# Corso di INFORMATICA
# Corso di Laurea Triennale in Matematica

# Calcolo Proposizionale

## Breve introduzione

# Vero e Falso

Una **proposizione** è una affermazione che può essere **vera** oppure **falsa**

- **Es.:**
  - **“Mia_madre_mi_vuole_bene”**
  - **“Il_cielo_è_blu”**
  - **“A”**

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

| | | | | |
|---|---|---|---|---|
| ***vero*** | ***V*** | ***true*** | ***T*** | ***1*** |
| ***falso*** | ***F*** | ***false*** | ***F*** | ***0*** |

# I Connettivi Logici

- Le proposizioni si possono connettere fra loro per 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 connettono (***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

# La Negazione: NOT

"NOT" È UN OPERATORE BOOLEANO UNARIO (l'unico connettivo mono-argomentale)

- Se P è una proposizione, ci sono due casi possibili:
- Per la negazione di P si ci sono 2 casi corrispondenti:

P = VERO → NOT P = FALSO

P = FALSO → NOT P = VERO

# NOT: esempio

- 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 fondamentali

# Connettivi bi-argomentali

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

- Si usano anche con valori numerici
- In genere si fa corrispondere ***“F”*** a ***0*** e ***“V”*** a ***1***

- ***“=“*** corrisponde a “se e solo se” ( $\leftrightarrow$ )
  - **Es**. promosso “se e solo se” preparato
- “implicazione” ( $\rightarrow$ )
  - **Es**. multiplo_di_10 “implica” multiplo_di_2
- $\neq$ corrisponde alla ***<u>o</u>*** alternativa: o … o ….
  - **Es**. 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 è 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)

AND **(∧)** - Corrisponde alla congiunzione italiana ***e***
Esempio: per andare a Parigi, devo stare bene ***e*** devo 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 è 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)

- OR (∨) - Corrisponde alla disgiunzione ***o***
- Esempio: per essere promosso, devo essere preparato ***o*** devo 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 è 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**

**< = >** ≤ ≠ ≥

# Proposizioni Composte

ES.: (not B) OR (B AND NOT A) = A

CON: **A vero e B falso**

| | (NOT B) | OR | (B | AND | NOT A) | = | A |
|---|---|---|---|---|---|---|---|
| ordine di applicazione | 1 | 4 | 0 | 3 | 2 | **5** | 0 |
| risultato delle applicazioni | V | V | F | F | F | **V** | V |

# Proposizioni Composte

- Il metodo è simile a quello per le espressioni aritmetiche.

- Es.: x = 5, y = 20, z = 10

```
                          x + (y - z) / x
ordine di applicazione      3     1     2
```

<table>
<tr><th>x</th><th>+</th><th>(y</th><th>-</th><th>z)</th><th>/</th><th>x</th><th>passo</th></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

Esempio precedente: **not B or (B and not A) = A**

*Con* ***A=Vero*** ***B=Falso***

<table>
<tr><td>not</td><td>B</td><td>or</td><td>(B</td><td>and</td><td>not</td><td>A)</td><td>=</td><td>A</td><td>passo</td></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>

# Tabelle di Verità

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

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

Sfruttiamo le TABELLE (o TAVOLE) di VERITÀ

# Tabelle 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) |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

# Tabelle 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 | | | | | | |

# Tabelle 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 | | | | |

# Tabelle 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 | | |

# Tabelle 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 |

# Tabelle 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 |

# Tabelle di Verità

… si opera allo stesso modo con il secondo AND

| (P | AN D | 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 |

# Tabelle 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 |

# Tabelle di Verità

… sotto “OR”, che è il “connettivo principale” di questa proposizione, troviamo i valori di verità della proposizione intera.

| (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 |

# 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. **$(A < B) = (A \geq B)$**

1 3 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 |

# Connettivi tri-, .. , n- argomentali

BASTA, non se ne può più !

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 ognuna di queste combinazioni 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:

  - **SU:** i semi sono uguali
  - **B2:** il secondo giocatore ha tirato una briscola
  - **M1:** il valore della carta del I giocatore è maggiore del valore della carta del II giocatore

# Forme Disgiuntive Normali

Condizione di presa nella briscola per il primo giocatore (elenco delle 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)**