

| Principali informazioni sull'insegnamento | |
|--|------------------------------|
| Titolo insegnamento | Calcolabilità e Complessità |
| Corso di studio | Triennale Informatica |
| Crediti formativi | 6 |
| Denominazione inglese | Computability and complexity |
| Obbligo di frequenza | no |
| Lingua di erogazione | Italiano |

| Docente responsabile | Nome Cognome | Indirizzo Mail |
|-----------------------------|---------------|--------------------------|
| | Giovanni Pani | giovanni.pani @ uniba.it |

| Dettaglio credi formativi | Ambito disciplinare | SSD | Crediti |
|----------------------------------|---------------------|---------|---------|
| | informatica | Info 01 | 6 |

| Modalità di erogazione | |
|-------------------------------|---------------------------------|
| Periodo di erogazione | Secondo semestre |
| Anno di corso | Secondo |
| Modalità di erogazione | Lezioni frontali Laboratorio |

| Organizzazione della didattica | |
|---------------------------------------|-----|
| Ore totali | 170 |
| Ore di corso | 62 |
| Ore di studio individuale | 108 |

| Calendario | |
|----------------------------|-------------|
| Inizio attività didattiche | 26 Febbraio |
| Fine attività didattiche | 1 Giugno |

| Syllabus | |
|---|--|
| Prerequisiti | |
| Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di apprendimento trasversali) | <ul style="list-style-type: none"> • <i>Capacità di comprensione dei limiti del calcolo, distinguendo ciò che è intrinsecamente non computabile rispetto a quello che può e potrà essere realizzato</i> • <i>Conoscenza e capacità di costruzione di grammatiche che definiscono linguaggi di programmazione.</i> • <i>Autonomia di giudizio nello scegliere strumenti, macchine astratte, per risolvere problemi di calcolo.</i> • <i>Abilità nel comunicare problemi di calcolo e loro soluzione.</i> • <i>Capacità di apprendere nuovi algoritmi relativi a macchine astratte.</i> |
| Contenuti di insegnamento | Introduzione. Automati, computabilità e complessità. Terminologia e |

notazione matematica. Definizioni, teoremi dimostrazioni.
Tipi di dimostrazioni.

Linguaggi regolari.

Definizione formale di automa finito. Esempi. Definizione formale di computazione. Progettare automi finiti. Le operazioni regolari. Definizione formale di automa finito non deterministico. Equivalenza tra automi deterministici e non deterministici, complessità della dimostrazione. Pumping lemma per i linguaggi regolari. Chiusura rispetto alle operazioni regolari. Definizione di espressione regolare. Dimostrazioni di equivalenza tra automi regolari, espressioni regolari, grammatiche di tipo 3 e complessità delle dimostrazioni. Il pumping lemma per i linguaggi regolari. Esercizi.

Linguaggi context free.

Definizione di grammatiche context free. Pumping lemma per context free come generalizzazione del pumping lemma per i linguaggi regolari. Applicazioni del pumping lemma. Esercizi.

Macchine di Turing.

Definizione della macchina di Turing. Esempi. Macchine di Turing multinastro, multitraccia, non deterministiche. Equivalenza tra i vari modelli. Macchina di Turing Universale, definizione, progettazione e implementazione. Tesi di Church. Linguaggi decidibili. Problemi decidibili per gli automi finiti, problemi decidibili per gli automi pushdown (cjk), problemi decidibili per le macchine di Turing. Linguaggi non decidibili, diagonalizzazione. Problema dell'alt, problemi semidecidibili.

Complessità.

Complessità temporale, complessità temporale polinomiale, complessità temporale esponenziale, complessità temporale non deterministica, complessità temporale polinomiale non deterministica. La classe P, la classe NP, Np completezza. Complessità spaziale, il teorema di Savitch. Pspace, NPspace.

Laboratorio: il software <http://www.jflap.org/>. Automi finiti deterministici e non deterministici, automi push down deterministici e non deterministici, Macchine di Turing deterministiche e non deterministiche. Macchine di Turing multinastro. La macchina universale di Turing

| | |
|---|--|
| Programma | |
| <p>Testi di riferimento:</p> <p>M.Sipser Introduzione alla teoria della Computazione.</p> <p>G.Ausiello, F. D'Amore, G.Gambosi. Linguaggi, modelli, Complessità.</p> | |
| Note ai testi di riferimento | |
| Metodi didattici | Lezioni frontali e esercitazione in laboratorio |
| Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro) | Scritto e Orale |
| Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello) | <ul style="list-style-type: none"> - Capacità di definire algoritmi calcolabili con una macchina di Turing. - Definire una macchina di Turing universale. - Capacità di verificare se una funzione è calcolabile e se si definire la sua complessità. |
| Altro | |