



A.A. 2010-2011
Corso di laurea magistrale in Informatica

Contenuti di massima degli insegnamenti

Classe LM-18 (Informatica)
ex D.M. 270/2004

Insegnamenti obbligatori.....	2
Insegnamenti curriculari.....	6
Curriculum <i>Algoritmi</i>	6
Curriculum <i>Matematica discreta e Informatica Teorica</i>	7
Curriculum <i>Metodi formali per il software</i>	9
Curriculum <i>Intelligenza artificiale e multimedia</i>	10
Curriculum <i>Reti e sicurezza</i>	13



Insegnamenti obbligatori

Algoritmi e strutture dati (6CFU)

Obiettivi formativi:

Il corso si propone di introdurre metodologie di progettazione e di analisi di algoritmi utili alla risoluzione di problemi che nascono in vari campi dell'informatica, nonché presentare algoritmi e strutture dati avanzati che trovano effettivo uso nella risoluzione efficiente di problemi applicativi di rilievo.

Programma di massima

Analisi ammortizzata e analisi nel caso medio.

Algoritmi e strutture dati probabilistici.

Cenni di geometria computazionale e di ottimizzazione combinatoria.

Elenco dei possibili argomenti da trattare nel corso:

- Skip list e loro applicazione in contesti paralleli e concorrenti.
- Splay tree come strutture dati utili nell'implementazione di caches e negli algoritmi di garbage collection.
- Suffix tree e loro uso in ricerca avanzata di stringhe, biologia computazionale e compressione dati.
- Probabilistic counting e sue applicazioni in data mining.
- Nearest neighbor search e sue applicazioni in pattern recognition e image retrieval.
- Diagrammi di Voronoi e triangolazione di Delaunay.
- Min Cut e sue applicazioni per il branch-and-bound.

Compilatori (6CFU)

Obiettivi formativi

Introdurre le tecniche di traduzione automatica del codice. Formalizzare il controllo semantico del codice e i ragionamenti sulla sua struttura. Apprendere a scrivere piccoli traduttori automatici.

Programma sintetico

Struttura di un compilatore. Analisi lessicale. Analisi sintattica: analisi top-down e bottom-up. Strumenti per la generazione automatica di analizzatori lessicali (scanner) e sintattici (parser). Traduzione guidata dalla sintassi. Analisi semantica del codice: controllo dei tipi, posizione dei costrutti sintattici, determinazione del codice morto. Ambiente di esecuzione: passaggio dei parametri, tavole dei simboli, allocazione della memoria. Generazione del codice intermedio. Generazione del codice oggetto. Analisi e ottimizzazione del codice.

**Calcolabilità e complessità (6CFU)****Obiettivi formativi**

Al termine del corso gli studenti conosceranno i principali metodi e risultati delle Teorie della Calcolabilità e della Complessità e poterli applicare per individuare la complessità di problemi in diversi campi.

Linguaggi e automi

I linguaggi della gerarchia di Chomsky.

- Automi a stati finiti: concetti e teoremi fondamentali:
 - chiusure, automi a stati finiti non deterministici e problemi di decisione. Espressioni regolari e grammatiche regolari.
- Automi a pila: concetti e teoremi fondamentali:
 - chiusure, automi a pila nondeterministici e problemi di decisione. Grammatiche acontestuali. Ambiguità.

Teoria della Calcolabilità

- Macchina di Turing.
 - Macchina di Turing non deterministiche. Variante a k nastri. Modelli equivalenti. Tesi di Church. Cenni a modelli alternativi: paralleli, quantistici,... Problemi decidibili, semidecidibili e indecidibili. Riduzioni tra problemi.

Teoria della Complessità.

- Classi di complessità. P e NP .
- Riduzioni polinomiali e NP -completezza.
 - Teorema di [Cook-Levin](#) e altri problemi NP -completi.
 - Teorema di Savitch e l'equivalenza tra $NPSPACE$ e $PSPACE$.
 - Classi sublineari. $NL=coNL$



Metodi formali per il software (6CFU)

Il corso introduce gli strumenti formali più diffusi per la specifica, l'analisi e la verifica di sistemi sequenziali e concorrenti. Le metodologie proposte sono supportate da tools open source usati per lo sviluppo di esempi concreti.

