

Necessità → Rappresentare l'esecuzione di funzioni

$$f: \text{Input} \rightarrow \text{Output}$$

⇒ Vogliamo descrivere COME calcolare  $f$

ovvero vogliamo rappresentare ALGORITMI

→ Sequenza finita di passi discreti che descrivono come calcolare  $f$

Strategie per il calcolo di  $f$

⇒ Abbiamo bisogno di uno strumento che permette di manipolare dati descrivendo Algoritmi

## PROGRAMMA

⇒ Lingue di programmazione (PL) : linguaggi formali che permettono di scrivere programmi

Programmi : frasi ben formate di un PL, sono una rappresentazione FINTA di una sequenza potenzialmente INFINITA di effetti su una macchina

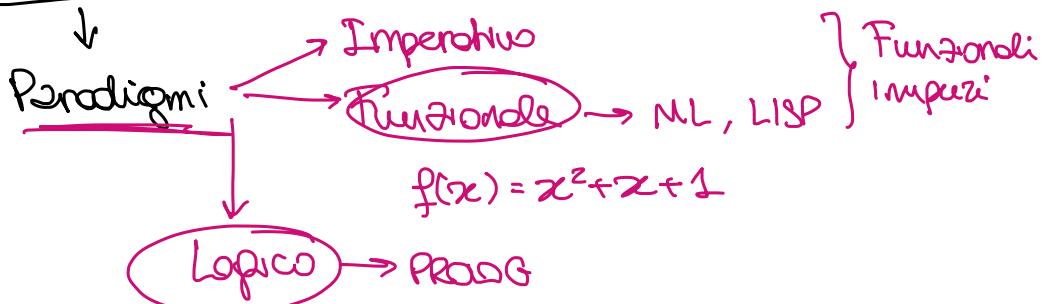
## APPLICAZIONI

- Applicazioni scientifiche FORTRAN
- Applicazioni economiche COBOL
- Applicazioni AI LISP
- Applicazioni Sistemi C
- Software Web JAVA
- ⋮

## LIVELLO

→ Basso Livello (Linguaggio Machine  
↳ Assembly)

→ Alto Livello

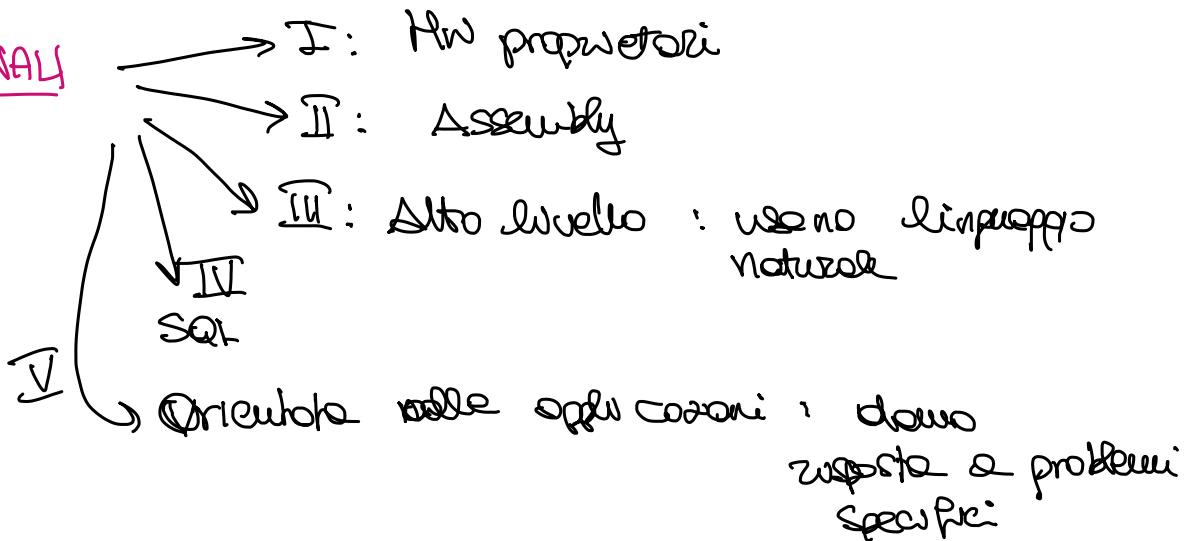


Procedurale → Imperativo / strutturato  
(procedurale senza solti)

OOP → Java

Dichiarative → Logico / Funzionale  
(basate matematiche)

## GENE RAZZONALI



## CARATTERISTICHE

- sequenziali
- concorrenti
- parallelo
- distribuiti
- OO
- scripting
- ⋮

### LEGGIBILITÀ

### WRITABILITY

### AFFIDABILITÀ

- quantità limitata di costanti
- poco molteplicazione di elementi
- poco overloading di operatori

Type checking

Gestione eccezioni

Aliasing (negativo)

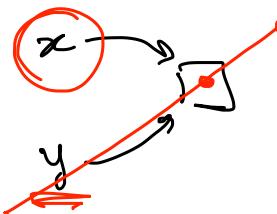
Orientalità

Astrosione (OO)

