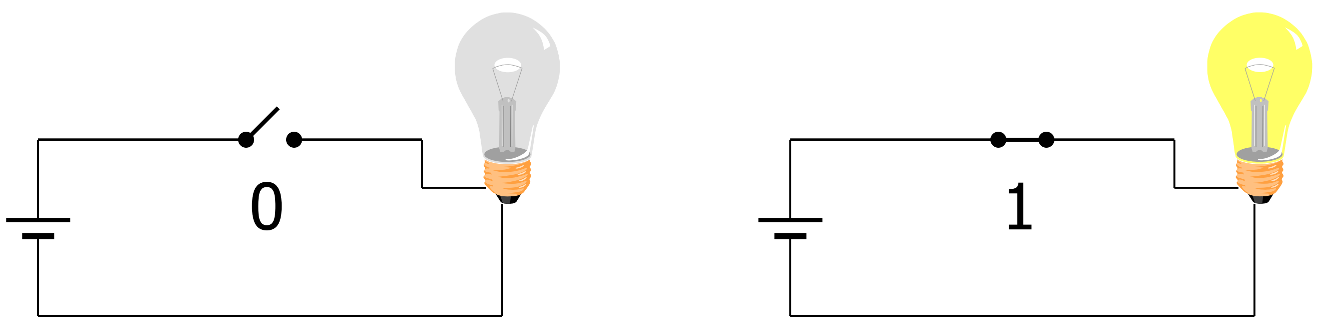
Programma del corso

- *Introduzione agli algoritmi*
- *Rappresentazione delle Informazioni*
- ***Elementi di Programmazione***
- *Architettura del calcolatore*
- *Reti di Calcolatori*

Calcolo proposizionale

Algebra Booleana

- Contempla due costanti 0 e 1 (falso e vero)
- Corrispondono a due stati che si escludono a vicenda
- Possono descrivere lo stato di apertura o chiusura di un generico contatto o di un circuito a più contatti



- Si definiscono delle operazioni fra i valori booleani: AND, OR, NOT sono gli operatori fondamentali

Vero e falso: logica binaria

- Una **proposizione** è una affermazione (formula ben formata di un linguaggio), che può essere **vera** oppure **falsa**
- **Es. "Mia madre mi vuole bene"**
- Non esiste una terza possibilità

Proposizioni

- **Proposizione semplice** =
  complesso linguistico o segnico per il quale ha senso attribuire un valore di verità

- **Valore di verità** (principio di bivalenza)

può essere	***vero***	***V***	***true***	***T***	***1***
	falso	***F***	***false***	***F***	***0***

- **Esempio1** la_luna_è_verde_a_pallini_blu
- **Esempio2** A

Connettivi

- Le proposizioni si possono connettere fra loro a formare ***proposizioni composte***

- Il valore di verità della proposizione composta dipende dal tipo di ***connettivo*** e dal valore di verità delle proposizioni componenti

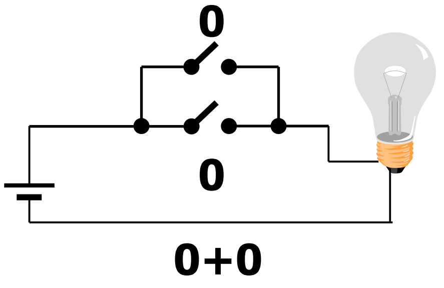
- I connettivi si distinguono dal numero di proposizioni (1, 2, … n) che possono connettere (***mono-, bi-, … n- argomentali***)

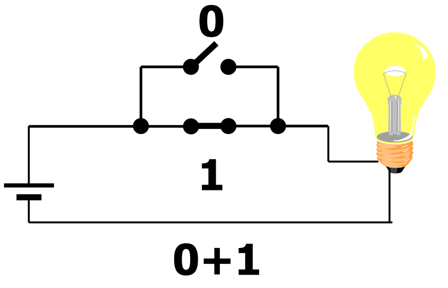
- **L'effetto** dei connettivi si specifica **elencando** tutti i possibili casi di combinazioni di valori vero e falso delle proposizioni componenti