Metodi di Specifica dei Requisiti di Sistemi Software

Approfondimento delle principali metodologie per la definizione e la validazione dei requisiti sia a livello di sistema che a livello del software (tipi di dato astratti, metodi algebrici e logici).

Metodi di Modellazione di Sistemi Software

Uso dei principali metodi formali per la progettazione di sistemi software. (Algebre di processo, sistemi di transizione, Timed/Hybrid Automata, Statecharts, Reti di Petri).

Metodi di Analisi e Sintesi di Sistemi Software

Presentazione delle principali metodologie per l'analisi, la verifica e la sintesi di sistemi software (static analysis, testing, model checking, sintesi automatica, trasformazione di programmi e modelli).

Casi di studio: Uso di tools per la definizione di requisiti, per modellazione della dinamica e per la verifica di sistemi software.



Sistemi distribuiti (6CFU)

In questo corso si espongono alcune delle idee fondamentali usate per costruire i moderni sistemi distribuiti. Queste idee sono state sviluppate dalla fine degli anni '70 e rappresentano oggi un corpo di risultati coerenti e di grande impatto e interesse teorico e applicativo. Questi risultati hanno influenzato e influenzano oggi in modo fondamentale l'informatica moderna. Sono usati nella progettazione dei sistemi peer-to-peer, come Kademia, nella progettazione dei protocolli e dei servizi di rete, come nei servizi di Akamai, nella progettazione delle basi di dati distribuite, fino alla scelta delle primitive hardware nei moderni microprocessori multi-core (grazie ai risultati di Herlihy, 1991, sulla gerarchia wait-free nei sistemi a memoria condivisa, le ultime linee di microprocessori Intel implementano la Compare & Swap invece della classica Test and Set).

Obiettivi:

Questo corso ha l'obiettivo di dare agli studenti una buona familiarità con le problematiche fondamentali, in termini di fondamenti, protocolli e limiti insormontabili, dei sistemi distribuiti e del loro impatto nella progettazione nei sistemi moderni.

Programma di massima:

1. Introduzione ai sistemi distribuiti
 - a. definizioni, modelli di computazione distribuita (sincroni vs asincroni, scambio di messaggi vs memoria condivisa, etc)
 - b. ordinamento degli eventi in un sistema distribuito;
 - c. Lamport clocks, vector clocks; stati globali e protocolli di snapshot;
2. Protocolli
 - a. protocolli di commit;
 - b. roll-back recovery;
 - c. leader election;
 - d. broadcast affidabile;
3. Il problema del consenso
 - a. impossibilità del consenso in presenza di nodi guasti in un sistema asincrono;
 - b. il consenso in presenza di nodi bizantini;
 - c. failure detectors;
4. Distributed shared memory
5. Sistemi P2P
 - a. Distributed hash tables; esempio: Chord;
 - b. File system distribuiti; esempi: Oceanstore, Google File System;
 - c. Kademia.



Insegnamenti curriculari

Curriculum Algoritmi

Insegnamento	Crediti	SSD	Ambito
Algoritmi avanzati	6	Inf/01	Caratt.
Algoritmi per le reti	6	Inf/01	Caratt.
Complessità	6	Inf/01	Caratt.
Ingegneria degli algoritmi	6	Inf/01	Caratt.

Algoritmi Avanzati

Algoritmi probabilistici, algoritmi paralleli, algoritmi distribuiti.

Algoritmi per le reti

In questo corso verranno trattati molti problemi che nascono dall'utilizzo di reti di comunicazione di vario tipo (cablate e senza fili, fisse e mobili); per ciascuno di essi, verrà discusso come sia possibile modellarlo tramite un problema algoritmico classico e verrà proposta una o più soluzioni. Poiché l'approccio è comunque algoritmico, si porrà l'accento sul metodo con cui sia possibile trovare la soluzione ottima o, almeno, la più efficiente possibile.

Complessità

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Ingegneria degli algoritmi

Lo scopo principale dell'area di ricerca conosciuta come "Ingegneria degli Algoritmi" (da cui il nome del corso) è di definire metodologie standard e modelli computazionali realistici per la progettazione e l'analisi di algoritmi / programmi.

