

# Lezione 1

Terminologie : → da qui in avanti verrà abbreviato con LSP

Linguaggi di programmazione  $\rightarrow$  È un linguaggio usato per esprimere un algoritmo.

È un linguaggio, elaborato da un processore ( processore = qualcosa con capace di capire il linguaggio, anche l'uomo )

Programma  $\rightarrow$  codifica di un algoritmo in un linguaggio

Processore  $\rightarrow$  macchina, virtuale o reale, che esegue l'algoritmo del programma.  
↳ intesa come ADE

Linguaggio completo :

Gli LSP sono computazionalmente completi se possono esprimere qualunque funzione calcolabile ( grand purpose ). "Devono poter simulare ogni Turing machine".

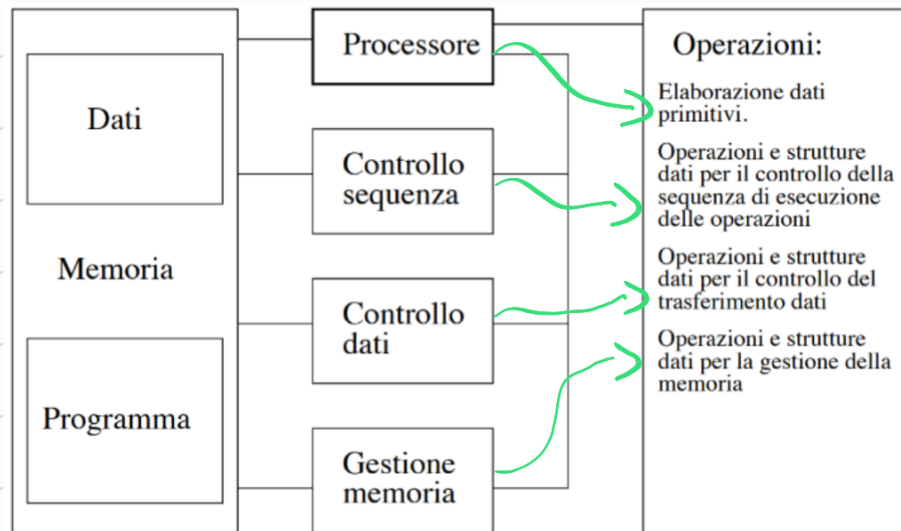
Sono completi solo se riuscano ad esprimere programmi indecidibili. ( SQL e HTML non sono completi perché terminano sempre ).

Macchine astratte, definizione :

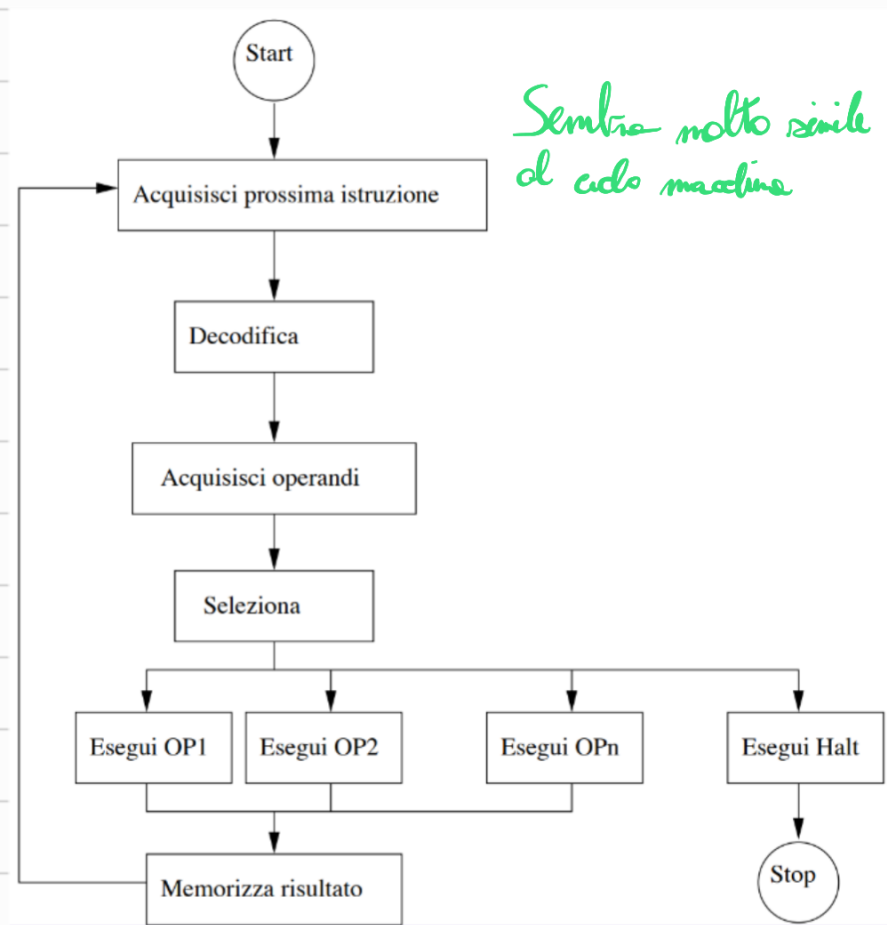
Dato un LSP "P" e una macchina astratta M ( computabile ), si dice che

