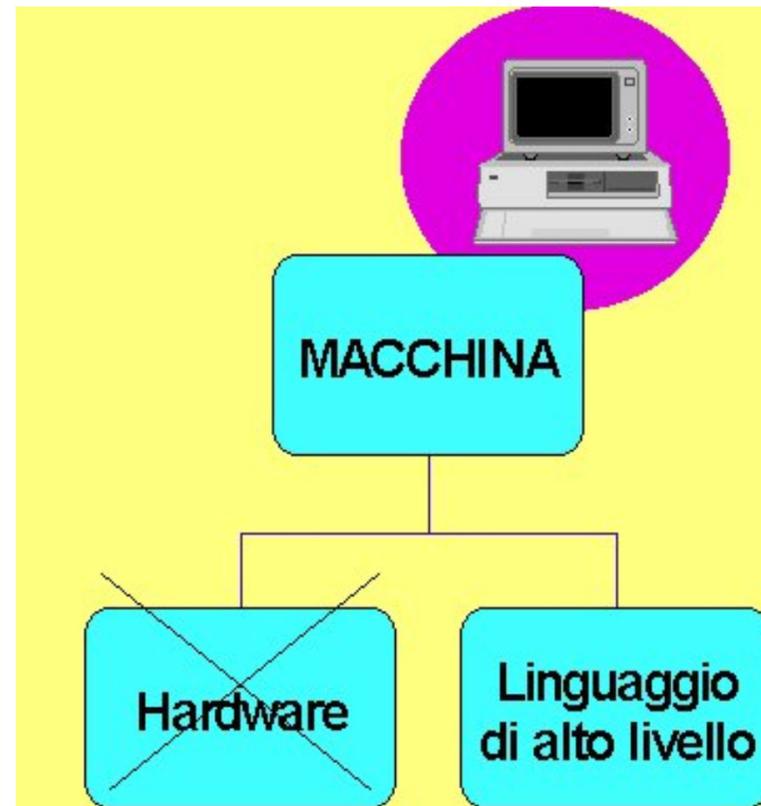
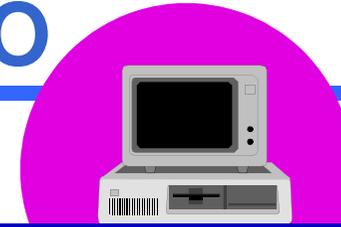


LINGUAGGI DI ALTO LIVELLO

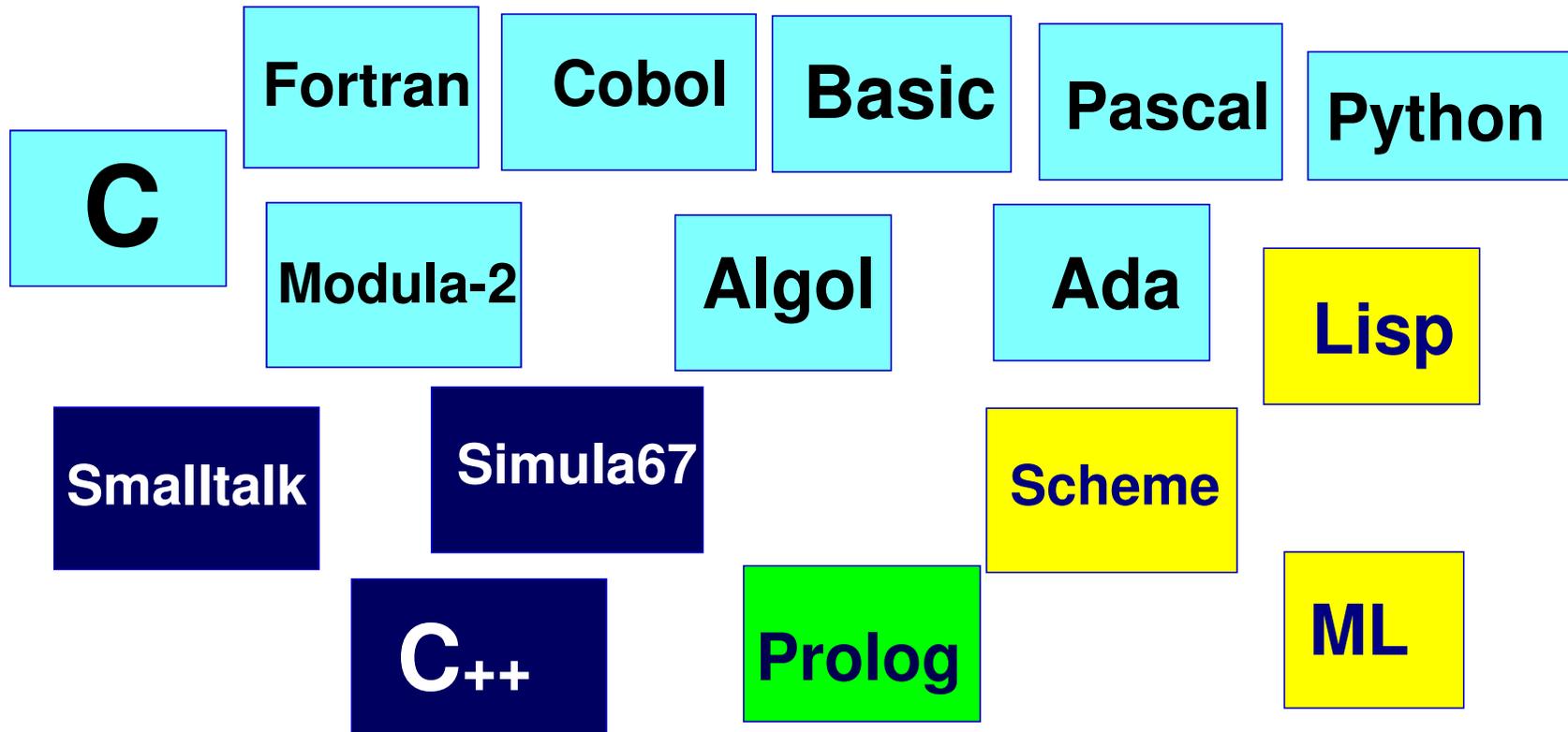
Si basano su una *macchina virtuale* le cui “mosse” non sono quelle della macchina hardware



