ALGEBRA BOOLEANA

Stefania Bandini

GEORGE BOOLE

(1815-1864)

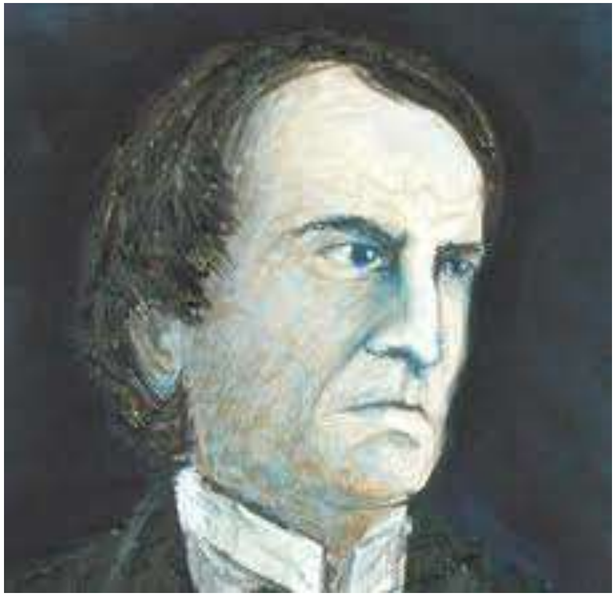
Matematico e logico britannico, ed è considerato il fondatore della **logica matematica**. La sua opera influenzò anche settori della filosofia.

A causa dello stato di povertà della sua famiglia, fu praticamente un autodidatta, e studiò il greco, il latino, il francese, il tedesco, l'italiano oltre alla matematica fin da giovane sui testi di Laplace e Lagrange. Assillato da problemi economici, svolse varie attività.

Morì all'età di soli 49 anni per una polmonite causata da un banale raffreddore.

GEORGE BOOLE
(1815-1864)

Incoraggiato ed indirizzato da Duncan Gregory, curatore del *Cambridge Mathematical Journal*, **Boole** si dedicò allo studio di metodi algebrici per la risoluzione di equazioni differenziali.



La pubblicazione dei suoi risultati gli fece ottenere una medaglia della Royal Society e, successivamente, nel 1849, la nomina alla cattedra di matematica al *Queen's College* di Cork. In questa sede egli insegnò per il resto della sua vita.

GEORGE BOOLE
(1815-1864)

Con l'opera *The Mathematical Analysis of Logic* (1847), scritta sulla scia della polemica insorta fra Augustus **De Morgan** e Sir William Rowan **Hamilton** sulla quantificazione del predicato, propose una sua interpretazione del rapporto fra **matematica**, **logica** e **filosofia** che prevedeva una associazione tra logica e matematica al posto di quella fra logica e metafisica.

Boole considerava la logica alla stregua della scienza delle leggi dei simboli attraverso i quali si esprimono i pensieri, e applicò parte della filosofia algebrica cantabrigense ad un settore inesplorato come quello della logica formale.

Primo estimatore e continuatore dell'opera di Boole fu lo stesso de Morgan.

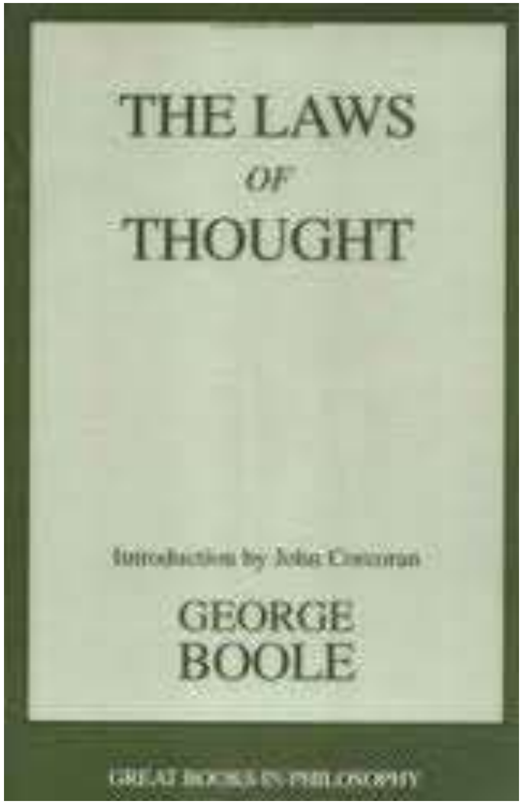
Augustus De Morgan (1806-1871)

"LE LEGGI DEL PENSIERO"

Nel 1854 pubblicò la sua opera più importante, indirizzata alle **leggi del pensiero**, con la quale propose una nuova impostazione della logica.

