

UNIVERSITÀ DEGLI STUDI "ROMA TRE"
CORSO DI STUDI IN MATEMATICA
IN2 - MODELLI DI CALCOLO – A.A. 2002-2003

M. PEDICINI

APPUNTI DI INFORMATICA: MACCHINE, LINGUAGGI E
MODELLI

INDICE

1. Informazioni	1
Lavoro Guidato	1
Valutazione	1
Bibliografia	1
Bibliografia Addizionale	1
2. Introduzione	1

Il corso rivolto a chi si interessa agli aspetti matematici della Computer Science, fornirà una panoramica sui risultati e sui principali problemi aperti in questo campo.

1. INFORMAZIONI

Pagina Web: <http://www.iac.rm.cnr.it/~marco/in2002>

e-mail: marco@iac.rm.cnr.it

Lavoro Guidato.

- lavoro guidato (scadenza mensile);
- esercizi di programmazione (programmazione funzionale ed object oriented);
- esami/esoneri (esonero parziale (novembre) ed esonero finale (gennaio)).

Valutazione.

- (1) (per chi segue il corso)
 - 35% esame finale (secondo esonero);
 - 65% media voti lavoro guidato (scartando il voto peggiore ad uno degli esercizi) + 1 esonero + progetto;
 - eventuale orale per "aggiustare" il voto finale.
- (2) (per chi non segue il corso)
 - esercizi di programmazione;
 - esame scritto;
 - esame orale.

Nota: **validità temporale degli esoneri**, l'esonero ha validità per un numero di sessioni che dipende dal voto x :

- $15 \leq x < 18$: valido per una sessione,
- $18 \leq x < 24$: valido per due sessioni,
- $24 \leq x < 27$: valido per tre sessioni,
- $27 \leq x < 30$: valido per tutti gli appelli dell'anno accademico.

Bibliografia.

- M. Pedicini, *Appunti di Informatica Teorica* (questi appunti) essenzialmente basati su: Patrick Dehornoy, *Calculabilite et Decidabilite* (1993), Springer-Verlag in francese;
- A. Bernasconi, B. Codenotti, *Introduzione alla complessità computazionale*, Springer-Verlag.
- J.-L. Krivine, *Lambda Calculus: Types and Models*, (Masson).
- P. H. Wintson, S. Narasimhan, *On to Java*, Addison-Wesley (1998).

Bibliografia Addizionale.

- R. Sethi, *Programming Languages: concepts and constructs*, Addison-Wesley (ed. italiana Zanichelli).
- Aho, Hopcroft, Ullman, *Design and Analysis of Computer Programming*.
- H. Hermes, *Enumerability, Decidability, Computability*, Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen, n. 127, Springer-Verlag.
- N. D. Jones, *Computability and Complexity from a Programming Perspective*, MIT Press.
- G. Ausiello, G. Gambosi, F. d'Amore - *Linguaggi, Modelli, Complessità* (draft scaricabile in rete).

2. INTRODUZIONE

L'argomento di studio, di questo corso è dato dai *modelli di calcolo* che sono stati via via proposti per catturare il concetto di calcolo automatico. Gli aspetti cui ci interessiamo sono essenzialmente quelli di natura matematica (laddove il concetto di calcolo ha interessato ed interessa molto la filosofia della scienza). Per fare questo una prima parte del corso riguarderà la presentazione di alcuni dei formalismi più utilizzati e dei legami che intercorrono tra essi.

Una prima osservazione che si può fare, è che i modelli proposti non necessariamente fanno riferimento all'idea ed alla forma che si è andata definendo per i calcolatori, e infatti alcuni dei modelli più usati sono stati proposti prima dell'avvento stesso dal calcolo meccanico, ed anzi forse lo hanno influenzato anticipandone le proprietà.

Se dovessimo ordinare rispetto ad una qualche scala evolutiva i modelli matematici del concetto di computazione, dovremmo situare in basso i modelli che più somigliano ai calcolatori elettronici e più in alto situare i modelli più astratti. Una caratteristica dei modelli più semplici risiede nel fatto di catturare il concetto in maniera più *intensionale*, intendendo con questo termine che il modello cattura il concetto andando a descrivere i passi elementari di calcolo permessi alla macchina più che fornire una descrizione del calcolabile. I modelli più astratti sarebbero invece quelli che ripongono la propria espressività su una descrizione *estensionale*, ovvero che considera due oggetti identici se il comportamento è indistinguibile; in questo caso una descrizione funzionale risulta certamente più vicina alla pratica del linguaggio matematico.

I modelli che analizzeremo sono: il modello RAM (Random Access Machine) una formalizzazione dell'architettura di Von Neumann alla base dei calcolatori elettronici, il modello di Macchina di Turing identificato con la definizione stessa di calcolabilità, il modello delle funzioni ricorsive e quello del calcolo delle lambda-espressioni alla base dei linguaggi di programmazione funzionale.

Dal punto di vista applicativo, affronteremo lo studio della programmazione funzionale ed object-oriented. In particolare, considereremo il linguaggio java per studiare alcune caratteristiche dell'OOP.