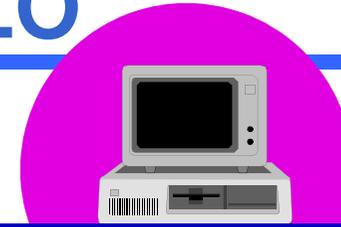
LINGUAGGI DI ALTO LIVELLO



Barriera di astrazione

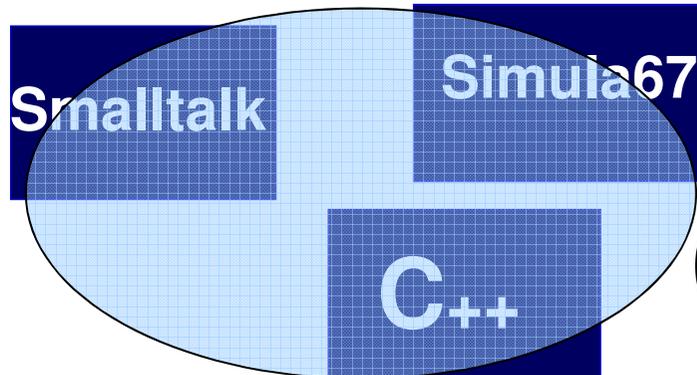
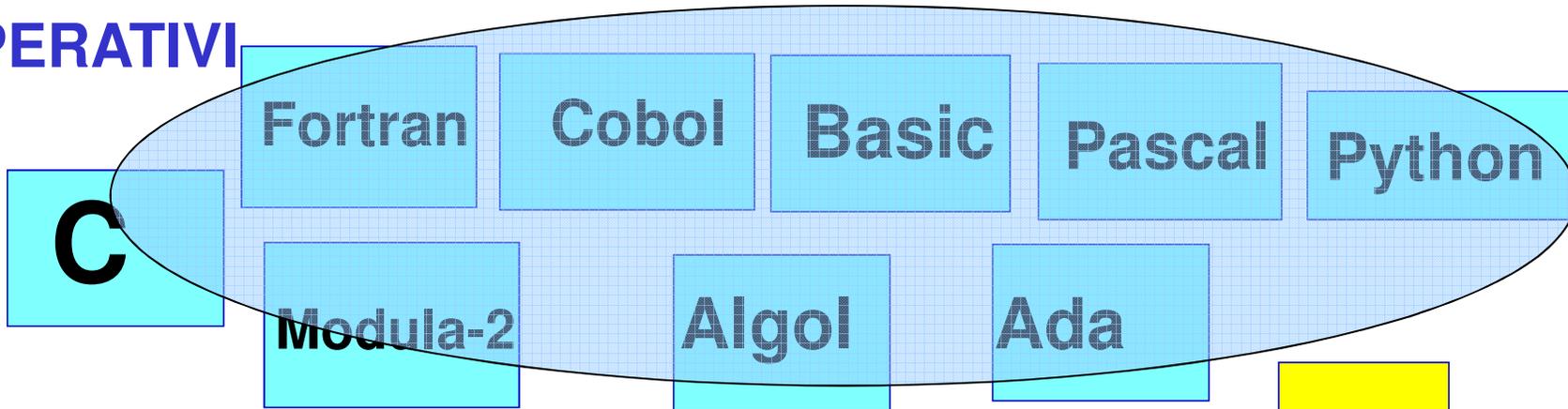


