UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2018/2019
ANNO ACCADEMICO EROGAZIONE	2019/2020
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	INFORMATICA TEORICA
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	16671
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	CASTIGLIONE GIUSEPPA Ricercatore Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	153
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	72
PROPEDEUTICITA'	05880 - PROGRAMMAZIONE E LABORATORIO C.I.
MUTUAZIONI	
ANNO DI CORSO	2
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	**CASTIGLIONE GIUSEPPA** Martedì 14:00 15:00 Dipartimento di Matematica e Informatica. Stanza 209 secondo piano. Giovedì 14:00 15:00 Dipartimento di Matematica e Informatica. Stanza 209 secondo piano.

DOCENTE: Prof.ssa GIUSEPPA CASTIGLIONE

PREREQUISITI	Nozioni di base di Logica e Matematica discreta.
RISULTATI DI APPRENDIMENTO ATTESI	Conoscenza e capacita' di comprensione Acquisizione dei concetti fondamentali della Teoria degli Automi, dei Linguaggi Formali e della Teoria della calcolabilita' con particolare riferimento ai principali modelli matematici utilizzati: automi a stati finiti, espressioni regolari, grammatiche. Sviluppo delle capacita' di formalizzazione, astrazione, modellazione di sistemi e analisi di problemi complessi. Comprendere la chiara distinzione tra aspetti sintattici e semantici. Conoscere l'esistenza di problemi non risolubili o di problemi "difficilmente" risolubili, in termini di risorse di calcolo, e quindi la necessita' di una loro classificazione in classi di complessita. Capacita' di applicare conoscenza e comprensione Capacita' di utilizzare le conoscenze acquisite alla costruzione di automi e grammatiche in campi applicativi specifici, con particolare riferimento all'elaborazione di testi e alla descrizione astratta di un algoritmo. Acquisizione degli strumenti per leggere e comprendere gli aspetti basilari della letteratura specialistica della disciplina. Capacita' di utilizzare il linguaggio tecnico proprio della disciplina. Autonomia di giudizio Essere in grado di valutare la rilevanza degli argomenti della disciplina, e di contestualizzare gli aspetti teorici della teoria degli automi, dei linguaggi formali e della teoria della calcolabilita' ad ambiti applicativi. Abilita' comunicative Capacita' di esporre in modo matematicamente chiaro e rigoroso le tematiche trattate della teoria degli automi, dei linguaggi formali e della teoria della calcolabilita' anche a un pubblico non esperto, mostrando come metodi e risultati matematici si rapportano a ambiti applicativi specifici. Capacita' d'apprendimento Capacita' di aggiornamento con la consultazione delle pubblicazioni scientifiche proprie del settore. Capacita' di seguire, utilizzando le conoscenze acquisite nel corso, i corsi di laurea magistrale, i master di secondo livello, corsi d'approfondimento e seminari specialistici nei settori trattati.
VALUTAZIONE DELL'APPRENDIMENTO	Una prova in itinere scritta (opzionale per gli studenti) svolta durante la pausa didattica, una prova finale scritta ed una prova orale. La prova in itinere ha durata di 1 ora e mezza e consiste nello svolgimento di tre esercizi su automi ed espressioni regolari. Ogni esercizio avra' una valutazione da 0/30 a 5/30, la prova risulta superata con una valutazione totale di almeno 9/30. La prova scritta finale ha la durata di tre ore e consiste nello svolgimento di sei esercizi (tre se la prova in itinere e' superata). Ogni esercizio avra' una valutazione da 0/30 a 5/30 da sommare eventualmente alla valutazione della prova in itinere. La prova risulta superata con una valutazione totale di almeno 18/30. Le prove scritte sono finalizzate ad accertare le abilita' a trattare il riconoscimento di linguaggi con gli automi e grammatiche, la capacita' di comprendere un problema ed individuarne la calcolabilita. Gli esercizi inoltre prevedono l'applicazione di algoritmi su automi e grammatiche studiati durante il corso, la verifica della proprieta' di riconoscibilita' e generabilita. Durante la prova orale lo studente dovra' rispondere a minimo tre domande poste oralmente, su argomenti del corso con riferimento ai testi consigliati e al materiale didattico fornito (slide, dispense, esercizi svolti). La prova ha l'obbiettivo di valutare le conoscenze acquisite, la comprensione degli argomenti, e l'acquisizione del linguaggio specifico. La soglia della sufficienza sara' raggiunta se lo studente mostrera' di conoscere i concetti fondamentali e sara' in grado di esporli ed argomentarli autonomamente. Al di sotto di tale soglia, l'esame risultera' non superato. I livelli di valutazione saranno proporzionali alle abilita' di linguaggio, all'esposizione corretta e dettagliata degli argomenti e alla capacita' di applicare, collegare e contestualizzare ii concetti. In particolare si individuano quattro fasce di valutazione della prova orale: 18-21/30 lo studente espone gli aspetti essenziali degli argomenti con linguaggio semplice, sa applicare i concetti a problemi semplici, contestualizza solo se guidato, 22-24/30 lo studente espone tutti gli aspetti degli argomenti con linguaggio adeguato, sa applicare i concetti a problemi semplici, contestualizza solo se guidato, 25-27/30 lo studente espone tutti gli aspetti degli argomenti con linguaggio

