Back-to-school Meeting

Master 2 ICA

Aurélien Garivier

ENS Lyon - Computer Science Department

Sept. 7th, 2026

Outline

General Information

Internships and Computer Science Department

Short Presentations of the Courses

Communication

Website http://informatique.ens-lyon.fr/en/academic-programs/master/m2

Contact: aurelien.garivier@ens-lyon.fr
zoe.michal-sihalath@ens-lyon.fr

Students: m2if-pair@listes.ens-lyon.fr

Teachers: cours.m2.info.pair@listes.ens-lyon.fr

⚠ Pad: https://avalon.ens-lyon.fr/etherpad/p/r.9a4e4731bbe8bb24be57345842bd18af

$\Longrightarrow$ check "every day"

TODO

- Registration at ENS
  $\Longrightarrow$ contact `zoe.michal-sihalath@ens-lyon.fr`
- Rules: see website
  - 30 ECTS + internship
  - Semester 1: two waves (Sep. 7th → Nov. 14 and Nov. 16 → Feb. 3)
  - each of the CR courses is worth 5 ECTS
  - you may (under condition) take at most 10 ECTS outside of the CR
    requires a relevant with a project! needs my validation
  - recommended: 7 CR or equivalent
- Declaring your choices, your internship, etc: follow-up pad

Internship: 5-6 months starting Feb. 4th

Tutorship

- To contact in case of difficulty, or simply to help choosing
- Meeting with tutors on Monday Sept. 23rd at 1:15pm
- Mandatory: discuss you internship with your tutor!
- Goal in mind: PhD

Internship

- Responsible: `edouard.bonnet@ens-lyon.fr`
- information meeting will be organized in October
- 5-6 months
- you find the internship (some suggestions provided)
- `https://perso.ens-lyon.fr/edouard.bonnet/internshipsM2.htm`
- requires the validation of your tutor!
- requires time to establish the "convention"!



Importance of the M2 internship

- More than half of the year
- 30 credits out of 60
- Likely to be the place where you later start your PhD (reunion in October)

Importance of the M2 internship

- More than half of the year
- 30 credits out of 60
- Likely to be the place where you later start your PhD (reunion in October)

Not meant to stress you but to realize that

The time and energy you spend for the internship (before/during/after) is well spent.

Schedule

- **October:** You start actively looking for an internship
- **November-December:** Your convention has been validated
- **End of January-Start of February:** Your internship starts
- **Mid-June:** Your internship ends, and your report is due
- **Early to mid-July:** Internship defenses

Who/what can help you?

- **To decide the area of your internship:** Mainly yourself.

Who/what can help you?

- **To decide the area of your internship:** Mainly yourself.
- **To find an advisor/internship:** Your tutor, lecturers in relevant classes.

Who/what can help you?

- **To decide the area of your internship:** Mainly yourself.
- **To find an advisor/internship:** Your tutor, lecturers in relevant classes.
- **To have an internship that goes well:** Us, write to us if something goes wrong that cannot be simply fixed by talking to your advisor.

Who/what can help you?

- **To decide the area of your internship:** Mainly yourself.
- **To find an advisor/internship:** Your tutor, lecturers in relevant classes.
- **To have an internship that goes well:** Us, write to us if something goes wrong that cannot be simply fixed by talking to your advisor.
- **To write a good report:**
  - Your advisor: ask them for well-written papers on your topic
  - Your advisor: send them an early draft of your report
  - Taking possible feedbacks from L3/M1 seriously
  - Some material/pointers on the dedicated webpage.

Who/what can help you?

- **To decide the area of your internship:** Mainly yourself.
- **To find an advisor/internship:** Your tutor, lecturers in relevant classes.
- **To have an internship that goes well:** Us, write to us if something goes wrong that cannot be simply fixed by talking to your advisor.
- **To write a good report:**
  - Your advisor: ask them for well-written papers on your topic
  - Your advisor: send them an early draft of your report
  - Taking possible feedbacks from L3/M1 seriously
  - Some material/pointers on the dedicated webpage.
- **To make a good oral presentation:**
  - Your advisor: ask if you can rehearse your talk with them
  - Rehearsing several times
  - Some material/pointers on the dedicated webpage.

Ending with an introduction

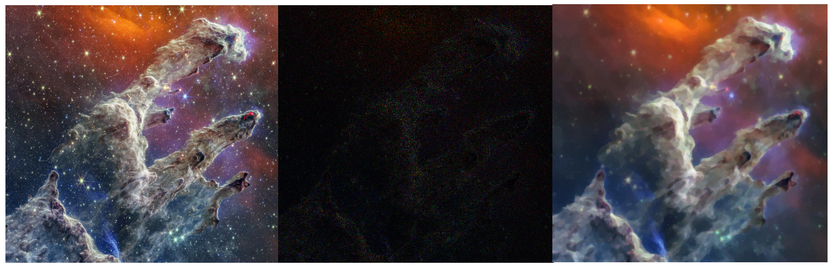
Édouard Bonnet, researcher here at LIP in structural and algorithmic graph theory

"Dedicated webpage" will be accessible from my homepage:
`https://perso.ens-lyon.fr/edouard.bonnet/`
and also from the pad.

Short Presentations of the Courses

- ~~18~~ 17 courses
- 2 minutes presentation + 2 minutes questions
- in increasing order of CR

CR01: Multiresolution in image restoration



Direct problem

$$y = A\bar{x} + \epsilon$$

Inverse problem

$$\text{Find } \hat{x} \in \arg\min_x \|Ax - y\|^2 + \lambda \|Dx\|_*$$

Solution methods :