LINGUAGGI DI ALTO LIVELLO

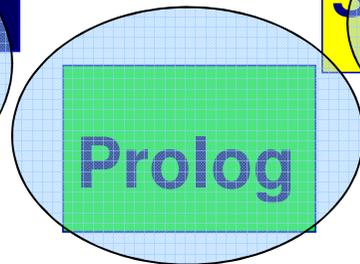


Barriera di astrazione

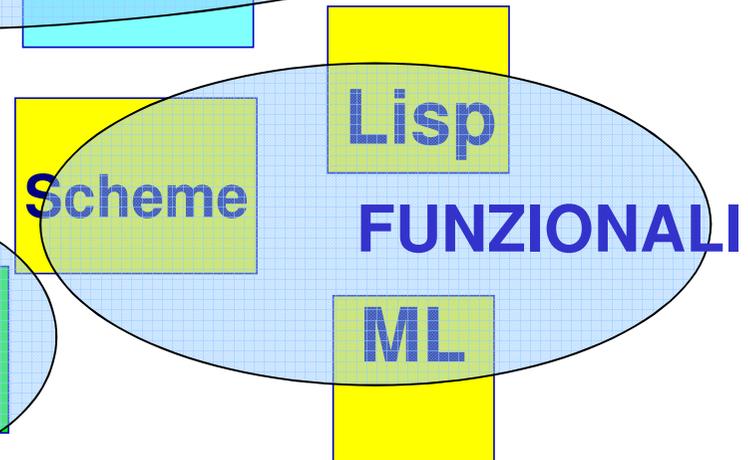
IMPERATIVI



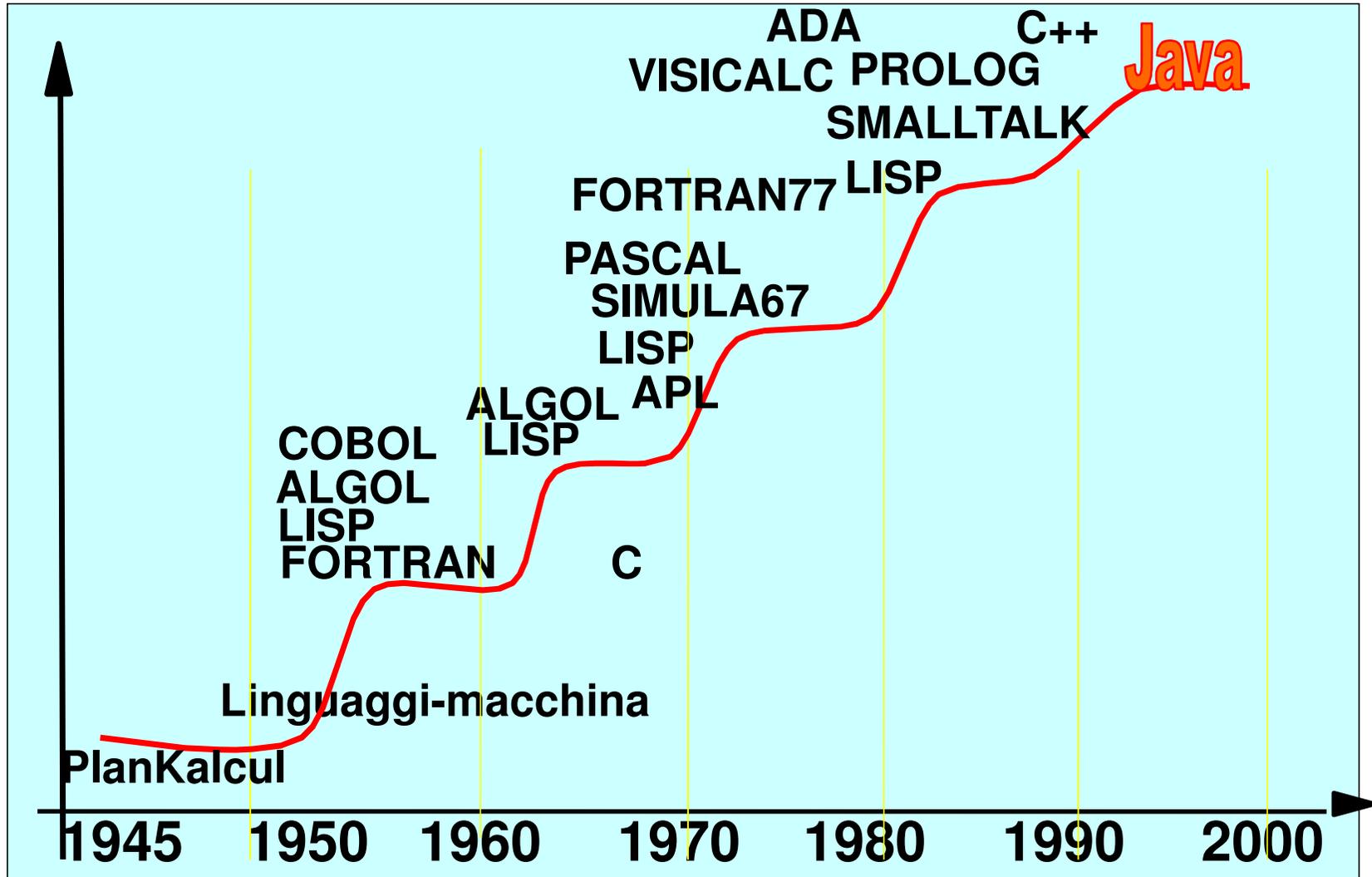
A OGGETTI



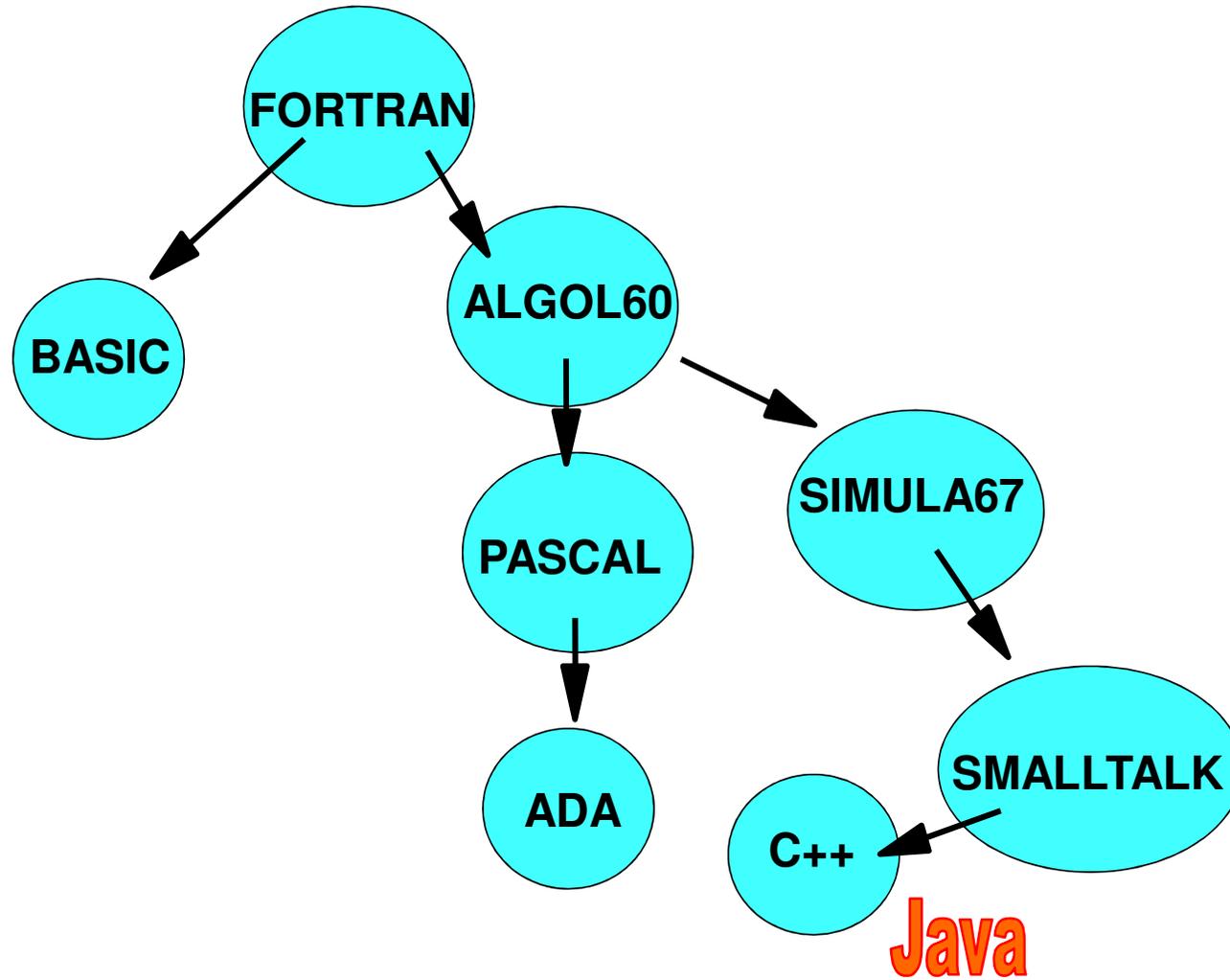
DICHIARATIVI



EVOLUZIONE DEI LINGUAGGI



EVOLUZIONE DEI LINGUAGGI



CHE COS'È UN LINGUAGGIO?

*“Un linguaggio è un **insieme di parole e di metodi di combinazione delle parole** usate e comprese da una comunità di persone”*

- È una definizione **poco precisa**:
 - *non evita le ambiguità* dei linguaggi naturali
 - non si presta a descrivere processi computazionali *meccanizzabili*
 - *non aiuta a stabilire proprietà*

LA NOZIONE DI LINGUAGGIO

- Occorre una **nozione di linguaggio *più precisa***
- **Linguaggio come *sistema matematico***