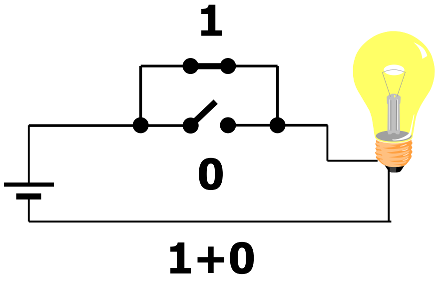
L'operazione di OR

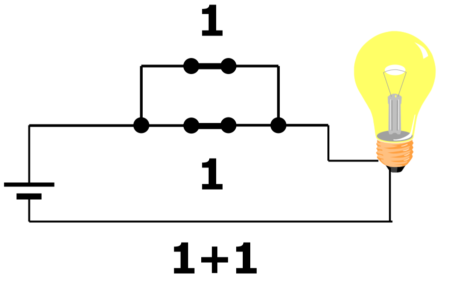
- Si definisce l'operazione di somma logica (OR): il valore della somma logica è il simbolo 1 se il valore di almeno uno degli operandi è il simbolo 1

0+0 = 0
0+1 = 1
1+0 = 1
1+1 = 1





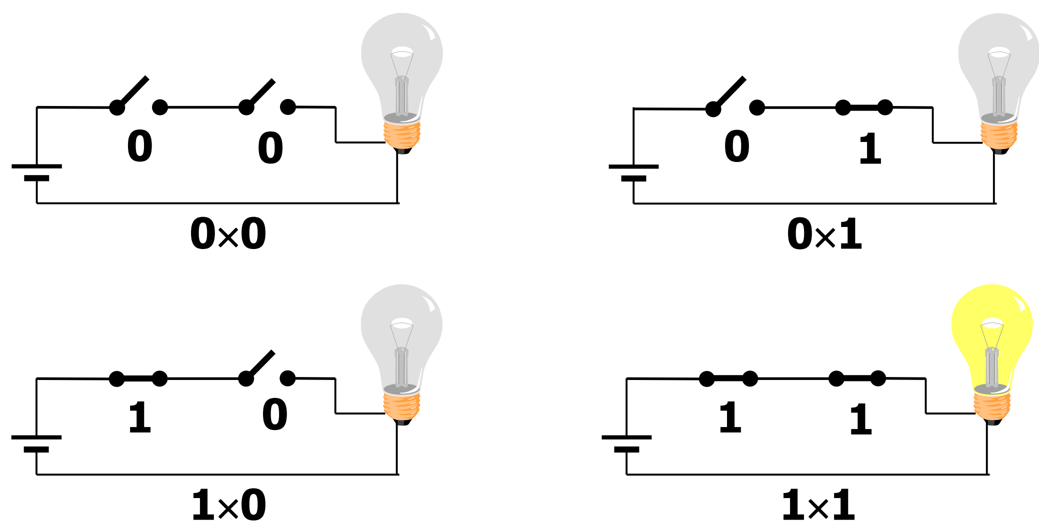




L’operazione di AND

- Si definisce l’operazione di prodotto logico (AND): il valore del prodotto logico è il simbolo 1 se il valore di tutti gli operandi è il simbolo 1

$0 \times 0 = 0$
$0 \times 1 = 0$
$1 \times 0 = 0$
$1 \times 1 = 1$



La negazione NOT

- Si definisce l'operatore di negazione (NOT): l'operatore inverte il valore della costante su cui opera

$$\overline{0} = 1$$
$$\overline{1} = 0$$

- Dalla definizione…

$$\overline{\overline{0}} = 0$$
$$\overline{\overline{1}} = 1$$

Variabili binarie

- Una variabile binaria indipendente può assumere uno dei due valori **0** e **1**

$$\begin{tikzcd} & 0 \\ x \arrow[ur] \arrow[dr] & \\ & 1 \end{tikzcd}$$

- Date *n* variabili binarie indipendenti, la loro somma logica (OR) è

$$x_1 + x_2 + \ldots + x_n = \begin{cases} \mathbf{1} \ \text{ se almeno una } x_i \text{ vale } \mathbf{1} \\ \\ \mathbf{0} \ \text{ se } x_1 = x_2 = \ldots = x_n = \mathbf{0} \end{cases}$$

AND e NOT con variabili binarie

- Date *n* variabili binarie indipendenti, il loro prodotto logico (AND) è

$$x_1 \times x_2 \times \ldots \times x_n = \begin{cases} \mathbf{0} & \text{se almeno una } x_i \text{ vale } \mathbf{0} \\ \mathbf{1} & \text{se } x_1 = x_2 = \ldots = x_n = \mathbf{1} \end{cases}$$