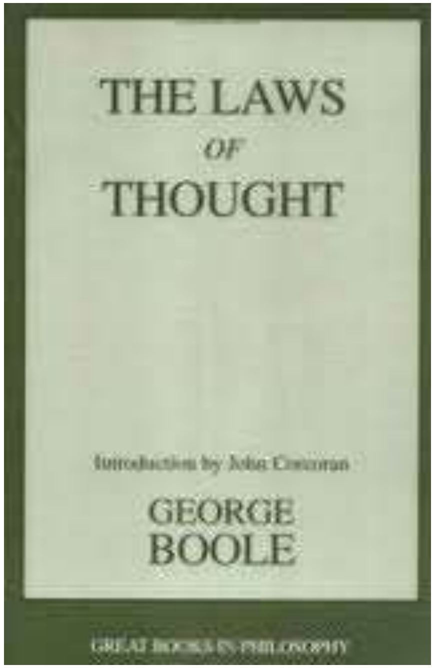
Scopo dell'opera fu quello di studiare le leggi delle operazioni mentali alla base del ragionamento esprimendole nel linguaggio simbolico del calcolo e di istituire, di conseguenza, una disciplina scientifica della logica sorretta da un metodo.

Dopo aver rilevate le analogie fra oggetti dell'algebra e oggetti della logica, ricondusse le composizioni degli enunciati a semplici operazioni algebriche.



LE LEGGI DEL PENSIERO

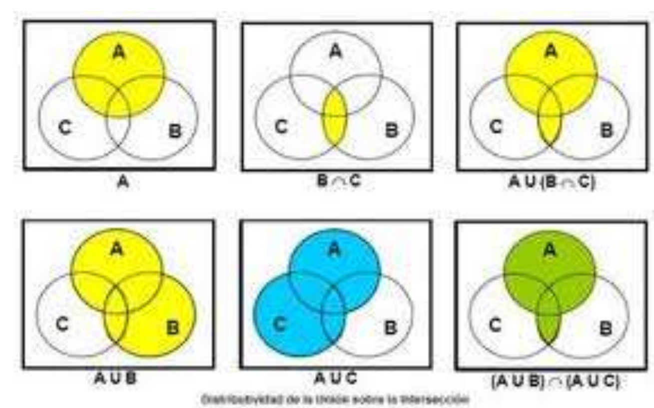
"Scopo di questo trattato è d'indagare le leggi fondamentali di quelle operazioni della mente per mezzo delle quali si attua il ragionamento; di dar loro espressione nel linguaggio simbolico di un calcolo e d'istituire, su questo fondamento, la scienza della logica costruendone il metodo; di fare, di questo stesso metodo, la base di un metodo generale per l'applicazione della dottrina matematica della probabilità e, in ultimo, di ricavare dai diversi elementi di verità portati alla luce nel corso di queste indagini alcune indicazioni probabili sulla natura e la costituzione della mente umana."



ALGEBRA DI BOOLE

Con questo lavoro fondò la teoria di quelle che attualmente vengono dette **algebre di Boole** (o, semplicemente, **algebra booleana**).

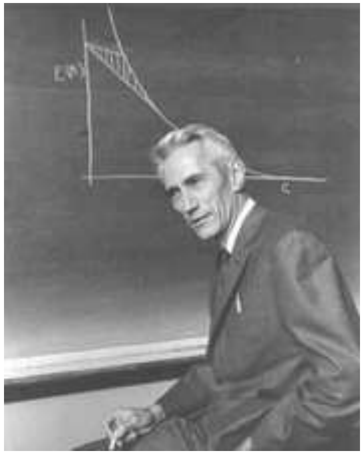
Pur mantenendo distinte le operazioni mentali da quelle algebriche, il compito di Boole fu quello di dare alla logica un “abito matematico” algebrico.



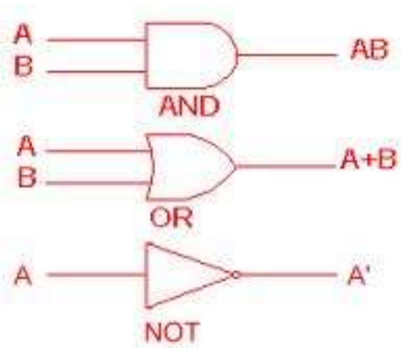
ALGEBRA DI BOOLE

L'opera maggiore di Boole è stata la base per gli studi sui **circuiti elettronici** e sulla **commutazione**, e ha costituito un passo importante verso la concezione dei moderni computer, grazie soprattutto a **Claude Elwood Shannon**, che ha riconosciuto la coincidenza tra il funzionamento dei **circuiti commutatori** e **la logica proposizionale**.

L'algebra di Boole è infatti rimasta pressoché ignorata per oltre 80 anni, fino al 1937, Shannon propose per primo di applicarla all'analisi e alla sintesi di circuiti a relè, che sono caratterizzati dai due stati di funzionamento **aperto** e **chiuso**.



Claude E. Shannon (1916-2001)



ALGEBRA DI BOOLE

L'algebra booleana originaria era definita su un insieme costituito da due elementi (successivamente è stata generalizzata per insiemi costituiti da un numero di elementi uguale a una qualsiasi potenza del 2), che a seconda dei casi vengono chiamati **vero**, **falso**.

