Starter Kit

Master Degree in Computer Science - a.y. 2015/2016

Introduction

The University of Pisa has a long scientific tradition that dates back to famous scientists as Fibonacci and Galileo Galilei, to name a few. In the small town of Pisa, almost half of the population is composed by university students, thus making it an ideal environment for obtaining a degree while enjoying a good student life. The first Italian university degree in Computer Science has been offered here in Pisa since the end of the 60s, as a follow-up of the intuition of Enrico Fermi who suggested to build the first Italian computer for scientific purposes in the 50s. We are committed to our tradition in offering high-quality undergraduate and graduate classes in Computer Science.

Starting up

Your academic life starts here at the University of Pisa for your career in Computer Science (CS). Probably your scientific background is different, as it does for the academic rules that you followed so far in your previous university. To help with your stay here, we provide some quick facts that can facilitate your impact with our CS classes at the University of Pisa. After that, the instructors of the courses of the first year provide some suggested readings, very helpful to know before attending that course. Please read carefully the information below at your earliest convenience before arriving in Pisa.

Quick facts

- Try to get your Student ID card as soon as possible. It will contain the ID enrollment number (called "*numero di matricola"* in Italian) that is asked everywhere.
- Our Master degree has various classes to attend in two years, with two semesters per year, plus a final dissertation. Each classes corresponds to a certain amount of ECTS credits, for a total of 120 ECTS (dissertation included) in two years.
- The first semester of the first year starts on September 21, 2015.

Courses

- The courses are aimed at a Master level and taught in English. As expected, they require some background from the attending students. For this, we suggest you to consider the material below.
- Indeed, you can take an initial self-test before starting our classes. The purpose of this self- test is to help you to get smoothly into our teaching system. If you know in advance which topics are required, you can follow our suggested readings and online courses below to get synchronized with our prerequisites. This is a helping hand that we hope you will be able to take.

Advanced algorithms (9 ECTS)

Instructor: prof. Roberto Grossi

Picture URL: http://www.di.unipi.it/~grossi/me.jpg

Course topics and goals:

The course will focus on the design and analysis of advanced algorithms and data structures for the efficient solution of combinatorial problems involving all basic data types, such as integers, strings, (geometric) points, trees and graphs. This course deepens and extends the algorithmic notions and skills of students, who are exposed to complex problems that require a significant effort in problem solving. One "brainstorming" lecture per week is devoted to a full immersion in this activity, and it is highly recommend to attend it. The purpose is not that of teaching/learning

further N algorithms, but to refine students' skills. The final written exam will be based on the topics discussed during the “brainstorming” lectures.

Background:

We expect the students to have attended an introductory course on algorithms and data structures during their Bachelor (or equivalent) degree.

Self-test URL: **http://www.di.unipi.it/~grossi/alg2test.html**

Suggested readings:

T. Cormen, C. Leiserson, R. Rivest and C. Stein. Introduction to Algorithms, 3rd Edition, MIT Press. (http://www.amazon.com/Introduction-Algorithms-Edition-Thomas-Cormen/dp/0262033844)

- Chapter 1: The Role of Algorithms in Computing
- Chapter 2: Getting Started
- Chapter 7: Quicksort
- Chapter 10: Elementary data structures
- Chapter 11: Hash Tables
- Chapter 12: Binary Search trees
- Chapter 22: Elementary Graph Algorithms
- Chapter 34: NP-Completeness

Suggested online course:

Khan Academy: Computer Science: Algorithms, https://www.khanacademy.org/computing/computer-science/algorithms

Numerical methods and Optimization (12 ECTS)

Instructors: prof. Roberto Bevilacqua, prof. Giancarlo Bigi

Course page URL: http://www.di.unipi.it/~bigig/dida/mno.html

Course topics and goals:

The course introduces some of the main techniques and methods for the solution of numerical problems. These methods often require the joint exploitation of the typical techniques of numerical analysis and of optimization algorithms. We show some of the main situations in which optimization methods are applied to solve problems of numerical analysis and some of the main situations in which the techniques of numerical analysis are essential to solve optimization problems. We also show the application of these methods to some specific problems chosen, for instance, in the following areas: regression and parameter estimation in statistics, approximation and data fitting, machine learning, data mining, image and signal reconstruction, economic equilibria and finance. The final oral exam will be a discussion concerning some of the topics developed during the course.

Background:

We expect the students to have attended introductory courses on Calculus, Linear Algebra, and Numerical Analysis during their Bachelor (or equivalent) degree.