- La negazione di una variabile *x* è

$$\overline{x} = \mathbf{0} \quad \text{se} \quad x = \mathbf{1}$$
$$\overline{x} = \mathbf{1} \quad \text{se} \quad x = \mathbf{0}$$

Alcune identità

- Si verificano

$$x + 1 = 1 \qquad x \times 1 = x$$
$$x + 0 = x \qquad x \times 0 = 0$$
$$x + x = x \qquad x \times x = x$$

- Ad esempio…

$$\begin{array}{ccc} & x \times 1 = x & \\ x = 0 \swarrow & & \searrow x = 1 \\ 0 \times 1 = 0 & & 1 \times 1 = 1 \end{array}$$

OK!

Configurazioni delle variabili

- Date n variabili binarie indipendenti $x_1, x_2, \ldots, x_n$, queste possono assumere 2^n configurazioni distinte

Ad esempio per $n=3$ si hanno 8 configurazioni

$$x_1x_2x_3 \Rightarrow \begin{matrix} 000 & 001 & 010 & 011 \\ 100 & 101 & 110 & 111 \end{matrix}$$

- Una configurazione specifica è individuata univocamente da un AND (a valore 1) di tutte le variabili, dove quelle corrispondenti ai valori 0 compaiono negate

$$010 \longleftrightarrow \overline{x}_1 x_2 \overline{x}_3$$

Funzioni logiche

- Una variabile y è una funzione delle n variabili indipendenti x_1, x_2,…, x_n, se esiste un criterio che fa corrispondere in modo univoco ad ognuna delle 2^n configurazioni delle x_i un valore di y

$$y = \mathrm{F}(x_1, x_2, \ldots, x_n)$$

- Una rappresentazione esplicita di una funzione è la tabella di verità, in cui si elencano tutte le possibili combinazioni di x_1, x_2, …, x_n, con associato il valore di y

$$\boxed{y = x_1 + x_2} \rightarrow$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Una tabella di verità

- Date tre variabili booleane (**A,B,C**), si scriva la funzione **F** che vale 1 quando solo due di esse hanno valore 1

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Si può scrivere la funzione come somma logica delle configurazioni corrispondenti agli 1

$$F(A,B,C) = \bar{A}BC + A\bar{B}C + AB\bar{C}$$

Forma canonica: somma di prodotti (OR di AND)

– tutte le funzioni logiche si possono scrivere in questa forma

Un esempio: lo XOR

- La funzione XOR verifica la disuguaglianza di due variabili

x_1	x_2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

- L'espressione come somma di prodotti è quindi...

$$XOR = \overline{x}_1 x_2 + x_1 \overline{x}_2$$

- Torneremo a fine lezione su una applicazione sulle funzioni logiche nell’Intelligenza Artificiale

La negazione “NOT”

- Se P è una proposizione, si danno due casi possibili:
- Di conseguenza, per la negazione di P si avranno pure 2 casi corrispondenti:

Se:

P = VERO ⟶ NOT P = FALSO

P = FALSO ⟶ NOT P = VERO

“NOT” È UN OPERATORE BOOLEANO UNARIO (l’unico connettivo mono-argomentale)

Esempio *not*

- Il connettivo NOT nega il valore delle proposizioni

piove		***not*** **piove**
		not
V		**F**
F		**V**

Connettivi bi-argomentali

A	B		$\underline{V}$	=A	=B	and	or	=	<	→	$\underline{F}$	≠A	≠B	nand	nor	≠	>	≥
V	V		V	V	V	V	V	V	F	V	F	F	F	F	F	F	F	V
V	F		V	V	F	F	V	F	F	F	F	F	V	V	F	V	V	V
F	V		V	F	V	F	V	F	V	V	F	V	F	V	F	V	F	F
F	F		V	F	F	F	F	V	F	V	F	V	V	V	V	F	F	V

- Alcuni sono banali: rimandano tutti V, tutti F, stessi valori di A o B, valori negati di A e B
- Altri sono parte fondamentale del nostro linguaggio

Connettivi bi-argomentali:

$< \;=\; > \rightarrow \neq \geq$

- Si usano anche con valori numerici
- Si ricordi la corrispondenza di "***F*** a ***0*** " e di "***V*** a ***1***"

- **=** corrisponde a "se e solo se" ($\leftrightarrow$)

  Es. sarò_promosso "se e solo se" imparerò_la_materia
- "implica" ($\rightarrow$)

  Es. essere_multiplo_di_10 "implica" essere_multiplo_di_2
- $\neq$ corrisponde alla ***<u>o</u>*** alternativa: o … o ….