che consenta di rispondere a domande come:
 - quali sono le ***frasi lecite***?
 - si può stabilire se una frase ***appartiene al linguaggio***?
 - come si stabilisce il **significato** di una frase?
 - **quali elementi linguistici primitivi**?

LINGUAGGIO & PROGRAMMA

- Dato un algoritmo, **un programma** è la sua **descrizione *in un particolare linguaggio*** di programmazione
- **Un linguaggio di programmazione** è una **notazione formale** che può essere usata per descrivere algoritmi. Due aspetti del linguaggio:
 - SINTASSI
 - SEMANTICA

SINTASSI & SEMANTICA

- **Sintassi**: l'insieme di regole formali per la **scrittura di programmi** in un linguaggio, che dettano le *modalità per costruire frasi corrette* nel linguaggio stesso
- **Semantica**: l'insieme dei **significati** da attribuire alle frasi (sintatticamente corrette) costruite nel linguaggio

NB: una frase può essere **sintatticamente corretta** e tuttavia *non avere significato!*

SINTASSI

Le regole sintattiche sono espresse attraverso *notazioni formali*:

- ◆ **BNF (Backus-Naur Form)**
- ◆ **EBNF (Extended BNF)**
- ◆ **diagrammi sintattici**

SINTASSI EBNF: ESEMPIO

Sintassi di un *numero naturale*

`<naturale> ::=`

`0 | <cifra-non-nulla>{<cifra>}`

`<cifra-non-nulla> ::=`

`1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`

`<cifra> ::=`

`0 | <cifra-non-nulla>`

ESEMPIO DI SINTASSI: numeri naturali

`<naturale> ::=`

`0 | <cifra-non-nulla>{<cifra>}`

Intuitivamente significa che un numero naturale si può riscrivere come 0 oppure (|) come una cifra non nulla seguita da zero o più ({}) cifre