Self-test URL: **http://www.di.unipi.it/~bigig/dida/mno/1516/test.pdf**

Suggested readings:

Richard L. Burden, J. Douglas Faires, Annette M. Burden, Numerical Analysis, Brooks/Cole CENGAGE Learning.
(http://tinyurl.com/pma5h99)

· Chapter 1: Mathematical Preliminaries and Error Analysis
  1.1 Review of calculus
  1.2 Round-off and Computer Arithmetics
  1.3 Algorithms and Convergence

- Chapter 2: Solutions of Equations in One Variable
  2.2 Fixed-Point Iteration
  2.3 Newton's Method
- Chapter 3: Interpolation and Polynomial Approximation
  3.1 Interpolation and Lagrange Polynomial
- Chapter 6: Direct Methods for Solving Linear Systems
  6.1 Linear Systems of Equations
  6.2 Pivoting Strategies
  6.3 Linear Algebra and Matrix Inversion
- Chapter 7: Iterative Techniques in Matrix Algebra
  7.1 Norm of Vectors and Matrices
  7.2 Eigenvalues and Eigenvectors

Models of computations (9 ECTS)

Instructor: prof. Roberto Bruni
Picture URL: http://www.di.unipi.it/~bruni/me.jpg
Course topics and goals:

The objective of the course is to present:
different models of computation,
their programming paradigms,
their mathematical descriptions, both concrete and abstract, and also to present some intellectual tools/techniques:
for reasoning on models, and
for proving (some of) their properties.
To this aim, a rigorous approach will be followed for both the syntax and the semantics of the paradigms we work on:
IMP: imperative models
HOFL: functional models
CCS, pi-calculus: concurrent, non-deterministic and interactive models
PEPA: probabilistic/stochastic models
The focus will be on basic proof techniques (there will be not time to introduce more advanced topics in detail).

In doing so, several important notions will be overviewed (not necessarily in this order): proof systems (axioms and inference rules), goal-driven derivations, various incarnations of well-founded induction, structural recursion, lambda-calculus, program equivalence, compositionality, completeness and correctness, type systems, domain theory (complete partial orders, continuous functions, fixpoint), labelled transition systems, bisimulation equivalences, temporal and modal logics, Markov chains, probabilistic reactive and generative systems.

Background:

We expect the students to have some familiarity with context-free grammars, first-order logic formulas and code fragments in imperative and functional style.

Self-test URL: **http://didawiki.di.unipi.it/doku.php/magistraleinformatica/mod/start/pretest**

Suggested readings:

Introduction to Automata Theory, Languages, and Computation, 3/E John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman Pearson, 2007
ISBN-10: 0321455363, ISBN-13: 9780321455369 Chapter 1 Automata: The Methods and the Madness Chapter 2 Finite Automata Chapter 4 Properties of Regular Languages Chapter 5 Context-Free Grammars and Languages Chapter 9 Undecidability

Foundations of Computer Science
http://infolab.stanford.edu/~ullman/focs.html
Chapter 1 Computer Science: The Mechanization of Abstraction Chapter 2 Iteration, Induction, and Recursion Chapter 10 Patterns, Automata, and Regular Expressions Chapter 11 Recursive Description of Patterns Chapter 12 Propositional Logic Chapter 14 Predicate Logic

Suggested online course:

Introduction to Automata Theory, Languages, and Computation
http://infolab.stanford.edu/~ullman/ialc.html

Principles of programming languages (9 ECTS)

Instructor: prof. Andrea Corradini
Picture URL: http://www.di.unipi.it/~andrea/Andrea_Corradini.jpg
Course topics and goals:

The course presents principles and techniques for the implementation and usage of programming languages. Starting from the formal definition of the syntax of such languages, the first part introduces the main phases of a compiler with main emphasis on the lexical, syntactical and semantical analysis phases of the front-end. The second part covers the main topics of the structure of programming languages from the viewpoint of the runtime support of its abstract machine and of the expressiveness of the supported linguistic constructs. The focus will be on constructs of imperative, functional and object oriented languages.

Background:

We expect students to have some familiarity with the theory of formal languages (string grammars, in particular context-free ones, regular expressions, finite state automata), and some programming experience with one object-oriented language (like Java, C++, C#, ...) and with one functional language (like ML, Scheme, Haskell, ...).

Self-test URL: **http://www.di.unipi.it/~andrea/Didattica/PLP_PreliminaryTest.pdf**

Suggested readings:

Introduction to Automata Theory, Languages, and Computation, 3/E John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman Pearson, 2007ISBN-10: 0321455363, ISBN-13: 9780321455369 Chapter 1 Automata: The Methods and the Madness Chapter 2 Finite Automata Chapter 4 Properties of Regular Languages Chapter 5 Context-Free Grammars and Languages
Programming Languages: Principles and Paradigms by Maurizio Gabbrielli and Simone Martini, Springer, 2010. URL: http://www.springer.com/computer/swe/book/978-1-84882-913-8.

Chapters 1, 2 and 3.

Suggested online course:

Introduction to Automata Theory, Languages, and Computation
http://infolab.stanford.edu/~ullman/ialc.html

Advanced programming (9 ECTS)

Advanced data base systems (9 ECTS)