  Es. ***o*** ti_mangi_questa_minestra ***o*** ti_butti_dalla_finestra

Operatori booleani binari

AND	congiunzione
OR	disgiunzione inclusiva
XOR	disgiunzione esclusiva

La congiunzione “AND”

- Date due proposizioni P e Q l’operatore “AND” permette di costruire una nuova proposizione “P AND Q” che sarà VERA solo se P e Q sono entrambe vere

P	Q	P AND Q
V	V	V
V	F	F
F	V	F
F	F	F

La congiunzione “AND”

- Corrisponde alla congiunzione italiana ***e*** **(• ∧)**
- Esempio: per andare a Parigi, debbo stare bene ***e*** debbo avere i soldi per viaggio, vitto ed alloggio

Sto_bene	**Ho_i_soldi**		**Sto_bene AND Ho_i_soldi**
			AND
V	**V**		**V**
V	**F**		**F**
F	**V**		**F**
F	**F**		**F**

La disgiunzione inclusiva “OR”

- Date due proposizioni P e Q l’operatore “OR” permette di costruire una nuova proposizione “P OR Q” che sarà FALSA solo se P e Q sono entrambe false.

P	Q	P OR Q
V	V	V
V	F	V
F	V	V
F	F	F

La disgiunzione inclusiva “OR”

- Corrisponde alla disgiunzione ***o*** **(+** ∨)
- Esempio: per essere promosso, debbo essere preparato ***o*** debbo essere raccomandato

essere_preparato	essere_raccomandato		essere_promosso OR essere_raccomandato
			OR
V	V		V
V	F		V
F	V		V
F	F		F

La disgiunzione esclusiva “XOR”

- Date due proposizioni P e Q l’operatore “XOR” permette di costruire una nuova proposizione “P XOR Q” che sarà VERA quando P e Q hanno valori diversi.

P	Q	P XOR Q
V	V	F
V	F	V
F	V	V
F	F	F

Proposizioni composte

- Si valutano attribuendo valori di verità alle proposizioni semplici e applicando i connettivi
- L'ordine di applicazione dei connettivi rispecchia, a meno di parentesi "(" e ")" la seguente gerarchia:

not
and
or
$\mathbf{<} \quad \mathbf{=} \quad \mathbf{>} \quad \leq \ \neq \ \geq$

- **Es.:** per A vero e B falso not B or (B and not A) = A

	not B	or	(B and	not A)	= A
ordine di applicazione	3	4	2	1	5
risultati delle applicazioni	V	V	F	F	**V**

Proposizioni composte

- Vediamo meglio il metodo che è lo stesso che per le espressioni aritmetiche:

$$x + (y - z) / x$$

(con ordine di applicazione) 3 1 2

e con x = 5, y = 20, z = 10

<table>
<tr><td>x</td><td>+</td><td>(y</td><td>-</td><td>z)</td><td>/</td><td>x</td><td>passo</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td>+</td><td>(20</td><td>-</td><td>10)</td><td>/</td><td>5</td><td>0</td></tr>
<tr><td>5</td><td>+</td><td colspan="3">10</td><td>/</td><td>5</td><td>1</td></tr>
<tr><td>5</td><td>+</td><td colspan="5">2</td><td>2</td></tr>
<tr><td colspan="7">7</td><td>3</td></tr>
</table>

