Département d'Informatique

COMPUTER SCIENCE STUDIES

AT ÉCOLE NORMALE SUPÉRIEURE

University year 2012/2013

Some courses descriptions are not translated into English or are not updated
(last modif 11 september 2012)

Département d’informatique – ENS, 45 rue d’Ulm, F-75230 Paris Cedex 05, France

http://diplome.di.ens.fr

Version: 11 September 2012

CONTENTS

1 Studies in computer science at ENS and the ENS Diploma

1.1 Presentation of the ENS Diploma

Following the harmonisation of European higher education, the Ecole Normale Supérieure has created in 2005 its own degree entitled *ENS Diploma*. It constitutes the pedagogical and scientific framework into which the ENS offers to set the predoctoral studies, beyond the academic curriculum. Its purpose is to offer a diversity of curricula combining an education of excellence in the main discipline and a flexible and ambitious opening in other disciplines.

The Diploma is open to students from preparatory classes (preparing for entrance examinations to Grandes Ecoles) and to students from French or foreign universities willing to receive the same education as the civil servant students “normaliens” and international selection scholars). The students are subject to a specific selection procedure (cf. *Admission conditions and procedures)*.

The ENS Diploma is awarded after three academic years (as a general rule[1]), during which each student validates:

- a high level academic curriculum sanctioned by a Masters degree in a discipline named the “main speciality of the Diploma”. As a general rule[1], this curriculum comprises the third year of the Bachelor (L3) and the two years of the Masters (M1 and M2), and each one of the three years corresponds to the validation of 60 ECTS units (European Credit Transfer System);
- additional courses for at least 36 ECTS units validated during the three years of studies. These can (i) be chosen in the main speciality (courses followed in the discipline of the Masters), (ii) constitute the “secondary speciality of the Diploma” (coordinated courses in another discipline) or (iii) take advantage of the diversity of courses proposed by the ENS to the students.

1.2 Computer science studies at École normale supérieure

Within the ENS Diploma, the studies in computer science differ according to whether this discipline is the main speciality of the student or not:

- *Computer science students*: students for which computer science is the main speciality are pedagogically and scientifically attached to the computer science department. The students registered in the computer science department follow the *computer science speciality of the ENS Diploma*, which is a specific high level curriculum comprising the third of year of the Bachelor (L3) and the two years of the Masters (M1 and M2).
- *Students from another scientific department:* the computer science department also offers courses to the students registered for other disciplines in other departments of the ENS. They can be arranged either as a coherent set of courses in computer science and constitute the secondary speciality of the student, or as a set of independent courses that the student validates for his/her Diploma with the consent of his/her tutor and the professors of these courses.

[1] French or foreign students may also enter in second year, i.e., at the beginning of the Master: the ENS Diploma is then awarded after the two years of the Master (M1 and M2).)

There are bridges between the different departments of the ENS. With the consent of the directors of studies, students of the computer science speciality may be reoriented during their curriculum, either to other disciplines in the ENS Diploma, or to other academic curricula outside the Diploma.

2 Persons in charge

Director of studies:

Patrick Cousot

Director of teaching activities:

Jean Vuillemin

Secretary:

Isabelle Delais
École normale supérieure
Département d'informatique
45, rue d'Ulm – 75230 Paris cedex 05
Tel: +33 (0)1 44 32 20 45
Fax: +33 (0)1 44 32 20 75
Email: diploma AT di.ens.fr or isabelle.delais AT ens.fr

3 Objectives and opportunities

3.1 Objectives of the computer science speciality of the ENS Diploma

The students registered in the computer science speciality of the Diploma are administratively linked to the computer science department of the ENS. This department offers for their main speciality an original academic curriculum with a small number of students (about twenty students per year, including the civil servant students and the international selection scholars), which are trained as computer scientists with a solid background in pure and applied mathematics. A close supervision of the students enables a faster pace and a deeper reflection than in other curricula. The courses are completed by mandatory research internships.

Objectives of the studies in computer science at ENS:

- integrating, in a top-level curriculum in computer science, civil servant students, international selection scholars and students selected by the computer science department to follow their studies in the ENS Diploma (cf *Admission conditions and procedures*);
- training for and by research, aiming at bringing each student to a high professional level, with top-level scientific courses. Research internships abroad, which are targeted and mandatory, open international opportunities for the student in his/her chosen domains;
- orienting each student according to its specific background and personal choices, thanks to a flexible structure. For this purpose, each student of the computer science speciality is personally monitored by a tutor all along his/her curriculum; he/she is also invited to follow introductory courses in other disciplines, which customize his/her curriculum.

3.2 Opportunities for graduating students of the computer science speciality of the ENS Diploma

Each graduate of the ENS Diploma will have a research Master's. A graduate from the computer science speciality of the Diploma can then start a PhD in mathematics or computer science, which s/he will achieve in principle after two or three years of research work after his curriculum. S/He may also work immediately after obtaining his/her Diploma.

Possible opportunities for graduates, after completing a PhD, are as follows:

- researcher in computer science in a public research institution (CNRS, CEA, INRIA, ONERA, CNES, etc.) ;
- professor/researcher in a French or foreign university ;
- researcher in computer science in the industry (France Telecom, EADS, etc.) ;
- computer science engineer in the industry in France or abroad ;
- teacher in preparatory classes or, more generally, in higher education (IUT, CNAM, etc.)

4 Admission conditions and procedures

4.1 Civil servant students and international selection scholars

After they have obtained 120 ECTS units (L2) by following the preparatory classes and passed the competitive exam for the ENS, civil servant students wishing to follow a curriculum in computer science start their studies in computer science at ENS by registering in the first academic year (L3).

The international selection scholars also have access to the studies in computer science at ENS, either at the first year level (L3) or directly at the second year level (M1), depending on their previous studies in their former university.

4.2 The students of the ENS diploma

Students from either a preparatory class or French or foreign university, having obtained (directly or by equivalence) 120 ECTS units (L2) or 180 ECTS units (L3) are allowed to apply for the ENS Diploma to follow studies in computer science.

For the computer science speciality as in other disciplines within the ENS Diploma, the general rule is to select students at the third year of the Bachelor (L3). Access may however be granted at the Masters level (M1), in particular for students from foreign universities.

4.3 Application to the computer science speciality of the ENS Diploma

The application dates (between mid-April and mid-July), conditions and procedure are available at the following URL:

http://diplome.di.ens.fr/Candidature.html

Nota Bene – Accommodation and Grants

There are very few rooms at the ENS for the students admitted to prepare the ENS diploma. Therefore, the candidates to the ENS diploma must see to looking for accommodation before the selection results are published.

Accommodation and/or grant applications via the CROUS (Centre Régional des Œuvres Universitaires et Scolaires) are to made before 30 April (usual deadline).

5 Registrations, tutors and programs of studies

In mid September, the students admitted to prepare the ENS diploma in computer science, the civil servants students and the international selection scholars who wish to study computer science are requested to go and meet the Computer science studies secretary in order to see to the various registrations and to have a personal tutor chosen for them.

5.1 Administrative and pedagogical registrations

The students of the computer science speciality of the Diploma must register, for each one of the three years of their curriculum, both (i) to the university degree corresponding to their current academic year and (ii) to the ENS diploma:

	University degree	**ENS Diploma**
1st year:	registration in L3 at University Paris 7	registration at ENS
2nd year:	registration in M1 (M.P.R.I.) at ENS	registration at ENS
3rd year:	registration in M2 (M.P.R.I.) at ENS	registration at ENS

***University degree**:* after they have been selected for the studies in computer science at ENS, the students register in the third year of the Bachelor (L3) - via the computer science studies secretary - at University Paris 7, which awards them the Bachelor's diploma at the end of the first year of the curriculum. During the next two years, the students register in the *Master parisien de recherche en informatique* (M.P.R.I.) at ENS, which awards them the Master's diploma at the end of their curriculum. It is also possible to register in another Master's (such as the research Master's « Mathématiques appliquées - Mathématiques /vision /apprentissage » (Applied mathematics – mathematics/vision/learning) of École normale supérieure de Cachan).

***ENS Diploma**:* the student registers in the « Diplôme de l'École normale supérieure » every year of the curriculum. The computer science studies secretary provides some of the mandatory documents for the registration.

5.2 Tutors

A tutor, teacher or researcher in the computer science department, is chosen for each student. The tutors will help the students organize their studies and will advise them for their internships, their research, their careers.
Students are requested to meet their tutors regularly and not only for the signatures of programs of studies or the assessments of results.
Students may ask to have their tutor changed.

5.3 Programs of studies

Each year, the students sign a program of studies (also called contract of studies) for the current year. It is given to the direction of studies of the ENS after it has been signed by their tutors and by the director of studies of the computer science department.
The program of studies contains not only the mandatory courses of the chosen curriculum but also the additional courses or activities that might be validated for the ENS diploma.
The Computer Science department may demand additional programs of studies.

6 Organization of the curriculum

The studies in computer science within the ENS Diploma are organized over three years, corresponding to the academic years L3 (Licence), M1 and M2 (Master). The academic degrees are awarded provided 60 ECTS are validated each year. At the end of the three years, the student who has passed his/her Master's and validated additional courses for 36 ECTS units will be awarded the Diploma of the École normale supérieure. Courses validated for a national university degree (Bachelor or Master) cannot be validated a second time as an additional course of the ENS Diploma.

The additional courses of the ENS Diploma are divided into three categories:

- **a language course (3 ECTS)**. There is a structure at ENS, named ECLA (Espace des Cultures et Langues d'Ailleurs), specialized in teaching language courses, which allows the students to validate this course in first, second or third year of the Diploma. Courses of English for scientists are also offered. This language course is not mandatory if the student has done an internship lasting at least 6 months in a foreign laboratory during his diploma.
- **mandatory courses of the computer science speciality for 12 ECTS units**. These two courses must be chosen among the M1 and M2 courses of the Diploma which have not been already chosen for the M.P.R.I, or among M1 or M2 mathematics courses.
- freely chosen courses, **including at least 12 ECTS units outside computer science**.