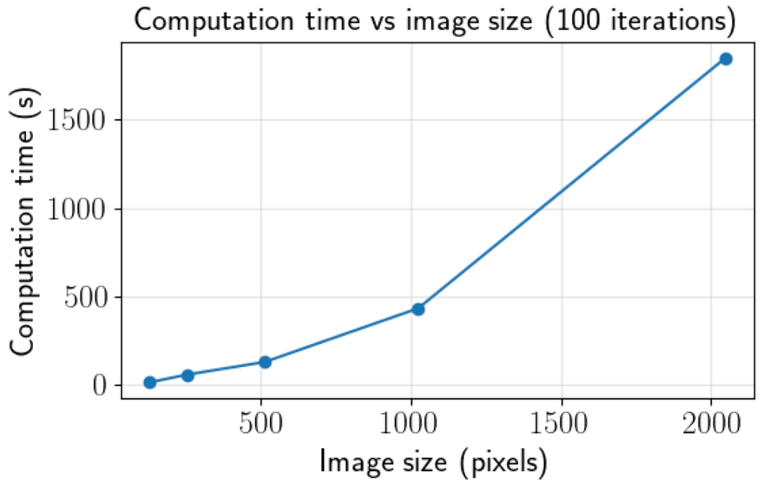
- Proximal gradient schemes

$$x^{[k+1]} = \text{prox}(x^{[k]} - \gamma_k \nabla f(x^{[k]}))$$

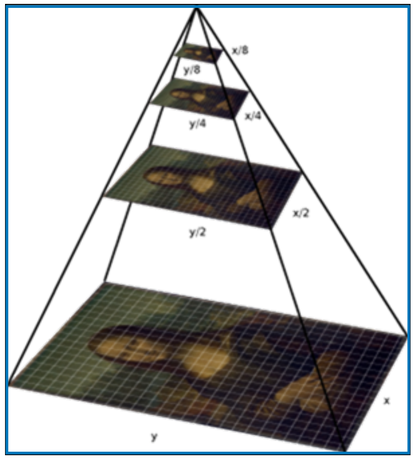
- Approches involving neural networks

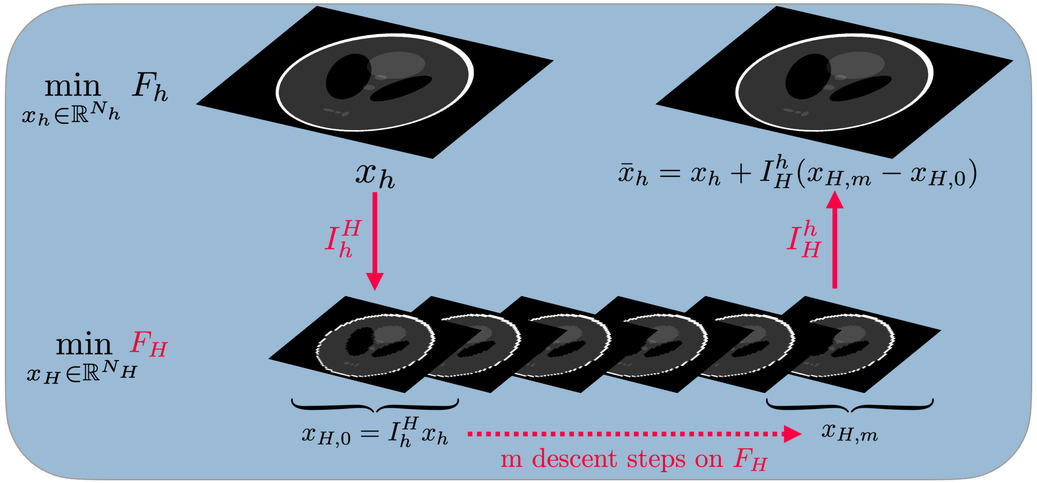
$$x^{[k+1]} = D_\sigma(x^{[k]} - \gamma_k \nabla f(x^{[k]}))$$

Bottleneck: scale with image size



Multilevel proximal methods





Mathematical Optimization with Multiple Players





Classical optimization solves

$$\min_{\boldsymbol{x} \in X} \; f(\boldsymbol{x}), \quad X \subseteq \mathbb{R}^{n_r} \times \mathbb{Z}^{n_z}$$

Two major extensions involve additional players ...

Robust Optimization

An **adversarial** player controls parameters $\boldsymbol{\xi}$

$$\min_{\boldsymbol{x} \in X} \; \max_{\boldsymbol{\xi} \in \Xi(\boldsymbol{x})} \; \min_{\boldsymbol{y} \in Z(\boldsymbol{\xi}, \boldsymbol{x})} \; f(\boldsymbol{x}, \boldsymbol{\xi}, \boldsymbol{y})$$

Models parameters uncertainty/variability

- Uncertain travel times
- Variable demand forecasts
- Disaster management

Bilevel Optimization

A **follower** solves her own problem

$$\min_{\boldsymbol{x} \in X(\boldsymbol{y}), \boldsymbol{y}} \; f(\boldsymbol{x}, \boldsymbol{y}) \text{ s.t. } \boldsymbol{y} \in \operatorname*{arg\,min}_{\boldsymbol{y}' \in Y(\boldsymbol{x})} \; g(\boldsymbol{x}, \boldsymbol{y}')$$

Models hierarchical decision-making

- Toll setting problems
- Stackelberg competition
- Hyperparameter optimization

CR03: **Computational Aspects of Quantum Simulation**

- **Instructors:** Omar Fawzi and Samuel Slezak (QINFO, ENS Lyon)
- **Pre-requisites:** Linear algebra and probability
  No knowledge of quantum theory is assumed
- **Contents:**
  - Introduction to the formalism of quantum theory
  - Algorithms for simulating quantum dynamics (Clifford circuits, bounded-entanglement dynamics,...)
  - Algorithms for simulating ground state properties of quantum systems (Complexity-theoretic aspects, semidefinite programming based methods,...)
  - Algorithms for simulating finite-temperature properties of quantum systems (Classical Gibbs states, counting complexity, quantum Gibbs sampling,...)

Evaluation: Two assignments + project

CR4: Theory of (Linear) Optimization and Approximation Algorithms

- Instructors: Sophie Huiberts (LIMOS) and Alantha Newman (LIP).
- Topics:
  - Basics of Linear Programming, Simplex Method including worst-case complexity, modern analysis, and practical aspects.
  - Approximation Algorithms: classic (randomized, iterative) and modern rounding techniques for linear relaxations, applied to problems such as: traveling salesman problem, maximum cut and graph partitioning, coloring and scheduling.
- Course website: `https://perso.ens-lyon.fr/alantha.newman/course_proposal.html`

CR05: Proof Assistants

Master 2 · ENS de Lyon · 2026–2027

Cyril Cohen · Filippo A. E. Nuccio · Enrico Tassi

2nd wave, 15 sessions, Tue & Wed, 15:45 – 17:45

PREREQUISITES

Functional programming (**Projet Fonctionnel**, L3, or equivalent), and a course covering typed λ-calculus together with the Curry–Howard correspondence (**Proofs and Programs**, M1 ICA, or equivalent). **No prior exposure to a proof assistant is assumed.**

1 · FOUNDATIONS — SESSIONS 1–5 · CC, ET

The Calculus of Constructions syntax, typing rules and models; inductive types, records and infuctive families; architecture of a proof assistants, elaboration, tactics and decision procedures.

2 · METAPROGRAMMING — SESSIONS 6–8 · ET

Logic programming in metaprogramming: λProlog and Elpi, writing a simple typechecker; the representation of Rocq terms and the Rocq-Elpi API; writing tactics that manipulate goals and proof terms.

3 · ROCQ AND MATHCOMP — SESSIONS 9–11 · CC

How a large library is held together: from naive encodings of algebraic structures to Hierarchy Builder; Different notions of extensionality in Type Theory; topology in proof assistants (definition, filters and practical use).

4 · LEAN AND MATHLIB — SESSIONS 12–15 · FN

An introduction to Lean, its library-search tools and Mathlib's conventions; function, sets and quotients in Mathlib, then two sessions on categories.

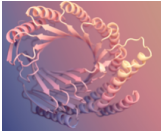
Assessment: graded lab sessions and an exam.

CR06: Generative models

- A powerful tool
- Also very hyped



Stable diffusion/Dall-E/ Midjourney



Protein generation

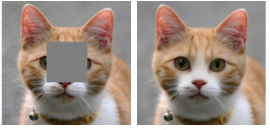


Image reconstruction (unmasking)

Goal and scope:

- Get a clearer picture of the techniques, open the black box
- Focus on image and text generation (Transformers (GPT), diffusion, flow matching)
- Tools: probas, ML, deep learning, optim, python, hands on in pytorch
- Teachers: Quentin Bertrand, Ségolène Martin, Mathurin Massias (Inria OCKHAM)

All info at `https://generativemodels.github.io/`

Floating-Point Arithmetic

Claude-Pierre Jeannerod and Guillaume Melquiond

The most common way to approximate real numbers.
Almost all numerical software relies on it.

Performing efficient, accurate, and rigorous computations is really challenging and is the main motivation for offering this course.

Contents:

- History and basic principles of FPA
- The IEEE 754 standard
- Hardware implementation of FPA, compiler issues
- Rigorous error analysis, stability theorems
- Formal proofs
- Techniques for extending precision
- AI-driven recent alternatives: stochastic, low precision, ...

Evaluation: 2 written homeworks (50%) + analysis and presentation of a research paper (50%).

