

FACOLTÀ	Scienze MM.FF.NN.
ANNO ACCADEMICO	2014-2015
CORSO DI LAUREA	INFORMATICA
INSEGNAMENTO	INFORMATICA TEORICA
AMBITO DISCIPLINARE	Discipline Informatiche
CODICE INSEGNAMENTO	03946
ARTICOLAZIONE IN MODULI	SI
NUMERO MODULI	2
SETTORI SCIENTIFICO DISCIPLINARI	INF/01
DOCENTE RESPONSABILE (MODULO 1)	Antonio Restivo Professore Ordinario Università di Palermo
DOCENTE RESPONSABILE (MODULO 2)	Settimo Termini Professore Ordinario Università di Palermo
CFU	12
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	204
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	96
PROPEDEUTICITÀ	Analisi Matematica, Programmazione e Laboratorio, Metodi Matematici per l'Informatica
ANNO DI CORSO	II
SEDE DI SVOLGIMENTO DELLE LEZIONI	Da definire nel Calendario Didattico
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, Esercitazioni in aula
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova Scritta, Prova Orale, Presentazione di alcuni argomenti integrativi e complementari del programma sotto forma di seminari degli studenti
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Primo semestre (modulo 1), Secondo semestre (modulo 2)
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Da definire nel Calendario Didattico
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Martedì e Giovedì dalle 15.00 alle 17.00 (modulo 1), Contattare il docente (modulo 2)

<p>RISULTATI DI APPRENDIMENTO ATTESI</p> <p>Conoscenza e capacità di comprensione</p> <p>Acquisizione dei concetti fondamentali della Teoria degli Automi, dei Linguaggi Formali e della Teoria della calcolabilità con particolare riferimento ai principali modelli matematici utilizzati: automi a stati finiti, espressioni regolari, grammatiche. Acquisizione degli strumenti avanzati per leggere gli aspetti basilari della letteratura specialistica della disciplina. Capacità di utilizzare il linguaggio tecnico proprio della disciplina.</p>
--

Capacità di applicare conoscenza e comprensione

Capacità di riconoscere, ed organizzare in autonomia argomenti base dell'informatica teorica. Capacità di utilizzare le conoscenze acquisite (in particolare, le metodologie per la costruzione di automi e grammatiche) in campi applicativi specifici, con particolare riferimento all'elaborazione di testi.

Autonomia di giudizio

Essere in grado di valutare la rilevanza generale di argomenti della disciplina, e di collegare gli aspetti teorici della teoria degli automi, dei linguaggi formali e della teoria della calcolabilità con gli aspetti pratici di diversi ambiti applicativi.

Abilità comunicative

Capacità di esporre in modo chiaro e rigoroso le tematiche generali della teoria degli automi, dei linguaggi formali e della teoria della calcolabilità anche a un pubblico non esperto, mostrando come metodi e risultati matematici si rapportano a ambiti applicativi specifici.

Capacità d'apprendimento

Capacità di aggiornamento con la consultazione delle pubblicazioni scientifiche proprie del settore. Capacità di seguire, utilizzando le conoscenze acquisite nel corso, sia master di secondo livello, sia corsi d'approfondimento sia seminari specialistici nei settori trattati.

OBIETTIVI FORMATIVI DEL MODULO 1

Conoscere le capacità computazionali degli automi a stati finiti e la capacità generativa delle grammatiche non contestuali. Rapporti tra modelli deterministici e non deterministici. Capacità di convertire un formalismo in un altro equivalente: ad esempio, grammatiche e automi, automi e espressioni regolari, automi deterministici e non deterministici. Saper progettare automi che riconoscono linguaggi fissati. Saper progettare grammatiche che generano linguaggi fissati. Saper usare automi e grammatiche nella progettazione di algoritmi. Conoscere l'utilizzo degli automi e delle grammatiche come modello in alcune importanti di applicazioni: ad esempio, progetto di compilatori, software per progettare circuiti digitali, software per esaminare vaste collezioni di testi. Riportati nel Regolamento Didattico del Corso di Studio

