

FONDAMENTI DELL'INFORMATICA

a.a. 2005/2006

Prof. Donato Malerba
Dipartimento di Informatica
Università degli Studi Bari

tel.: 080 5443269
email: malerba@di.uniba.it
<http://www.di.uniba.it/~malerba/>



**"All science is either physics
or stamp collection"**
(Ernest Rutherford)

Il famoso fisico metteva a contrasto una semplice raccolta di conoscenze con delle teorie formulate matematicamente.

*L'informatica è parente stretta della filatelia,
oppure può assurgere al rango di scienza?*

Cosa è una disciplina?

Lat. *Disciplīna(m)*, originariamente
'insegnamento, educazione', da *dīscere*
'imparare'

- Materia di studio – Ramo del sapere
(da Zanichelli, Vocabolario della lingua italiana, 1994)
- Quando un insieme di conoscenze
diventa 'disciplinato'?
- Quando queste sono generate mediante
paradigmi

Cos'è un paradigma?

Paradigma: dal greco παράδειγμα 'modello, esempio',
da παρά 'presso, accanto' (indica similarità) +
δεικνύναι 'mostrare, indicare'

- Sebbene il termine paradigma sia frequentemente
assimilato al concetto di *modello*, esso è
caratterizzato dal fatto che può essere utilizzato per
produrre qualcosa.

Esempio: paradigma dei verbi.

- Nella scienza un paradigma è un modello utilizzato
per produrre nuova conoscenza.
- Esso si connota come una teoria aperta, ampia,
verificabile, senza precedenti, che getta le
fondamenta di una disciplina.

Thomas Kuhn (filosofo, 1922-1996)

Il termine paradigma fu introdotto nella filosofia della scienza da T. Kuhn per indicare quell'insieme di regole, modelli di problemi e criteri risolutivi che caratterizza una comunità scientifica (in termini di disciplina, era storica, luogo geografico, ecc.).

T. Kuhn

*The Structure of Scientific
Revolutions: A Paradigm Shift in
the History of Science*

1962



Thomas Kuhn (filosofo, 1922-1996)

I paradigmi aiutano le comunità scientifiche a delimitare la loro disciplina in quanto aiutano lo scienziato a

- creare percorsi di ricerca
- formulare domande
- selezionare metodi con i quali esaminare le domande
- definire aree di rilevanza
- [stabilire/creare significati?]

"In assenza di un paradigma o di qualche candidato a paradigma, tutti i fatti che potrebbero riferirsi allo sviluppo di una data scienza appariranno molto probabilmente ugualmente rilevanti."

"Una **rivoluzione** (scientifica) è un cambio di paradigma".

Paradigmi per la costruzione della conoscenza

- Teoria
- Astrazione
- Progettazione

Paradigma della teoria

Ha le sue radici nella **matematica** ed è costituito da quattro passi che vengono seguiti nello sviluppo di una teoria valida e coerente:

1. *Caratterizzare gli oggetti dello studio* (definizioni)
2. *Ipotizzare possibili relazioni fra loro* (teoremi)
3. *Determinare se le relazioni sono vere* (dimostrazioni)
4. *Interpretare i risultati*

Paradigma dell'astrazione (o sperimentazione)

Ha le sue radici nel **metodo scientifico sperimentale** (tipico della fisica) ed è costituito da quattro passi che vengono seguiti nell'investigazione di un fenomeno:

1. *Formare un'ipotesi*
2. *Costruire un modello e fare una previsione*
3. *Disegnare un esperimento e raccogliere i dati*
4. *Analizzare i risultati*

Paradigma della progettazione

Ha le sue radici nell'**ingegneria** ed è costituito da quattro passi che vengono seguiti nella costruzione di un sistema per risolvere un dato problema:

1. *Formulare i requisiti di stato*
2. *Specificare lo stato*
3. *Progettare e implementare il sistema*
4. *Test del sistema*

Informatica come disciplina

Le conoscenze informatiche sono in verità generate da **un mix di questi tre paradigmi**.

In questo l'informatica è atipica rispetto ad altre discipline scientifiche dove uno dei tre paradigmi è prevalente.

- Il paradigma della teoria ha contribuito alla maturazione delle conoscenze su computabilità, complessità computazionale, algoritmi e strutture dati, automi, linguaggi formali.

Informatica come disciplina

- Il paradigma dell'astrazione ha contribuito allo studio di metriche per programmi e sistemi, alle simulazioni di sistemi e di processi fisici, all'analisi delle prestazioni di sistemi, al test di protocolli, al confronto delle architetture, all'apprendimento automatico.
- Il paradigma della progettazione ha contribuito allo sviluppo di ambienti di sviluppo, simulatori, sistemi CAD, CAM, VLSI, database, sistemi operativi.

Informatica come disciplina

Nei corsi di programmazione e di algoritmi e strutture dati vi è stato insegnato a progettare delle soluzioni a partire da delle specifiche, a implementarle e a testarle. → [paradigma della progettazione](#)

Nei corsi di architettura e di sistemi operativi vi è stato insegnato a costruire il modello di un calcolatore a vari livelli di astrazione, a valutare l'efficienza di una istruzione o di un insieme di istruzioni, o di un sistema di gestione della memoria. → [paradigma dell'astrazione](#)

Nel corso di basi di dati vi è stato insegnato a progettare il modello concettuale e logico di una base di dati. → [paradigma della progettazione](#)

Informatica come disciplina

Nei corso di ingegneria del software vi è stato insegnato a modellizzare i processi di produzione del software (vedi metodologie) → [paradigma dell'astrazione](#)

ma anche a progettare il software e a integrare le componenti → [paradigma della progettazione](#)

Nel corso di linguaggi di programmazione vi è stato insegnato a studiare modelli (automi e grammatiche) in grado di tradurre, riconoscere e generare dei linguaggi → [paradigma della teoria](#)

e a modellizzare i linguaggi sulla base dei loro costrutti sintattici → [paradigma dell'astrazione](#)

Occorre completare la vostra formazione con un corso che insegni a generare sistematicamente conoscenze informatiche con il paradigma della teoria.