CR8 : Independent Sets in Graphs

Main instructor: Stéphan Thomassé
With contributions of : Édouard Bonnet, Jean-Florent Raymond, Nicolas Trotignon and Rémi Watrigant

Prerequisites: basic notions in graph theory, algorithms, complexity theory, linear algebra and discrete probability

Evaluation : article presentation + final exam

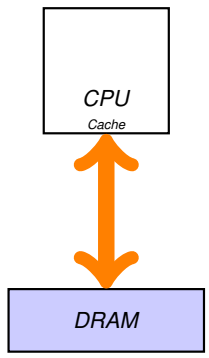
Topics :

Structural graph theory :	Algorithmics :
• Graph classes • Perfect graphs • Ramsey theory (Erdős-Pósa, Erdős-Hajnal) • Polyhedral method • Geometrical graphs	• Reduction techniques • Approximation • Fixed-parameter tractability • Moderately exponential algorithms • Optimization

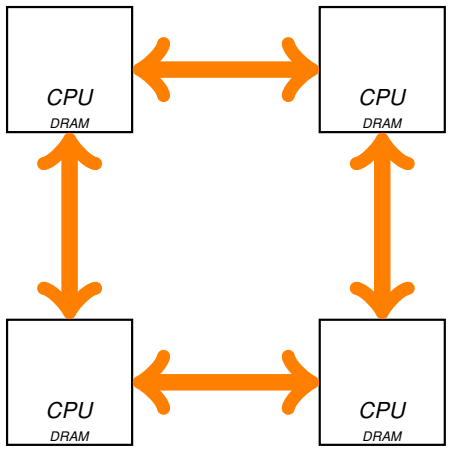
Tools and techniques :

- VC-dimension
- Regularity Lemma
- Containers
- Semi-Definite Programming
- Graph parameters
- ...

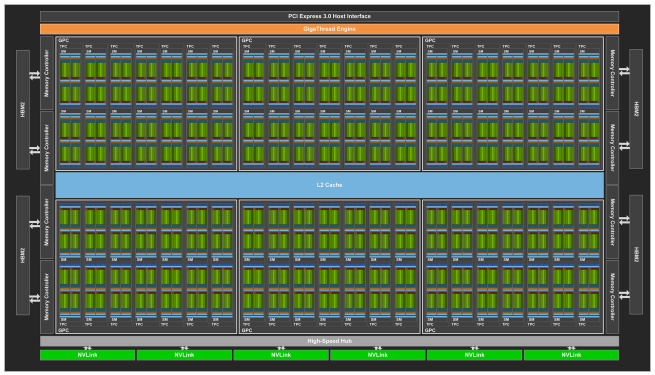
CR09 : Resource-aware computations on CPUs and GPUs



Sequential machine



Parallel machine



Nvidia Volta 100 architecture

Contents

- Models, algorithms and theoretical results for memory-bounded computations
- GPU fundamentals and CUDA programming

Prerequisite : experience with C/C++

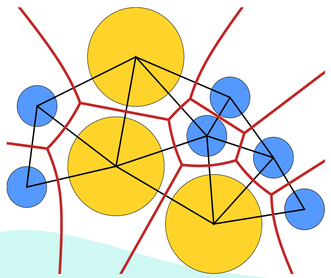
When : in the second wave

Evaluation

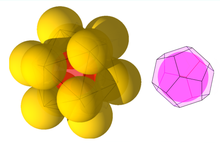
- 2 pen-and-paper-based and 3 small programming assignments
- Report/Presentation

CR10: Computer-assisted proofs

Wave 2



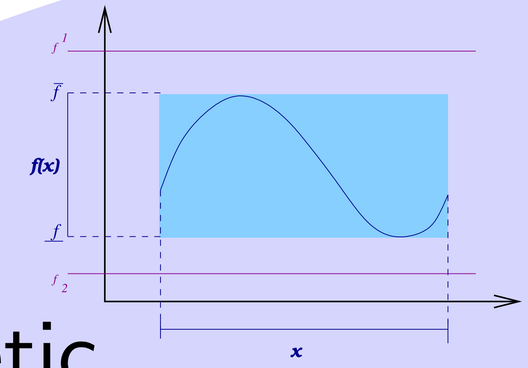
Daria Pchelina , Michael Rao , Damien Pous , Nathalie Revol



optimal disk & sphere packings

Kepler conjecture

geometry



interval arithmetic

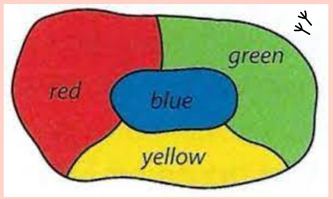
computer

tilings

graphs

discrete math

4-color theorem



formal proofs

CR11 – Privacy-preserving machine learning with homomorphic encryption (G. Hanrot, J.H. Park)

Context:

- Homomorphic encryption computes on *encrypted data*;
- Increasing ML usage requires techniques for privacy preservation;
- HE computational model requires specific algorithmic design.

Outline:

- LWE/RLWE encryption [crypto]
- The CKKS HE system [computer algebra, numerics]
- Homomorphic tools for AI algorithms [algorithms, numerics]
- Advanced topics [crypto, algorithms]
- + 2 tutorials on CodeHEaaN (`heaan.io`).

Prerequisites:

- **Programming (C)**, computer algebra, cryptography, probabilities.

Evaluation:

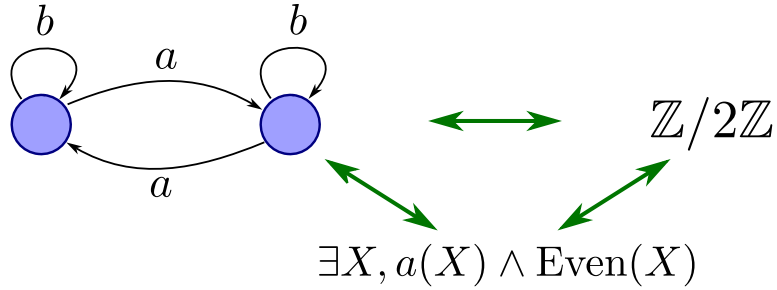
- Project (eg. from fhe-benchmarking, 50%) / Paper presentation / Course question.

HEAAN CRYPTO LAB

CR12: Graphs, Machines and Logic

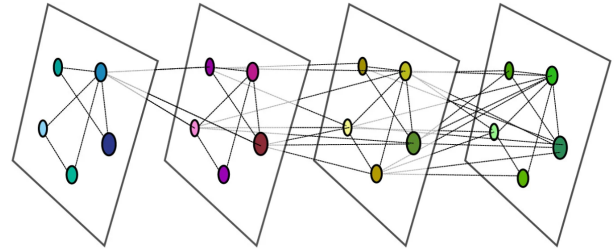
Automata, logic and algebra

Denis Kuperberg



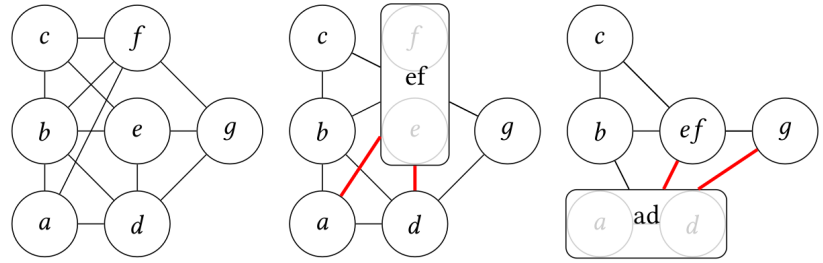
Graph Neural Networks and logics with counting

