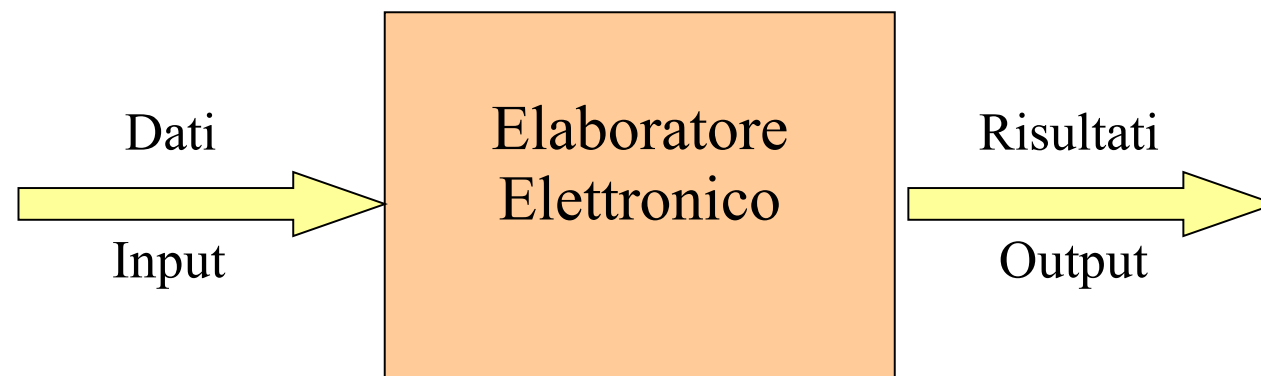


Linguaggi di programmazione

Programmazione

- L'attività con cui si predispone l'elaboratore ad eseguire un ***particolare insieme di azioni*** su ***particolari dati***, allo scopo di ***risolvere un problema***



Linguaggio di programmazione

- Insieme di regole per la descrizione formale di un algoritmo eseguibile da un calcolatore:
 - **Lessico** (alfabeto): insieme dei termini disponibili
 - **Sintassi**: forma delle frasi
 - **Semantica**: significato delle frasi

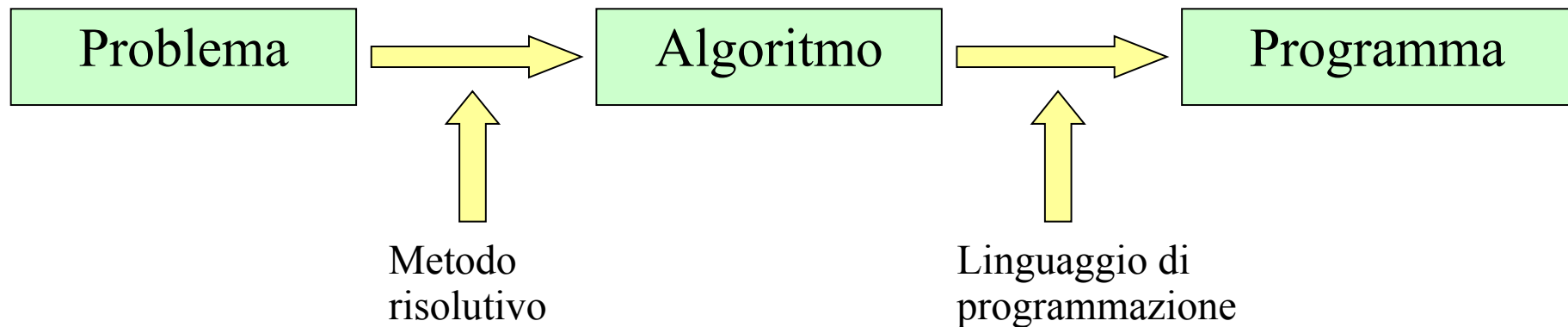
Programma

- Un **programma è un testo** scritto in accordo alla **sintassi** e alla **semantica** di un linguaggio di programmazione
- Un **programma** è la *formulazione testuale*, in un certo linguaggio di programmazione, di un algoritmo che risolve un dato **problema**
 - *È composto da un numero finito di istruzioni*
 - *Ogni istruzione descrive una operazione*

