ENGINEERING PROGRAMME

2024-2025

Year 2 / Year 3

Specialisation option

Computer Science for Artificial Intelligence

OD INFOIA

PROGRAMME SUPERVISOR
Didier LIME

Autumn Semester

Course unit	ECTS Credits	Track	Course code	Title
UE 73	12	Core course	ALGOA INDUR PAPY STASC	Advanced algorithmics Sustainability, ethics and computing Advanced programming in Python Introduction to statistics and data science with Python
UE 74	13	Core course	AGATH DEEP PAMEF PIIA1 QCM	Algorithmic game theory Deep Learning Parallelism and model checking Project 1 Quality, design and modelling

Spring Semester

Course unit	ECTS Credits	Track	Course code	Title
UE 83	14	Core course	GPGPU GRAAL MPAR PIIA2 PRLOG	Programming on graphical processor units Graphs and algorithms Probabilistic modelling and reinforcement learning Project 2 Logic programming

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 73 / 93

Advanced algorithmics [ALGOA]

LEAD PROFESSOR(S): Didier LIME

Requirements

Objectives

At the end of this course, students will be able to:

1. Analyze algorithms
   - assess their complexity
   - prove their functional and non-functional properties

2. Design efficient algorithms:
   - by choosing adapted data structures
   - by using generic solving methods

Course contents

This course consists of three main parts:

1. Program analysis
   - basic properties of algorithms: termination, correctness, completeness, complexity
   - inductive proofs and invariants
   - computation models for complexity
   - decidability and complexity of algorithmic problems
   - complexity assessment : worst case, base case, recursive algorithms
   - complexity in average et randomized algorithms

2. Algorithm design paradigms
   - exhaustive enumeration
   - backtracking
   - divide and conquer
   - dynamic programming
   - greedy algorithms
   - problem transformations

3. Data structures
   - arrays and lists
   - amortized complexity
   - stacks and queues
   - priority queues and heaps
   - binary search trees and AVL trees
   - hash tables

Lab work will enable the practical application of these notions through the design and development of a file compression tool.

Course material

Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest; Clifford Stein (2009) [1990]. Introduction to Algorithms (3rd ed.). MIT Press and McGraw-Hill. ISBN 0-262-03384-4. 1320 pp.

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	18 hrs	0 hrs	12 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 73 / 93

Sustainability, ethics and computing [INDUR]

LEAD PROFESSOR(S): Morgan MAGNIN

Requirements

Basic skills in computer science (methodology and technologies)

Objectives

Objectives in terms of knowledge:
1. Regulation and ethics:
    - Legal framework for automatic data processing
    - Ethics and responsibility
        - Possible biases of artificial intelligence (AI)
        - Autonomous agents and ethics
        - Moral reasoning and ethical responsibility in AI
        - Social issues

2. Green computing:
    - Identify the principles of green computing and the associated levers:
    - Direct and indirect issues and impacts
    - Hardware and software energy consumption
    - Digital Sobriety
        - Algorithms and green software: software eco-design
        - Green data centers

Objectives in terms of skills:
1. Know how to build an ethical and responsible analysis of automatic information processing (identification of the impact of this processing, possible biases, etc.)
2. Know how to implement a diagnosis of the energy consumption of computer applications

Course contents

Course outline:
1. General introduction to the concepts of sustainability and ethics in computer science
2. Case studies introduced at the beginning of the course and taken up again at the end to complete the knowledge acquired in the field of analysis and implementation of more responsible and sustainable solutions.
3. Series of lectures/lectures/talks on "green computing" and "ethics and responsibility"