Per esempio, nel calcolo proposizionale si usano come elementi di partenza delle proposizioni semplici, a ciascuna delle quali si possa attribuire in modo univoco il valore di verità **vero** o **falso**.

"5 è un numero dispari "

ha valore **vero**

"5 è un numero pari "

ha valore **falso**.

ALGEBRA DI BOOLE

Una proposizione semplice suscettibile di assumere i due soli valori, **vero** o **falso** si dice ***variabile booleana*** o *di* ***commutazione***.

Valore	Simboli alternativi
vero	1 - true - ON - SI - YES
falso	0 - false - OFF - NO

Unendo più proposizioni tramite nessi logici si formano proposizioni complesse, il cui valore di verità è ricavabile in maniera puramente formale dai valori delle proposizioni costituenti.

ALGEBRA DI BOOLE

Operatori AND (&&), OR (||), NOT (!)

Le operazioni più semplici che si possono compiere sulle proposizioni sono:

a) *congiunzione*, tramite il connettivo "e". Esempio:

"5 è un numero dispari **e** 6 è un numero pari"

b) *disgiunzione*, tramite il connettivo "o". Esempio:

"5 è un numero dispari **o** 6 è un numero pari"

c) *negazione*, tramite l' avverbio "non". Esempio:

"5 **non** è un numero dispari".

A queste operazioni corrispondono rispettivamente gli operatori booleani **AND** (**&&**), **OR** (**||**), **NOT** (**!**)

ALGEBRA DI BOOLE

AND (&&) produce una variabile con valore **v** se e solo se collega due variabili entrambe con valore **v**

f && f = f
f && v = f
v && f = f
v && v = v

OR (||) produce una variabile con valore **v** se collega due variabili di cui una almeno abbia valore **v**

f || f = f
f || v = v
v || f = v
v || v = v

NOT (**!**) produce una variabile con valore **v** se anteposto a una variabile con valore **f**, e una con valore **f** se anteposto a una **v**

!v = f
!f = v

- **AND** e **OR** collegano tra loro due variabili e sono detti perciò operatori *binari* o *diadici*;
- **NOT** opera su una sola variabile ed è detto perciò operatore *unario* o *monadico*.

ALGEBRA BOOLEANA

Tabelle di verità

Nell'applicare l'algebra di Boole ai circuiti elettronici si sostituisce spesso il valore **v** con **1** e il valore **f** con **0**. Perciò i tre operatori booleani visti in precedenza possono essere descritti anche con le seguenti *tabelle di verità*:

AND (&&)

x	y	x&&y
0	0	0
0	1	0
1	0	0
1	1	1

OR (||)

x	y	x||y
0	0	0
0	1	1
1	0	1
1	1	1

NOT (!)

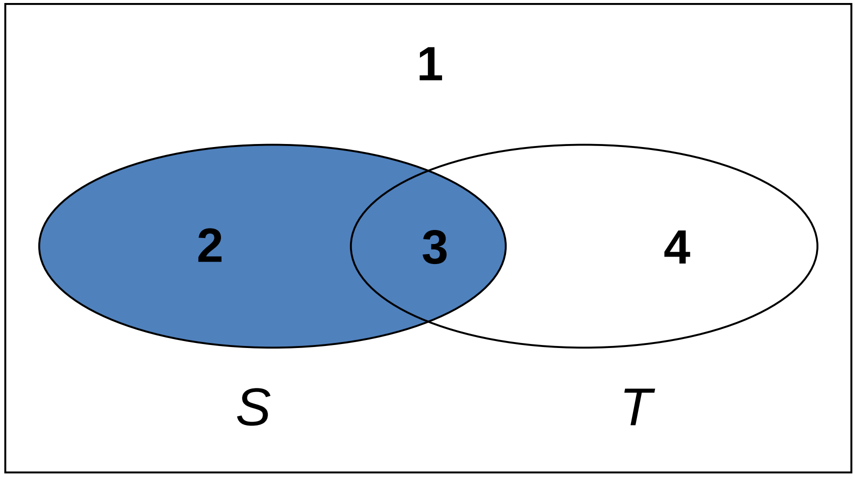
x	!x
0	1
1	0

In esse le righe corrispondono a tutte le possibili combinazioni dei valori di verità delle variabili booleane, e le colonne corrispondono alle variabili booleane e al valore dell'espressione risultante.

ALGEBRA BOOLEANA

Esiste una stretta somiglianza tra le tabelle di verità e i diagrammi di Venn impiegati nelle operazioni tra insiemi.

La figura seguente è un diagramma di Venn che mostra due insiemi, *S* e *T*, ognuno rappresentato da un’ellisse.