François Schwarzentruber

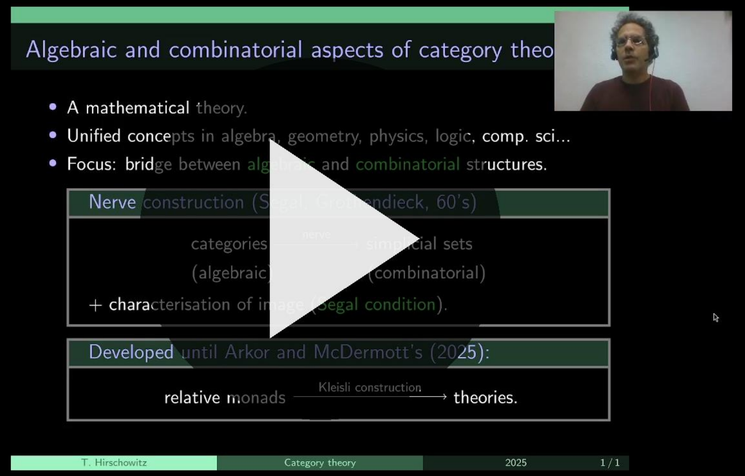


Algorithmic meta-theorems on graphs

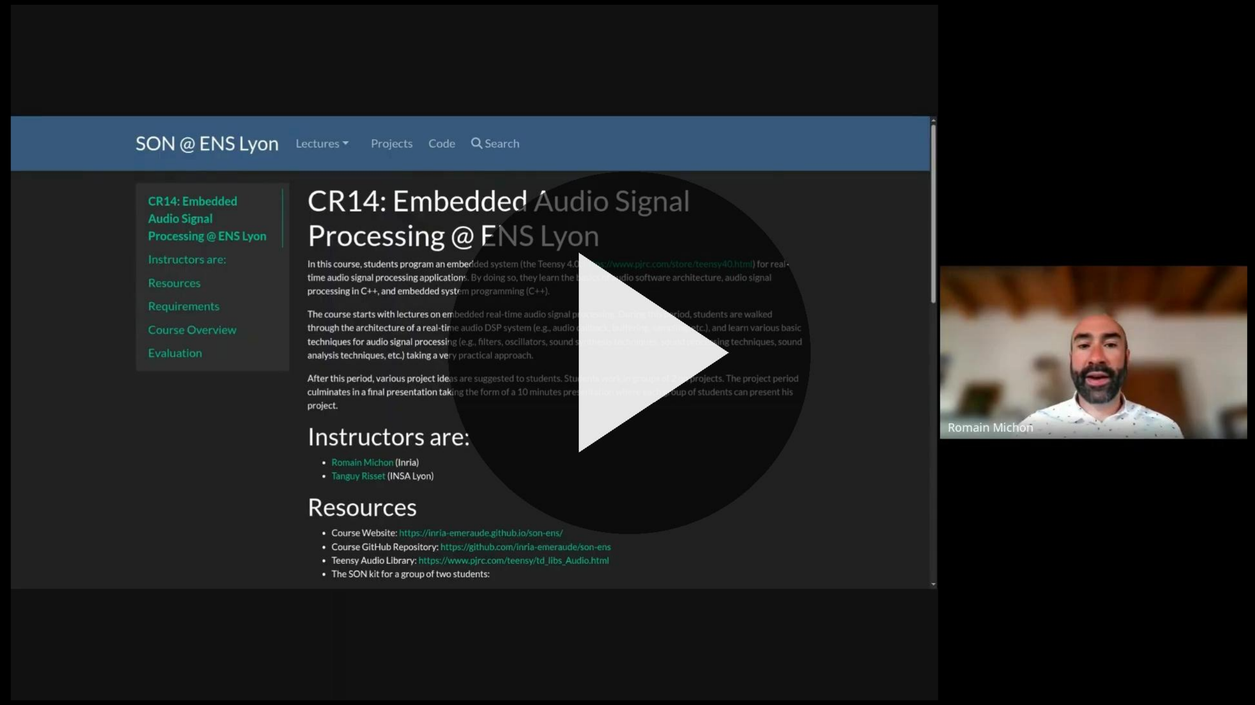
Stéphan Thomassé



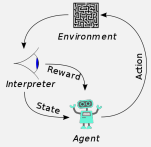
Evaluation modalities: homework, literature review, final exam


Algebraic and combinatorial aspects of category theo
A mathematical theory.
Unified concepts in algebra, geometry, physics, logic, comp. sci...
Focus: bridge between algebraic and combinatorial structures.
Nerve construction (Segal, Duskin, Illusie, Grandjean, 60's)
categories
nerve
simplicial sets
(algebraic)
(combinatorial)
+ characterisation of image (Segal condition).
Developed until Arkor and McDermott's (2025):
relative monads
Kleisli construction
theories.
T. Hirschowitz
Category theory
2025
1 / 1

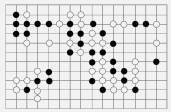
CR13.mp4


SON @ ENS Lyon
Lectures
Projects
Code
Search
CR14: Embedded Audio Signal Processing @ ENS Lyon
Instructors are:
Resources
Requirements
Course Overview
Evaluation
CR14: Embedded Audio Signal Processing @ ENS Lyon
Instructors are:
Romain Michon (Inria)
Tanguy Risset (INSA Lyon)
Resources
Course Website: https://inria-emeraude.github.io/son-ens/
Course GitHub Repository: https://github.com/inria-emeraude/son-ens
Teensy Audio Library: https://www.pjrc.com/teensy/td_libs_Audio.html
The SON kit for a group of two students:
Romain Michon

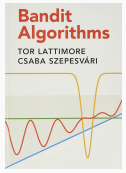
CR14.mp4

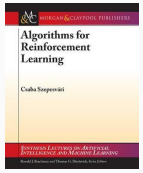


CR15 Reinforcement Learning









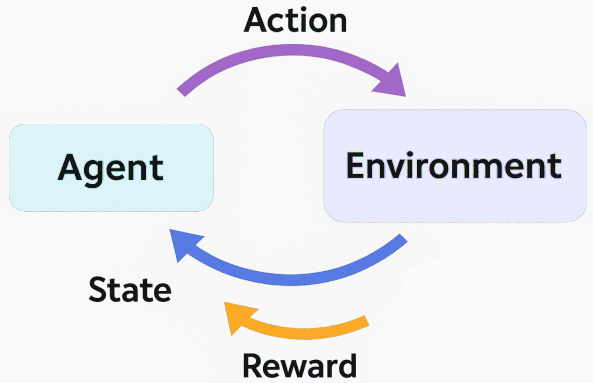


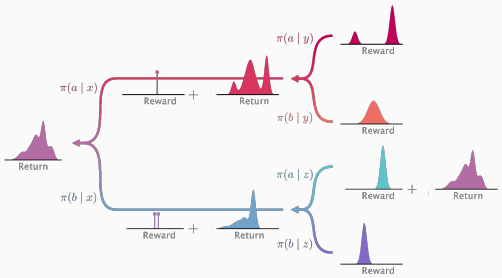


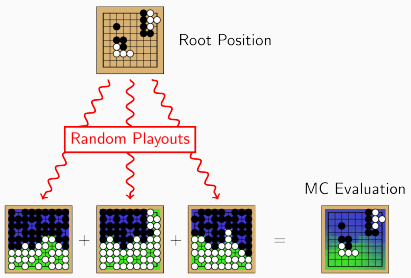
Planning and Dynamic Programming:
Markov Decision Processes

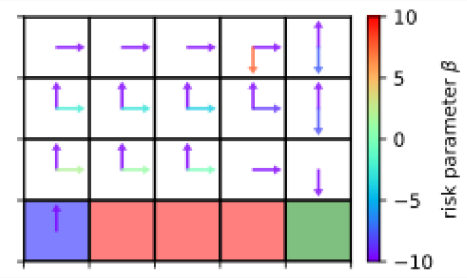
Learning and Stochastic Approximation:
bandits, Q-learning, deep RL

Prerequisites: probability, linear algebra, basic programming





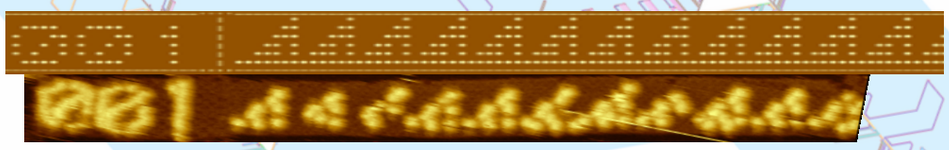


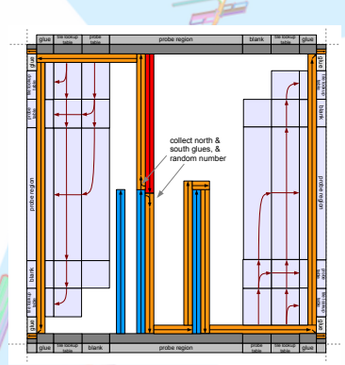


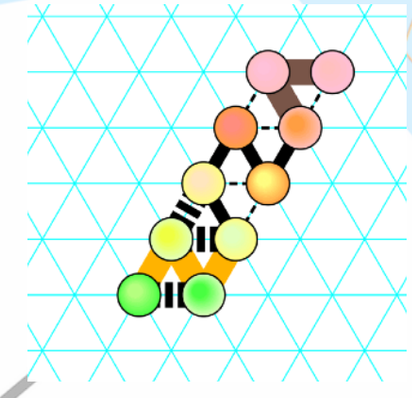
CR16
Molecular programming: Theory & experiment

