

## Informatica Teorica

### Presentazione del corso

### Obiettivi e motivazioni

Perché l'informatica "teorica"?

La teoria stimolata dalla pratica:

generalità, rigore, "controllo"

- Comprendere a fondo e in maniera critica i principi dell'informatica (riesame approfondito di concetti elementari)
- Costruire basi solide per comprendere e dominare l'innovazione futura (esempio: multimedialità e modellazione della computazione concorrente)

- Gettare un ponte tra corsi applicativi di base e avanzati (ingegneria del SW (1-2), calcolatori-architetture HW, sistemi distribuiti...)
  - Informatica teorica nel contesto del “percorso formativo” del corso di studi
- Applicazione “diretta” a casi pratici: prosiegua attraverso il corso di metodi formali e tesi/ne
  - ... disponibilità di temi per prova finale del primo livello

### Il programma

- La modellizzazione informatica  
(Come descrivo un problema e la sua soluzione):  
non tanto singoli modelli specialistici  
piuttosto creare la capacità di capire e “inventare”  
modelli vecchi e nuovi
- La teoria della computazione:  
che cosa so/posso fare con lo strumento informatico  
(quali problemi so risolvere)?

## Il programma (continua)

- La teoria della complessità:  
quanto costa risolvere un problema con lo strumento informatico?
- Gli sviluppi nel(i) corso(i) a valle (II livello)  
(corso integrato/congiunto con il master Polimi/UIC):  
I metodi formali nella progettazione informatica

## L'organizzazione

- Prerequisiti:
  - Le basi essenziali di Informatica (Informatica 1,[2,3], Ing. del SW)
  - Elementi di matematica non continua ([Algebra e Logica])
- Lezioni e esercitazioni (stile classico)
  - Interazione auspicata vivamente:
    - Intervento diretto a lezione
    - Ricevimento
    - Email

### L'organizzazione (continua)

- Esame basato sulla *capacità di applicare, non di ripetere*:  
principalmente scritto  
possibilità di consultare testi  
non ripetitivo; stimolante (?)  
punteggio max > 30
- Prove in itinere standard

### L'organizzazione (continua)

- Il materiale didattico:
  - Testo ufficiale (in italiano [UTET] e in inglese [Wiley]):  
Ghezzi/Mandrioli: Informatica teorica
  - Eserciziario (Mandrioli, Lavazza, Morzenti, San Pietro, Spoletini, terza edizione, Esculapio)
  - Sito del corso  
(<http://home.dei.polimi.it/rossi/IT.html>)
  - Temi d'esame risolti  
(<ftp://ftp.elet.polimi.it/outgoing/Matteo.Giovanni.Rossi/Didattica/IT/>)
  - Lucidi e (non!) appunti  
(Lucidi ≠ bigino!)  
(<ftp://ftp.elet.polimi.it/outgoing/Matteo.Giovanni.Rossi/Didattica/IT/>)

## I modelli dell'informatica

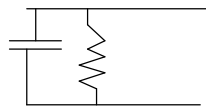
- Non (sol)tanto discreti rispetto a continui  
(bit e byte rispetto a numeri reali ed equazioni varie)
- Quanto:

- Generali:  
il sistema informatico nel contesto (spesso) di un sistema più ampio: impianto, organizzazione, sistema “embedded”, ...
- Flessibili:  
spesso non esiste il “modello già pronto”:  
occorre saper adattare modelli esistenti ad esigenze specifiche e impreviste
- esistono molti (troppi) modelli specialistici:  
occorre saper studiare/inventare nuovi modelli



- Occorre (maggiore) attitudine dinamica e critica:
  - confronto modello-realtà
  - analisi e sintesi del modello/progetto
- rispetto a:

$$\dot{x} = Ax + Bu$$



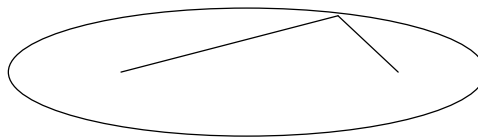
$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = g(x, y, z)$$

- Spesso la vera difficoltà di un problema sta nel .... formularlo!
- Che significa:
  - “automatizzare una procedura d’ufficio”
  - “evitare incidenti ferroviari/aerei/...”
  - ...?

- Modelli operazionali  
(macchine astratte, sistemi dinamici, ...)  
basati sul concetto di stato e di meccanismi  
(operazioni) per la sua evoluzione
- Modelli descrittivi  
tesi a formulare le proprietà desiderate o temute del  
sistema piuttosto del suo funzionamento

### Esempi

- Modello operativo dell'ellisse:



- Modello descrittivo dell'ellisse:

$$a.x^2 + b.y^2 = c$$

- Descrizione operativa dell'ordinamento:
  - Calcola il minimo e mettilo al primo posto;
  - Calcola il minimo degli elementi rimasti e mettilo al secondo posto;
  - ...
- Descrizione non-operativa dell'ordinamento:
  - Individua una permutazione della sequenza data tale che  $\forall i, a[i] \leq a[i+1]$

- In realtà le differenze tra modellizzazione operativa e modellizzazione descrittiva non sono così nette: più che altro si tratta di un utile riferimento nel classificare un tipo di modello



## Un primo, fondamentale, “meta” modello: il *linguaggio*

- Italiano, francese, inglese, ...
- C, Pascal, Ada, ...  
ma anche:
- Grafica
- Musica
- Multimedialità, ...