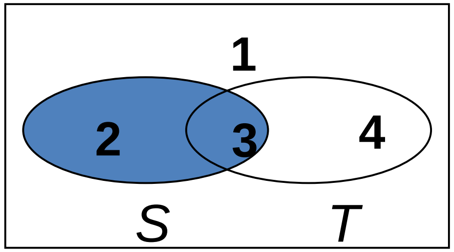


I due insiemi potrebbero rappresentare, rispettivamente:

S l’insieme delle persone di questa aula con i capelli castani
T l’insieme delle persone di questa aula con gli occhi azzurri

Le due ellissi dividono il piano in 4 regioni, numerate da 1 a 4:

ALGEBRA BOOLEANA



1. la regione 1 rappresenta gli elementi che non appartengono né a *S* né a *T*, cioè le persone che non hanno né i capelli castani né gli occhi azzurri
2. la regione 2 rappresenta la differenza *S*-*T*, cioè le persone che hanno i capelli castani e non gli occhi azzurri
3. la regione 3 rappresenta l'intersezione ($\cap$) di *S* e *T*, cioè le persone con capelli castani e occhi azzurri
4. la regione 4 rappresenta la differenza *T*-*S*, cioè le persone che hanno gli occhi azzurri e non i capelli castani
5. le regioni 2, 3 e 4 rappresentano l'unione ($\cup$) di *S* e *T*, cioè le persone con capelli castani o con occhi azzurri.

ALGEBRA BOOLEANA

L' operazione di congiunzione (**AND**) sui valori di verità corrisponde alla intersezione ($\cap$) tra insiemi, la disgiunzione (**OR**) corrisponde alla unione e la negazione (**NOT**) alla complementazione ($\bar{}$).

Algebra di Boole			**Teoria degli insiemi**	
Operatore	operazione	simbolo	Operazione	simbolo
AND	congiunzione	$\wedge$ (&& nel C)	intersezione (prodotto)	$\cap$ $(\cdot)$
OR	disgiunzione	$\vee$ (|| nel C)	unione (somma)	$\cup$ $(+)$
NOT	negazione	$\neg$ (! nel C)	complemento	$\sim$ $(\bar{\ })$

ALGEBRA BOOLEANA

Gli operatori **AND**, **OR**, **NOT** godono di due importanti proprietà, note come *teoremi di De Morgan*:

$$\overline{x \cdot y} = \overline{x} + \overline{y} \quad ; \quad \overline{x+y} = \overline{x} \cdot \overline{y}$$

che si possono dimostrare per *esaustione*, ossia scrivendo le tavole della verità dei due termini e osservando che sono uguali per tutte le possibili combinazioni dei valori assunti dalle variabili **x**, **y**.

Per esempio, per la prima si avrebbe:

x	y	$\overline{x}$	$\overline{y}$	$x \cdot y$	$\overline{x \cdot y}$	$\overline{x}+\overline{y}$
1	1	0	0	1	0	0
1	0	0	1	0	1	1
0	1	1	0	0	1	1
0	0	1	1	0	1	1

PROPRIETA' DELL'ALGEBRA BOOLEANA

Le operazioni tra le variabili booleane rispondono alle seguenti leggi dell'algebra booleana.

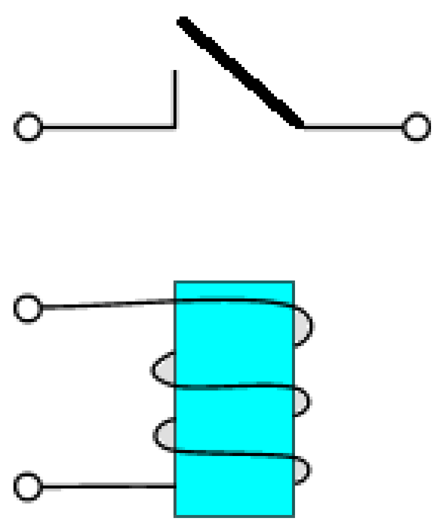
$x = \overline{\overline{x}}$	doppio complemento
$x + y = y + x$ $x \cdot y = y \cdot x$	commutatività
$(x + y) + z = x + (y + z)$ $(xy)z = x(yz)$	associatività
$x + yz = (x + y)(x + z)$ $x(y + z) = xy + xz$	distributività
$x + 0 = x$ $x \cdot 1 = x$	elemento neutro
$\overline{x + y} = \overline{x}\ \overline{y}$ $\overline{xy} = \overline{x} + \overline{y}$	leggi di De Morgan
$x + x = x$ $x \cdot x = x$	indepotenza
$x + \overline{x} = 1$ $x \cdot \overline{x} = 0$	inversi
$x + 1 = 1$ $x \cdot 0 = 0$	dominanza
$x + xy = x$ $x(x + y) = x$	assorbimento

ALGEBRA BOOLEANA

Porte logiche

Gran parte dell'importanza dell'algebra di Boole deriva dal fatto che essa trova applicazione nella teoria dei circuiti elettrici, in quanto è possibile realizzare dei dispositivi fisici abbastanza semplici che funzionano secondo le sue regole.