They can be chosen among the courses offered by the computer science department or by any other department of ENS, but also in other university curricula, with the agreement of the tutor and of the director of studies of the computer science department.

Students are strongly advised to validate additional courses for at least 12 ECTS each year.

6.1 First year

The licence (L3) is awarded by University Paris 7. It requires 60 ECTS, divided in 48 ECTS of courses at levels L3 (first and second semester) and M1 (second semester), and 12 ECTS of research internship. These courses are organized and taught at ENS. They are regularly renewed so as to closely follow the advancement of science. The diversity of the topics and teachers allows many different future opportunities for the students.

The students also validate additional courses for the ENS Diploma (at least 12 ECTS recommended every year).

Research internships in laboratories (in universities or in the industry) are planned at the end of the first year (during the summer) for students, priority being given to locations in France outside Paris.

At the end of the first year, the board of examiners, within the partnership with University Paris 7, decides on awarding the Bachelor diploma (L3) to the student, and on his/her admission into the second year of ENS diploma.

As from September 2012, **Computer science-Maths and Maths-Computer Science curricula** are offered. The civil servant students or international selection scholars who choose them will be registered **either in L3 of Mathematics OR in L3 of Computer Sciences and will have to validate only one L3** (cf. 7.1.4 Computer science-Maths &Maths- Computer Science curricula of the 1st year)

6.2 Second year

The second year consists of M2 courses for 30 ECTS during the first semester and of a research internship lasting about 5 months in a foreign laboratory, for 30 ECTS.

In parallel, the students are offered mini-courses of research level which are taught by specialists (most often foreign professors invited by ENS). The students also validate additional courses for the ENS Diploma (at least 12 ECTS recommended every year).

The board of examiners meets again at the end of the second year and decides on awarding the first year of the Master‘s degree (M1) to the student, and on his/her admission to the third year of the ENS diploma.

6.3 Third year

During the third year, the student completes his/her Master’s degree by following M2 level courses during the first semester for 30 ECTS and, during the second semester, he/she spends at least 5 months doing a research internship in France or abroad, for 30 ECTS. The students also validate additional courses for the ENS Diploma (at least 12 ECTS recommended every year).

At the end of the year, the student usually chooses a PhD supervisor and subject. At this level, the students progressively integrate a research laboratory. In order to facilitate the integration in the research field, it is often advisable to spend the whole year or part of the year in a laboratory in France outside Paris or in a European country.

The board of examiners of the Master’s degree decides on awarding the Master‘s degree to the student. The student is then proposed to the direction of studies of ENS to obtain the Diploma. If the student has validated additional courses for 36 ECTS units over the course of his/her curriculum, the ENS awards him/her the ENS Diploma with the main speciality corresponding to the one of the Master’s and, if relevant, with a secondary speciality in another discipline (cf. *Presentation of the ENS Diploma*).

6.4 Internships

Besides the mandatory internships of L3, M1 and M2, students may as from the 2nd year spend a whole year abroad doing an internship.

Please note that internships need to be prepared several months in advance in order to obtain permissions of the directors of studies, authorizations of the foreign lab or company, visas to spend a long time in a foreign country, signatures of the internship agreements and sometimes authorizations to take long leaves of absence (“Congé Sans Traitement” for student-civil servants and international selection scholars or “Césure” for students of the ENS diploma).

7 Courses for university year 2012-2013

In the list below, the French titles of the courses are between brackets. A detailed description of the contents of each course is given in the last pages of this brochure.
The contact details of most teachers can be found in the Intranet directory of ENS and on Internet. The ECTS (European Credit Transfer System) units are indicated for each course.

7.1. First year: licence (L3)

7.1.1 First semester

The 4 following courses are mandatory:

-ALGORITHMS AND PROGRAMMING (ALGORITHMIQUE ET PROGRAMMATION) – 6ECTS
Jacques Stern, Claire Mathieu

-PROGRAMMING LANGUAGES AND COMPILATION (LANGAGES DE PROGRAMMATION ET COMPILATION) – 6ECTS
Jean-Christophe Filliâtre, Louis Mandel

-FORMAL LANGUAGES, COMPUTABILITY AND COMPLEXITY (LANGAGES FORMELS, CALCULABILITÉ ET COMPLEXITÉ) – 6ECTS
Eugène Asarin, Anne Bouillard

-DIGITAL SYSTEM: FROM ALGORITHM TO CIRCUIT (SYSTÈME DIGITAL : DE L'ALGORITHME AU CIRCUIT) – 6ECTS
Jean Vuillemin

Students must choose and validate at least 1 of the following mathematics or computer-science courses during the 1st year:

-ALGEBRA 1 (ALGÈBRE 1)
Olivier Biquard
(Maths course: 12 ECTS.1st semester)

-INTEGRATION AND PROBABILITY (INTÉGRATION ET PROBABILITÉS)
Zhan SHI
(Maths course: 12 ECTS.1st semester)

-RANDOM STRUCTURES AND ALGORITHMS (STRUCTURES ET ALGORITHMES ALÉATOIRES)
Anne Bouillard, Pierre Bremaud
(Computer science course: 6 ECTS.1st semester)

-COMPLEX AND HARMONIC ANALYSIS (ANALYSE COMPLEXE ET HARMONIQUE)
Wendelin Werner
(Maths course: 12 ECTS.2nd semester)

-STATISTICAL MACHINE LEARNING (APPRENTISSAGE STATISTIQUE)
Francis Bach, Olivier Catoni
(Maths course: 12 ECTS.2nd semester)

7.1.2 Second semester

The following computer course is mandatory:

-OPERATING SYSTEMS AND COMPUTER NETWORKS (SYSTÈMES ET RÉSEAUX) – 6ECTS
Marc Pouzet, Louis Mandel

Students must choose and validate at least 2 M1 level computer science courses, among the following:
One of these courses may be replaced by a mathematics course.

- DATABASES (BASES DE DONNÉES) – 6ECTS
Serge Abiteboul
(This course takes place at ENS Cachan)

-GEOMETRIC FOUNDATIONS OF COMPUTER SCIENCE (BASES GÉOMÉTRIQUES DE L'INFORMATIQUE) – 6ECTS
Michel Pocchiola, Jean Ponce

-SOFTWARE ENGINEERING AND CLOUD COMPUTING (GÉNIE LOGICIEL ET CLOUD COMPUTING) – 6ECTS
Joannes Vermorel

-L'INFORMATIQUE SCIENTIFIQUE PAR LA PRATIQUE – 6ECTS
David Naccache

-INITIATION TO CRYPTOLOGY (INITIATION À LA CRYPTOLOGIE) – 6ECTS
Jacques Stern, Damien Vergnaud

-COMPUTER SCIENCE LOGIC (LOGIQUE ET INFORMATIQUE) – 6ECTS
Jean Goubault-Larrecq
*(This **highly recommended** course takes place at ENS Cachan)*

-COMMUNICATION NETWORKS (RÉSEAUX DE COMMUNICATION) – 6ECTS
Anna Busic

-INFORMATION THEORY AND CODING (THÉORIE DE L'INFORMATION ET CODAGE)
Marc Lelarge

- SIGNAL PROCESSING (TRAITEMENT DU SIGNAL) – 6ECTS
Stéphane Mallat

7.1.3 Computer science internship

Students must do an internship lasting 2 to 3 months in a public or private research laboratory, either in France (preferably outside Paris) or in a European country, between June and end of August 2013.
This internship (which also includes writing an internship report and making a presentation) counts 12 ECTS for the licence (L3).
L3 internships of previous years: http://www.di.ens.fr/~vergnaud/stages.html

7.1.4 Computer science-Maths & Maths–Computer science curricula of the 1st year (as from September 2012)

The civil servant students or international selection scholars who choose them will be **registered in either in a L3 of Computer Science OR a L3 of Mathematics.**

These curricula are jointly organized by the departments of mathematics and computer science of ENS.

They offer:

- a double curriculum in both mathematics and computer science to motivated students.
- the possibility to postpone by one year the choice between mathematics and computer science.

---Computer science-Maths curriculum for the students registered in L3 of Computer science and attached to the Computer science department (DI)

- Computer science courses:

-- 4 computer science courses in the 1st semester

- Algorithmique et programmation (Algorithms and Programming)
- Langages formels, calculabilité et complexité (Formal languages, computability and complexity)
- Langages de programmation et de compilation (Programming languages and compilation)
- Structures et algorithmes aléatoires (Random structures and Algorithms)

-- 2 computer science courses in the 2nd semester

-- + the 2nd semester course specific to these curricula: Apprentissage statistique (Statistical Machine Learning)

- Mathematics courses:

-- 2 courses to choose among the 5 courses below:

- Logique (Logic) 1st semester
- Intégration et probabilités (Integration and probability) 1st semester
- Algèbre I (Algebra I) 1st semester
- Algèbre II (Algebra II) 2nd semester
- Processus aléatoires (Stochastic processes) 2nd semester

-- + Analyse complexe et harmonique (Complex and Harmonic Analysis) 2nd semester

- Internship (12 ECTS) and Presentation (exposé) of the maths-computer science curriculum (12 ECTS)

This personal work will be supervised by a teacher in mathematics and a teacher in computer science and will consist in:

- bibliographical work similar to the presentation (exposé) of the 1st year of the mathematics curriculum. This work will take place in the 2nd semester, will be supervised by a mathematics teacher and/or a computer science teacher and will be focused on a subject connected to that of the internship.

- an internship in a computer science laboratory.
- the writing of a report in two parts and a presentation before a board of examiners belonging to the mathematics and computer science departments.

At the end of the 1rst year, the students who wish to continue their studies in mathematics will have to obtain the consent of the mathematics department and will have to validate the fundamental maths courses not validated during the 1rst year.