MODULO	TEORIA DEGLI AUTOMI E DEI LINGUAGGI FORMALI
ORE FRONTALI	LEZIONI FRONTALI
48 (6 CFU)	
8 ore	<i>Automi a Stati Finiti</i> Motivazioni e descrizione informale. Definizione di automa a stati finiti deterministico (DFA). Linguaggio riconosciuto da un DFA. Rappresentazione di un DFA con grafo degli stati. Automi a stati finiti non deterministici (NFA). Teorema di equivalenza tra DFA e NFA. La "subset construction". Discussione sulla "state complexity" di DFA e NFA. Applicazioni alle ricerche testuali. Automi con ϵ -transizioni. Eliminazione delle ϵ -transizioni.
6 ore	Espressioni regolari. Linguaggi regolari. Equivalenza tra linguaggi regolari e linguaggi riconosciuti da DFA (Teorema di Kleene). Algoritmo di eliminazione degli stati per convertire un automa in un'espressione. Algoritmo di Berry e Sethi per convertire un'espressione in un automa.
2 ore	Il "pumping lemma" per i linguaggi regolari. Applicazioni del pumping lemma.
8 ore	Equivalenza e minimizzazione di automi. La relazione di indistinguibilità degli stati. Automa ridotto. Equivalenza tra automa ridotto e automa minimale. Teorema di Myhill-Nerode. Unicità dell'automa minimale.

	Algoritmo di minimizzazione di un DFA. Algoritmo per decidere l'equivalenza di due DFA
2 ore	Automati bidirezionali (2-DFA). Equivalenza tra 2-DFA e 1-DFA (Teorema di Rabin-Shepherdson).
4 ore	Problemi di decisione per i linguaggi regolari
8 ore	<i>Grammatiche e Linguaggi Liberi dal Contesto (CF)</i> Motivazioni e descrizione informale. Definizione di grammatica. Derivazioni delle grammatiche. Linguaggio generato da una grammatica. La gerarchia di Chomsky. Le grammatiche e i linguaggi CF. Alberi sintattici. Ambiguità nelle grammatiche e nei linguaggi CF: grammatiche ambigue, eliminazione delle ambiguità, ambiguità inerente.
6 ore	Forme normali. Forma normale di Chomsky. Pumping lemma per i linguaggi CF. Applicazioni del pumping lemma. Proprietà di chiusura dei linguaggi CF. Proprietà di decisione per i linguaggi CF
4 ore	Automati a Pila (PDA). Linguaggi riconosciuti da PDA. Equivalenza di PDA e grammatiche CF.
TESTI CONSIGLIATI	<i>J. E. Hopcroft, R. Motwani, J. D. Ullman, Automati, Linguaggi e Calcolabilità, Addison-wesley (PearsonEducation Italia) 2003.</i> <i>R. McNaughton, Elementary Computability, Formal Languages and Automata, Prentice-Hall, 1082</i> <i>D. Perrin, Finite Automata, Capitolo 1 del Vol.2 del Handbook of Theoretical Computer Science, Elsevir, 1990.</i>

OBIETTIVI FORMATIVI DEL MODULO 2
L'obiettivo del modulo è quello di fornire agli studenti gli elementi di base, concettuali e formali, della teoria della calcolabilità, mettendo in evidenza i rapporti esistenti tra alcuni risultati teorici di carattere generale e alcuni problemi e domande che sorgono a partire da aspetti apparentemente solo "tecnici" della programmazione.
Riportati nel Regolamento Didattico del Corso di Studio

MODULO	TEORIA DELLA CALCOLABILITÀ'
ORE FRONTALI	LEZIONI FRONTALI
48 (6 CFU)	
8 ore	<i>Aspetti generali della nozione di calcolabilità</i> Centralità della nozione di calcolabilità. Analisi di Turing del processo di calcolo. Enunciato e discussione della tesi di Church- Turing. Primi esempi di funzioni Turing-calcolabili. Definizione di produttività di una Macchina di Turing (MdT). Definizione della funzione p (produttività massima delle MdT a n stati, o "Busy Beaver Problem"). Dimostrazione della non Turing calcolabilità della funzione p . Sue conseguenze concettuali e formali. Presentazione intuitiva di vari "explicitata" formali dell' "explicitandum" <i>informale</i> di funzione calcolabile. Le funzioni ricorsive primitive. Costruzione della funzione di Ackermann. Definizione di funzione epsilon-ricorsiva e μ -ricorsiva.
	Le funzioni ricorsive primitive. Definizione e proprietà principali. Dimostrazione della ricorsività primitiva di varie funzioni elementari. Metodi di codifica ricorsivi primitivi. Numeri di Godel. Il linguaggio di programmazione S di Davis/Weyuker Dimostrazione della