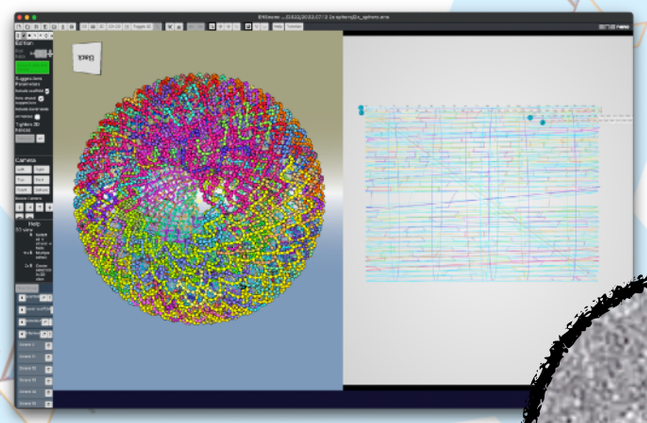


Nicolas SCHABANEL
Directeur de Recherches CNRS — LIP, ENS de Lyon

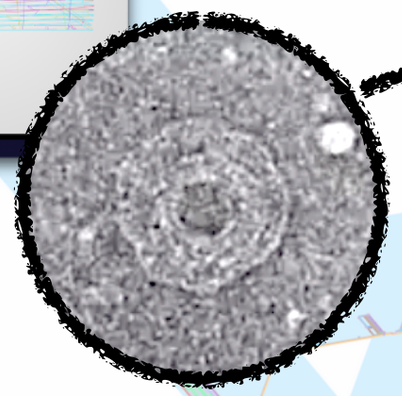


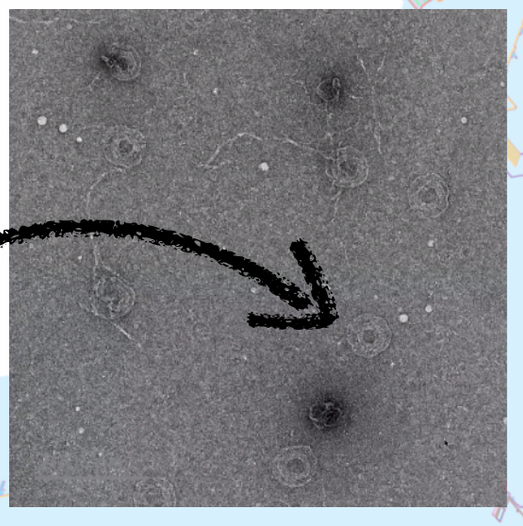






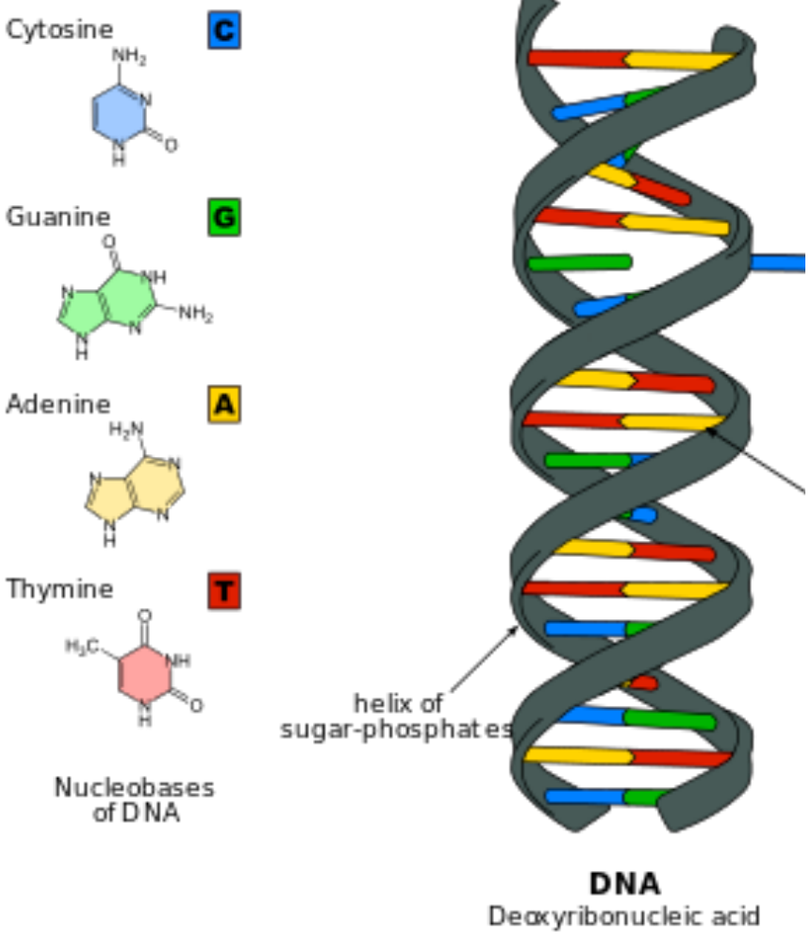
ens nano

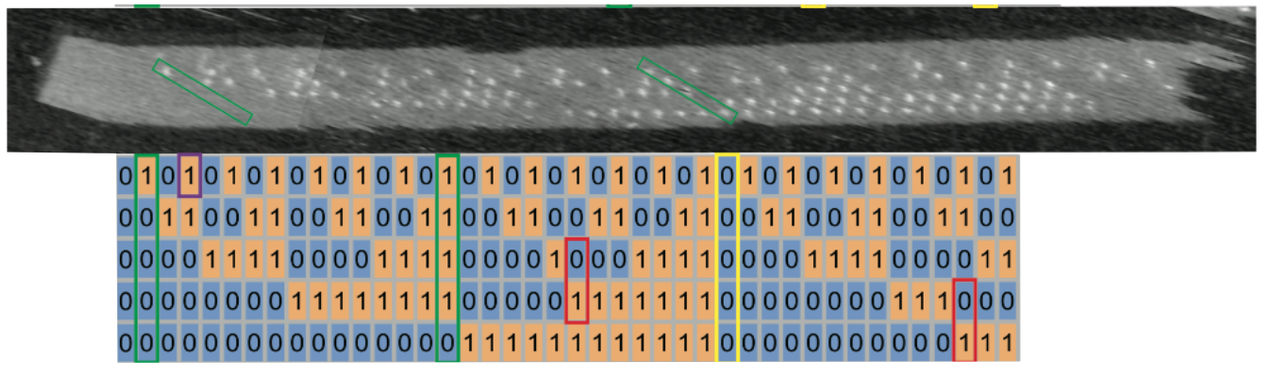


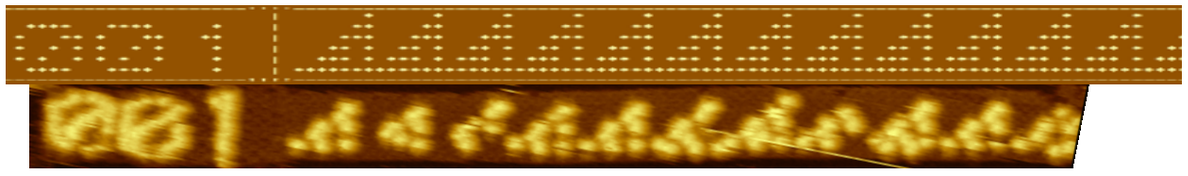


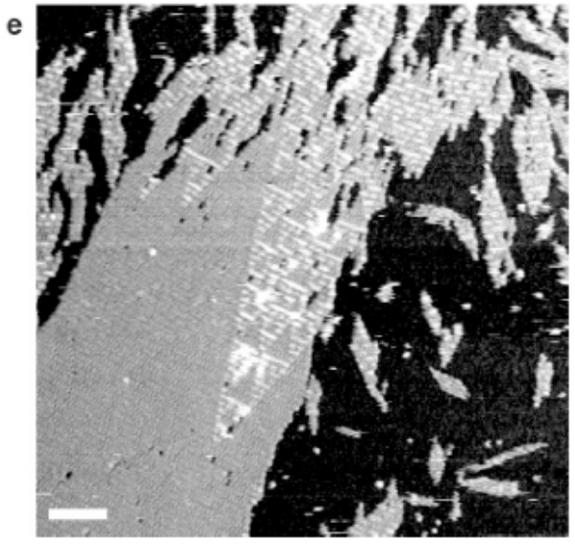
CR16 Molecular programming: Theory & experiment

