

FACOLTÀ	Scienze MM.FF.NN.
ANNO ACCADEMICO	2013/2014
CORSO DI LAUREA	INFORMATICA
INSEGNAMENTO	Informatica Teorica
TIPO DI ATTIVITÀ	Caratterizzante
AMBITO DISCIPLINARE	Discipline Informatiche
CODICE INSEGNAMENTO	16671
ARTICOLAZIONE IN MODULI	NO
SETTORI SCIENTIFICO DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	Antonio Restivo Professore Ordinario Università di Palermo
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	153
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	72
PROPEDEUTICITÀ	Analisi Matematica, Programmazione e Laboratorio C.I., Metodi matematici per l'informatica
ANNO DI CORSO	II
SEDE DI SVOLGIMENTO DELLE LEZIONI	Consultabile nel sito www.cs.unipa.it
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, Esercitazioni
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova Scritta, Prova Orale
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	II semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Come da calendario disponibile www.cs.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Martedì e Giovedì dalle 15.00 alle 17.00

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione

Acquisizione degli strumenti avanzati per leggere gli aspetti basilari della letteratura specialistica della disciplina. Capacità di utilizzare il linguaggio tecnico proprio della disciplina.

Capacità di applicare conoscenza e comprensione

Capacità di riconoscere, ed organizzare in autonomia argomenti base dell'informatica teorica.

Autonomia di giudizio

Essere in grado di valutare la rilevanza generale di argomenti della disciplina.

Abilità comunicative

Capacità di esporre le tematiche generali dell'informatica teorica anche a un pubblico non esperto..

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

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
8	<i>Breve introduzione alla teoria della calcolabilità</i> Formalizzazione della nozione di algoritmo. La macchina di Turing. Funzioni calcolate da una macchina di Turing. Linguaggi riconosciuti da una macchina di Turing. La tesi di Turing-Church. La macchina universale di Turing. Esistenza di funzioni non calcolabili. Il problema della "fermata" di una macchina di Turing. Problemi decidibili e indecidibili. Modelli particolari di macchine di Turing. La gerarchia di Chomsky.
8	<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.
8	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	Il "pumping lemma" per i linguaggi regolari. Applicazioni del pumping lemma.
8	Equivalenza e minimizzazione di automi. La relazione di indistinguibilità degli stati. Automa ridotto. Equivalenza tra automa ridotto e automa minimale. Unicità dell'automa minimale. Algoritmo di minimizzazione di un DFA. Algoritmo per decidere l'equivalenza di due DFA. Il teorema di Myhill-Nerode. Il monoide sintattico.
2	Automi bidirezionali (2-DFA). Equivalenza tra 2-DFA e 1-DFA (Teorema di Rabin-Shepherdson).
4	Proprietà di chiusura dei linguaggi regolari. Problemi di decisione per i linguaggi regolari
8	<i>Grammatiche e Linguaggi Liberi dal Contesto (CF)</i> Motivazioni e descrizione informale. Definizione di grammatica. Derivazioni delle grammatiche. Linguaggio generato da un 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	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
6	Automi a Pila (PDA). Linguaggi riconosciuti da PDA. Equivalenza di PDA e grammatiche CF. Automi a pila deterministici. Cenni sull'analisi sintattica.
	ESERCITAZIONI
12	Esercitazioni pratiche sugli argomenti svolti.
TESTI CONSIGLIATI	<p><i>J. E. Hopcroft, R. Motwani, J. D. Ullman, Automi, Linguaggi e Calcolabilità, Addison-wesley (PearsonEducation Italia) 2003.</i></p> <p><i>R. McNaughton, Elementary Computability, Formal Languages and Automata, Prentice-Hall, 1082</i></p> <p><i>D. Perrin, Finite Automata, Capitolo 1 del Vol.2 del Handbook of Theoretical Computer Science, Elsevir, 1990.</i></p>