--- Maths-Computer science curriculum for the students registered in L3 of Mathematics and attached to the Mathematics department(DMA)

- Mathematics courses:

-- 3 courses to choose among the 5 courses below:

- Logique (Logic) 1st semester
- Intégration et probabilités (Integration and probability) 1st semester
- Algèbre I (Algebra I) 1st semester
- Algèbre II (Algebra II) 2nd semester
- Processus aléatoires (Stochastic processes) 2nd semester

-- + the 2nd semester course specific to these curricula: Apprentissage statistique (Statistical Machine Learning)

-- + Analyse complexe et harmonique (Complex and Harmonic Analysis) 2nd semester

- Computer science courses:

-- 2 computer science courses in the 1st semester

- Langages formels, calculabilité et complexité (Formal languages, computability and complexity)
- Algorithmique et programmation (Algorithms and Programming)
  OR
  Langages de programmation et de compilation (Programming languages and compilation)

-- + 1 computer science courses in the 2nd semester

- Internship (12 ECTS) and Presentation (exposé) of the maths-computer science curriculum (12 ECTS)

This personal work will be supervised by a teacher in mathematics and a teacher in computer science and will consist in:

- bibliographical work similar to the presentation (exposé) of the 1st year of the mathematics curriculum. This work will take place in the 2nd semester, will be supervised by a mathematics teacher and/or a computer science teacher and will be focused on a subject connected to that of the internship.
- an internship in a computer science laboratory.
- the writing of a report in two parts and a presentation before a board of examiners belonging to the mathematics and computer science departments.

At the end of the 1rst year, the students who wish to continue their studies in computer science will have to obtain the consent of the computer science department and will have to validate some mandatory computer science courses not validated during the 1rst year.

7.2. Second year: Master (M1)

7.2.1 First semester

As from 2012-2013,
the M1 computer science student must attend and validate at least 1 Mathematics course unless s/he has already validated 2 mathematics courses and/or Logic during the 1st year.

Logique (Logic) is mandatory except if validated during the the 1st year.
-LOGIC (LOGIQUE)
Martin Hils
(Maths course: 12 ECTS.)

The M1 computer science student is strongly advised to attend and validate at least 1 additional mathematics ou computer science-maths course among the following:

-ALGEBRA 1 (ALGÈBRE 1)
Olivier Biquard
(Maths course: 12 ECTS.)

-INTEGRATION AND PROBABILITY (INTÉGRATION ET PROBABILITÉS)
Zhan SHI
(Maths course: 12 ECTS.)

- INTRODUCTION TO MODELLING AND NUMERICAL SIMULATION (INITIATION À LA MODÉLISATION ET À LA SIMULATION NUMÉRIQUE)
Erwan Faou-David Lannes
(Maths course: 12 ECTS.)

-RANDOM STRUCTURES AND ALGORITHMS (STRUCTURES ET ALGORITHMES ALÉATOIRES)
Anne Bouillard, Pierre Bremaud
(Computer science course: 6 ECTS.)

-STATISTICS (STATISTIQUE)
Gérard Biau
(Maths course: 12 ECTS.)

-TOPOLOGY AND DIFFERENTIAL CALCULUS (TOPOLOGIE ET CALCUL DIFFERENTIEL)
Patrick Bernard
(Maths course: 12 ECTS.)

For his/her Master M1, the student must validate courses for 30 ECTS during the 1st semester. Besides the maths courses, the student must choose computer science courses in the following list or among the MPRI courses (Parisian Master of Research in Computer Science). It is also possible, with the agreement of the tutor and of the director of studies, to choose courses in other Masters such as for instance the MVA (Mathématiques, Vision, Apprentissage) master of ENS Cachan.

It is also highly recommended to choose and validate one or two additional courses in the following list for the ENS diploma.

ARITHMETIC ALGORITHMS FOR CRYPTOLOGY (ALGORITHMES ARITHMÉTIQUES POUR LA CRYPTOLOGIE)
MPRI course–M2-12-2 (3 ECTS)
François Morain

DISTRIBUTED ALGORITHMS FOR NETWORKS (ALGORITHMIQUE DISTRIBUÉE POUR LES RÉSEAUX)
MPRI course–M2-18-1 (3 ECTS)
Pierre Fraigniaud

ALGORITHMS FOR EMBEDDED GRAPHS (ALGORITHMES POUR LES GRAPHES PLONGÉS)
MPRI course–M2-38-1 (3 ECTS)
Eric Colin de Verdière, Claire Mathieu

CATEGORIES AND LAMBDA-CALCULI (CATÉGORIES, LAMBDA-CALCULS)
MPRI course–M1-20 (6 ECTS)
Paul-André Melliès

ADVANCED COMPLEXITY (COMPLEXITÉ AVANCÉE)
MPRI course–M1-17 (6 ECTS)
Jean Goubault-Larrecq

FOUNDATIONS OF NETWORK MODELS (FONDEMENTS SUR LA MODÉLISATION DES RÉSEAUX)
MPRI course–M2-17-1 (3 ECTS)
François Baccelli, Jean Mairesse

ABSTRACT INTERPRETATION: APPLICATION TO VERIFICATION AND STATIC ANALYSIS (INTERPRÉTATION ABSTRAITE : APPLICATION À LA VÉRIFICATION ET À L'ANALYSE STATIQUE)
MPRI course–M2-6 (6 ECTS)
Patrick Cousot, Radhia Cousot

INTRODUCTION TO COMPUTER VISION (INTRODUCTION À LA VISION ARTIFICIELLE)
Computer science course (4 ECTS)
Jean Ponce

MATHEMATICAL METHODS FOR NEUROSCIENCE (MÉTHODES MATHÉMATIQUES POUR LES NEUROSCIENCES)
UPMC Maths et Applications & MVA &Computer science course (4 ECTS)
Olivier Faugeras

ROBOT MOTION PLANNING: COMBINATORIAL ISSUES VIA CONTROL THEORY (PLANFICATION DE MOUVEMENT EN ROBOTIQUE ET EN ANIMATION GRAPHIQUE : DU CONTINU AU COMBINATOIRE VIA LA COMMANDABILITÉ DES SYSTÈMES)
Computer science course (6 ECTS)
Jean-Paul Laumond

CRYPTOGRAPHIC PROTOCOLS: FORMAL AND COMPUTATIONAL PROOFS (PROTOCOLES CRYPTOGRAPHIQUES : PREUVES FORMELLES ET CALCULATOIRES)
MPRI course–M2-30 (6 ECTS)
Hubert Comon-Lundh, David Pointcheval

OBJECT RECOGNITION IN COMPUTER VISION (RECONNAISSANCE D'OBJETS ET VISION ARTIFICIELLE)
MVA &Computer science course (4 ECTS)
Ivan Laptev, Josef Sivic, Cordelia Schmid

SEMANTICS, LANGUAGES AND ALGORITHMS FOR MULTICORE PROGRAMMING
(SÉMANTIQUE, LANGAGES ET ALGORITHMES POUR LA PROGRAMMATION MULTICORE)
MPRI course–M2-37-1 (3 ECTS)
Albert Cohen

SYNCHRONOUS SYSTEMS (SYSTÈMES SYNCHRONES)
MPRI course–M2-23-1 (3 ECTS)
Marc Pouzet, Jean Vuillemin

TECHNIQUES IN CRYPTOGRAPHY AND CRYPTANALYSIS
MPRI COURSE–M2-12-1 (3 ECTS)
Michel Abdalla, Phong Nguyen, Vadim Lyubashevsky

7.2.2 Second semester

The student must do an internship abroad lasting about 5 months to validate 30 ECTS for his Master M1.

7.3. Third year: Masters (M2)

7.3.1 First semester

The student must validate MPRI M2 courses for 30 ECTS. It is possible to validate courses from another academic programme, with the agreement of his/her tutor and of Damien VERGNAUD who represents the ENS on the MPRI studies committee.

Instead of the MPRI M2, the student may prepare and validate the MVA (Mathématiques, Vision, Apprentissage) Master M2 of ENS Cachan with the agreement of his/her tutor and of the Computer Science director of studies.

The student must also validate additional courses for the ENS Diploma, unless he/she has already validated the 36 ECTS required.

7.3.2 Second semester: internship

For the MPRI M2, the student must do an internship in France or abroad, lasting about 5 months. The internship counts 30 ECTS for the MPRI M2 diploma.

For the MVA M2, the mandatory internship lasts at least 4 months and takes place between April and September.

8 Computer science courses of the ENS Diploma (outside the computer science speciality)

8.1. Computer science as a « secondary speciality » of the ENS Diploma

A student registered in a speciality of the Diploma different from the computer science speciality (and therefore attached to another department of ENS) can choose to validate a coherent set of courses in computer science, which can constitute the secondary speciality of his/her ENS Diploma. This coherent set of courses must represent a total of 24 ECTS units.

8.2. Computer science in the ENS Diploma

The students who already have some knowledge in computer science may choose L3 courses during the first semester.

The computer science department also offers a course in programming, open to all students:

PROGRAMMING FOR THE NON-COMPUTER SCIENTIST
(INITIATION À LA PROGRAMMATION POUR NON-INFORMATICIENS)

(3 ECTS)
(Course taught in the 2nd semester)

Damien Vergnaud

This introductory course is open to students from all Science and Humanity majors. No prior knowledge of computer science is required. The course does not aim towards any specific application of programming but will adapt to the needs of the students. Students that will be required to program software during their research may be interested in taking this course.

Details of the courses for the academic year 2012/2013

Abstract interpretation: application to verification and static analysis (Interprétation abstraite : application à la vérification et à l'analyse statique)

(Patrick Cousot, Radhia Cousot)

The static analysis of programs consists in verifying statically (without execution) dynamic program properties (at runtime).

The classes of properties to be verified are diversified ranging from safety (for example the absence of runtime errors), liveness (such as the guarantee of response to a signal) and security (for example, the confidentiality of information handled by a program).

The major difficulty to automatically prove these dynamic properties is to find inductive arguments to make the proof (for example, by induction on the number of program elementary steps). Various solutions can be considered: asking the end-user (deductive methods), using finitary models (model-checking) or compute the inductive argument by approximation of the program semantics (using the fixpoint approximation techniques of abstract interpretation).