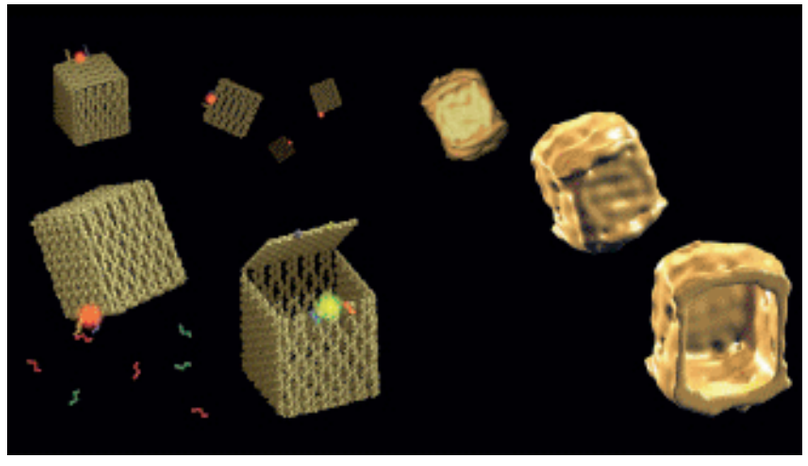
Use DNA complementary to build *computing* nanoobjects (~50-100nm wide, that is ~500-1000 atoms wide)





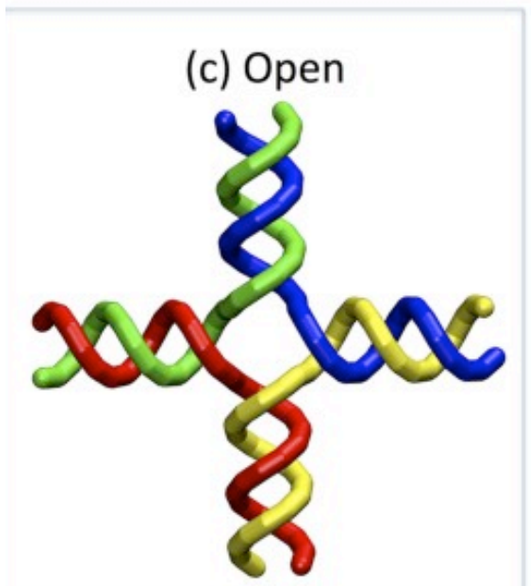


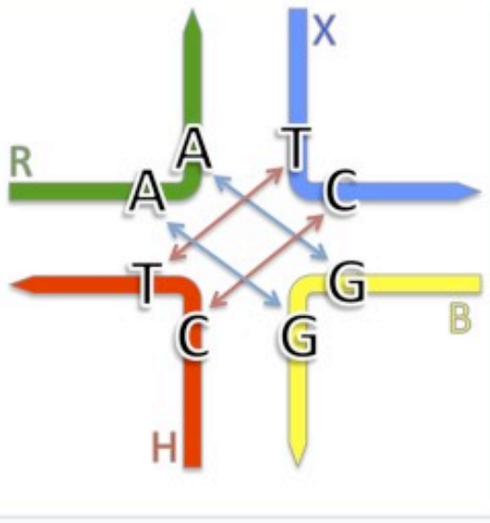


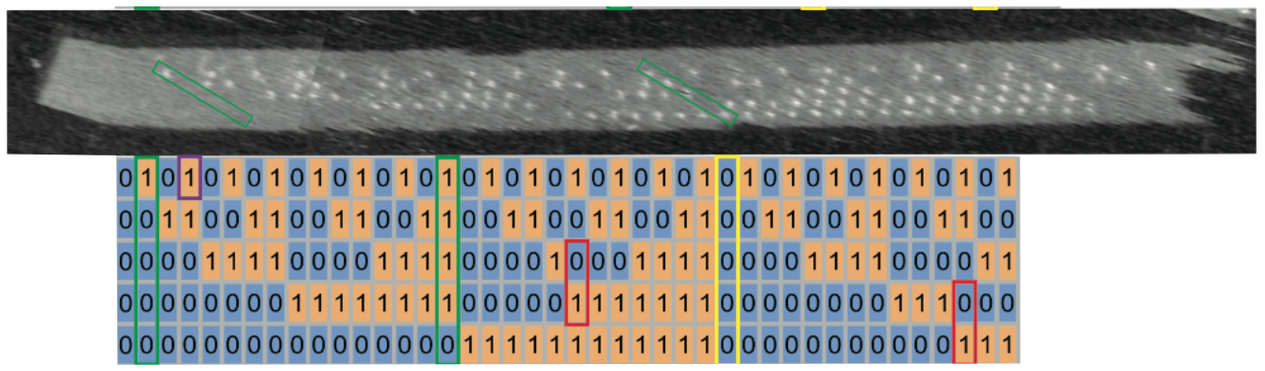


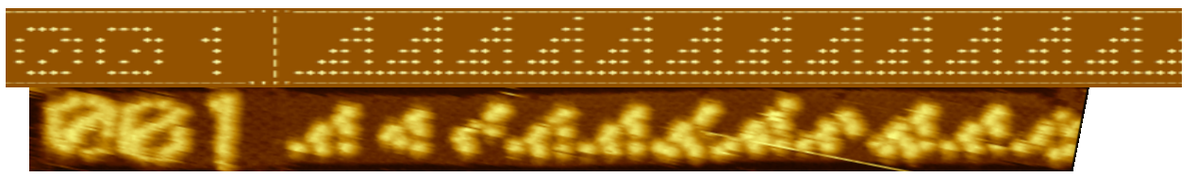
CR16 Molecular programming: Theory & experiment

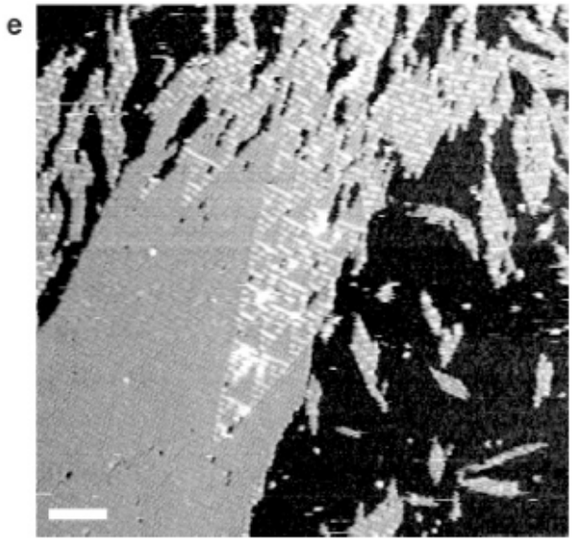
Use DNA complementary to build *computing* nanoobjects (~50-100nm wide, that is ~500-1000 atoms wide)