Proposizioni composte

- Vediamo meglio l'esempio precedente: ***A=V, B=F***

la proposizione **not B or (B and not A) = A**

con ordine di applicazione 3 4 2 1 5

<table>
<tr><th>not</th><th>B</th><th>or</th><th>(B</th><th>and</th><th>not</th><th>A)</th><th>=</th><th>A</th><th>passo</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>not</td><td>F</td><td>or</td><td>(F</td><td>and</td><td>not</td><td>V)</td><td>=</td><td>V</td><td>0</td></tr>
<tr><td>not</td><td>F</td><td>or</td><td>(F</td><td>and</td><td colspan="2">F)</td><td>=</td><td>V</td><td>1</td></tr>
<tr><td>not</td><td>F</td><td>or</td><td colspan="4">F</td><td>=</td><td>V</td><td>2</td></tr>
<tr><td colspan="2">V</td><td>or</td><td colspan="4">F</td><td>=</td><td>V</td><td>3</td></tr>
<tr><td colspan="7">V</td><td>=</td><td>V</td><td>4</td></tr>
<tr><td colspan="9">V</td><td>5</td></tr>
</table>

Tavole di verità

Per calcolare i valori di verità di una proposizione non elementare come:

(P AND Q) OR (NOT P AND NOT Q)

Tavole di verità

... si assegnano i valori di ingresso alle varie occorrenze di P e di Q

(P	AND	Q)	OR	(NOT P	AND	NOT Q)

Tavole di verità

… si assegnano i valori di ingresso alle varie occorrenze di P e di Q

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V						
V						
F						
F						

Tavole di verità

... si assegnano i valori di ingresso alle varie occorrenze di P e di Q

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V		V				
V		F				
F		V				
F		F				

Tavole di verità

… si assegnano i valori di ingresso alle varie occorrenze di P e di Q

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V		V		F		
V		F		F		
F		V		V		
F		F		V		

Tavole di verità

… si assegnano i valori di ingresso alle varie occorrenze di P e di Q

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V		V		F		F
V		F		F		V
F		V		V		F
F		F		V		V

Tavole di verità

... si calcolano poi i valori del primo AND e si cancellano le colonne dei valori usati

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V	V	V		F		F
V	F	F		F		V
F	F	V		V		F
F	F	F		V		V

Tavole di verità

… si opera allo stesso modo con il secondo AND

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V	V	V		F	F	F
V	F	F		F	F	V
F	F	V		V	F	F
F	F	F		V	V	V

Tavole di verità

... si calcola infine OR utilizzando come valori di ingresso le due colonne rimaste...

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V	V	V	V	F	F	F
V	F	F	F	F	F	V
F	F	V	F	V	F	F
F	F	F	V	V	V	V

Tavole di verità

... sotto OR, che è il "connettivo principale" troviamo la tavola di verità della proposizione.

(P	AND	Q)	OR	(NOT P	AND	NOT Q)
V	V	V	V	F	F	F
V	F	F	F	F	F	V
F	F	V	F	V	F	F
F	F	F	V	V	V	V

Teorema o tautologia

- Proposizione composta che è vera per qualsiasi combinazione di valori di verità attribuita alle sue proposizioni semplici
- Es. *La prima legge di De Morgan*

not (A and B) = not A or not B

2 1 6 3 5 4

ordine di esecuzione

not	(A	and	B)	=	not	A	or	not	B
F	V	V	V	V	F	V	F	F	V
V	V	F	F	V	F	V	V	V	F
V	F	F	V	V	V	F	V	F	V
V	F	F	F	V	V	F	V	V	F

Contraddizione

- Proposizione composta che è falsa per qualsiasi combinazione di valori di verità attribuita alle sue proposizioni semplici
- Es. $(A < B) = (A \geq B)$

$\quad\quad\quad\;\; 1 \quad\quad\; 3 \quad\quad 2$ ordine di esecuzione

(A	<	B)	=	(A	≥	B)
V	F	V	F	V	V	V
V	F	F	F	V	V	F
F	V	V	F	F	F	V
F	F	F	F	F	V	F

REGOLE DI INFERENZA

Consideriamo 2 importanti tautologie:

$((P \Rightarrow Q) \text{ AND } P) \Rightarrow Q$

MODUS PONENS

$((P \Rightarrow Q) \text{ AND NOT } Q) \Rightarrow \text{NOT } P$

MODUS TOLLENS

Esercizi:
dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q

Esercizi: dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q
V				V		
V				V		
F				F		
F				F		

Esercizi: dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q
V		V		V		V
V		F		V		F
F		V		F		V
F		F		F		F

Esercizi: dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q
V	V	V		V		V
V	F	F		V		F
F	V	V		F		V
F	V	F		F		F