12 ore	<p>definibilità di varie funzioni in S. Introduzione del concetto di macro. Codifiche dei programmi di S. Calcolabilità in S delle funzioni ricorsive primitive. Il teorema della "fermata". Esistenza di programmi "universali". Definizione di insieme ricorsivamente enumerabile (r.e.) e di insieme ricorsivo. Studio delle relazioni intercorrenti tra insiemi ricorsivamente enumerabili e insiemi ricorsivi. Il teorema di Post. Esistenza di insiemi ricorsivamente enumerabili ma non ricorsivi.</p> <p>Il teorema s-m-n (o del parametro) di Kleene. Alcune sue conseguenze. il teorema di ricorsione, il teorema del punto fisso, il teorema di Rice.</p>
8 ore	<p>Il linguaggio di programmazione LOOP di Meyer e Ritchie . Dimostrazione della LOOP-calcolabilità delle funzioni ricorsive primitive. Dimostrazione dell'equivalenza tra funzioni ricorsive primitive e funzioni calcolabili da programmi LOOP. Teoremi di limitazione alla crescita delle funzioni ricorsive primitive. Profondità di nidificazione dei cicli LOOP. La gerarchia L_n. Dimostrazione della non ricorsività primitiva della funzione di Ackermann. Inverso del teorema di limitazione alla crescita. Dimostrazione della calcolabilità in S della funzione di Ackermann. Introduzione del linguaggio WHILE come estensione del linguaggio LOOP. Dimostrazione della equivalenza tra il linguaggio S e il linguaggio WHILE.</p>
12 ore	<p>Linguaggi di programmazione S_n per il calcolo di stringhe su un alfabeto di n simboli. Simulazione in S_n delle funzioni calcolabili in S. Introduzione del linguaggio di T di Post -Turing e dimostrazione della calcolabilità in T delle funzioni parzialmente calcolabili in S_n.</p> <p>Dimostrazione della calcolabilità in S delle funzioni calcolabili da programmi di Post-Turing. Dimostrazione dell'equivalenza tra MdT a quadruple, MdT a quintuple e programmi di Post Turing. Dimostrazione dell'equivalenza tra MdT con nastro infinito bidirezionale e MdT con nastro infinito in una sola direzione. Macchine di Turing non deterministiche. Processi di Thue e simulazione di MdT non deterministiche mediante processi di Thue. Definizione di grammatica. Dimostrazione dell'equivalenza tra i linguaggi accettati da MdT non deterministiche e i linguaggi generati da grammatiche.</p> <p>Ricorsività primitiva degli operatori di derivabilità in una grammatica. Dimostrazione dell'equivalenza tra linguaggi ricorsivamente enumerabili e linguaggi generati da una grammatica. Varie caratterizzazioni degli insiemi ricorsivamente enumerabili. Il teorema della forma normale di Kleene.</p> <p>Dimostrazione dell'equivalenza tra funzioni calcolabili in S e funzioni μ-ricorsive. Non ricorsiva enumerabilità dell'insieme di indici delle funzioni ricorsive totali.</p>
8 ore	<p>Il problema della corrispondenza di Post e sua insolubilità algoritmica. Cenni al decimo problema di Hilbert, agli insiemi diofantei e al teorema di Matjasievic.</p> <p>I sette <i>Problemi del Millennio</i> come riproposizione dei problemi di Hilbert al Convegno del 1900.</p> <p>Il problema della complessità. Difficoltà di fornire modelli formali generali ed onnicomprensivi della nozione di complessità. La complessità astratta. Gli assiomi di Manuel Blum. La complessità secondo Kolmogorov. Cenni alla logica <i>fuzzy</i>.</p>
TESTI	<i>M. Davis, E. Weyuker, Computability, Complexity and Languages,</i>

CONSIGLIATI	Academic Press (1983). <i>G. S. Boolos, R. C. Jeffrey, Computability and Logic</i> , Cambridge University Press (1989). Si suggerisce, inoltre, la lettura di <i>E. Casari, Computabilità e ricorsività</i> , Quaderni della Scuola Superiore di Idrocarburi dell'ENI (1959).
--------------------	--