The course explores this last technique, first recalling the bases, then exploring a number of infinitary abstractions so as to handle a great number of applications to the analysis of infinite state systems, whether emerging, classical or industrialized.

Contents of the course:

- Introduction to abstract interpretation ;
- Numerical abstract domains ;
- Symbolic abstract domains;
- Combination and refinement of abstract domains;
- Design of an abstract interpreter by abstract interpretation;
- Static analysis of sequential, procedural, recursive and modular programs;
- Probabilistic abstract domains;
- Static analysis of asynchronous parallel programs;
- Static analysis of mobile code;
- Verification by parameterized abstraction of predicates;
- Static verifications on real-time, safety-critical embedded control-command software;
- Practical and theoretical opened problems in static analysis by abstract interpretation, perspectives.

(MPRI course : 6 ECTS)

Please see description updates and more details of this MPRI M2-6 course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-6

Advanced Complexity (Complexité avancée)

(Jean Goubault-Larrecq)

Complexity theory goes well beyond NP-completeness. The aim of this course is to have a look at several other fundamental complexity-theoretic constructions: space complexity, alternating machines, randomized machines. We shall see a few fascinating theorems: that alternating time is equivalent to deterministic space for example, or Shamir's IP=PSPACE theorem.

Outline:

- The polynomial hierarchy, alternating macines, PSPACE. QBF is PSPACE-complete.
- Alternating complexity classes, games. The theorems by Chandra-Kozen-Stockmeyer: AL=P, AP=PSPACE. Log-space reductions. HORNSAT is P-complete.
- Directed graph reachability is NL-complete. The Immerman-Szelepcsényi theorem: non-reachability in directed graphs is also NL-complete. So NL=coNL.
- Randomized complexity classes: RP, coRP, BPP, ZPP. Reducing the error. The P/poly class. The Bennett-Gill theorem: BPP is included in P/poly. The Karp-Lipton theorem: if NP is included in BPP then PH collapses at the second level. Sipser and Gács' theorem: BPP is at the second level of the polynomial hierarchy.
- Arthur vs Merlin games. The classes MA and AM. Babai's theorem: MA is included in AM, the Arthur vs Merlin hierarchy collapses. BP . NP = AM. Arthur vs Merlin games through alternation between the E and the existential quantifiers. Interactive proofs. GRAPH-NON-ISOMORPHISM is in IP [1]. AM is at the second level of the polynomial hierarchy. The theorem of Goldwasser-Sipser: IP [k] is included in AM [k+1]. The Boppana-Håstad-Zachos theorem: if coNP is included in AM then PH collapses at the second level. Consequence for GRAPH-NON-ISOMORPHISM.
- Universal hashing techniques, GRAPH-NON-ISOMORPHISM is in AM (direct proof). The error can be made zero in case x is in L, for every language L in AM. AM is at the second level of the polynomial hierarchy.
- Classes with a polynomial number of rounds: ABPP, IP. Shamir's theorem: ABPP=IP=PSPACE.
- (optional) Fagin's theorem: the NP-complete problems on graphs are those definable by existential second-order formulae.
- Approximation problems. The approximation thresholds of NODE COVER, TSP, KNAPSACK, MAXSAT. The Arora-Safra theorem: NP=PCP(O (logn), O (1)) (without proof). Equivalence of the Arora-Safra theorem with the fact that MAX3SAT cannot be approximated.

(MPRI course : 6 ECTS)
Please see description updates of thisMPRI 1-17 course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-17

Algebra 1 (Algèbre 1)

(Olivier Biquard)

1) Groups, symmetric groups, action of groups on sets.
   Normal subgroups and quotients. Semi-direct products of groups and extensions of

groups.
The group of invertible elements of a cyclic group with some arithmetic applications.
2) Groups and geometry, linear groups, orthogonal groups, classical groups.
Quadratic forms, Hermitian forms and alternating forms.
3) Multilinear algebra: tensor product, tensor algebra, symmetric algebra, exterior algebra.
4) Introduction to representation theory. Characters

(Maths course : 12 ECTS)

Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=41#

Algebra 2 (Algèbre 2)

(Olivier Debarrre)

(Maths course : 12 ECTS)
Please see description this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=46

Algorithms for embedded graphs (Algorithmes pour les graphes plongés)

(Eric Colin de Verdière, Claire Mathieu)

The course revolves around exact and approximation algorithms for embedded graphs: planar graphs and graphs drawn without crossings on surfaces. It is at the frontier of “classical” graph algorithms and computational geometry, and combines several directions of active research that share common background and techniques:

- exact algorithms for planar graphs;
- approximation algorithms for planar graphs;
- algorithms for graphs embedded on surfaces, using topological methods.

(MPRI course : 3 ECTS)
Please see description update and more details of this course MPRI 2-38-1 at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-38-1

Algorithms and programming (Algorithmique et programmation)

(Jacques Stern, Claire Mathieu, Maribel Fernandez et Damien Vergnaud)

This course deals simultaneously with the fundamentals of data structures and the principles of algorithm design, together with some more advanced topics. Students are expected to have had a minimal exposure to algorithms. Each lecture is divided into two parts, the first devoted to basic knowledge and the second to one more advanced result (or more exceptionally).

Algorithms: design and evaluation.

- basic course: termination, complexity, programming strategies,

- advanced course: bin packing, dynamic memory allocation.

First part: data structures

- Sorting and hashing
  - basic course: examples of sorting algorithms, hashing, collisions, open hashing,
  - advanced course: sorting networks.
- Pattern matching
  - basic course: Rabin-Karp, Knuth-Morris-Pratt,
  - advanced course: algorithms for biocomputing.
- Trees
  - basic course: search trees, examples,
  - advanced course: binomial heaps, Fibonacci heaps.
- Graphs
  - basic course: transitive closure, connected components, shortest paths.
  - advanced course: eigenvalues and expansion graph.
- Flows
  - basic course: Ford-Fulkerson, Edmonds-Karp,
  - advanced course: unit flows, Dinic, matching algorithms.
- Reductions
  - basic course: introduction to P, NP, NP-completeness,
    advanced course: Proofs of NP-completeness by reductions.

Second part: numerical and symbolic algorithms

- Integers
  - basic course: multiplication, exponentiation,
  - advanced course: primality tests.
- Fast Fourier Transform
  - basic course: FFT, complexity,
  - advanced course: fast multiplication.
- Linear programming
  - basic course: simplex, complexity,
  - advanced course: the ellipsoid algorithm.
- Linear algebra and geometry of numbers
  - basic course: LUP decomposition, least squares,
  - advanced course: lattices, the LLL algorithm.
- Polynomial factorisation
  - basic course: polynomials with integer coefficients, gcd, binary polynomials,
  - advanced course: the algorithms of Berlekamp and Cantor-Zassenhaus.
- Systems of polynomial equations
  - basic course : basic standard algorithms,
  - advanced course: exp-space complexity.

(Computer science course: 6 ECTS)

Web page of the course in 2011-2012: http://www.di.ens.fr/~bouillaguet/teaching.html

Arithmetic algorithms for cryptology (Algorithmes arithmétiques pour la cryptologie)

(François Morain)

The goal of this course is to present the concepts and tools of modern public-key cryptology, whose mathematical building-blocks include finite fields and algebraic curves. This course aims to present algorithmic number theory, together with classical number theory and complexity theory, with a view to applications in cryptology. It forms a coherent whole with the other MPRI courses involving cryptography, such as 2-30, 2-13-1 and 2-13-2.

Prerequisites

Specific Prerequisites

We expect students to have already followed an introductory course in cryptology. The general principles of cryptology (integrity, authenticity, confidentiality) must be known, as well as some artisanal techniques (eg. Vigénère). We will assume that the students have already acquired some notions about PKI, secure channels (with an example such as SSL), and about Proofs of Knowledge.

In Number Theory, the basic results on finite fields, on integer arithmetics and on modular integers should be known, and also some results on polynomial arithmetics (Belekamp algorithm).

Students should have seen examples of symmetric cryptographic primitives: hash functions, block ciphers (DES, AES), stream ciphers (RC4, A5), modes of operation (ECB, CBC). Classical assymetric primitives (RSA and Diffie-Hellman) should also be known.

Note: No specific knowledge of algebraic curves or cryptanalysis is necessary.

General Prerequisites

The prerequisites are not specific to cryptology and are essentially included in the general list.

Notions of complexity classes, Turing machines, and NP problems. Some knowledge in algebra and probability. Basic algorithmic tools.

(MPRI 2-12-2 course : 3 ECTS)
Please see description updates of this course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-12-2

Categories and lambda-calculi (Catégories, Lambda-calculs)

(Paul-André Melliès)

This course is concerned with the syntax and semantics of programming languages, starting from the lambda-calculus. This formalism, which was introduced in logic in the 1930's, has met computer science in the 60's, when it was used for the formal specification of programming languages such as ALGOL, or for the design of languages like LISP, Scheme, or CAML. The lambda-calculus offers a rich dictionary of correspondences between programming and logic: proofs and programs, types and formulas or specifications.

We shall prove the main syntactical theorems of the lambda-calculus: confluence, standardisation, termination. Then we shall present the models of the lambda-calculus: to this aim, the language of category theory will be used.

Interpreting a language in a model is akin to a compilation, and as a matter of fact models offer occasions to return to syntax: abstract machines for program execution, proofs of properties of programs. Similarly, observations on a particular model of the lambda-calculus have lead Girard to linear logic, which has connectives allowing to control hypotheses viewed as resources.
We shall also discuss extensions of the lambda-calculus with imperative features such as references and exceptions, both at a syntactical and semantical level. Here too, categories provide suitable abstractions such as monads, which allow for a modular and uniform view of different notions of effects in programming.
Finally, links with concurrency theory (pi-calculus, join calculus) will be sketched.

Recommended reading for the course:
Domains and Lambda-calculi. R. Amadio et P.-L. Curien. Cambridge University Press,1998.
And also:
- Semantics of programming languages. C. Gunter. MIT Press, 1992.
- Categories, types and structures. A. Asperti and G. Longo. MIT Press, 1991 (sold out, but available on the web page of G. Longo (di.ens.fr)).
- Theories of programming languages. J. Reynolds. Cambridge University Press, 1992.