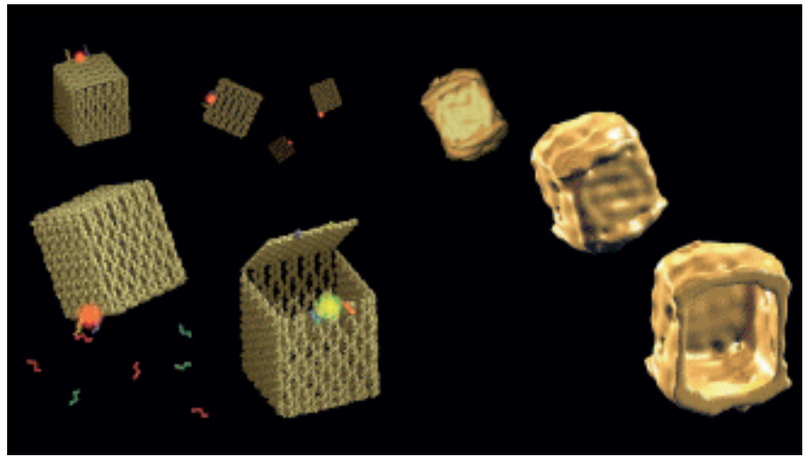












CR16 Molecular programming: Theory & experiment

THEORY

MODELS

Abstract Tile Assembly (aTAM)

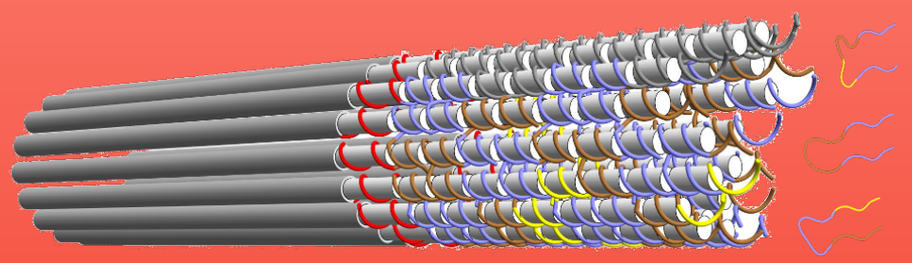
THEOREMS

Any CA can be simulated by a self-assembling boolean circuit

WET LAB EXPERIMENTS

MOLECULAR MECHANISMS

Single stranded tile nanotubes



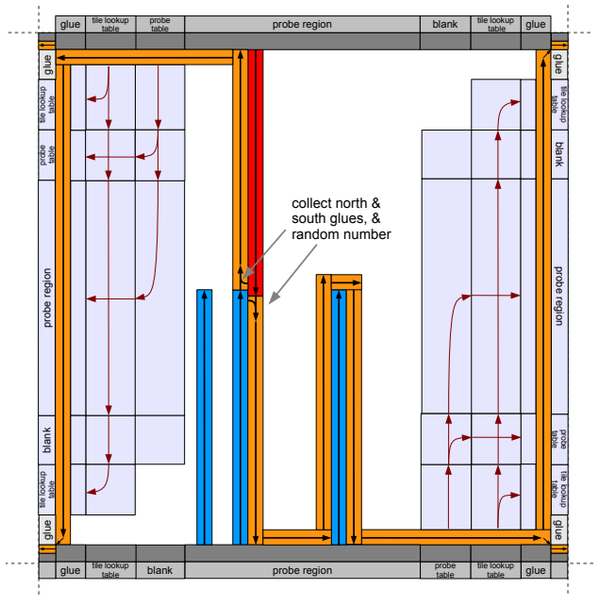
MOLECULAR DESIGNS

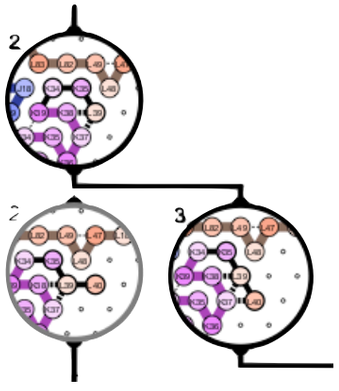
Boolean gates implemented as single DNA strand

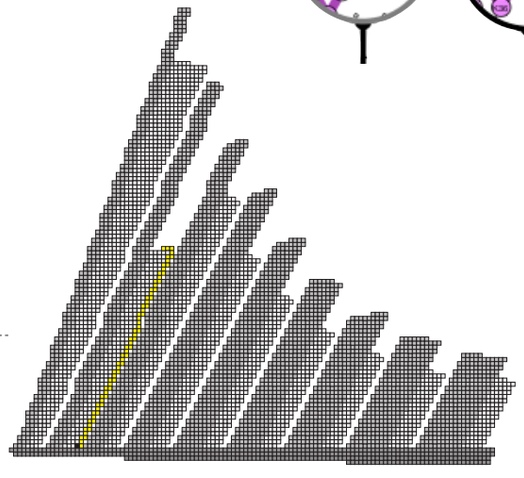


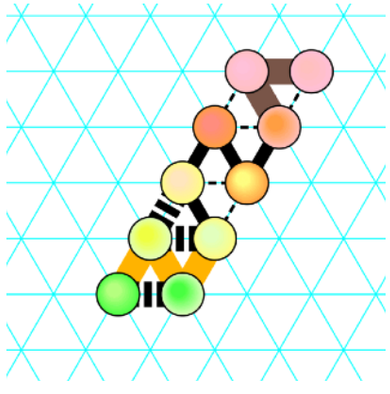
CR16 Molecular programming: Theory & experiment

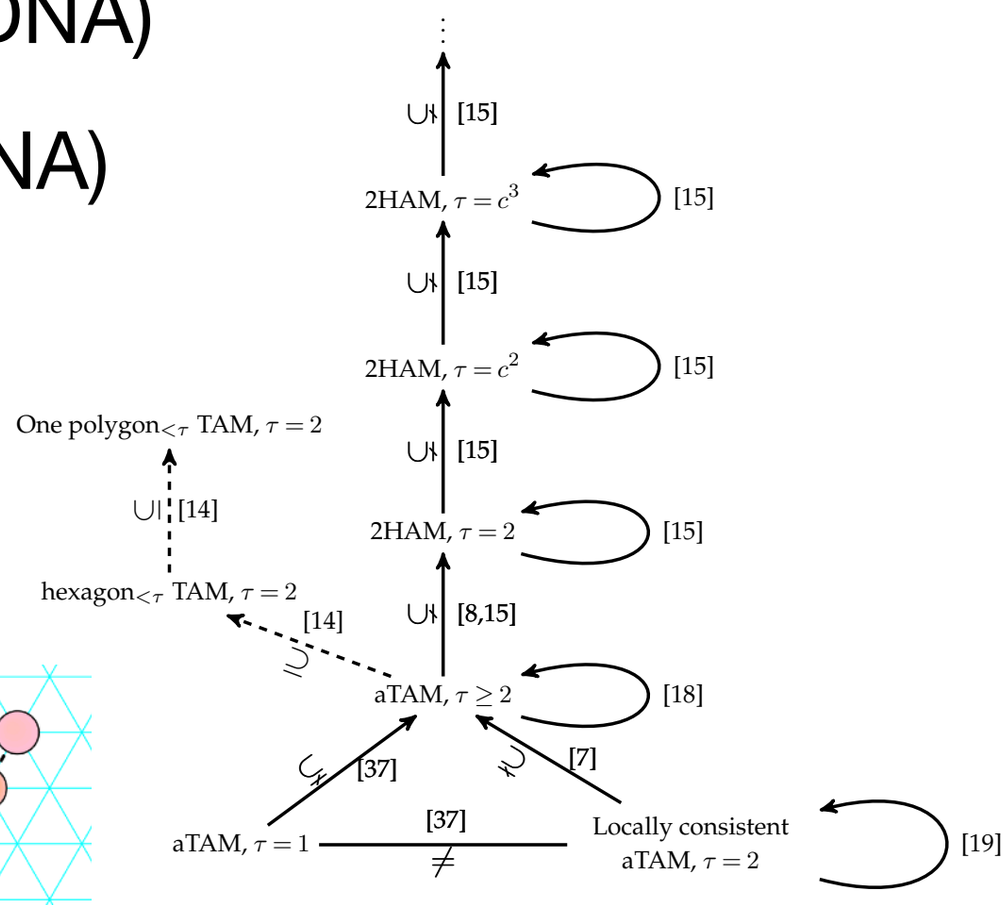
- **Theory: models, universality and classification**
  - Various tile assembly systems (DNA)
  - Oritatami systems & Turedos (RNA)