Il corso si propone di affrontare questo aspetto (non coperto altrove nel nostro corso di laurea), approfondendo aspetti prestazionali legati allo sviluppo del software e coniugando il progetto e



l'analisi teorica di algoritmi e strutture dati efficienti con la loro effettiva codifica in un linguaggio di programmazione reale e la loro validazione sperimentale. Questo comporta:

- da una lato, introdurre modelli di costo teorici piu' avanzati rispetto al classico modello RAM usato nei corsi algoritmici di base, e progettare algoritmi efficienti in tali modelli. Uno scopo importante di tali modelli (e.g., memoria esterna, cache-oblivious, multi-core, data streaming) e' di catturare aspetti architetturali come la presenza di memorie gerarchiche e di piu' core;
- dall'altro lato, integrare e rafforzare l'approccio teorico tradizionale con metodologie sperimentali standard su piattaforme e dati di test reali.

Curriculum Matematica discreta e Informatica Teorica

Insegnamento	Crediti	SSD	Ambito
Complessità	6	Inf/01	Caratt.
Teoria dei grafi	6	Inf/01	Caratt.
Teoria della computazione	6	Inf/01	Caratt.
Teoria dell'Informazione	6	Inf/01	Caratt.

Complessità

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Teoria dei grafi

Questo corso offre un' introduzione generale alla teoria di grafi. Gli argomenti coperti sono: Matching, sia da un punto di vista teorico che algoritmico; Connetivita', Teorema di Menger e sistemi di cammini disgiunti; Grafi planari e le basi della teoria dei grafi topologica; Graph coloring; la teoria probabilistica dei grafi; introduzione alla teoria dei minori di grafi.

Teoria della computazione

Obiettivi nozionistici

Il corso si propone di approfondire tecniche, nozioni e risultati recenti della Teoria della Computazione, prendere in esame differenti modelli di calcolo e si studiarne i limiti, introducendo le appropriate misure di complessità e analizzando le loro relazioni. In questo modo si vuole fornire allo studente le basi per orientarsi nel campo della teoria della



computazione ed avere il background iniziale per avviarsi ad un'attività di ricerca. L'obiettivo è fornire una conoscenza che consenta di saper applicare le tecniche apprese sia in contesti applicativi che teorici.

Metodologia e obiettivi formativi

Il corso sarà "flessibile" a seconda degli interessi degli studenti, degli argomenti attuali di ricerca, di particolari applicazioni recenti e significative e dei possibili percorsi in cui il corso può essere consigliato. Da un lato può essere portato avanti in maniera ortogonale ai vari modelli di computazione, dall'altro potrebbe privilegiare la specificità di uno di essi a scapito di una conoscenza più generale degli altri. Si intende invogliare gli studenti a lavorare su problemi di frontiera. E quindi sarà in parte pensato come un corso di "laboratorio", dove privilegiare il lavoro degli studenti, soli, in gruppo o con l'ausilio del docente, su problemi concreti e dove le nozioni vengono introdotte in funzione dei problemi affrontati.

Collocazione e Prerequisiti

Laurea magistrale di un corso di laurea in Informatica o Matematica. Sono da preferirsi come prerequisito corsi durante i quali gli studenti abbiano appreso tecniche di dimostrazione, specialmente in matematica discreta e combinatoria e corsi introduttivi ai linguaggi, complessità e calcolabilità. Un corso di complessità computazionale può essere considerato sia complemento che prerequisito di un tale corso.

Contenuti

Si presentano due esempi di organizzazione del corso.

In un'organizzazione ortogonale ai modelli di calcolo il corso potrebbe seguire la seconda parte del libro di Arora-Barak (cap. 12--16) affrontando i seguenti temi: Alberi di decisione, Branching programs, Circuiti logici e loro limiti, Complessità di Comunicazione, Dimostrazioni in sistemi logici, Modelli di computazione algebrici.