## Gli elementi di un linguaggio

- Alfabeto o vocabolario  
(sinonimi, matematicamente parlando):  
Insieme finito di simboli base
  - {a,b,c, ...z}
  - {0,1}
  - {Do, Re, Mi, ...}
  - {abate, abbaino, ..., zuzzurellone}
  - ASCII
  - ...

- Stringa (su un alfabeto  $A$ ):  
sequenza ordinata e finita di elementi di  $A$ , anche con ripetizioni  
– a, b, aa, alfa, giovanni, alla, nel mezzo del cammin, ...
- Lunghezza di una stringa:  
 $|a| = 1$ ,  $|ab| = 2$
- La stringa nulla  $\epsilon$ :  $|\epsilon| = 0$
- $A^*$  = insieme di tutte le stringhe, inclusa  $\epsilon$ , su  $A$ .  
–  $A = \{0,1\}$ ,  $A^* = \{\epsilon, 0, 1, 00, 01, 10, \dots\}$

- Operazione su stringhe:  
concatenazione:  
 $x \cdot y$   
 $x = \text{abb}$ ,  $y = \text{baba}$ ,  $x \cdot y = \text{abbbaba}$   
 $x = \text{Quel ramo}$ ,  $y = \text{del lago di Como}$ ,  
 $x \cdot y = \text{Quel ramo del lago di Como}$   
“.” : associativa, noncommutativa
- $A^*$ :  
*monoide libero* costruito su  $A$  mediante “.”
- $\epsilon$ : unità rispetto a “.”

## Linguaggio

- L sottoinsieme di  $A^*$ : ( $L \subseteq A^*$ )
  - Italiano, C, Pascal, ... ma anche:
  - sequenze di 0 e 1 con numero pari di 1
  - l'insieme degli spartiti in fa minore
  - le matrici quadrate il cui determinante è 0
  - ...
- Concetto estremamente ampio, in un certo senso *universale*

## Operazioni su linguaggi

- Operazioni insiemistiche:  
 $\cup, \cap, L_1 - L_2, \neg L = A^* - L, (\bar{L} = \neg L)$
- Concatenazione (tra linguaggi):  
 $L_1 \cdot L_2 = \{x.y \mid x \in L_1, y \in L_2\}$ 
  - $L_1 = \{0,1\}^*, L_2 = \{a,b\}^*$  ;  
 $L_1 \cdot L_2 = \{\epsilon, 0,1, 0a, 11b, abb, 10ba, \dots \text{Non } ab1!\}$

- $L^0 = \{\epsilon\}$ ,  $L^i = L^{i-1} \cdot L$
- $L^* = \bigcup_{n=0}^{\infty} L^n$
- NB:  $\{\epsilon\} \neq \emptyset$  !  
 $\{\epsilon\} \cdot L = L$ ;  
 $\emptyset \cdot L = \emptyset$
- $+ = \text{"*-0"}: L^+ = \bigcup_{n=1}^{\infty} L^n$

### Alcuni risvolti “pratici”

- $L_1$  : insieme dei documenti “Word/Mac”
- $L_2$  : insieme dei documenti “Word/PC”
- $L_1 \cap L_2$  : insieme dei documenti Word “compatibili Mac-PC”  
 (=  $\emptyset$ ??)
- Composizione di un messaggio su rete:
  - $x \cdot y \cdot z$ :
  - $x$  = testata (indirizzo, ...)
  - $y$  = testo
  - $z$  = “chiusura”

- Il linguaggio: strumento di espressione ...
- di un *problema*
- $x \in L$ ?
  - Un messaggio è corretto?
  - Un programma è corretto?
  - $y = x^2$ ?
  - $z = \text{Det}(A)$ ?
  - Il sonoro di un film è ben sincronizzato con il video?

- $y = \tau(x)$ 
  - $\tau$  : traduzione: funzione da  $L_1$  a  $L_2$ 
    - $\tau_1$  : raddoppio degli “1” ( $1 \mapsto 11$ ):  
 $\tau_1(00101110) = 0011011110, \dots$
    - $\tau_2$  : scambio a con b ( $a \leftrightarrow b$ ):  
 $\tau_2(abbbaa) = baaabb, \dots$
    - ma anche:
    - compressione di files
    - protocolli autocorrettori
    - compilazione da linguaggi di alto livello in linguaggi oggetto
    - traduzione italiano  $\mapsto$  inglese

## Conclusione

- Il concetto di linguaggio e le operazioni base ad esso associate forniscono un mezzo espressivo estremamente generale per descrivere sistemi di ogni tipo, loro proprietà e problemi ad essi connessi:
- Calcolare il determinante di una matrice;
- Stabilire se un ponte crollerà sotto un certo carico;
- ....
- In fin dei conti nel calcolatore ogni informazione è una stringa di bit ...