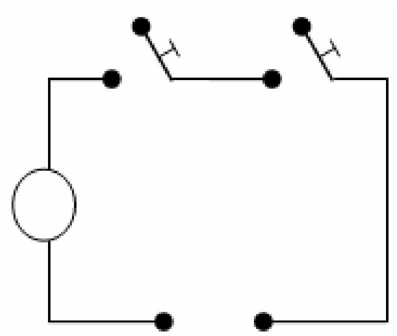
Tali dispositivi, che si chiamano ***porte logiche*** o ***gate***, si potrebbero realizzare in linea di principio con dei semplici interruttori comandati da relé: ogni interruttore si trova normalmente nello stato di aperto (in cui cioè non fa passare corrente), e viene chiuso fornendo una tensione opportuna (*di soglia*) al proprio relè.



ALGEBRA BOOLEANA

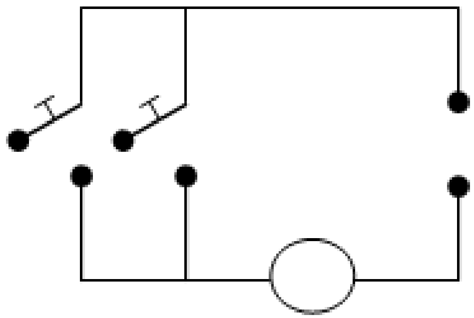
L'interruttore è inserito in un circuito comprendente un generatore che eroga la stessa tensione; questa corrisponde alla variabile booleana **1** (o vero), mentre una tensione inferiore corrisponde alla variabile **0** (o falso).

Se colleghiamo due di questi interruttori **in serie** con il generatore otteniamo un circuito equivalente all'operatore **AND**.



Infatti questo circuito fornisce in uscita la tensione **1** se e solo se entrambe le tensioni applicate ai due relè agli ingressi hanno valore **1** (e quindi fanno chiudere i corrispondenti interruttori).

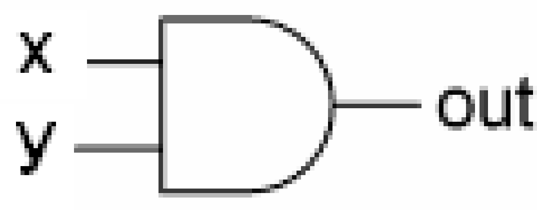
In alternativa, si possono collegare due interruttori **in parallelo** con un generatore, ottenendosi l'equivalente dell'operatore **OR**. Infatti adesso sarà presente la tensione **1** in uscita se almeno una delle due tensioni in ingresso assume il valore **1**.

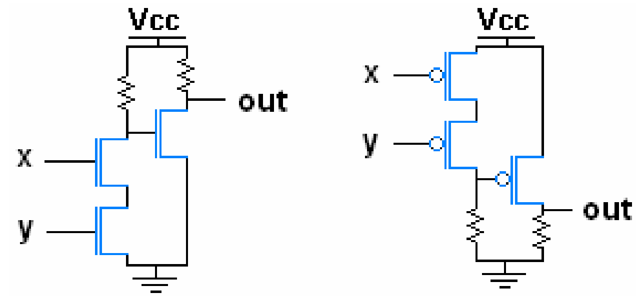


ALGEBRA BOOLEANA

Circuiti logici

Le funzioni dell'operatore **AND** possono essere svolte dal seguente circuito logico:



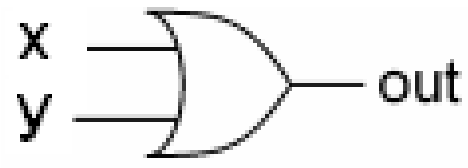


Circuito AND nMOS

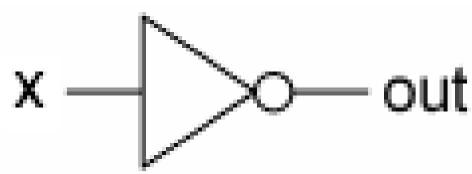
Circuito AND pMOS

ALGEBRA BOOLEANA

Le funzioni dell'operatore **OR** possono essere svolte dal seguente circuito logico:



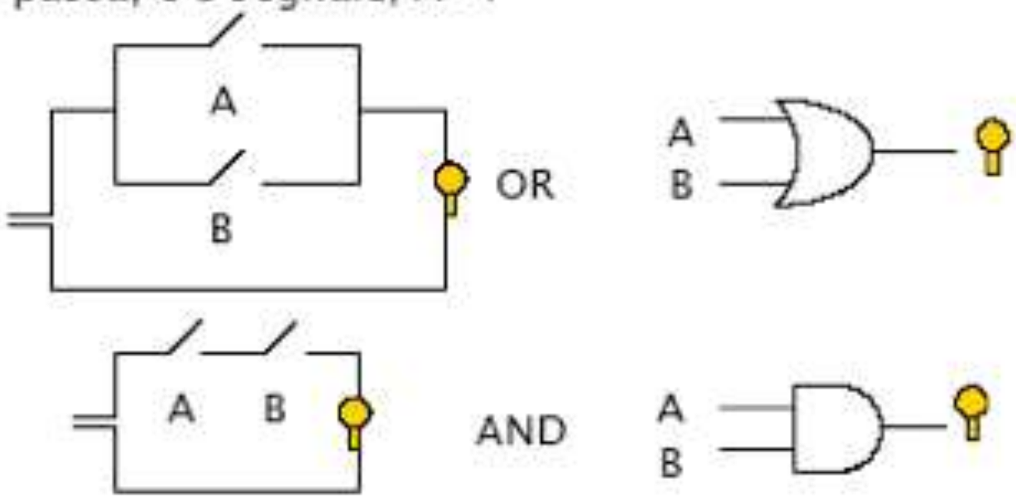
Le funzioni dell'operatore **NOT** possono essere svolte dal seguente circuito logico, detto anche *invertitore*:



ALGEBRA BOOLEANA

Esempio

- Circuito con interruttori e lampadina:
- Se un interrutore A è aperto allora la corrente non può passare, non c’e’ segnale, A=0
- Se un interruttore A è chiuso allora la corrente passa, c’è segnale, A=1



ALGEBRA BOOLEANA

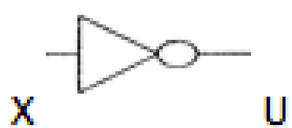
NOT

L'operatore NOT restituisce l'opposto del valore della variabile di ingresso.

Tabella:

X	U
0	1
1	0

Circuito:



Algebra: $U=\overline{X}$

ALGEBRA BOOLEANA

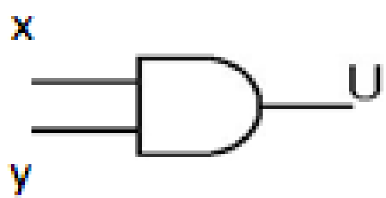
AND

L'operatore AND si applica a due variabili di ingresso. Ritorna 1 solo se entrambe sono 1.

Tabella:

X	Y	U
0	0	0
0	1	0
1	0	0
1	1	1

Circuito:



Algebra: U=X · Y

ALGEBRA BOOLEANA

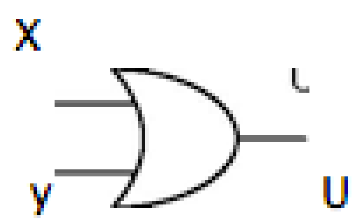
OR

L'operatore OR si applica a due variabili di ingresso. Restituisce 0 solo se entrambe sono 0.

Tabella:

X	Y	U
0	0	0
0	1	1
1	0	1
1	1	1

Circuito:



Algebra: U=X+Y

ALGEBRA BOOLEANA

EXOR

L'operatore EXOR o XOR si applica a due variabili di ingresso.

Restituisce 1 solo se solo una delle due è 1.

Tabella:

X	Y	U
0	0	0
0	1	1
1	0	1
1	1	0

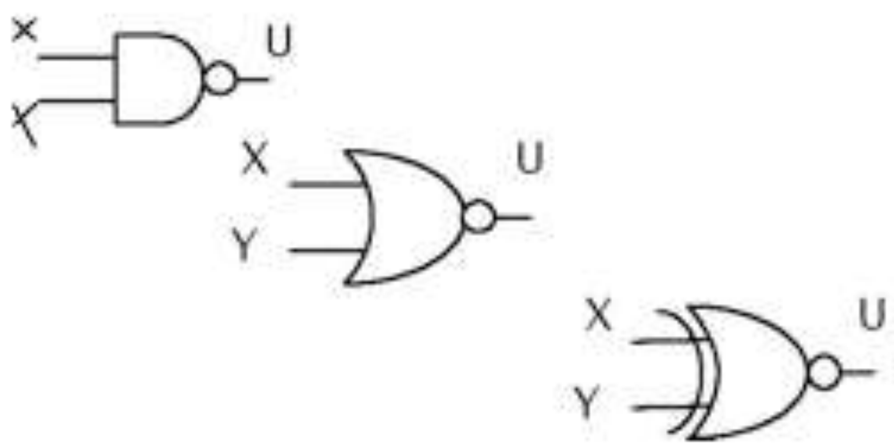
Circuito:

Algebra: $U=X \oplus Y$

ALGEBRA BOOLEANA

Un operatore + l'operatore NOT

Si può rappresentare la successione di un operatore AND, OR, XOR e l'operatore NOT in modo compatto con:



ALGEBRA BOOLEANA

Esempio

Ad una festa in maschera il padrone di casa costruisce un rilevatore automatico di costume per determinare quali persone ammettere.
La regola è la seguente:

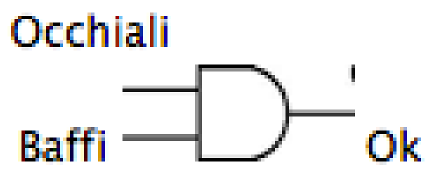
- le persone devono avere baffi finti e occhiali
- inoltre devono avere o scarpe da pagliaccio o cappello da pirata
- oppure devono avere un vestito in maschera completo
- ma non è ammesso chi ha un cane

Un custode preme i tasti di un circuito e in risposta ottiene/non attiene un segnale se la persona è ammessa/non ammessa

ALGEBRA BOOLEANA

Esempio...