	adeguato, sa applicare i concetti a tutti i problemi proposti, contestualizza solo se guidato, 28-30/30 lo studente espone tutti gli aspetti degli argomenti con linguaggio adeguato, sa applicare i concetti a tutti i problemi proposti, contestualizza autonomamente. La lode verra' attribuita agli studenti capaci di dimostrare i teoremi esposti a lezione. La valutazione avviene in trentesimi e sara' la media delle valutazioni delle due prove (scritta itinere+finale e prova orale).
OBIETTIVI FORMATIVI	Conoscere le capacita' computazionali degli automi a stati finiti e la capacita' generativa delle grammatiche non contestuali. Rapporti tra modelli deterministici e non deterministici. Capacita' 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.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali
TESTI CONSIGLIATI	J. E. Hopcroft, R. Motwani, J. D. Ullman, Automi, Linguaggi e Calcolabilita, Addison-wesley (PearsonEducation Italia) III edizione 2009. J. E. Hopcroft, R. Motwani, J. D. Ullman, Automi, Linguaggi e Calcolabilita, Addison-wesley (PearsonEducation Italia) III edition 2009.

PROGRAMMA

ORE	Lezioni
6	Automi a Stati Finiti. Motivazioni, applicazioni e descrizione informale. I concetti centrali della teoria degli automi. Definizione di automa a stati finiti deterministico (DFA). Automi riconoscitori. Rappresentazione di un DFA con grafo degli stati e tabella delle transizioni. Automi a stati fini 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 epsilon-transizioni. Eliminazione delle epsilon-transizioni.
6	Espressioni regolari. Linguaggi regolari. Applicazioni di espressioni 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 regolare. Algoritmo di Berry e Sethi per convertire un'espressione in un automa.
6	Chiusura dei linguaggi regolari rispetto alle operazioni booleane e reverse. Il "pumping lemma" per i linguaggi regolari. Applicazioni del pumping lemma. Problema di decisione se un linguaggio regolare e' vuoto. Problema di inclusione ed equivalenza dei linguaggi regolari.
6	Equivalenza tra automi. Problema di decisione dell'equivalenza di due DFA. Minimizzazione di automi deterministici tramite gli algoritmi classici di minimizzazione. La relazione di indistinguibilita' degli stati. Automa ridotto. Equivalenza tra automa ridotto e automa minimale. Teorema di Myhill-Nerode. Unicita' dell'automa deterministico minimale.
6	Grammatiche e Linguaggi Liberi dal Contesto (CF). 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. Ambiguita' nelle grammatiche e nei linguaggi CF: grammatiche ambigue, eliminazione delle ambiguita, ambiguita' inerente. Alcune applicazioni delle grammatiche libere dal contesto.
6	Automi a pila. Definizione, linguaggio accettato. Equivalenza tra automi a pila e grammatiche context-free.
6	Forme normali. Forma normale di Chomsky. Pumping lemma per i linguaggi CF. Applicazioni del pumping lemma. Proprieta' di chiusura dei linguaggi CF. Problemi di decisione per i linguaggi CF
6	Breve introduzione alla teoria della calcolabilita. La macchina di Turing. Funzioni calcolate da una macchina di Turing. Linguaggi riconosciuti da una macchina di Turing. La tesi di Turing-Church.
6	La macchina universale di Turing. Esistenza di funzioni non calcolabili. Il problema della fermata di una macchina di Turing. Problemi decidibili e indecidibili. Problemi intrattabili. P ed NP. Problemi NP-completi. Modelli particolari di macchine di Turing. Gerarchia di Chomsky e decidibilita.
6	Uso di strumenti software per la manipolazione di automi. Determinizzazione, minimizzazione di automi con Jflap
3	Uso di strumenti software per la manipolazione di espressioni regolari.
6	Uso di strumenti software per la manipolazione di grammatiche. Alberi di derivazione con Jflap.
3	Pumping lemma per i linguaggi context-free con l'uso di Jflap.