Course material

- Cerna Collectif. Éthique de la recherche en apprentissage machine. [Rapport de recherche] CERNA; ALLISTENE. 2017, pp.51. hal-01643281 https://hal.inria.fr/hal-01643281/document
- Panorama de formations et de ressources pédagogiques existantes sur le thème « informatique verte » : https://ecoinfo.cnrs.fr/2019/06/21/formations-abordant-les-aspects-environnementaux-du-numerique/
- MOOC « Impacts environnementaux du numérique » https://www.fun-mooc.fr/fr/cours/impacts-environnementaux-du-numerique/
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR), 54(6), 1-35. https://arxiv.org/pdf/1908.09635

- Acar, Hayri. Software development methodology in a Green IT environment. PhD thesis. Université de Lyon, 2017. https://tel.archives-ouvertes.fr/tel-01724069/file/TH2017ACARHAYRI.pdf

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	12 hrs	10 hrs	8 hrs	0 hrs	2 hrs

Advanced programming in Python [PAPY]

LEAD PROFESSOR(S): Lucas LESTANDI

Requirements

Objectives

At the end of this course, student will have the skills to:
1. Set up a robust Python environment and a suitable development environment.
2. Write efficient Python programmes using libraries and modern abstractions.
3. Ensure readability and maintenance by following modular architecture and community style guidelines.
4. Package their code for distribution on other machines including dependencies.

Course contents

1. The right way to work with Python
- setting up the right environment: IDE, jupyter notebooks, Python environments (conda, etc.)
- language philosophy, syntax
- Python interpreter
- good practice for programming in Python (PEP8 style guidelines, etc.)
2. Code structure and data types
- variables, memory and references
- data types and structures
- object oriented programming (OOP): classes
- writing robust code: architecture, introspection, exceptions, etc.
3. Using modules
- native modules: os, sys, subprocess
- external libraries: using pip and conda
- useful examples: numpy, scipy, matplotlib
- create your own modules
4. Towards production code
- decorators
- integration with other langages (C++, etc.)
- testing and debugger
- distribution and portability

Course material

Assessment

Collective assessment: EVC 1 (coefficient 0.5)

Individual assessment: EVI 1 (coefficient 0.5)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	8 hrs	0 hrs	22 hrs	0 hrs	2 hrs

Introduction to statistics and data science with Python [STASC]

LEAD PROFESSOR(S): Bertrand MICHEL

Requirements

Objectives

This lecture is an introduction to statistical learning.
Main objectives:
- understanding the main concepts of statistical learning
- introduction to standard methods in statistical learning
- practice on real data using standard Python libraries

Course contents

- introduction to statistical learning
- standard methods for classification
- CART random forests and boosting
- unsupervised learning
- introduction to kernel methods

Course material

- The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Trevor HastieRobert TibshiraniJerome Friedman, 2009 Springer.
- Hands-On Machine Learning with Scikit-Learn and TensorFlow
by Aurélien Géron, O'Reilly 2017.

Assessment

Collective assessment: EVC 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	15 hrs	17 hrs	0 hrs	0 hrs	0 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 74 / 94

Algorithmic game theory [AGATH]

LEAD PROFESSOR(S): Didier LIME

Requirements

Objectives

Ath the end of this course, students will know how to :
- model decision problems involving several agents in various environments as strategic games
- algorithmically compute strategies giving the best personal gain or the best global equilibrium
- program artificial intelligence agents based on these concepts

Course contents

The course is divided in three main parts:

1. Normal form games
   - games, payoffs, strategies
   - solution concepts : dominated strategies, Nash equilibria, maximal regret minimisation, correlated equilibria
   - two-player zero-sum games

2. Extensive form games
   - sequential games
   - subgame perfect equilibria
   - backward induction
   - approximations : Monte Carlo and statistical evaluation

3. Repeated games
   - finitely and infinitely repeated games
   - regret-based learning

Course material

- Ken Binmore. Playing for Real : A Text on Game Theory, OUP USA, 2007.
- Algorithmic Game Theory, Nisan, Roughgarden, Tardos, and Vazirani, Cambridge University Press, 2007.
- Multiagent Systems, Y. Shoham, K. Leyton-Brown, Cambridge University Press, 2009.
- Michael Maschler, Eilon Solan, Shmuel Zamir. Game Theory, Cambridge University Press, 2013.