Algoritmo e programma

Passi per la risoluzione di un problema:

- Individuazione di un procedimento risolutivo
- Scomposizione del procedimento in un insieme ordinato di azioni → **ALGORITMO**
- Rappresentazione dei dati e dell'algoritmo attraverso un formalismo comprensibile dal calcolatore → **PROGRAMMA**



Linguaggi

- Diversi tipi di linguaggi:
 - Imperativi
 - Funzionali
 - Dichiarativi
 - A oggetti

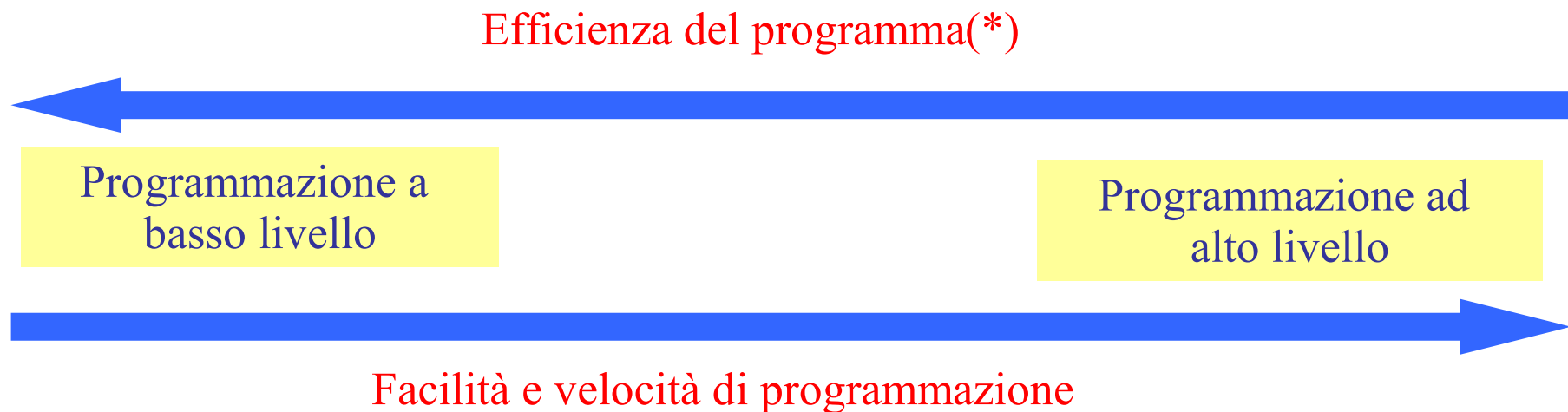
- Tutti basati sulla traduzione nell'unico linguaggio eseguibile dal calcolatore:
il Linguaggio Macchina

Linguaggio Macchina

- Insieme di istruzioni eseguibili dalla CPU
- Dipende dalla CPU:
 - cablata al suo interno, ogni istruzione genera una sequenza di segnali di controllo
- Complesso da utilizzare
 - ogni istruzione esegue un'operazione semplicissima come somme registri spostamenti da memoria a registri

