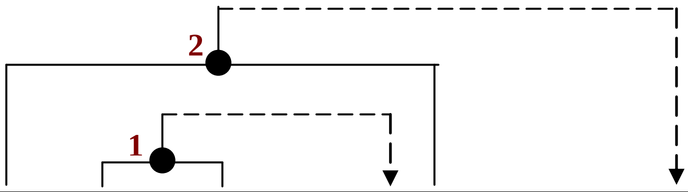


p	***q***	***NOT q***	***p AND (NOT q)***	***NOT(p AND (NOT q))***
V	**V**	**F**	**F**	**V**
V	**F**	**V**	**V**	**F**
F	**V**	**F**	**F**	**V**
F	**F**	**V**	**F**	**V**

Esempio:

Supponiamo di voler conoscere i valori di verità della seguente forma enunciativa

p OR (q AND r))



p	***q***	***r***	***q AND r***	***p OR (q AND r))***
V	**V**	**V**	**V**	**V**
V	**V**	**F**	**F**	**V**
V	**F**	**V**	**F**	**V**
V	**F**	**F**	**F**	**V**
F	**V**	**V**	**V**	**V**
F	**V**	**F**	**F**	**F**
F	**F**	**V**	**F**	**F**
F	**F**	**F**	**F**	**F**

L'EQUIVALENZA LOGICA

DEF: Due forme enunciative si dicono **logicamente equivalenti** se hanno la stessa tavola di verità.
L'equivalenza logica si indica con il simbolo $\equiv$

N.B. Le tavole di verità da mettere a confronto sono costituite solo dalle colonne relative agli enunciati semplici partecipanti più la sola colonna del risultato finale, escludendo eventualmente qualsiasi colonna intermedia utilizzata per svolgere i passaggi risolutivi.

Esempio:
Proviamo che le due forme enunciative

(p AND q) OR (NOT p) *e* ***(NOT p) OR q***

sono equivalenti.
Dobbiamo innanzitutto costruire le due tavole di verità con seguendo i passi prima specificati.

Tavola di verità di: ***(p AND q) OR (NOT p)***

p	*q*	*p AND q*	*NOT p*	*(p AND q) OR (NOT p)*
V	**V**	**V**	**F**	**V**
V	**F**	**F**	**F**	**F**
F	**V**	**F**	**V**	**V**
F	**F**	**F**	**V**	**V**

Tavola di verità di: ***(NOT p) OR q***

p	*q*	*NOT p*	*(NOT p) OR q*
V	**V**	**F**	**V**
V	**F**	**F**	**F**
F	**V**	**V**	**V**
F	**F**	**V**	**V**

Poichè le due forme enunciative hanno la stessa tavola di verità esse risultano essere equivalenti.

Quindi possiamo scrivere che ***(p AND q) OR (NOT p)*** $\equiv$ ***(NOT p) OR q***

LEGGI O PROPRIETÀ DELL'ALGEBRA BOOLEANA

1) Legge di **IDEMPOTENZA** **p OR p ≡ p**

p AND p ≡ p

2) Legge della **DOPPIA NEGAZIONE** **NOT (NOT p) ≡ p**

3) Principio di **NON CONTRADDIZIONE** **p AND (NOT p) è sempre FALSO**

4) Leggi di **ASSORBIMENTO** **p AND (p OR q) ≡ p**

p OR (p AND q) ≡ p

5) Proprietà **ASSOCIATIVA** **(p OR q) OR r ≡ p OR (q OR r)**

(p AND q) AND r ≡ p AND (q AND r)

6) Proprietà **COMMUTATIVA** **p OR q ≡ q OR p**

p AND q ≡ q AND p

7) Proprietà **DISTRIBUTIVA** **p OR (q AND r) ≡ (p OR q) AND (p OR r)**

p AND (q OR r) ≡ (p AND q) OR (p AND r)

8) Leggi di **DE MORGAN** **NOT (p OR q) ≡ (NOT p) AND (NOT q)** *(prima legge)*

NOT (p AND q) ≡ (NOT p) OR (NOT q) *(seconda legge)*

Nota Bene

Tutte queste leggi (o proprietà) si dimostrano costruendo le tavole di verità di entrambi i membri e verificando che esse, secondo la definizione, coincidano.