Un corso in cui si privilegia l'approfondimento, ad esempio, della sola teoria della complessità delle dimostrazioni, può prevedere i seguenti temi: Sistemi di dimostrazione, complessità delle dimostrazioni, relazioni con P vs NP. Codifica proposizionale di principi combinatori. Sistemi logici tradizionali: Sistemi di Hilbert e Calcolo dei Sequenti Sistemi Logici e loro limiti: Risoluzione e sue estensioni Sistemi Geometrici e loro limiti: Iperpiani di Chvatal e Sistemi Lovasz-Schröder. Sistemi Algebrici e loro limiti: Calcolo dei Polinomi, Nullstellensatz Automatizzabilità di sistemi di dimostrazione e generazione automatica di dimostrazioni.

Alcuni riferimenti bibliografici

- M. Sipser. Theory of computation
- J.L Balcazar, J. Diaz, J. Gabarrò. Structural Complexity Theory



- S. Arora, B. Barak. Complexity theory: a Modern Approach
- C. Papadimitriou. Computational Complexity
- E. Kushilevitz, N. Nisan. Communication Complexity
- M. Lee, P. Vitanyi. An Introduction to Kolmogorov Complexity and Its Applications
- J. Krajicek. Bounded Arithmetic, propositional Logic and Computational Complexity

Teoria dell'informazione

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Curriculum Metodi formali per il software

Insegnamento	Crediti	SSD	Ambito
Logica matematica per informatica	6	Inf/01	Caratt.
Metodi di verifica del software	6	Inf/01	Caratt.
Modelli di calcolo	6	Inf/01	Caratt.
Teoria della concorrenza	6	Inf/01	Caratt.

Logica matematica per Informatica

Il corso ha lo scopo di esporre lo studente ai risultati e alle tecniche dimostrative fondamentali della logica matematica contemporanea (teoria della dimostrazione, teoria della ricorsività, teoria dei modelli), con particolare attenzione alle loro applicazioni informatiche.

Obiettivi formativi: conoscenza dei risultati e padronanza delle tecniche fondamentali della logica matematica contemporanea, capacità di applicarle a nuovi contesti di ricerca informatica.

Metodi di verifica del software

L'attuale insegnamento di Verifica automatica.

Modelli di calcolo

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.



Teoria della concorrenza

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Curriculum Intelligenza artificiale e multimedia

Insegnamento	Crediti	SSD	Ambito
Apprendimento automatico	6	Inf/01	Caratt.
Interazione multimodale	6	Inf/01	Caratt.
Interazione per il web	6	Inf/01	Caratt.
Metodi di estrazione delle informazioni dal web	6	Inf/01	Caratt.
Visione artificiale	6	Inf/01	Caratt.

Apprendimento automatico

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Interazione multimodale

L'interazione uomo-uomo è essenzialmente multimodale; infatti, noi sfruttiamo diversi canali per comunicare, utilizzandoli sia in modo alternativo, tramite la parola, la scrittura, i gesti, il tatto, ecc., che simultaneamente, come ad esempio durante la comunicazione faccia a faccia. D'altro canto, l'interazione Uomo-Computer (HCI – Human-Computer Interaction) si è sviluppata in un ambito unimodale, con l'input dell'utente immesso tramite tastiera e successivamente tramite mouse, e l'output a video da parte del computer, nella forma di testi o icone.

Con lo sviluppo di nuove tecnologie, anche i paradigmi di interazione stanno mutando per sfruttarne i vantaggi, portando al superamento delle interfacce WIMP (Window-Icon-Menu-Pointer). Nuove tecniche di HCI fanno uso di nuove modalità di comunicazione. Questo corso intende fornire una visione d'insieme delle nuove interfacce e degli attuali argomenti di ricerca sulle tecnologie multimodali. Saranno introdotti i fondamenti della percezione e della comunicazione umana, allo scopo di comprendere le possibilità ed i limiti delle diverse interfacce multimodali.