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	16 hrs	0 hrs	14 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 74 / 94

Deep Learning [DEEP]

LEAD PROFESSOR(S): Bertrand MICHEL / Didier LIME

Requirements

Objectives

At the end of this course, students will know the theory and algorithmics underlying feed-forward artificial neural networks, as well as the main theory elements for different representatives of the family of deep neural networks (recurrent networks, auto-encoders, generative networks).
They will also be acquainted with their practical use through dedicated software libraries.
Finally, they will see how these models can be applied to solve real-life problems through several case-studies.

Course contents

1. Feed-forward networks
   - Response computation
   - Optimization and learning
   - Convolutional networks
   - Principles of transfer learning and fine tuning

2. Recurrent networks
   - Long term Short Term Memories (LSTM)
   - Models for natural language processing: embeddings, attention, and transformers

3. Auto-encoders et generative models
   - Auto-encoders
   - Generative adversarial networks (GAN)

4. Case-studies
   - Autonomous vehicles
   - Biomedical applications.

Course material

- The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Trevor HastieRobert TibshiraniJerome Friedman, 2009 Springer.
- Hands-On Machine Learning with Scikit-Learn and TensorFlow
by Aurélien Géron, O'Reilly 2017.

Assessment

Collective assessment: EVC 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	17 hrs	7 hrs	8 hrs	0 hrs	0 hrs

Parallelism and model checking [PAMEF]

LEAD PROFESSOR(S): Olivier ROUX

Requirements

Objectives

At the end of this course, students will know and be able to apply the principles of programming on Graphics Processing Units (GPU) to accelerate and optimize calculations. They will be able to implement parallel algorithms taking into consideration the locality of memory and data as well as the hardware specificities of the control flow on GPUs. The practical lab work will allow students to implement, validate and evaluate classical parallel programming algorithms in order to optimize computation.

Course contents

The course will be broken down into four main parts:
1. Generalities about programming on GPU: — concurrent vs parallel programming
— principle and architecture of GPUs
2. Operators, memories and data structures: — Data allocation and control flow
— memory types and data locality
— atomic operations and synchronization
3. Study of algorithm patterns for parallel programming
4. Using their knowledge to optimize a neural network calculation

Course material

— Programming massively parallel processors: a hands-on approach (3rd Edition), David Kirk et Wen-mei W. Hwu, Morgan Kaufmann, 2017
— CUDA by Example: An Introduction to General-Purpose GPU Programming, Jason Sanders et Edward Kandrot, Addison Wesley, 2010
— Nvidia Online training https://developer.nvidia.com/cuda-education

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	20 hrs	0 hrs	10 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 74 / 94

Project 1 [PIIA1]

LEAD PROFESSOR(S): Didier LIME

Requirements

Objectives

Apply in practice the knowledge and skills acquired during the year.

Course contents

Starts in late September, ends early January. Can be merged with the second project (PIIA2).

Course material

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	1	0 hrs	0 hrs	0 hrs	32 hrs	0 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Autumn Semester - Course Unit 74 / 94

Quality, design and modelling [QCM]

LEAD PROFESSOR(S): Myriam SERVIÈRES

Requirements

Objectives

The course covers three key themes of computer development in business: Software Engineering, Databases, and Computer Development Methods. The objectives of these three topics are, respectively:
- To acquire the basics of Software Engineering and IT Project Management.
- To acquire the basics of software engineering and computer project management, to master the elements of design and use of relational databases.
- To develop using tools and methods that allow the construction of software applications of industrial quality.

Following this course, the skills acquired should allow to:
- Design and model software and write specifications.
- Master the elements of design and use of relational databases.
- Collaborate in developing software, and automated test suites, and guarantee the quality of the code.

