



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Computabilità, Complessità e Logica

Prof. Adriano Peron

Riferimenti

- ▶ Dipartimento di Matematica Informatica e Geoscienze, Via Economo 12/3, studio 430 (terzo piano)
- ▶ N. Tel. 040-5582619
- ▶ Email: adriano.peron@units.it

- ▶ Orario del corso:

Lunedì, 14-15

Martedì, 9-11

Mercoledì, 11-13

Giovedì, 9-11

- ▶ Ricevimento studenti:

Mercoledì, 14-16

Modalità di esame

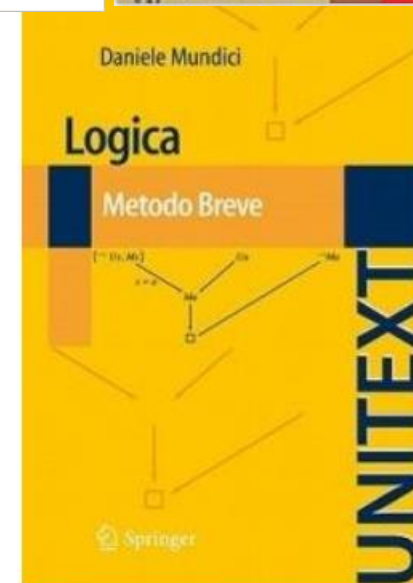
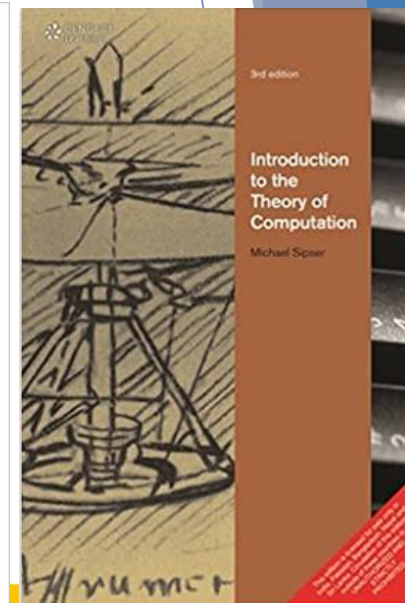
- ▶ L'esame consiste di una prova scritta ed una orale.
- ▶ La prova scritta riguarda le parti di calcolabilità e logica.
- ▶ E' possibile sostenere l'esame scritto:
- ▶ **Nella sua interezza**: mediante uno scritto che se superato sarà seguito da un colloquio orale
- ▶ **Per parti**: sostenendo separatamente le parti di computabilità e quella di logica.
- ▶ La parte di complessità è oggetto di solo esame orale.
- ▶ All'esame orale si accede dopo aver superato la prova scritta.
- ▶ L'esame orale riguarda l'interezza del programma con riguardo alle parti non oggetto della parte scritta: la parte di complessità, le prove dei teoremi concordati.
- ▶ E' possibile a richiesta prevedere prove intercorso scritte che esonerano dallo scritto finale.

Materiale didattico

- ▶ Slide del corso per i contenuti fondamentali in **File>Slide** del canale TEAMS.
- ▶ Esercizi per casa assegnati nelle «**Attività**» del canale TEAMS.
- ▶ Esercizi per casa annotati e corretti in **File>Esercizi** del canale TEAMS.
- ▶ Esempi di testi di esame in **File>Esami** del canale TEAMS.

Materiale didattico: testi

- ▶ Testo di riferimento per una trattazione completa e esaustiva:
- ▶ Parte di «**Computabilità e Complessità**»
- ▶ Michael Sipser, Introduction to the theory of computation, Thomson, 2006.
- ▶ Versione italiana.
- ▶ Michael Sipser, Introduzione alla teoria della computazione, Apogeo.
- ▶ Parte di «**Logica**»
- ▶ Daniele Mundici, Logica-Metodo breve, Springer, 2011.



Contenuti del corso

- ▶ L'obiettivo principale è di introdurre gli aspetti **fondazionali** dell'informatica studiando:
- ▶ **Modelli di computazione:**
 - ▶ strutture matematiche che permettono di proporre un modello di macchina capace di risolvere problemi, calcolare funzioni, descrivere il comportamento di sistemi, elaborare informazioni etc.
- ▶ **Espressività dei modelli di computazione:**
- ▶ Quali sistemi o funzioni il modello computazionale riesce a descrivere o computare? Quali problemi riesce a risolvere?
- ▶ **Esempio** (risoluzione problemi):
 - ▶ Posso verificare se le parentesi di una espressione algebrica sono bene accoppiate?
 - ▶ Posso verificare se una parola è palindroma?
 - ▶ Posso verificare se un numero intero è primo?

Contenuti del corso

- ▶ **Esempio** (descrivere il comportamento di sistemi):
 - ▶ Posso descrivere con precisione il comportamento del controllore di un ascensore che viene sollecitato dai pulsanti di chiamata interna ad esterna?
- ▶ **Esempio** (calcolare funzioni):
 - ▶ Posso calcolare la funzione $y=2x$?
- ▶ **Esempio** (elaborare informazioni):
 - ▶ Posso leggere un programma e stabilire se l'esecuzione del programma su uno specifico input terminerà?

Contenuti del corso: Computabilità

- ▶ C'è un modello **più espressivo** di altri?
- ▶ Tutti i problemi sono Risolvibili da una macchina?
- ▶ Oppure ci sono **limiti alla calcolabilità**?
- ▶ Come possiamo definire che cosa è **calcolabile** e cosa **non è calcolabile**?
- ▶ **Esempio**: E' possibile scrivere un programma A che riceve in ingresso un generico programma B e un input X per B e stabilisce se B termina con l'input X?

Contenuti del corso: Complessità

- ▶ Possiamo associare ad un problema un livello di difficoltà intrinseco di costo legato all'uso delle risorse per risolverlo?
- ▶ In altri termini un problema ha una precisa e intrinseca **complessità** indipendente dalle astuzie usate dal programmatore per risolverlo
- ▶ **Esempio di problemi su grafi:**
- ▶ 1) Stabilire se un nodo X del grafo è raggiungibile dal nodo iniziale.
- ▶ 2) Stabilire se un grafo ha un cammino Hamiltoniano (un cammino tra due nodi fissati che visita tutti i nodi rimanenti una sola volta).
 - ▶ 1) e 2) **hanno la stessa difficoltà?**

Requisiti di un modello

► Sintassi e semantica formali:

- **Sintassi formale**: è possibile stabilire senza ambiguità se una data descrizione (struttura) è una istanza del modello.
- **Semantica formale**: è possibile stabilire senza ambiguità quali siano le **computazioni** (o più in generale le strutture matematiche) associate al modello.

► Semplicità:

- Un modello computazionale è usualmente essenziale
- Non ha caratteristiche descrittive ridondanti.
 - Esempio: un linguaggio di programmazione ha diversi costrutti di controllo per esprimere iterazioni
- Svantaggio: Le descrizioni sono meno naturali e più intricate
- Vantaggio: E' più semplice studiare le proprietà del modello.

Piano di presentazione

► Computabilità (3 CFU)

- Dai modelli meno espressivi ai modelli più espressivi
- Macchine a stati finiti (Automati regolari)
- Macchine a stati illimitati (Automati con una struttura a pila)
- Macchine di Turing (Modello generale di computazione)

► Complessità (3 CFU)

- Le principali classi di complessità dei problemi in tempo e spazio.
- P
- NP
- PSPACE ed NPSPACE

► Logica (3 CFU)

- Calcolo proposizionale
- Calcolo dei predicate al primo ordine