For the lambda-calculus :
- The Lambda-calculus. H. Barendregt. North Holland, 1984.
- Lambda-calcul, types et modèles. J.-L. Krivine. Masson, 1990.

For category theory, read the first chapters of a book such that:
- Toposes, Triples and Theories. M. Barr and C. Wells. Springer, 1985.
- Sheaves in Geometry and Logic: a first introduction to topos theory. S. Mac Lane and Ieke Moerdjik. Springer, 1992.

(MPRI 1-20 course : 6 ECTS)
Please see more information about this course at:
http://www.pps.univ-paris-diderot.fr/~mellies/mpri-ens.html

Communication networks (Réseaux de Communication)

(Anna Busic)

This course is an introduction to communication networks, which consists of:

1. Guided reading on network architectures and protocols;
2. A course on the basis of discrete event simulation;
3. The realization of a discrete event simulator and the analysis of a communication network based on the simulator.

I. Reading. This will be focused on the following topics:

• Multiple access in local area networks (book: Multiple Access Protocols,R. Rom et M. Sidi, Springer, 1990).
  – wireline networks: Aloha, Ethernet, tree protocols, TDMA, CSMA;
  – wireless networks: 802.11, spatial Aloha;
  – stability, throughput maximization.

- Congestion control (book: An Engineering Approach to Computer Networking, Addison-Wesley, 1997.)
  - TCP and its variants (Reno, Tahoe, Vegas);
  - Bandwidth sharing and resource allocation, max-min fairness, proportional fairness.

- IP routing (Book: An Engineering Approach to Computer Networking, Addison-Wesley, 1997.)
  - Shortest paths and Dijkstra’s algorithm;
  - BGP;
  - Routing in mobile networks.

II. Course on network simulation. The following topics will be covered:

- Random variable generation;
- Matthes’ schemas, event tables and simulation;
- Confidence intervals;
- Perfect simulation.

III. Projects. A list of simulation projects will be proposed. Here are some typical examples: instability of multiclass networks; perfect simulation of CSMA networks; interaction of TCP flows; epidemic diffusion and caching in peer to peer networks; random mobile networks.

Requirements. A first probability course.

(Computer science course : 6 ECTS)

Complex and harmonic analysis (Analyse Complexe et Harmonique)

(Wendelin Werner)

- Harmonic functions, conjugate harmonic functions, discrete holomorphic functions.
- Holomorphic functions, Cauchy formula and its (numerous) applications, analytic functions.
- Conformal transformations: Riemann's Theorem, examples, Poincaré metric.
- Meromorphic functions, factorization of entire functions, Hadamard's Theorem.
- Some results on the Gamma function, on the Riemann zeta function and on elliptic functions.

Ref.
E.M. Stein, R. Shakarchi, Complex Analysis, Princeton University Press,
L.V. Ahlfors, Complex Analysis, 3rd Ed. Mc Graw Hill

(Maths course : 12 ECTS)

Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=45#

Computer science logic (Loqique et Informatique)

(Jean Goubault-Larrecq)

This course explores the basics of the lambda-calculus, a tool invented by the logician Alonzo Church in the 1930s, and which is instrumental today both in semantics of programming languages (computer science) and in proof theory (logic).

- Lambda-calculus and functional languages:
  - Lambda-calculus, operational semantics (reduction).
  - Expressiveness. Fixpoint and recursion combinators.
  - Termination problems, finite developments, confluence and parallel reductions.
  - Reduction strategies: by name, by need. Standardisation.
  - Models of the lambda-calculus, Pomega.
  - Calculi with explicit substitutions, machines. Geometry of interaction (optional)
- Logical aspects:
  - simply-typed lambda-calculus;
  - Curry-Howard correspondence between the latter and proofs in propositional minimal logic;
  - extension to classical logic, capture of continuations and exception handling;
  - second-order typed lambda-calculus: Girard-Reynolds System F, correspondence with second-order intuitionistic logic;
  - strong normalization, cut elimination.

Lecture notes are available at
http://www.lsv.ens-cachan.fr/~goubault/Lambda/loginfoindex.html

Bibliography:

- Jean-Louis Krivine. Lambda-calcul, types et modèles. Masson, 1992.
- Jean-Yves Girard, Yves Lafont & Paul Taylor. Proofs and Types. Cambridge University Press 1989.
- Christian Queinnec. Les langages Lisp. InterÉditions, 1994.

(Computer science course : 6 ECTS)
This highly recommended course takes place at ENS Cachan in the 2nd semester.

Cryptographic protocols: formal and computational proofs (Protocoles cryptographiques : preuves formelles et calculatoires)

(Hubert Comon-Lundh, David Pointcheval)

Cryptographic protocols are distributed programs which aim at securing communications and transactions by the means of cryptographic primitives. The design of cryptographic protocols is difficult: numerous errors have been discovered in protocols after their publication. It is therefore particularly important to be able to obtain proofs that protocols are secure.

Two models of the protocols have been considered: the formal model and the computational model. We shall present these two models, the associated proof techniques, and results that relate them.

This course will be an opportunity to adapt and use formal tools, such as process calculi, semantics, and logic to the particular case of the study of cryptographic protocols.

(MPRI course : 6 ECTS)
Please see the bibliography and more details of this MPRI 2-30 course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-30

Databases (Bases de données)

(Serge Abiteboul)

1. Introduction: Databases and DBMS
2. Relational model: relational algebra and relational calculus, equivalence theorem
3. Languages used in practice, SQL
4. File management, access structures: B-trees, hashing
5. Query optimization
6. Concurrency and transactions: Serializability, two-phase locking, timestamping and failures
7. Distributed data management
8. Integrity constraints

This course takes place at ENS Cachan

(Computer science course : 6 ECTS)
Web page of the course at : http://abiteboul.com/2011/DBCOURSE/

Digital system: from algorithm to circuit (Système digital : de l'algorithme au circuit)

(Jean Vuillemin)

The ***course lectures*** present the ***hardware*** component in our information world. From the principles for designing and realizing ***circuits***, to various high-performance computing applications, from physics, electronics, algebra and telecommunications. Each application goes from the algorithm (software) to the circuit (hardware): same operations, different performances.

The ***practical part*** is a project, to be realized in small groups: each group must entirely design a ***microprocessor***, and realize it with elementary logic gates; one then simulates the gate network in action, and program the micro so as to turn it into a *digital clock*, and simulated in real-time.

1. ***Digital Synchronous Circuit***: Combinational circuit, logic gates. Clocked register; digital synchronous circuit; *Digital clock* example. Complexity and BDD circuit synthesis.
2. ***Binary Numbers***: from bits to 2adic integers; Boolean algebra and Ring; Hyper-cube and integer sets. 2adic arithmetic and bit-serial circuits. Logic and set operations over binary integers: Binary Algebra.
3. ***Electronic Circuits***: from gates to transistors; from logic to layout design rules. Electric schemas and micron drawing of a serial adder. RAM and RAM memory structures. Fabrication technologies; Moore's laws.
4. ***Silicon Arithmetic***: adders and multipliers; serial and parallel; minimal depth. Optimal area/time tradeoffs. Arithmetic and Logic Unit. Hensel's division; 2adic square root.
5. ***Universal Machines***: silicon Turing machine; Church programmable micro-processor. Computable real numbers, and limits to automatic computations. On-line arithmetic: 2adic vs. real numbers. Programmable logic FPGA and dynamically reconfigurable systems.
6. ***Digital Physics***: algorithm (Fast Hough Transform) and circuit realization to identify straight lines in high-frequency digital images from the ATLAS detector in LHC; realization of a circuit for simulating the heat flows in a running VLSI, by solving the Heat Equation through a software controlled massively parallel circuit co-processor.
7. ***Telecommunications***: Introduction to Shannon's theory, source and canal. Data entropy: Huffman's algorithm, LZW entropic compression. Error control, noise entropy; Hamming code; Viterbi codes.
8. ***Audio and Video***: A/D and D/A conversions; speed/resolution tradeoffs. Capture, codes and transmissions of digital images; compression within Just Noticeable Distortion: fixed images JPEG and MPEG video. MP3 code and sound transmission.

(Computer science course: 6 ECTS)
More information about this course at :
http://www.di.ens.fr/~jv/HomePage/teaching.html

Distributed algorithms for the networks (Algorithmique distribuée pour les réseaux)

(Pierre Fraigniaud, Laurent Viennot)

L'***algorithmique distribuée*** consiste à concevoir et analyser des algorithmes dédiés à un ensemble d'entités autonomes dont l'action conjointe doit contribuer à la réalisation d'une tâche commune.

Le champ d'applications de l'algorithmique distribuée est si vaste qu'il serait vain d'en proposer une liste exhaustive. A eux seuls, le domaine des réseaux (Internet bien sûr, mais aussi les systèmes pair-à-pair, les réseaux sociaux, les réseaux sans fil, les réseaux mobiles, etc.) et celui des multi-processeurs (machine multi-coeurs, grilles de calcul, etc.) fournissent déjà une source immense d'applications potentielles. On peut également citer de nombreux autres cadres d'applications, dont évidemment la biologie avec l'étude de différents systèmes naturellement distribués (nuée d'oiseaux, colonie de fourmis, population de bactéries, etc.).

Ce cours, couplé au cours 2.18-2, a pour objectif de fournir les bases essentielles à la conception, la compréhension, le contrôle, et l'analyse de systèmes tels que ceux listés ci-dessus.

(MPRI course : MPRI 2-18-1 : 3 ECTS)
Please see description updates of this course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-18-1

Formal languages, computability and complexity (Langages formels, calculabilité et complexité)

(Eugène Asarin)

1: Regular languages, their properties and their characterization by automata, regular expressions, logical formulae and monoids. Star-free languages.

Introduction to languages of infinite words.

2: Grammars and Chomski hierarchy. Context-free languages, their properties and their characterization by pushdown automata.