Course contents

1. Software engineering. We will cover here:
- Software development cycles: specification, life cycle, planning, quality, specification, production, acceptance.
- UML design models: use cases, class diagrams, sequence diagrams, state-transition diagrams, and activity diagrams.
2. Databases
- Conceptual and physical modeling, relational model
- SQL queries
- Design of database query programs
3. Methods of computer development.
We will discuss version management tools, unit tests, and code metrics here.

Course material

- Modélisation objet avec UML, Pierre-Alain Muller, Best of Eyrolles, 2005.
- UML2 et les design patterns, Craig Larman, Pearson Education, 2005.
- Software Engineering 8, Ian Sommerville, Addison Wesley, 2007.
- Le génie logiciel et ses applications, Ian Sommerville, InterEdition, 1988.
- Processus d'ingénieries du logiciel, méthodes et qualité, Claude Pinet, Pearson Education, 2002.
- UML2, Benoit Charroux, Aomar Osmani, Yann Thierry-Mieg, Pearson Education, 2005.

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	9 hrs	8 hrs	13 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Spring Semester - Course Unit 103 / 83

Programming on graphical processor units [GPGPU]

LEAD PROFESSOR(S): Pierre-Emmanuel HLADIK

Requirements

Objectives

Integrate the concepts of parallel computing (communication and synchronization),
Understand how these principles can be implemented,
Gain insight into higher-level mechanisms,
Practise to understand the specific features of concurrency
Learn about proving properties of concurrent systems.

Course contents

1. General
2 Introduction: why parallel programs?
3 Synchronization through shared variables
4 Synchronization through communication
5 Verification of properties of (distributed) computer systems
6 Temporal logics, model-checking
7 Time checking
8 Conclusion
9 Bibliography

Course material

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	18 hrs	0 hrs	23 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Spring Semester - Course Unit 103 / 83

Graphs and algorithms [GRAAL]

LEAD PROFESSOR(S): Didier LIME / Loig JEZEQUEL

Requirements

Objectives

This lecture takes the form of a collection of focused seminars. It aims at giving an overview of the practical utility of graphs as well as at presenting a portfolio of algorithms that can be used to analyze them.

Course contents

As an introduction to the course, basic notions about graphs and standard algorithms will be presented.

After that, a set of topics will be proposed to the students. Each of them (possibly as a group) will have to work on one of these subjects and present it to the class. Among the topics proposed, one could list: Graphical representation of graphs; Multi-agents systems: collaborative exploration of graphs; Path finding, the A algorithm; Build the graph of the Internet; Random walks on graphs; Planar graphs and excluded minors; Small-world graphs; Graph rewriting. One of these topics will be presented by the teacher as an example of what is expected from the students.

Course material

- E. Goodaire, M. Parmenter. Discrete Mathematics with Graph Theory. 2018.
- D. Beauquier, J. Berstel, P. Chrétienne. Eléments d'algorithmique. 1992. (disponible en ligne)
- M. Gondran, M. Minoux. Graphes et algorithmes. 2009.
- J.-C. Fournier. Théorie des graphes et applications. 2006.

Assessment

Collective assessment: EVC 1 (coefficient 0.5)

Individual assessment: EVI 1 (coefficient 0.5)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	16 hrs	16 hrs	0 hrs	0 hrs	0 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Spring Semester - Course Unit 103 / 83

Probabilistic modelling and reinforcement learning [MPAR]

LEAD PROFESSOR(S): Benoit DELAHAYE / Didier LIME

Requirements

Objectives

In the first part of the course, students will learn about probabilistic state modeling techniques such as Markov chains, Markovian decision processes and their timed and imperfect information extensions, as well as the specific formal verification techniques.
In the second part, students will discover the principles of reinforcement learning for these models, as well as several techniques to tackle them.

The methods and techniques of this course will be illustrated and evaluated during tutorials in Python.