...con cui si può costruire una macchina virtuale per il linguaggio L. La macchina virtuale è un'equivalente insieme di strutture dati e algoritmi capaci di memorizzare ed eseguire programmi scritti in un linguaggio L.

## Struttura di una macchina astratta (formate da processore e memoria)



## Processore di una macchina astratta



## Esempi di macchine astratte

Linguaggio	macchina astratta
C	gcc + OS + HW
JAVA	Javac + JVM + OS + HW
C#	exe + CLR + OS + HW

## Traduttori linguaggio $\rightarrow$ Macchina astratta

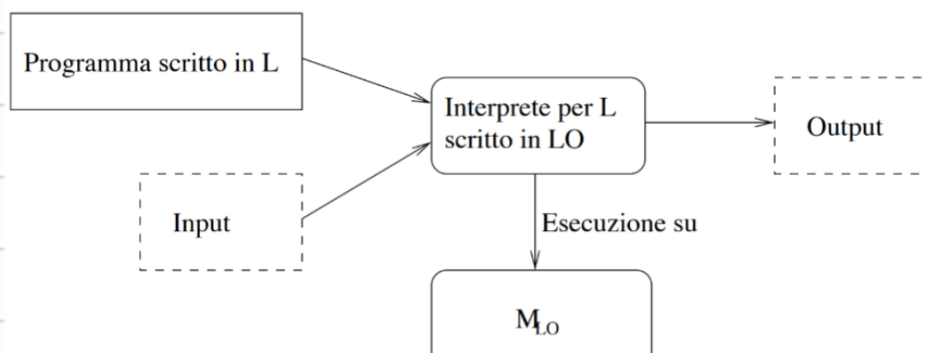
- Interpreti: traducono ed eseguono un costrutto alla volta.

Pro: debug - fase di sviluppo.

- Compilatori: traduzione di tutto il programma.