Obiettivi del corso:

Fornire concetti di base relativi alle proprietà fondamentali di *algoritmi* e *programmi* (proprietà sintattiche e semantiche) e di *problemi* (proprietà di calcolabilità e di complessità).

Introdurre *sistemi formali* tipici dell'informatica, per modellare dati, macchine, algoritmi e programmi, processi di calcolo.

Acquisire familiarità con sistemi formali e modelli di calcolo fondamentali e sviluppare capacità di astrazione e formalizzazione utili ad apprendere altri sistemi formali (e modelli) e a crearne di nuovi, se necessario.

Descrizione del corso:

Il corso introduce le nozioni fondamentali della teoria della calcolabilità e della complessità.

La prima parte delinea i concetti e la natura dei problemi che hanno soluzione effettiva.

La seconda parte caratterizza i problemi che sono risolvibili con risorse di calcolo limitate.

Il corso è di 6 CFU di tipo T1 = 48 ore di lezione frontale + 102 ore di studio individuale

1 CFU di tipo T1: 8 h di lezione in aula e 17 di studio individuale

Programma:

1. Concetti matematici di base (4 h)
2. Macchine di Turing e calcolabilità secondo Turing (8 h)
3. Modelli di calcolo per linguaggi imperativi e funzionali (8 h)
4. Teoria generale della calcolabilità (8 h)
5. Complessità di algoritmi sequenziali (10 h)
6. Complessità di problemi (10 h)

Testi consigliati (reperibili in biblioteca):

*Ausiello, D'Amore, Gambosi, **Linguaggi, modelli, complessità** (le lezioni si basano su questo libro).*

*Bovet, Crescenzi, **Teoria della complessità computazionale**.*

*Cormen, Leseirson, Rivest, **Introduzione agli algoritmi**.*

*Cutland, **Computability: an introduction to recursive theory**.*

*Hopcroft, Ullman, **Introduction to automata theory, and computation**.*

Orario lezioni: Martedì e Giovedì, 8.30 - 11.00.

Ricevimento: Mercoledì 9.00-11.00.

Trasparenze usate a lezione: sono a disposizione sulla home page del corso
<http://www.di.uniba.it/~malerba/courses/fdi/>
usatele come linea guida del corso ma assolutamente studiate dai testi consigliati.

Esame: la prova scritta consiste nella dimostrazione di teoremi o nella soluzione di esercizi riguardanti i temi trattati a lezione.

Alcuni esercizi svolti e le tracce delle prove scritte del precedente a.a. sono disponibili sul sito del corso.

Prove in itinere

Sono previste due prove in itinere.

1. La prima a metà aprile, verterà sulla teoria della computabilità.
2. La seconda a fine corso verterà sulla teoria della complessità.

Le prove in itinere verranno valutate e permetteranno di essere esonerati dalla prova scritta.

Prerequisiti

Conoscenze di matematica su:

- **principio di induzione**
- **Insiemi, relazioni e funzioni (dominio, codominio, composizione di funzioni, funzioni monotone, punto fisso)**
- **Strutture algebriche elementari (semigrupperi, monoidi (monoide libero), gruppi)**
- **Infiniti e infinitesimi, limiti di rapporti di polinomi, notazione asintotica $O(n)$**
- **Matrici e determinanti**
- **Funzioni elementari (potenza, radice, logaritmo, esponenziale), funzione proiezione,**
- **Serie numeriche e somme di serie convergenti**
- **Elementi di logica matematica (connettivi logici, quantificazione universale ed esistenziale, variabili libere)**
- **Definizioni implicite di valori (mediante equazioni)**
- **Funzionali**

Prerequisiti

Conoscenze di informatica su:

- **Algoritmi fondamentali**
- **Architettura degli elaboratori**
- **Linguaggi di programmazione**

Prerequisiti

Conoscenza dell'alfabeto greco:

Maiuscole	Minuscole	Trascriz. alfabeto latino	Nome greco	Nome latino
A	α	a	αλφα	alpha
B	β	b	βητα	beta
Γ	γ	g	γαμμα	gamma
Δ	δ	d	δελτα	delta
E	ε	e	ε ψιλον	epsilon
Z	ζ	z	ζητα	zeta
H	η	e	ητα	eta
Θ	θ	th	θητα	theta
I	ι	i	ιωτα	iota
K	κ	k	καππα	kappa
Λ	λ	l	λαμβδα	lambda

Maiusc.	Minusc.	Trascriz. alfabeto latino	Nome greco	Nome latino
M	μ	m	μυ	my
N	ν	n	νυ	ny
Ξ	ξ	x	ξει	xi
O	ο	o	ο μικρον	omikron
Π	π	p	πει	pi
P	ρ	r	ρω	rho
Σ	σ	s	σιγμα	sigma
T	τ	t	ταυ	tau
Υ	υ	y	υ ψιλον	ypsilon
Φ	φ	ph	φει	phi
X	χ	ch	χει	chi
Ψ	ψ	ps	ψει	psi
Ω	ω	o	ω μεγα	omega