Course contents

Course outline:
The course is divided into three main parts:

I. Probabilistic Modeling
1. Discrete-time Markov chains (DTMC)
2. Markov decision processes (MDP)
3. Imperfect information extensions (HMC and PO-MDP)

II. Formal verification for probabilistic models
1. Probabilistic Model-Checking
2. Statistical Model-Checking

III. Reinforcement learning
1. Multi-armed bandit and MDPs
2. Dynamic programming (DP)
3. Monte-Carlo methods
4. Temporal difference (TD)

Course material

- Christel Baier and Joost-Pieter Katoen. Principles of model checking. MIT press, 2008.
- Pedro R. D'Argenio, Arnd Hartmanns, and Sean Sedwards. Lightweight statistical model checking in nondeterministic continuous time. In Tiziana Margaria and Bernhard Steffen, editors, ISoLA'18, volume 11245 of LNCS, pages 336–353. Springer, 2018.
- David Henriques, Joao G Martins, Paolo Zuliani, André Platzer, and Edmund M Clarke. Statistical model-checking for markov decision processes. In 2012 Ninth international conference on quantitative evaluation of systems, pages 84–93. IEEE, 2012.
- Vikram Krishnamurthy. Partially observed Markov decision processes. Cambridge University Press, 2016.
- Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking: An overview. In International conference on runtime verification, pages 122–135. Springer, 2010.
- Martin L Puterman. Markov decision processes. Handbooks in operations research and management science, 2 :331–434, 1990.
- Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.
- Koushik Sen, Mahesh Viswanathan, and Gul Agha. On statistical model checking of stochastic systems. In International

Conference on Computer Aided Verification, pages 266–280. Springer, 2005.
- Marco A Wiering and Martijn Van Otterlo. Reinforcement learning, volume 12. Springer, 2012.

Assessment

Collective assessment: EVC 1 (coefficient 0.5)

Individual assessment: EVI 1 (coefficient 0.5)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	16 hrs	6 hrs	8 hrs	0 hrs	2 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Spring Semester - Course Unit 103 / 83

Project 2 [PIIA2]

LEAD PROFESSOR(S): Didier LIME

Requirements

Objectives

Apply in practice the knowledge and skills acquired during the year.

Course contents

Starts early January, ends late March. Can be merged with the first project (PIIA1).

Course material

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	2	0 hrs	0 hrs	0 hrs	48 hrs	0 hrs

ENGINEERING - OD INFOIA

Year 2 / Year 3 - Spring Semester - Course Unit 103 / 83

Logic programming [PRLOG]

LEAD PROFESSOR(S): Carito GUZIOLOWSKI

Requirements

Objectives

This course is an overview of the logic programming paradigm, which is a declarative paradigm of programming. We will use Answer Set Programming (ASP) as a modeling language and framework. The aim of ASP is to propose a solution to a problem (notably a combinatorial search problem) by modelling this problem instead of coding its solution. Modelling is performed in the form of logical rules (first order predicates) that have a defined syntax and semantics. The resolution of ASP logic programmes is carried out with very powerful solvers. In this course we will use gringo and clasp.

Course contents

The course presents an overview of the following aspects :
— Declarative programming
— ASP Syntax
— ASP Semantics
— Modelling of a problem
This course will be strongly articulated around practical tutorial and lab sessions. The main idea of these sessions will be to propose logic programmes to solve classical combinatorial search problems.

Course material

Knowledge representation, Reasoning, and Declarative Problem Solving. Chitta Baral. Cambridge University Press New York, NY, USA, 2003.

Gelfond, M., & Kahl, Y. (2014). Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge: Cambridge University Press. doi:10.1017/CBO9781139342124

Assessment

Individual assessment: EVI 1 (coefficient 1)

LANGUAGE OF INSTRUCTION	ECTS CREDITS	LECTURES	TUTORIALS	LAB	PROJECT	EXAM
French	3	20 hrs	0 hrs	10 hrs	0 hrs	2 hrs