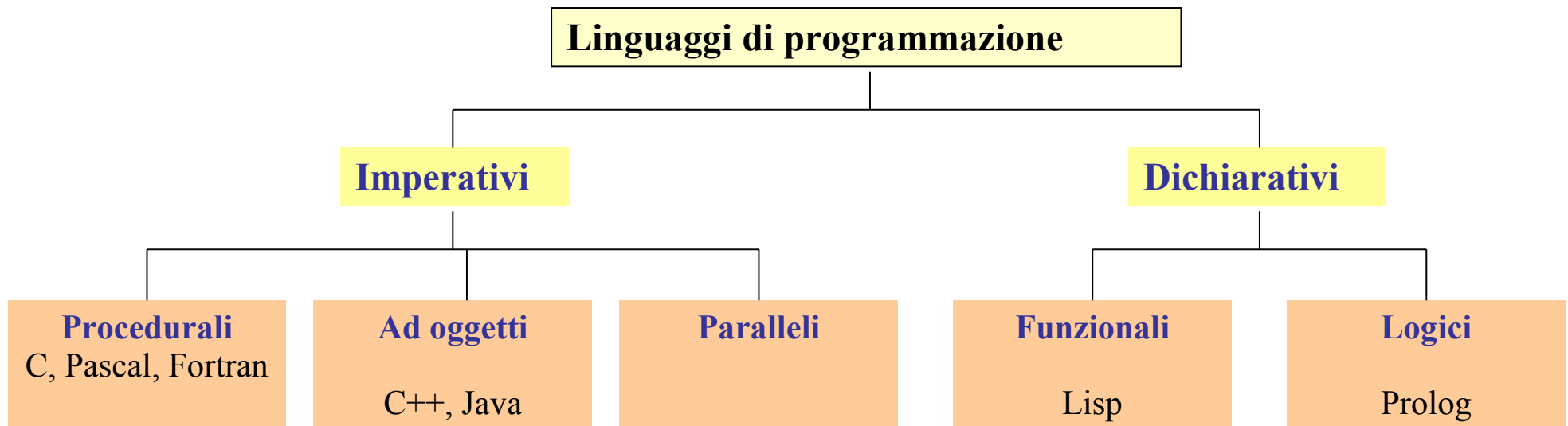
Linguaggi

- La programmazione a *basso livello* è più ardua e meno intuitiva, ma consente di sviluppare programmi efficienti.
- Ad *alto livello* la programmazione è più “naturale” e rapida, ma è possibile che non consenta di produrre software efficiente.
- (*)NB: Con l'aumentare delle prestazioni dei calcolatori e dei compilatori la programmazione a basso livello non risulta nemmeno così “efficiente” se non in casi particolari (es. Digital Signal Processors etc)



Tipologie di linguaggi

- Possiamo aggregare i numerosi linguaggi di programmazione esistenti (ad alto livello) sulla base del **modello astratto di programmazione** che sottintendono e che è necessario adottare per utilizzarli.



Traduzione dei programmi

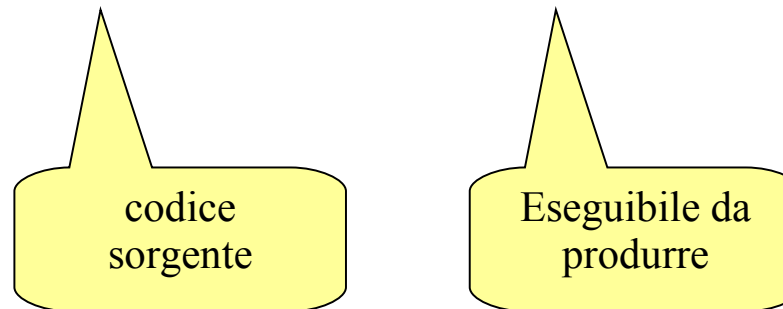
- Affinchè un programma scritto in un qualsiasi linguaggio di programmazione sia comprensibile (e quindi eseguibile) da un calcolatore, occorre **tradurlo** dal linguaggio originario al linguaggio della macchina
- Questa operazione viene normalmente svolta da speciali programmi detti **traduttori**

Compilatori e Interpreti

- E' necessario tradurre Il **testo** di un programma scritto in un particolare linguaggio di programmazione (**sorgente**) nella corrispondente **rappresentazione in linguaggio macchina** (programma **eseguibile**)
 - i **compilatori**: traducono l'intero programma (senza eseguirlo) e producono in uscita il programma convertito in linguaggio macchina
 - Gli **interpreti**: traducono ed eseguono immediatamente ogni singola istruzione del programma sorgente

Costruzione “*manuale*”

- La costruzione dell'eseguibile può essere effettuata “*a mano*”, attivando *compilatore* dalla *linea di comando* del sistema operativo
-
- */home/dpatti/\$ gcc pippo.c -o pippo*
(genera l'eseguibile *pippo*)



Ambienti Integrati

- Gli *ambienti di lavoro integrati* (**IDE – Integrated Development Environments**) automatizzano la procedura:
 - compilano i file sorgente (*se e quando necessario*)
 - invocano il linker per costruire l'eseguibile