Pro: velocità di esecuzione finale; più controlli in anticipo

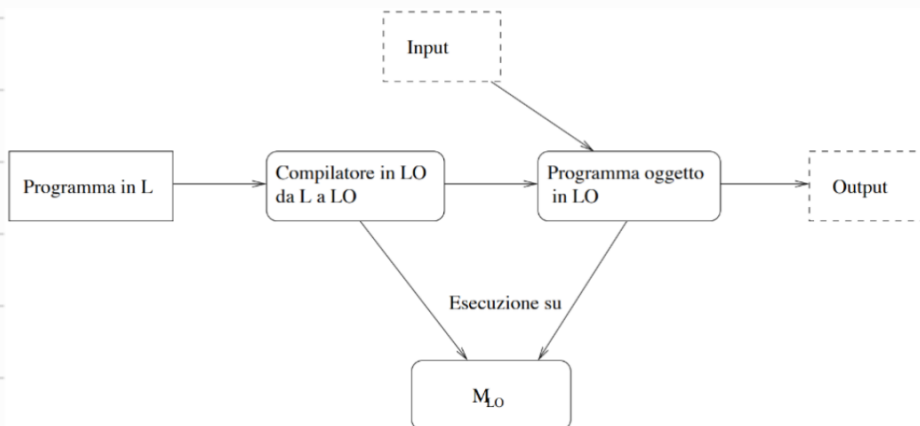
## Interpretazione pura



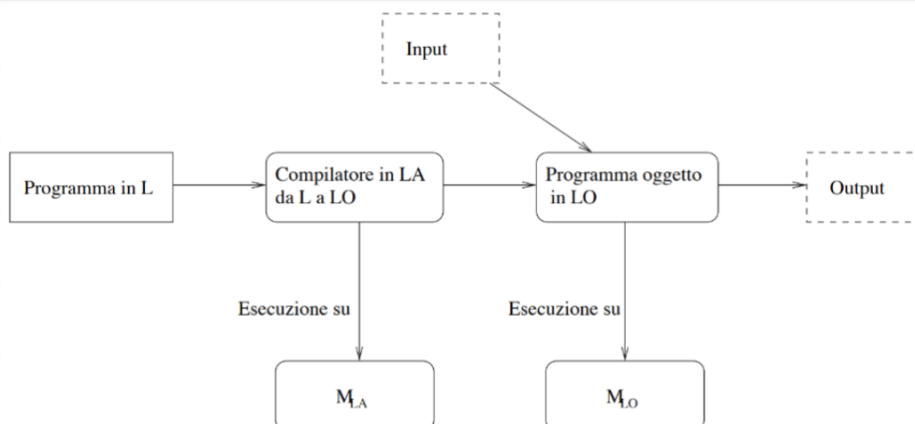
Come python.

Compilazione pura / caso semplice

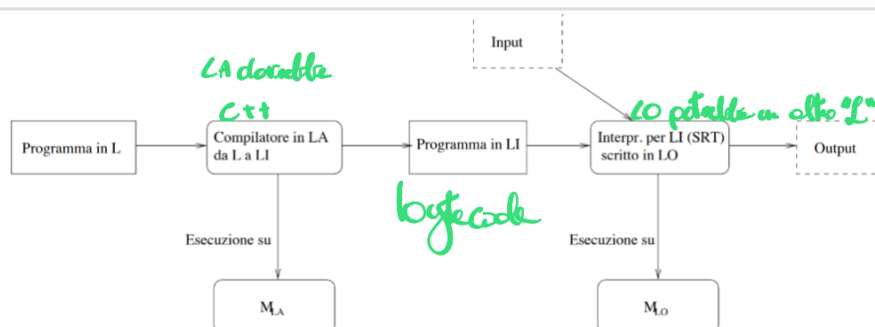
Questo potrebbe  
essere il caso di  
C



Compilazione pura, caso generale

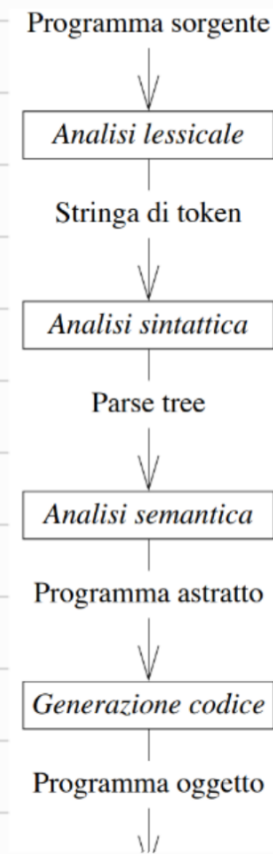


Compilazione per macchine intermedie



Esercizio: Se L è il linguaggio Java, chi sono LA, LI, e LO?

# Compilatore



→ Verifica delle caratteristiche lessicali

→ generazione dei token

→ Verifica delle caratteristiche sintattiche

↳ derivazione dalla grammatica

## Proprietà dei linguaggi:

- **Semplicità** - concisione vs leggibilità
- **Astrazione** - increspatura e chiarezza
- **Espressività** - ricchezza di costrutti vs semplicità
- **Ortogonalità** - poter ricorrere alle regole del linguaggio

Queste caratteristiche variano, in base o in modo, o anche del linguaggio