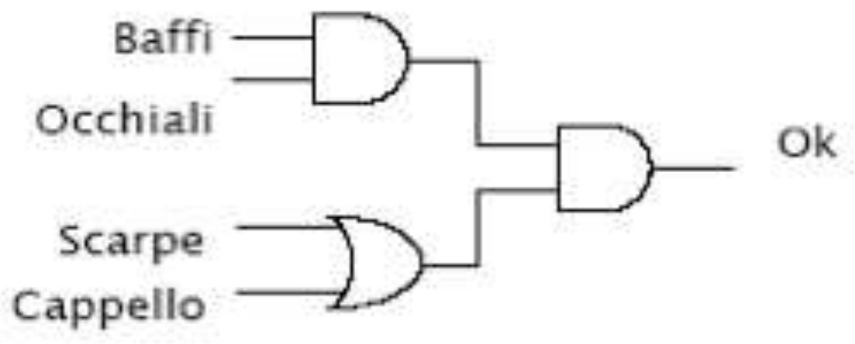
La prima condizione è che debbano avere sia baffi finti che occhiali



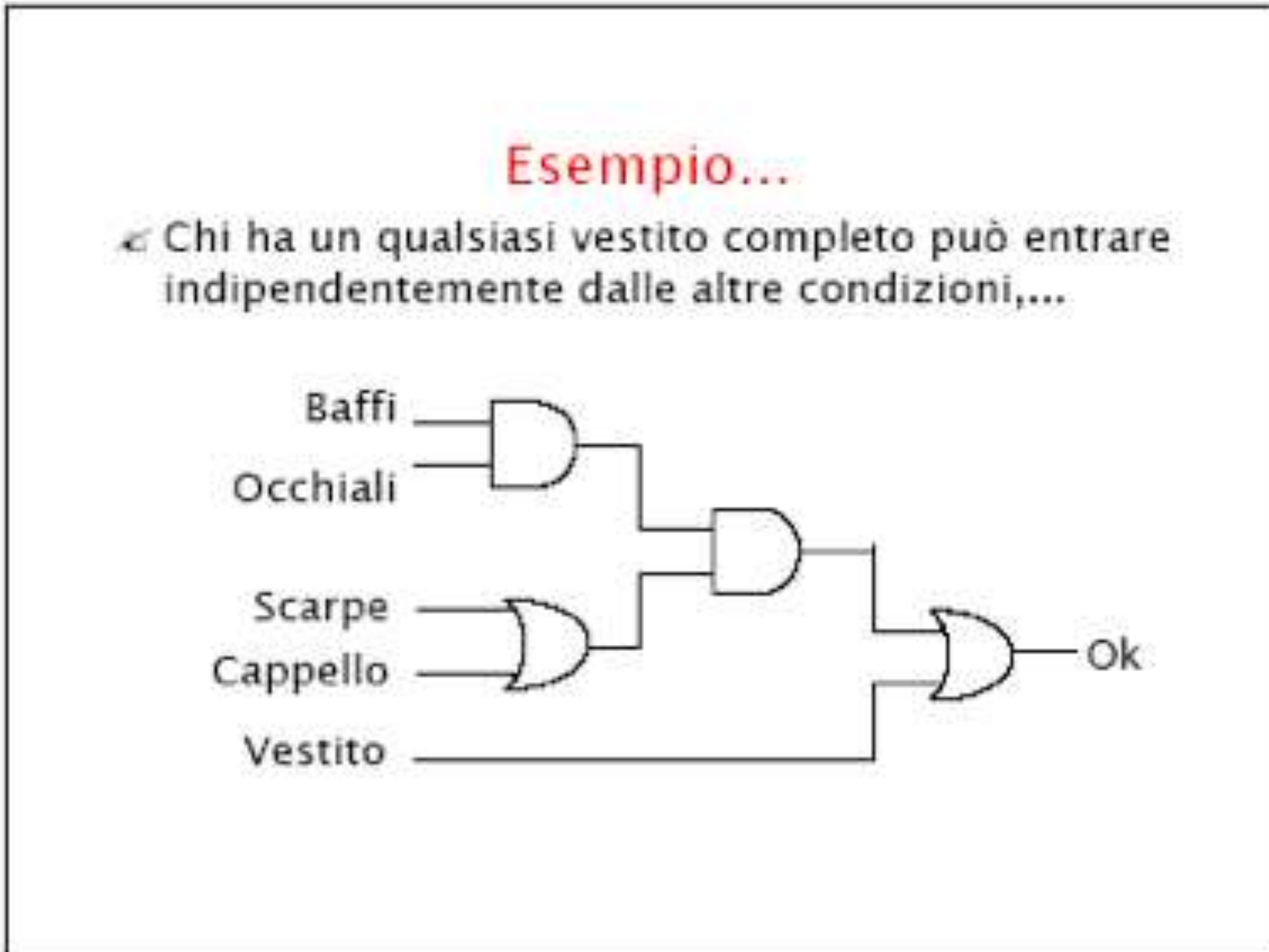
ALGEBRA BOOLEANA

Esempio...