`<cifra-non-nulla> ::=`

`1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9`

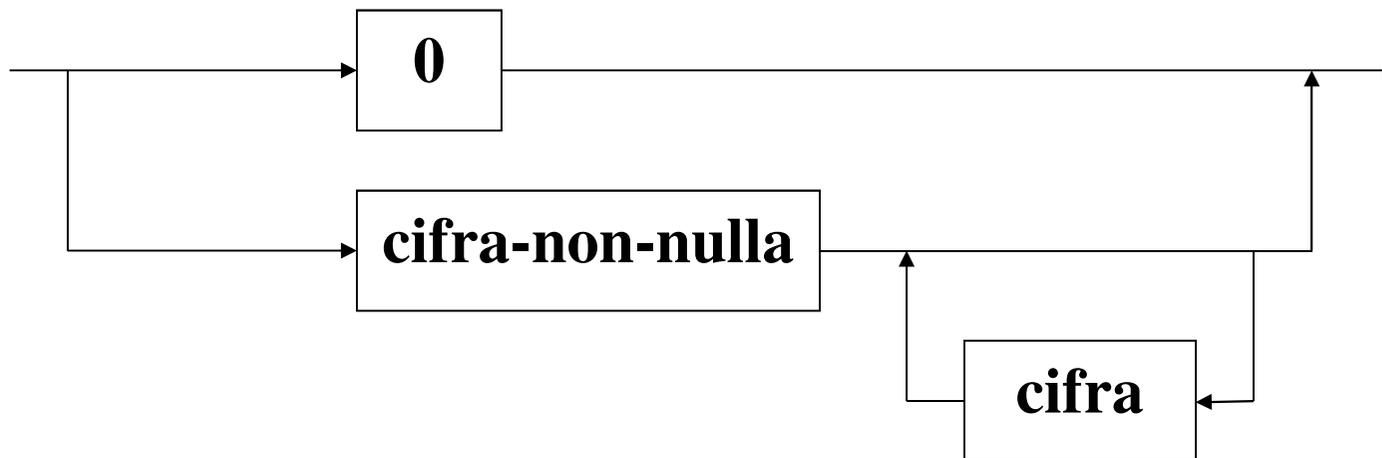
una cifra non nulla si può riscrivere come 1 oppure 2 oppure 3...

`<cifra> ::= 0 | <cifra-non-nulla>`

una cifra si può riscrivere come 0 oppure come una cifra non nulla (definita precedentemente)

DIAGRAMMI SINTATTICI: ESEMPIO

Sintassi di un *numero naturale*



SEMANTICA

La semantica è esprimibile:

- ◆ ***a parole*** (poco precisa e ambigua)
- ◆ mediante **azioni**
 - ***semantica operativa***
- ◆ mediante **funzioni matematiche**
 - ***semantica denotazionale***
- ◆ mediante **formule logiche**
 - ***semantica assiomatica***

DEFINIZIONE DI LINGUAGGIO

- Un linguaggio è un **insieme di frasi**
- Una frase è una **sequenza di simboli** appartenenti a un certo alfabeto

Proprietà desiderabili:

- Un linguaggio deve essere **effettivamente generabile**
- Un linguaggio di programmazione deve essere **decidibile**

ALCUNE DEFINIZIONI

Alfabeto V (o *vocabolario* o *lessico*)

- È *l'insieme dei simboli* con cui si costruiscono le frasi

Universo linguistico V^* di un alfabeto V

- È *l'insieme di tutte le frasi* (*sequenze finite di lunghezza arbitraria*) di elementi di V

Linguaggio L su un alfabeto V

- È *un sottoinsieme di V^**

ESEMPIO

$V = \{ \text{if, else, ==, A, 0, =, +, 1, 2, (,) } \}$

Allora:

$V^* = \{$
 if (A == 0) A = A + 2,
 if else A,
 do =A,
 ...
 }

ESEMPIO

$V = \{ \text{if, else, ==, A, 0, =, +, 1, 2, (,)} \}$

Allora:

$V^* = \{$
 if (A == 0) A = A + 2,
 if else A,
 do =A,
 ...
}



Non tutte queste
frasi faranno
parte del
linguaggio!

LINGUAGGI E GRAMMATICHE

- **Come specificare il sottoinsieme di V^* che definisce il linguaggio?**
- **Specificando il modo *formale e preciso* la sintassi delle frasi del linguaggio**

TRAMITE

una **grammatica formale**:
una **notazione matematica** che
consente di esprimere ***in modo***
rigoroso **la sintassi di un linguaggio**

GRAMMATICA FORMALE

Una *quadrupla* $\langle VT, VN, P, S \rangle$ dove:

- **VT** è un *insieme finito di simboli terminali*
- **VN** è un *insieme finito di simboli non terminali*
- **P** è un *insieme finito di produzioni, ossia di regole di riscrittura*
- **S** è un particolare *simbolo non-terminale* detto *simbolo iniziale* o *scopo* della grammatica

ESEMPIO COMPLESSIVO

G = $\langle \text{VT, VN, P, S} \rangle$

dove:

VT = { il, gatto, topo, sasso, mangia, beve }

VN = { <frase>, <soggetto>, <verbo>,
<compl-ogg>, <articolo>, <nome> }

S = <frase>

P = ...