Obiettivi

Al completamento del corso, gli studenti dovrebbero essere in grado di:



- evidenziare come i processi percettivi e cognitivi umani possono influenzare le possibilità ed i limiti di diverse interfacce multimodali
- descrivere le funzionalità di interfacce multimodali allo stato dell'arte
- valutare punti di forza e di debolezza di interfacce multimodali esistenti o proposte in letteratura
- proporre modelli efficienti per nuove interfacce che impiegano diverse modalità

Contenuti

Il corso darà agli studenti una presentazione teorica e pratica dei seguenti argomenti:

- percezione e cognizione umana
- comunicazione umana multimodale
- diversi tipi di interfacce HCI usate separatamente o in combinazione
- L'interesse si concentrerà quindi su:
 - input dell'utente: speech recognition, sensori di movimento, tracking di sguardo e gesti
 - output del computer: rappresentazioni visuali, tipi di display non convenzionali, sintesi del
 - linguaggio, oggetti sonori e dispositivi aptici
 - le modalità del futuro: adaptive brain interfaces

In particolare si esploreranno gli effetti della combinazione di differenti modalità, e il loro utilizzo in diversi campi applicativi.

Prerequisiti

Conoscenze di HCI e di programmazione.

Interazione per il web

Tecnologie del web: protocollo HTTP, ipertesto, metodi GET e POST, REST; XHTML, CSS, validazione, W3C; linguaggi server-side; linguaggi client side: javascript, flash; trasferimento dati: XML, JSON. Browser: funzionalità standard, interazione, compatibilità con gli standard e differenze tra browser.

Progetto di siti e applicazioni web: design e development; analisi dei requisiti; user research; modello agile; gruppi di lavoro interdisciplinari; User centred design; prototipazione e tool di prototipazione.



Tecniche di valutazione: expert based; con partecipazione dell'utente; lab / field / remote; query techniques; heatmaps; eye-tracking e metodi fisiologici. Analisi quantitativa: metodi statistici; analisi dei log e sistemi di web analytics; A/B testing; test multivariati.

Componenti interattivi per il web: widget standard; widget non standard; affordance.

Design per l'usabilità: home page design; content design; labelling; navigazione; search; form design.

Rich internet application: metodologie, problemi di usabilità.

Lavoro collaborativo e social web: interazione utente-utente.

Interazione su dispositivi mobili: limiti attuali della tecnologia web per dispositivi mobili; contesti d'uso; criteri di progettazione.

Tipologie di sito web: contesti per l'interazione e curve di apprendimento (informativo, commercio elettronico, applicazione, intranet, extranet).

Metodi di estrazione delle informazioni dal web

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Visione artificiale

L'obiettivo principale della visione artificiale e' quello di riprodurre la "vista" umana. I sistemi di visione artificiali quindi, mirano a creare un modello approssimato del mondo reale mediante l'analisi di immagini. Scopo del corso e' quello di fornire un'introduzione ai principi ed agli algoritmi per la costruzione di tali sistemi.



Curriculum Reti e sicurezza

Insegnamento	Crediti	SSD	Ambito
Crittografia	6	Inf/01	Caratt.
Prestazioni dei sistemi di rete	6	Inf/01	Caratt.
Reti avanzate	6	Inf/01	Caratt.
Sicurezza dei dati e delle reti	6	Inf/01	Caratt.

Crittografia

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Prestazioni dei sistemi di rete

Analisi quantitativa dei sistemi di elaborazione. Definizione di indicatori di qualità del servizio, di prestazione, di affidabilità ed efficienza. Il paradigma progettuale dell'autonomic computing applicato alle reti di elaboratori. Progettazione di sistemi di rete con proprietà self-* (self-configuration, self-adaptation, self-healing, self-protection). Il problema della misurazione e della predizione del carico di lavoro nei sistemi di rete. Valutazione analitica del carico di lavoro e predizione attraverso lo studio di serie temporali. Progettazione di sistemi auto-configuranti e adattativi basati su auto-apprendimento e sulla teoria dei sistemi di controllo automatico (cenni). Studio di alcuni problemi di prestazione e relative soluzioni algoritmiche e protocollari nel contesto dei sistemi geograficamente distribuiti e delle reti wireless. Efficienza energetica nei sistemi di routing e di calcolo distribuito, green data center, green cloud computing. Analisi di esempi concreti nelle reti di telecomunicazione.

Reti avanzate

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.

Sicurezza dei dati e delle reti

L'omonimo insegnamento dell'attuale laurea specialistica in Informatica.