Esercizi:
dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q
V	V	V	V	V		V
V	F	F	F	V		F
F	V	V	F	F		V
F	V	F	F	F		F

Esercizi:
dimostrare le due tautologie

MODUS PONENS

((P	$\Rightarrow$	Q)	AND	P)	$\Rightarrow$	Q
V	V	V	V	V	V	V
V	F	F	F	V	V	F
F	V	V	F	F	V	V
F	V	F	F	F	V	F

TAUTOLOGIA

Esercizi: dimostrare le due tautologie

MODUS TOLLENS

((P	$\Rightarrow$	Q)	AND	NOT Q)	$\Rightarrow$	NOT P

Esercizi: dimostrare le due tautologie

MODUS TOLLENS

((P	$\Rightarrow$	Q)	AND	NOT Q)	$\Rightarrow$	NOT P
V						F
V						F
F						V
F						V

Esercizi: dimostrare le due tautologie

MODUS TOLLENS

((P	$\Rightarrow$	Q)	AND	NOT Q)	$\Rightarrow$	NOT P
V		V		F		F
V		F		V		F
F		V		F		V
F		F		V		V

Esercizi: dimostrare le due tautologie

MODUS TOLLENS

((P	⇒	Q)	AND	NOT Q)	⇒	NOT P
V	V	V		F		F
V	F	F		V		F
F	V	V		F		V
F	V	F		V		V

Esercizi: dimostrare le due tautologie

MODUS TOLLENS

((P	$\Rightarrow$	Q)	AND	NOT Q)	$\Rightarrow$	NOT P
V	V	V	F	F		F
V	F	F	F	V		F
F	V	V	F	F		V
F	V	F	V	V		V

Esercizi:
dimostrare le due tautologie

MODUS TOLLENS

((P	$\Rightarrow$	Q)	AND	NOT Q)	$\Rightarrow$	NOT P
V	V	V	F	F	V	F
V	F	F	F	V	V	F
F	V	V	F	F	V	V
F	V	F	V	V	V	V

TAUTOLOGIA

IL MODUS PONENS

Il significato della prima tautologia è il seguente:

- Se ogni volta che accade un certo evento P allora accade anche un certo evento Q (ossia P è causa di Q)
- E io so che un evento P è realmente accaduto
- Allora posso inferire (= dedurre) che è avvenuto anche l'evento Q

$$((P \Rightarrow Q) \text{ AND } P) \Rightarrow Q$$

IL MODUS TOLLENS

Il significato della seconda tautologia è il seguente:

- Se ogni volta che accade un certo evento P allora accade anche un certo evento Q (ossia P è causa di Q)
- E io so che l'evento Q non è accaduto
- Allora posso inferire (= dedurre) che non è avvenuto neppure l'evento P

$$((P \Rightarrow Q) \text{ AND NOT } Q) \Rightarrow \text{NOT } P$$

QUINDI ....

Alla base della **logica matematica** ci sono:

•Le regole di composizione di proposizioni elementari per la costruzione di proposizioni complesse

•Il calcolo dei valori di verità delle proposizioni complesse

•Un ruolo importante è svolto dalle **tautologie**, che sono sempre vere, e dalle **contraddizioni**, che sono sempre false

Teorie Ipotetico-Deduttive

In una teoria ipotetico-deduttiva sono date alcune proposizioni iniziali (**ASSIOMI**) dalle quali, attraverso le regole di inferenza, potranno essere DEDOTTE nuove proposizioni (**TEOREMI**) attraverso procedure di DIMOSTRAZIONE, che sono una applicazione del **MODUS PONENS** e del **MODUS TOLLENS**.

Ma non sempre le cose vanno come ci aspettiamo …

I Paradossi

Un paradosso è una proposizione P tale che

- Se P è vera, allora P è falsa
- Se P è falsa, allora P è vera
- Ossia P è vera se e solo se P è falsa!

Esaminiamo dei paradossi famosi e un paradosso divertente

Epimenide di Creta afferma: "Tutti i cretesi mentono"

- Se Epimenide, che è cretese, dice il vero, allora sta mentendo
- Se Epimenide sta mentendo, allora dice il vero

Più semplicemente potrei dire:

"IO STO MENTENDO"

•Se dico il vero, allora sto mentendo, ma

•Se sto mentendo, allora ciò che dico è falso e dunque non sto mentendo, ossia dico il vero.