- Inoltre è necessario avere anche scarpe da pagliaccio o cappello da pirata



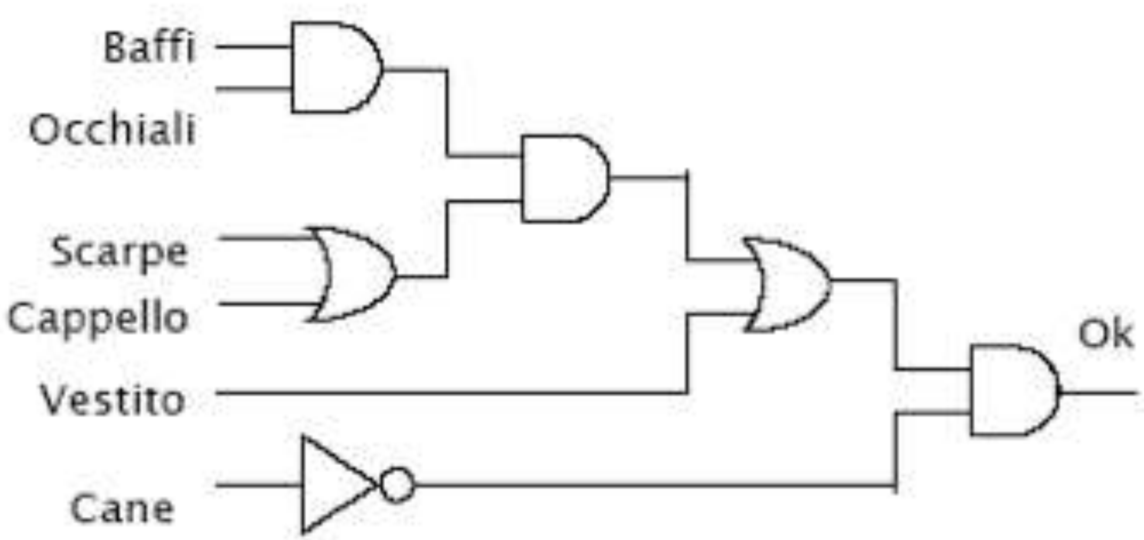
ALGEBRA BOOLEANA



ALGEBRA BOOLEANA

Esempio.

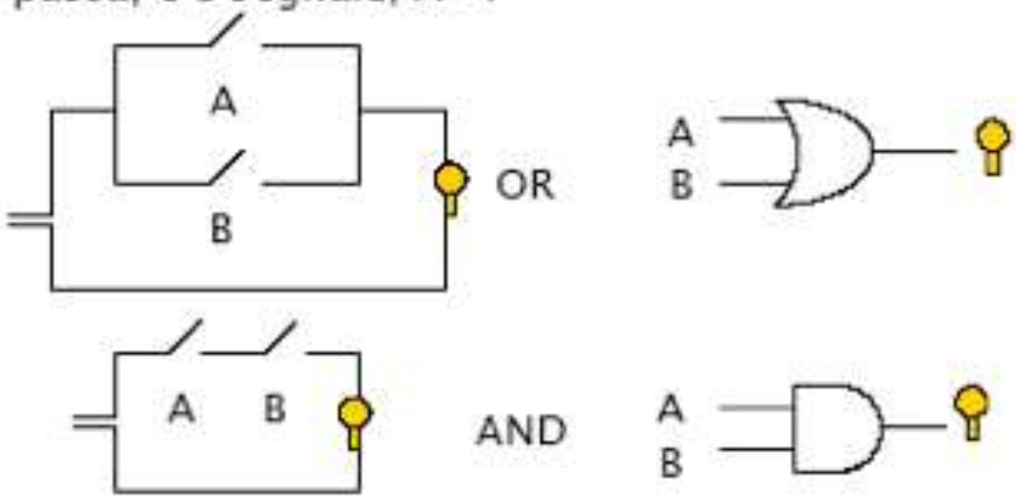
- ...ma chi ha un cane non può assolutamente entrare



ALGEBRA BOOLEANA

Esempio

- Circuito con interruttori e lampadina:
- Se un interrutore A è aperto allora la corrente non può passare, non c’e’ segnale, A=0
- Se un interruttore A è chiuso allora la corrente passa, c’è segnale, A=1



ALGEBRA BOOLEANA

END