3: Computability (recursive functions and Turing Machines). Decidable, undecidable, semi-decidable problems.

4: Time and space complexity. Complexity bounds. Complexity classes (NP, Pspace) and complete problems.

Course web page of Eugene Asarin:

http://www.liafa.univ-paris-diderot.fr/~asarin/ENS/lf.html

Manual for this course (in French):

http://www.liafa.univ-paris-diderot.fr/~carton/Lfcc/

Exercises web page of Anne Bouillard: http://www.di.ens.fr/~bouillar/enseignement.html

Previous page of this course:

http://www.liafa.univ-paris-diderot.fr/~carton/Enseignement/Complexite/ENS/

(Computer science course : 6 ECTS)

Foundations of network models (Fondements sur la modélisation des réseaux)

(François Baccelli, Alain Jean-Marie, Jean Mairesse)

Le but de ce cours est double :

* proposer des modèles mathématiques pertinents pour les réseaux de communications;

* donner les bases théoriques permettant de mener a bien l'analyse de la dynamique de ces modèles.

Le cours est structuré en thèmes, pouvant être plus ou moins développés suivant les années.

* Réseaux de files d'attente et modélisation markovienne (réseaux à commutation de paquets, réseaux à commutation de circuits).

* Dynamique des systèmes à événements discrets temporisés (semi-anneau max plus, inf convolutions, fonctions topicales, réseaux de Petri temporisés, modèles d'empilements de pièces, etc.).

* Contrôle de flux dans les réseaux de communication (TCP, contrôle de flux et de congestion, régulation, network calculus, ordonnancement etc.).

* Graphes aléatoires (à la Erdos-Renyi, géométriques) et modèles de percolation.

(MPRI course : 3 ECTS)

MPRI page of this course: https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-17-1

Geometric Foundations of Computer Science (Bases Géométriques de l'Informatique)

(Michel Pocchiola and Jean Ponce)

This course introduces the geometric and algorithmic foundations of computer science fields where geometry plays a fundamental role, in particular computational geometry and computer vision. The first part of thecourse is concerned with discrete geometry and the objects, techniques and applications of computational geometry. It focuses in particular on convex polyhedra, hyperplane arrangements, and randomized techniques. The second part of the course gives a concrete presentation of elementary notions of projective geometry and differential geometry, and their application to modeling camera systems in computer vision.

1. Cones, polyhedra and polytopes, face lattices, cyclic polytopes and the upper bound theorem, Voronoi diagrams, algorithms and applications.
2. Arrangements of hyperplanes, Clarkson's theorem on levels, zone theorem, cuttings, algorithms and applications.
3. Geometric hypergraphs, Vapnik-Chervonenkis dimension, epsilon-nets, algorithms and applications.
4. Euclidean, affine, and projective cameras: central perspective and parallel projection; elements of affine and projective geometry; projection and inverse projection of points and lines.

5. Multi-view geometry: epipolar geometry; trifocal tensor; projective calibration; affine and projective structure from motion; Euclidean calibration: Chasles's absolute conic and its relatives.

6. Smooth Euclidean surfaces and their outlines: elements of descriptive differential geometry; Koenderink's theorem; aspect graphs.

References

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. Computational Geometry: Algorithms and Applications. Springer-Verlag, Berlin, Germany, 2nd edition, 2000.
[2] D.A. Forsyth and J. Ponce. Computer Vision: A Modern Approach. Prentice Hall, 2003.
[3] J. E. Goodman and J. O'Rourke, editors. Handbook of Discrete and Computational Geometry. CRC Press, 2nd edition, 2004.
[4] J. Matou?sek. Lectures on Discrete Geometry. Number 212 in Graduate texts in Mathematics. Springer-Verlag, 2002.
[5] K. Mulmuley. Computational Geometry: An Introduction Through Randomized Algorithms. Prentice Hall, Englewood Cliffs, NJ, 1994.

(Computer science course: 6 ECTS)

Please see description updates of this course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-11

Information Theory and Coding (Théorie de l'information et codage)

(Marc Lelarge)

- Basics:

Entropy, mutual information, typical sequences, Fano's inequality.

- Data compression:

Optimal coding, Kraft's inequality, Huffman, Ziv-Lempel, distorsion theory

- Capacity of digital channels:

Shannon's theorem.

- Error correcting codes:

linear codes, cyclic codes, Hamming codes, BCH, Reed-Solomon codes.

References:

- R.J. McEliece, The Theory of Information and Coding, 1982.
- T. Cover, J. Thomas, Elements of Information Theory, Wiley, 1991.
- C. Shannon, A Mathematical Theory of Communication, 1948.
  http://cm.belllabs.com/cm/ms/what/shannonday/shannon1948.pdf

(Computer science course: 6 ECTS)

Webpage of the course: http://www.di.ens.fr/~lelarge/info.html

L'Informatique scientifique par la pratique

(David Naccache)

Ce cours aborde les disciplines scientifiques liées au traitement automatique de l'information à travers la microprogrammation et la mise en oeuvre optimisée.
L'objectif pédagogique du cours est double :

1. Initier les étudiants aux technologies de conception proches de la machine: matériel et microprogramme.
   Initiation puis maitrise de nouveaux outils et langages assembleur.
2. Ce faisant familiariser les étudiants avec certains algorithmes communs intervenant en informatique scientifique : compression, ramasse-miettes, hachage géométrique, correction d'erreurs, arithmétique entière, arithmétique en virgule flottante, solution par retraits itérés (backtracking), inférence de type, FFT et déconvolution. Elargissement, par la pratique, de la “culture informatique scientifique” de l'étudiant.

Les étudiants affronteront les contraintes rencontrées lors de programmation de ces algorithmes (e.g. taille de code, complexités mémoire et temps etc) ainsi que les différentes techniques permettant d'adapter les algorithmes à des architectures (e.g. mise en uvre en bit-slice, calcul partagé, évaluation paresseuse etc). A titre de projet les étudiants pourraient mettre en oeuvre de manière optimisée des algorithmes vus lors ou du cours ou lors d'autres cours dispensés dans le cadre de la formation interuniversitaire en informatique de l'ENS (traitement des images, optimisation de circuits électroniques, cryptographie, compilation etc).

1. Le microprocesseur 68HC05

- Présentation des outils de programmation en assembleur 68HC05. Architecture, Ports, registres, mémoires, ALU.
- TD : conception d'une librairie de calcul sur les nombres flottants en assembleur. Changement de la saturation des points d'une photographie du portique de l'ENS en utilisant une multiplication à virgule flottante arrondie.

2. La compression

- Introduction : Théorie de Shannon. Compression entropique. Compression par les méthodes RLE, LZW et Huffman.
- TD : mise en uvre de RLE, LZW et Huffman sur le 68HC05. Compression de la photographie du portique.

3. La correction d'erreurs

- Introduction : Contrôle des erreurs par codage algébrique. Codes de Viterbi. Turbo codes.
- TD : codeur / décodeur pour les codes de Hamming et de Reed-Solomon sur le 68HC05. Protection par code correcteur de l'image du portique. Bruitage de l'image à différents SNRs, affichage et correction.

4. Arithmétique sur les grands nombres : multiplication et réduction modulaire

- Introduction : Multiplication multi-précision et algorithmes de réduction de Montgomery et de Barrett. Preuve des deux algorithmes et leur analyse. Techniques de multiplication rapide (Solovay-Strassen) et multiplication à faible nombre de changements d'état.
- TD : Multiplication multi-précision et réduction modulaire Montgomery et Barrett sur le 68HC05. Codage de fonctions de calcul RNS (Residue Number System).

5. La cryptographie par la pratique.
   - Introduction : Signature et chiffrement, présentation et/ou rappel de RSA.
   - TD codage d'un chiffrement RSA et d'une vérification de signature à l'aide de la librairie de calcul sur les grands nombres codée lors du TD précédent. Vérification d'une signature numérique créée sur Mathématica sur l'image du portique.

6. Ramasse-miettes par la pratique.
   - Introduction au ramassage de miettes (garbage collection)
   - TD : Ramasse-miettes par la méthode mark & sweep. Découpage de l'image du portique en morceaux et assemblage des morceaux en un chemin de moindres mouvements à l'aide du programme de mark & sweep.

7. La machine de Minsky
   - Architecture.
   - TD : codage d'un simulateur de machine de Minsky et mise en uvre d'un programme très simple sur ce simulateur.

8. La technique des retraits itérés
   - Présentation de la technique des retraits itérés.
   - TD : Résolution du problème de lasermaze par la technique des retraits itérés. Détection du plus long chemin de reflets dans un extrait de l'image du portique par la méthode des retraits itérés.

9. La transformée de Hough et le hachage géométrique.
   - Le problème de la détection de droites, présentation des algorithmes.
   - TD : Codage en assembleur de la transformée de Hough. Extraction d'une droite de l'image du portique.

10. FFT et déconvolution. Régularisation de Tikhonov
    - TD : Codage en assembleur d'une routine de convolution et de déconvolution. Convolution de l'image du portique avec un filtre gaussien et déconvolution. Essais avec différents SNRs.

Séance finale :

- Exposition des résultats de tous les TDs du module (avec des posters et démonstrations) à l'intention des élèves et chercheurs internes et externes.

(Computer science course: 6 ECTS)

Initiation to cryptology (Initiation à la cryptologie)

(Jacques Stern, Damien Vergnaud)

This course is aimed at students interested in mathematical and practical aspects of algorithmics. Its goal is to teach the fundamentals of cryptology and the main tools that are used to solve security problems. The course is also proposed as a level 1 course for the MPRI and therefore serves as a preparation for the level 2 course in cryptology of the MPRI.

This course consists in 6 relatively independent parts.