Il paradosso del Barbiere
Bertrand Russell

Un generale ordina al barbiere della caserma (che è un soldato) di radere tutti e soli i soldati che non si radono da sé.

Il barbiere deve radersi o no?

Se si rade, allora non deve radersi.

Ma se non si rade, allora deve radersi …

Il Ponte dei Bugiardi

Pierino è un bugiardo.

Un giorno suo padre, stanco delle bugie, lo conduce davanti a un ponte e gli dice: “Questo è il ponte dei bugiardi, se un mentitore lo attraversa, crolla!”

Pierino, spaventato, giura di non dire più bugie, e torna a casa.

Il padre attraversa il ponte, e il ponte crolla.

Infatti il padre è un mentitore:

IL PONTE DEI BUGIARDI NON ESISTE!

Il paradosso dell’avvocato

In Academica (II, 95) Cicerone (106-43 a.C.) racconta il seguente caso, attribuito agli stoici.

Il filosofo Protagora accettò di avere come studente di legge un ragazzo che non poteva permettersi di pagarlo subito, con la clausola che egli l'avrebbe pagato dopo aver **vinto** la sua prima causa.

Poiché, dopo gli studi, lo studente non si decideva a praticare l'avvocatura e quindi non lo pagava, Protagora lo citò in giudizio.

Lo studente, che non poteva permettersi un avvocato, decise di difendersi da solo.

Il paradosso dell’avvocato

A. Protagora sosteneva che, se avesse vinto la causa, avrebbe dovuto essere pagato in base alla sentenza.

E se avesse perso, avrebbe dovuto essere pagato in base all'accordo.

Quindi, in ogni caso, doveva essere pagato!

B. Lo studente sosteneva che, se avesse vinto la causa, non avrebbe dovuto pagare in base alla sentenza.

E se avesse perso, non avrebbe dovuto pagare in base all'accordo.

Quindi, in ogni caso, **non** doveva pagare!

Conclusione paradossale

Non ho niente da dire,
e lo sto dicendo!

John Cage

Connettivi tri-, .. , n- argomentali

- BASTA, non se ne può più !
- Ma come descrivere situazioni **più complesse** in cui non sia intuitivo l'uso dei connettivi studiati?
- Sono sufficienti opportune combinazioni di ***not*** ***and*** ***or*** per esprimere un qualsiasi connettivo!
- Es.

A	>	B		A	and	not	B
V	F	V		V	F	F	V
V	V	F		V	V	V	F
F	F	V		F	F	F	V
F	F	F		F	F	V	F

Forme disgiuntive normali

- Utilizzano solo ***not*** ***and*** ***or*** per descrivere risultati dipendenti da valori di verità di proposizioni semplici
- Ci si riferisce alla **tavola di verità** in cui appaiono tutte le possibili assegnazioni di verità delle proposizioni semplici
- Si considerano le **combinazioni** per le quali si ottiene valore di verità V
- Per ogni tale combinazione le proposizioni con valore di **verità F** si fanno precedere da ***not***
- Le proposizioni così modificate vengono fra loro connesse con ***and*** (si creano congiunzioni)
- Tali congiunzioni vengono fra loro connesse con ***or*** , formando una forma disgiuntiva normale

Forme disgiuntive normali

- Es. A=B

A	B		=		
V	V		V	*	A and B
V	F		F		
F	V		F		
F	F		V	*	not A and not B

- Risultato: (A and B) or (not A and not B)

Forme disgiuntive normali

- **Esercizio**: condizione di presa nel **Gioco della Briscola** per il giocatore che tira per primo (**G1**)

- Proposizioni che definiscono la condizione:
  - i semi sono uguali (**SU**)
  - il secondo giocatore ha tirato una briscola (**B2**)
  - il valore della carta del I giocatore è maggiore del valore della carta del II giocatore (**M1**)

Forme disgiuntive normali

- Condizione di presa nella briscola per il primo giocatore (elenchiamo le varie combinazioni)

SU	B2	M1		G1
V	V	V		V
V	V	F		F
V	F	V		V
V	F	F		F
F	V	V		F
F	V	F		F
F	F	V		V
F	F	F		V

- **G1 = (SU and B2 and M1) or (SU and not B2 and M1) or (not SU and not B2 and M1) or (not SU and not B2 and not M1)**