ESEMPIO COMPLESSIVO

P = {
 <frase> ::= <soggetto> <verbo> <compl-ogg>
 <soggetto> ::= <articolo><nome>
 <articolo> ::= il
 <nome> ::= gatto | topo | sasso
 <verbo> ::= mangia | beve
 <compl-ogg> ::= <articolo> <nome>
}

ASTRAZIONE

Esistono linguaggi a *vari livelli di astrazione*

Linguaggio Macchina:

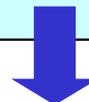
- implica la conoscenza dei metodi utilizzati per la rappresentazione delle informazioni

Linguaggio Macchina e Assembler:

- implica la conoscenza dettagliata delle *caratteristiche della macchina* (registri, dimensioni dati, set di istruzioni)
- semplici algoritmi implicano la specifica di molte istruzioni

Linguaggi di Alto Livello:

- Il programmatore può astrarre dai dettagli legati all'architettura ed esprimere i propri algoritmi in modo simbolico



Sono indipendenti dalla macchina hardware sottostante
ASTRAZIONE

ASTRAZIONE

- **Linguaggio Macchina:**

```
0100 0000 0000 1000
0100 0000 0000 1001
0000 0000 0000 1000
```

Difficile leggere e capire un programma scritto in forma binaria

- **Linguaggio Assembler:**

```
... LOADA H
   LOADB Z
   ADD
   ...
```

Le istruzioni corrispondono univocamente a quelle macchina, ma vengono espresse tramite nomi simbolici (parole chiave)

- **Linguaggi di Alto Livello:**

```
main()
{ int A;
  scanf("%d", &A);
  if (A==0) {...}
...}
```

Sono indipendenti dalla macchina

ESECUZIONE

Per eseguire sulla macchina hardware un programma scritto in un **linguaggio di alto livello** è necessario tradurre il programma in ***sequenze di istruzioni di basso livello***, direttamente eseguite dal processore, attraverso:

- **interpretazione** (ad es. BASIC)
- **compilazione** (ad es. C, FORTRAN, Pascal)

COME SVILUPPARE UN PROGRAMMA

Qualunque sia il linguaggio di programmazione scelto occorre:

- Scrivere il **testo del programma** e memorizzarlo su supporti di memoria permanenti (*fase di editing*)

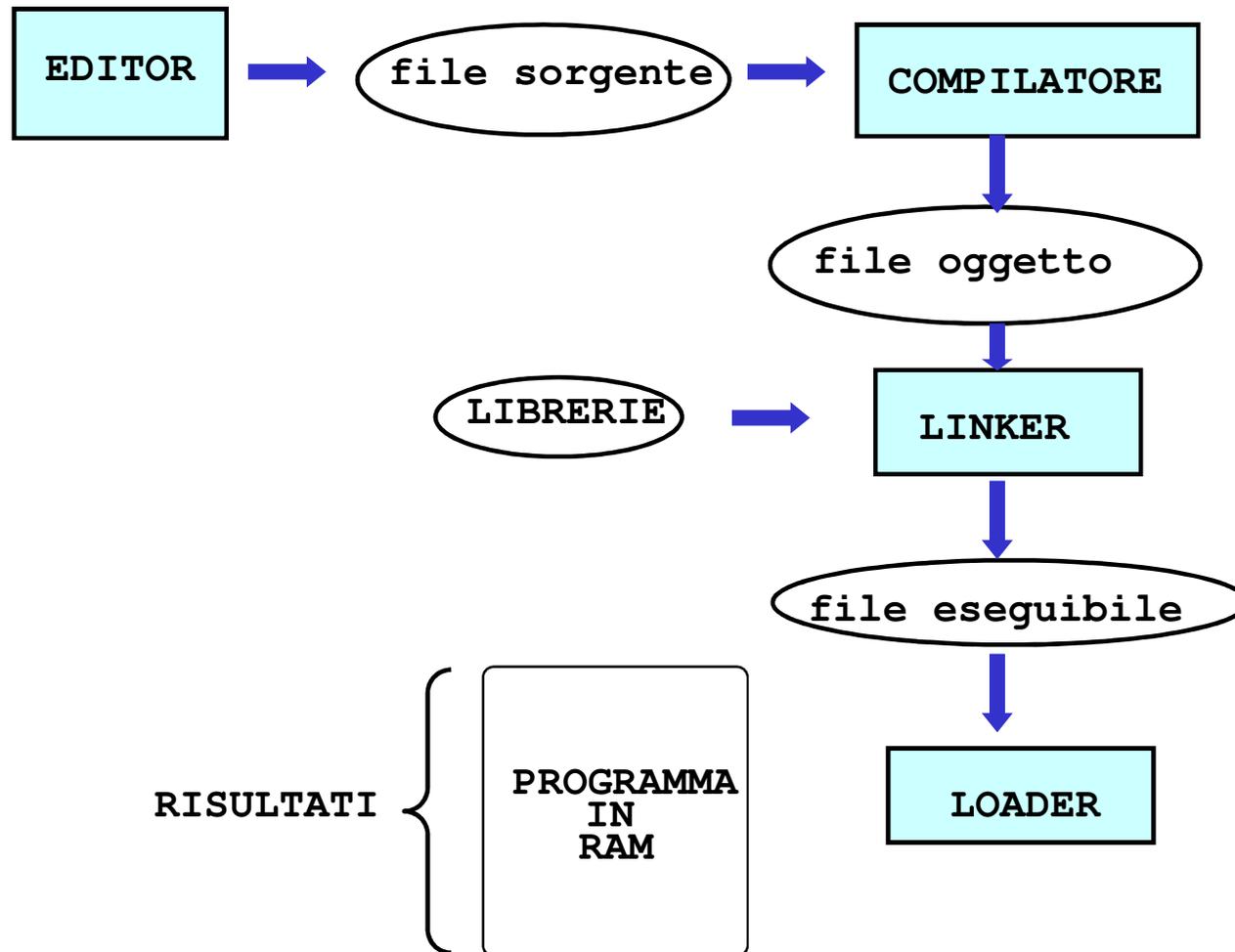
□ Se il linguaggio è compilato:

- ***Compilare il programma***, ossia utilizzare il compilatore che effettua una traduzione automatica del programma scritto in un linguaggio qualunque in un programma equivalente scritto in **linguaggio macchina**
- Eseguire il ***programma tradotto***

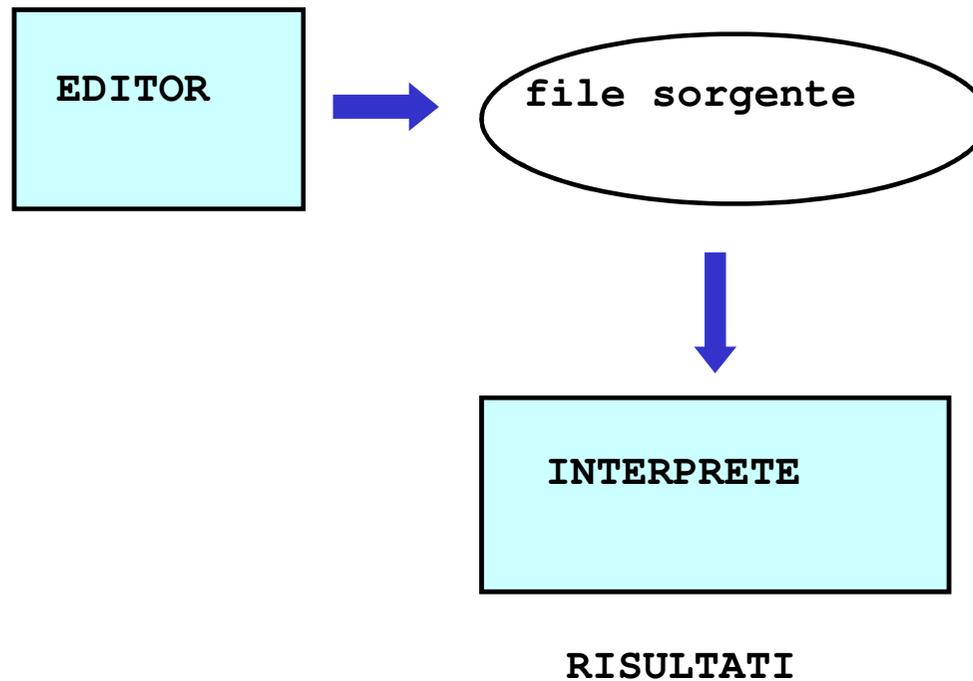
□ Se il linguaggio è interpretato:

- ***Usare l'interprete*** per eseguire il programma

APPROCCIO COMPILATO: SCHEMA



APPROCCIO INTERPRETATO: SCHEMA



COMPILATORI: MODELLO

La costruzione di un compilatore per un particolare linguaggio di programmazione è complessa

- La complessità dipende dal linguaggio sorgente

Compilatore: traduce il programma sorgente in programma oggetto

Due compiti:

- **ANALISI** del programma sorgente
- **SINTESI** del programma oggetto

COMPILATORI: MODELLO