- Introduction to cryptography
  - Permutations, substitutions, cryptanalysis (types of attacks).
  - Integrity, confidentiality, authenticity. One Time Pad.
- Symmetric cryptography
  - Stream ciphers.
  - Block ciphers.
  - Modes of operation (CBC, ECB, CTR).
  - Examples: DES, AES, RC4, A5/1.
  - Hash, MAC.
- Algorithmic techniques
  - Algorithmic of integers.
  - Modular arithmetic.
  - Finite fields.
- Asymmetric cryptography
  - RSA, Diffie-Hellman, El Gamal.
  - Multiplicativity of RSA (Hastad, attacks based on multiplicativity).
  - One-way Functions, trapdoors.
  - Pseudo random generators.
  - RSA signatures, El Gamal.
- Protocols
  - Introduction to Zero-Knowledge proofs.
  - Identification, signatures (FS, Schnorr).
- Applications
  - PKI, IPSEC.
  - Secure channel: SSL.

(Computer science course: 6 ECTS)
See more information about this course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-1-13

Integration and probability (Intégration et probabilities)

(Zhan SHI)

I. Integration

1. Measurable spaces
2. Integration with respect to a measure
3. Construction of measures
4. L_p spaces
5. Product measures
6. Signed measures
7. Change of variables formula

II. Probability

1. Foundations of probability theory

2. Independence
3. Convergence of random variables

(Maths course: 12 ECTS)
Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=42

Introduction to computer vision (Introduction à la vision artificielle)

(Jean Ponce)

This class presents an overview of the principles and techniques underlying computer vision, a scientific field whose aim is to equip computers with the ability to make sense of digital imagery (still images and video sequences).
The class will include programming assignments in Matlab/Scilab.

Outline:

1. Image formation: Camera models, light and color.
2. Local image processing: Filters, edge detection, features, texture.
3. Grouping: Clustering, fitting, and Segmentation.
4. Combining multiple images: Multi-view geometry, stereo, structure from motion.
5. Scene understanding: Face detection and recognition, bags of features for category-level object recognition.

Bibliography:

D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach",
Prentice-Hall, 2002.New course in 2010-2011. Summary to be communicated.
(Computer science course: 4 ECTS)

Introduction to modelling and numerical simulation (Initiation à la modélisation et à la simulation numérique)

(Erwan Faou - David Lannes)

The goal of this course is to present several partial differential equations coming from physics and to explain how mathematical analysis leads to numerical methods allowing their resolution through numerical computations.
Required: Basis of differential calculus, integration, functional analysis.

Program:
- Reminder on ODEs and their numerical approximation (Euler scheme, splitting methods, higher order schemes, stability and consistency, ...)

- Hamiltonian equations: symplectic schemes, numerical conservation of energy. Introduction to highly oscillating systems and infinite dimension systems.
- Spectral and pseudo-spectral methods. Their importance for the resolution of nonlocal pseudodifferentoal equations (FFT method). Curse of dimensionality and sparse grids.
- Nonlinear dispersive equations (examples of KdV and NLS). Hamiltonian structure. Solitary waves. Analysis of splitting schemes.
- Diffusion phenomena, methods of finite differences (stability analysis, explicit and implicit schemes, convection-diffusion...). Comparison with probabilistic methods
.- Hyperbolic systems with for instance the Saint-Venant equations (finite volumes, well-balances schemes, hydrostatic reconstruction).

(Maths Course : 12 ECTS)
Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=58

Logic (Logique)

(Martin Hils)

1) Naïve Set Theory
- Cantor-Bernstein Theorem
- Ordinals and Cardinals
- The Axiom of Choice and its equivalent forms

2) Model Theory
- Languages, structures, formulas, theories, models
- Syntax
- Completeness and Compacity Theorems
- Löwenheim-Skolem Theorems
- Quantifier elimination
- An application to algebra: Ax's Theorem on injective polynomial functions

3) Recursivity, Indicidability, Incompleteness
- Recursive Functions
- Peano arithmetics, indecidability of arithmetics
- Gödel incompleteness Theorems

4) Back to Set Theory
- Zermelo-Frankel Axioms
- Models of Set Theory and the continuous hypothesis

(Maths course: 12 ECTS)
Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=44

Mathematical methods for neuroscience (Méthodes mathématiques pour les neurosciences)

(Olivier Faugeras)

We present a number of mathematical tools that are central to modeling in neuroscience. The prerequisites to the course are a good knowledge of differential calculus and probability theory from the viewpoint of measure theory. The thrust of the lectures is to show the applicability to neuroscience of the mathematical concepts without giving up mathematical rigor. The concepts presented in the lectures will be illustrated by exercise sessions.

- Mesoscopic models of visual cortical areas: anatomical structure of the visual cortex (V1), functional architecture of V1, neural fields models.
- Introduction to dynamic systems: orbits and phase portraits, invariant manifolds, equivalence of dynamic systems, topological classification of equilibria, structural stability, center manifold in finite dimension.
- Introduction to bifurcation theory: dimension 1 (saddle-node, transcritical, pitchfork), dimension 2 (Hopf), center manifold, normal form, equivariant bifurcations.
- Applications: ring model of orientations, Turing mechanism for cortical pattern formation, geometric visual hallucinations.
- Neuronal models: aspatial Hodgkin-Huxley model, simplified models, synaptic models, spatial models.
- Importance of noise: Brownian motion, stochastic differential equations, application to neurons.
- Mean field models: theory of Sompolinsky-ben Arous-Guionnet of spin glasses, application to spiking neuronal models, theory of Mc-Kean-Tanaka-Sznitman of interacting particules, application to models of neurons with action potential , application to neural masses.

Web page of the 2011-2012 course :
http://www-sop.inria.fr/members/Olivier.Faugeras/MMN/MMN11
Former webpage:
http://www-sop.inria.fr/members/Olivier.Faugeras/MVA/MMN10

References:

- Wulfram Gerstner et W. Kistler, Spiking neuron models, Cambridge University Press, 2002.
- Yuri A. Kuznetsov, Elements of applied bifurcation theory.
- Eugene Izhikevich, Dynamical systems in neuroscience: the geometry of excitability and bursting, MIT Press, 2006.
- Jean-Pierre Françoise, Oscillations en biologie, Springer, 2000.
- Lawrence C. Evans, An introduction to stochastic differential equations, http://math.berkeley.edu/~evans/SDE.course.pdf
- Jean-François Le Gall, Mouvement brownien et calcul stochastique, http://www.dma.ens.fr/~legall/DEA96.pdf
- G. Bard Ermentrout et D. H. Terman, Mathematical Foundations of Neuroscience
- Sylvie Benzoni, Cours de M1 sur les EDOs, http://math.univ-lyon1.fr/~benzoni/

(UPMC Maths et Applications & MVA &Computer science course : 4 ECTS)

Object recognition in computer vision (Reconnaissance d'objets et vision artificielle)

(Ivan Laptev, Jean Ponce, Josef Sivic, Cordelia Schmid)

Automated object recognition --and more generally scene analysis—from photographs and videos is the grand challenge of computer vision. This course presents the image, object, and scene models, as well as the methods and algorithms, used today to address this challenge.

Course outline:

- Visual features: interest points, affine regions, invariants, Sift descriptors.
- Detecting specific objects and object classes: 2D and 3D alignment, voting methods, face detection and Adaboost.
- Image classification: bags of features and support vector machines, grids and pyramids, convolutional networks.
- Detecting object categories: constellations of visual features, part-based models, sliding window methods, weakly supervised model learning.
- Going further : scene analysis, understanding human activities in videos.

References:

-D.A. Forsyth and J. Ponce, ``Computer Vision: A Modern Approach'', Prentice-Hall, 2003.

-J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, ``Toward Category-Level Object Recognition'', Lecture Notes in Computer Science 4170, Springer-Verlag, 2007.

(MVA &Computer science course: 4 ECTS)

Operating systems and computer networks (Systèmes et réseaux)

(Marc Pouzet)

Le cours de systèmes présente les concepts fondamentaux des systèmes d'exploitation, leur utilisation et leur mise en œuvre dans un système UNIX.
Ce cours abordera, en autres, les points suivants :

- système de fichiers;
- gestion des processus;
- mémoire virtuelle;
- communication et synchronisation entre processus concurrents (mémoire partagée, signaux, sémaphores, sockets);
- ordonnancement préemptif et non-préemptif; OS temps réel;
- modèles de concurrence de haut niveau.

Les notions introduites seront illustrées par l'écriture d'applications en C ou en Ocaml, en utilisant l'interface POSIX.

La page du cours : http://www.di.ens.fr/~pouzet/cours/systeme/

(Computer science course: 6 ECTS)

Programming languages and compilation (Langages de programmation et compilation)

(Jean-Christophe Filliâtre)

This course is an introduction to the main concepts of programming languages through compilation, that is the translation from a high-level language to machine language. The course includes practical work where some notions are implemented. The evaluation includes a compilation project.

- OCaml programming
- Compilation principles / MIPS architecture
- Abstract syntax / Semantics / Interpreters
- Semantical analysis
  - Simple types
  - Polymorphism / type inference
  - Properties of a type system
- Lexing and parsing
- Compilation of sequential languages
  - Activation records
  - Parameter passing modes
- Compilation of functional languages
  - Functions as first-class values
  - Compilation of pattern-matching
- Compilation of object-oriented languages
  - Static vs dynamic typing
  - Representation of objects
- Garbage collection
- Production of efficient code
  - Intermediate languages RTL, ERTL, LTL
  - Register allocation

(Computer science course: 6 ECTS)

Web page of the course :http://www.lri.fr/~filliatr/ens/compil/

Programming for the Non-computer Scientist (Initiation à la programmation pour non-informaticiens)

(Damien Vergnaud)

This introductory course is open to students from all Science and Humanity majors. No prior knowledge of computer science is required. The course does not aim towards any specific application of programming but will adapt to the needs of the students. Students that will be required to program software during their research may be interested in taking this course.

- Python in command line (calculator, variables, types, …)
- Programming (scripts, conditions, loops, functions)
- Scientific computing with Python (Scipy/Numpy/Pylab)
- Efficient computing: array programming (as in Matlab)

Web page of the course: http://www.di.ens.fr/~vergnaud/initPython.html

(Computer science course taught in the 2nd semester: 3 ECTS)

Random Structures and Algorithms (Structures et Algorithmes Aléatoires)

(Anne Bouillard, Pierre Brémaud)

Objectives: this course aims at presenting the basics of discrete probabilities.