Tipi di dati

Espressività

Sintassi



## IMPLEMENTARE UN PL

↳ Architettura delle macchine

↳ Metodologie di progettazione di programmi

→ Dati e programmi in memoria (separata dalla CPU)

Istruzioni e dati che vengono tra memoria e CPU

```
program-counter  
repeat forever  
    fetch instruction nella memoria usando program-counter  
    aggiorniamo program-counter  
    decode instruction  
    execute instruction  
end repeat
```

Implementare PL significa costruire <sup>la</sup> Macchina astratta per il PL

Macchina astratta programma linguaggio alto livello

Macchina fisica programma Linguaggio macchina

PL → linguaggio L con i suoi costituti sintattici



Programma (fatto ben formato nel PL)

Per eseguirlo costruiamo la macchina astratta per L

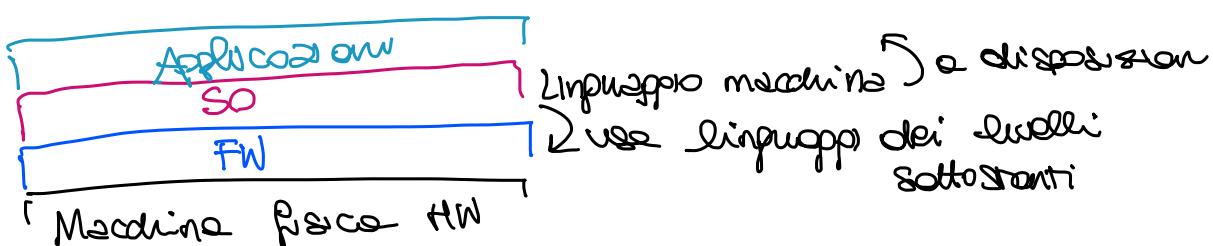
## MACCHINA ASTRATTA

Dato L linguaggio di programmazione  
 $M_L$  macchina astratta di L e' un  
insieme di strutture dati e  
algoritmi che permettono di eseguire  
programmi scritti in L

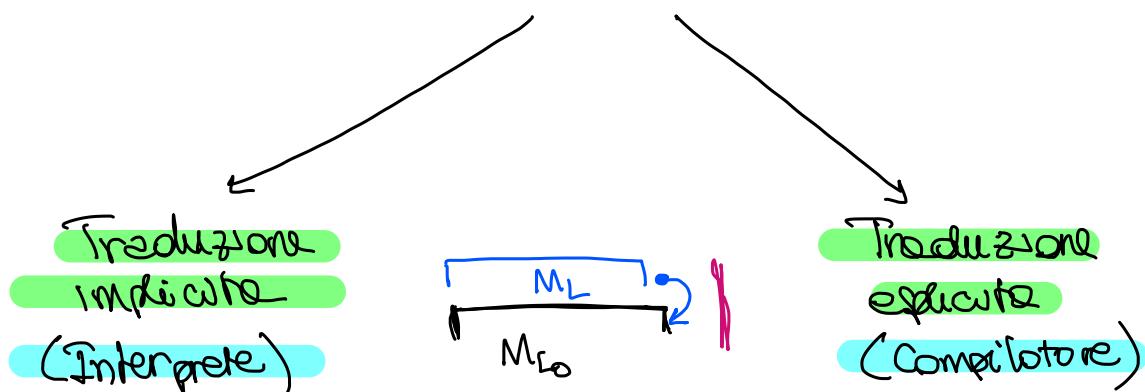
## LINGUAGGIO MACCHINA

Dato M macchine astratte se  
linguaggio  $L_M$  e' chiamato linguaggio  
macchina di M, ed e' l'insieme di  
tutte le strutture eseguibili (riconoscibili)  
da M.

Implementare un PL  $\rightarrow$  Realizzare  $M_L$



Implementazione di L



Realizziamo  $M_L$

Simulando ogni istruzione di L  
nel linguaggio di  $M_{L_o}$  (macchina  
sottostante)

Realizziamo  $M_L$   
producendo internamente  
il programma scritto in L  
in un programma scritto in lo

## Notazione

$\text{Prog}^L$  = insieme dei programmi scritti in L

D = insieme dei dati

$P^L \in \text{Prog}^L$  programma scritto in L

$$P^L : D \rightarrow D$$

$\forall \text{input} \in D \quad P^L(\text{input}) = \text{output} \in D$

esecuzione di  $P^L$

sul dato input restituisce il dato output

## INTERPRETE

### Macchina Universale

Un interprete, per il linguaggio L, scritto in Lo  
(interprete viene scritto in un linguaggio implementato da  $M_{Lo}$  sottostante) è un programma

$$I^{LoL} : \text{Prog}^L \times D \rightarrow D$$

$\forall \text{input} \in D. \quad I^{LoL}(P^L, \text{input}) = \text{output}$

$\forall P^L \in \text{Prog}^L$