Outline: there will be two parts:

Discrete probabilities and applications:
- random variables, independence and conditioning
- probabilistic method
- random graphs

Markovian models:
- Markov chains and asymptotic behavior
- Monte Carlo sampling, perfect sampling
- Gibbs fields

For each item, examples related to various fields of computer science will be given.

****

Course : Anne Bouillard et Pierre Brémaud, TD Anne Bouillard.
Webpage : http://www.di.ens.fr/~bouillar/SAA/index.html

(Computer science course : 6 ECTS)
(last update : July 2012)

Robot motion planning: combinatorial issues via control theory (planification de mouvement en robotique et en animation graphique : du continu au combinatoire via la commandabilité des systèmes)

(Jean-Paul Laumond)

Motion planning deals with algorithms to compute collision-free paths for a mechanical system (mobile robots, manipulators, virtual beings...) moving amidst obstacles. The approaches consist in exploring the so-called configuration space of the considered system: a configuration is the set of parameters required to locate all the system bodies in the 3D space. Obstacles in the workspace are then tranformed into obstacles in the configurations space. Thanks to this modeling, planning the motion for 3D bodies in the 3D space is equivalent to planning the motion of a point in some non necessarily connected manifold.

- Introduction: motion planning and applications in Robotics, CAD and Graphics.
- A geometrical formulation of the Piano Mover Problem
  - Configuration space searching: three methods to solve the problem of moving a polygon amidst polygonal obstacles
  - Algebraic geometry approach: the problem is decidable.
  - Algebraic topology and the problem of motions in contact.contact.
- Main searching methods

  - Cell decomposition. The exemple of two disks in the plane.
  - Retraction. The exemple of manipulation problem.
- Nonholonomic systems
  - Differential geometry elements (Vector fields, Lie brackets and Froebenius theorem). System controllability. Nonholonomic degree.
  - Optimal path for car-like robots. Nonholonomic motion planning via holonomic path approximation. Complexity.
  - Chained form systems and sinusoidal controls. Flat systems and geometric approaches.
  - The worked-out example of a mobile robot pulling a trailer.
- New approaches via random searches.

Bibliography:

- J.T. Schwarzt, M. Sharir and J. Hopcroft (Eds), Planning, Geometry and Complexity of Robot Motion, Ablex Series in Artificial Intelligence, Ablex Publishing, 1987.
- J.C. Latombe, Robot Motion Planning Kluwer Academic Publishers, 1991.
- J.P. Laumond (Ed), Robot Motion Planning and Control, Lectures Notes in Control and Information Science, 229, Springer Verlag, 1998 (out-of-print but available at http://www.laas.fr/~jpl).

(Computer science course: 6 ECTS)

Signal Processing (Traitement du Signal)

(Stéphane Mallat)

This course introduces the bases of digital signal processing and its applications to audio signals and image. Each course will be followed by a TD session of exercises or computer simulations. The following topics will be covered:

- Fourier integral and discrete Fourier transform
- Filtering and sampling theorem for analog to digital conversion
- Modelization of signals with stationary processes and applications to noise removal by Wiener filtering.
- Time-frequency analysis and audio processing
- Information theory, entropy, coding, and transform compression of audio signals (MP3) and images (JPEG)
- Introduction to non-linear signal processing

(Computer science course : 6 ECTS)

Software engineering and cloud computing (Génie logiciel et Cloud computing)

(Joannès Vermorel)

This course presents the fundamental concepts underlying software engineering, with a particular interest for complex and/or distributed systems. The course is associated to a software development project realized by the students organized in teams. Each session includes a regular talk followed by a collective evaluation of the student project status.

Pre-requisites: This course is not a computer programming course. Students should be comfortable with one or several programming languages before attending the course. Without being an absolute pre-requisite, previous participation to "Algorithms and programming" and "Digital systems: from algorithms to circuits" will be appreciated.

Software engineering is the study of the software production as an economic activity, where hardware, manpower and delays are always limited resources. The advances during the last decade within the domain of software engineering have permitted large productivity gains. The course will focus on understanding how development practices associated with appropriate tools influence (positively or negatively) the productivity of the software industry.

Distributed computing systems, with a strong focus on cloud computing, will be introduced and reviewed. Then, students will also work on a team project that involves intensive computations. The motivation of this choice is twofold: first, the evolution of computing hardware is the leading the market toward an "everything distributed" state; second, distributed systems tend to be very challenging both to develop and to debug.

Course notes: http://www.vermorel.com/softeng.html

References:

- AntiPatterns by William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick, Thomas J. Mowbray
- Joel on Software: And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, Designers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity by Joel Spolsky
- Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides

(Computer science course: 6 ECTS)

Statistical Machine Learning (Apprentissage statistique)

(Francis Bach, Olivier Catoni)

This course will consider machine learning on high-dimensional data, such as signals, images, biological data, or economical data, where large volumes of data need to be analyzed or classified.

The objective of the course is to present the main theories and algorithms of statistical machine learning. The considered frameworks will rely in particular on convex analysis and non-asymptotic deviation inequalities. Practical sessions (more than half of which will be realized with computers) will lead to simple implementations of the algorithms seen in class and applications to various domains such as bioinformatics and computer vision.

(Maths course: 12 ECTS)
Course specific to the L3 Maths-Computer Science and Computer Science-Maths curricula

More information on: http://www.math.ens.fr/cours-apprentissage/

And on: http://www.math.ens.fr/enseignement/fiche_cours.html?cours=62

Statistics (Statistique)

(Gérard Biau)

Objective: This course aims to give students the fundamentals of reasoning in statistical modelling. Emphasis is placed on practical use of the notions encountered.

Prerequisite: A good knowledge of probability theory and linear algebra.
Topics:

- Reminders on probability theory.
- Pointwise and interval estimation, tests.
- Maximum likelihood estimators.
- Linear model: estimation, confidence intervals and tests.
- Bayesian methods
- Introduction to nonparametric statistics, supervised learning and clustering.

(Maths course: 12 ECTS)
Please see description updates of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=57

Stochastic processes (Processus aléatoires)

(Josselin Garnier)

(Maths course: 12 ECTS)
Please see description of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=56

Synchronous systems (Systèmes synchrones)

(Marc Pouzet, Jean Vuillemin)

Description below is old
The synchronous methodology has been used successfully for the design and implementation of safe-critical embedded systems. Real examples can be found in various areas such as planes, trains, nuclear plants or mobile phones.

This course gives an introduction to synchronous systems from theoretical aspect through practical applications. It addresses both hardware and software aspects which are intrinsically related in the field.

Course outline:

- Foundations:
  Stream functions: causality, continuity, sequentiality. Kahn Process Networks, streams and clocks. Automata and circuits: equivalence and minimisation. Deterministic normal forms (Mealy/Moore) and non-deterministic (Fliess).
- Synchronous languages:
  Lustre: operational/denotational semantics. Clock calculus. Compilation. Program analysis: initialisation, causality. Esterel: operators and semantics. Causality. Compilation into circuits and sequential code. Real-time constraints: hardware/software co-design. Desynchronisation. Mixed systems and Mode-automata, functional synchronous languages.
- Vérification and proof
  Synchronous observers: coding safety properties in Lustre/Esterel. Model checking algorithms: forward/backward analysis. Representation of boolean formula and encoding of state transitions (BDD). Representation of streams and programs into a proof-assistant. Testing: automatic generation of test sequences from synchronous observers.
- Synchronous circuits:
  Electronic circuits: from analog circuits to digital circuits, from continuous time to discrete time. Combinatorial logic and memory. Programmable circuits: structure and programming FPGA. Reconfigurable circuits. Arithmetic circuits: representation of numbers, arithmetic operators, duality time/surface. Synthesis: hardware compilation of a synchronous program.

Prerequisite:
It is recommended to have followed a course on compilation and semantics of programming languages; to have elementary notions of operational semantics, on systems and circuits.

(MPRI course: 3 ECTS)
Please see description updates of this MPRI 2-23-1 course at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-23-1

Techniques in Cryptography and Cryptanalysis(Techniques en Cryptographie et Cryptanalyse)

(Michel Abdalla, Phong Nguyen, Vadim Lyubashevsky)

The goal of this course is to bring students to the frontier of current research in public-key encryption and computation on encrypted data.

The course is divided into two parts: functional encryption and fully-homomorphic encryption. In the first part, we consider the notion of functional encryption schemes, which is a generalization of the standard notion of encryption schemes, in which decryption keys allow a user to learn a function of the encrypted data. In particular, we examine several particular cases of functional encryption, such as identity-based encryption, searchable encryption, and attribute-based encryption.

In the second part of this course, we study the notion of fully-homomorphic encryption, which allows arbitrary computations on the encrypted data. Towards this goal, we review standard lattice hardness assumptions and lattice-based encryption schemes used in the constructions of fully-homomorphic encryption schemes.

(MPRI course : 3 ECTS)
Please see description updates of this course MPRI 2-12-1 at:
https://wikimpri.dptinfo.ens-cachan.fr/doku.php?id=cours:c-2-12-1

Topology and differential calculus (Topology and differential calculus)

(Patrick Bernard)

1) Complements of topology: Definitions, quotient topology, product topology, examples of topologies. Connected topological spaces, compactness, topological vector spaces, limits and limit points. Tietze-Urysohn theorem. Baire theory, Tychonoff theorem. Fixed point theorems, Ascoli theorem, Stone-Weierstrass theorem, Hann-Banach and Banach-Steinhaus theorem.

2) Banach differential calculus : local inversion theorem, implicit function and constant rank theorems.
3) Hilbert spaces: convexity, duality, Hilbertian basis, spectral theorem.

4) Ordinary differential equations : Cauchy-Lipschitz theorem, flows of vector fields, linearization.

(Maths course: 12 ECTS)
Please see description of this course at:
http://www.math.ens.fr/enseignement/fiche_cours.html?cours=43#