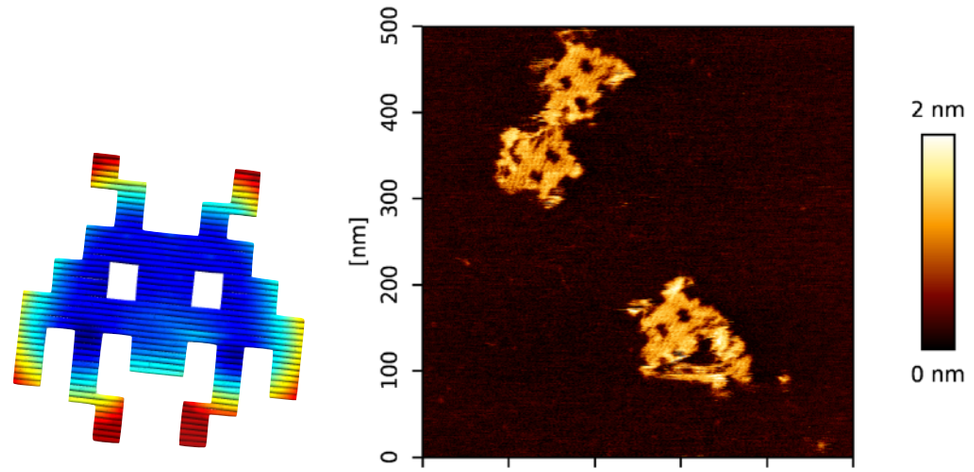


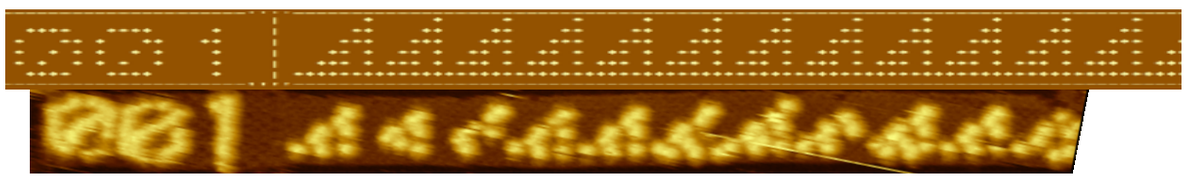


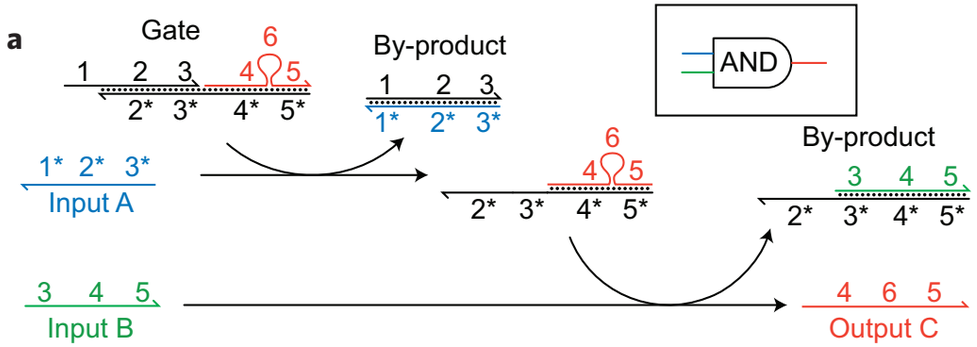
CR16 Molecular programming: Theory & experiment

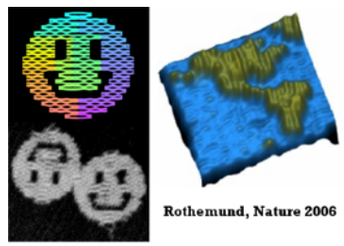
- **Experiments: Methods, protocols and… *pratice!***
  - How it works, experimental achievements
  - DNA origami
  - Single stranded tiles
  - Strand displacement circuits
  - Enzymatic molecular systems
- **Mini-project:**
  Design a DNA origami, Order, Assemble and Image it in the lab!

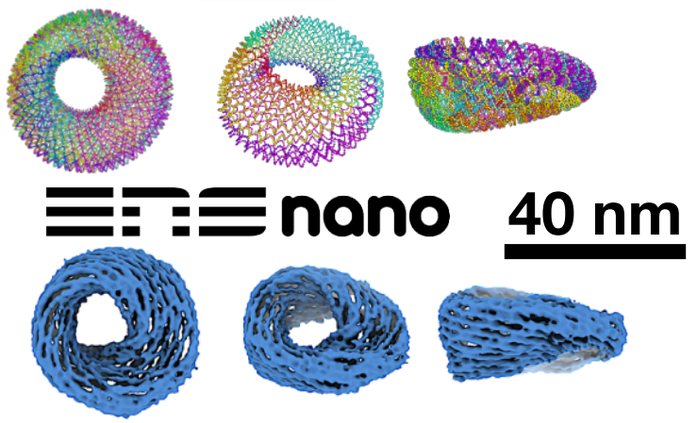
Designed and imaged during 2020 lecture !











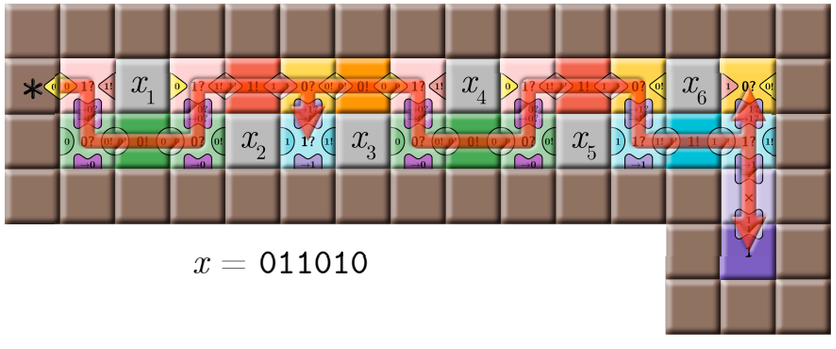
AFM microscopy

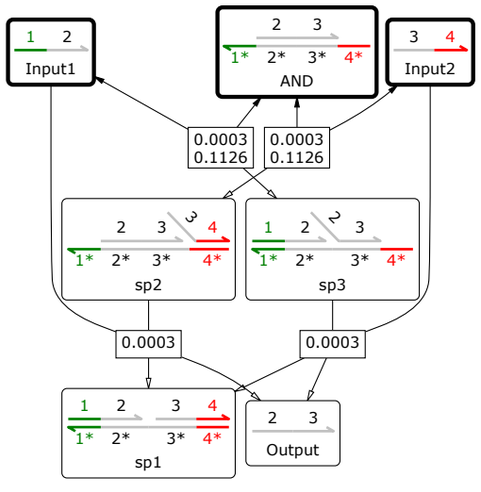


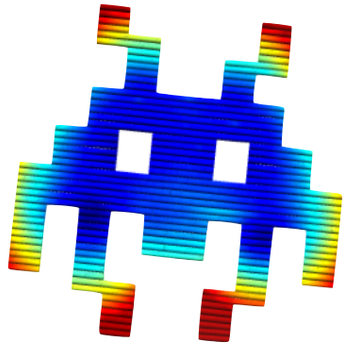
CR16 Molecular programming: Theory & experiment

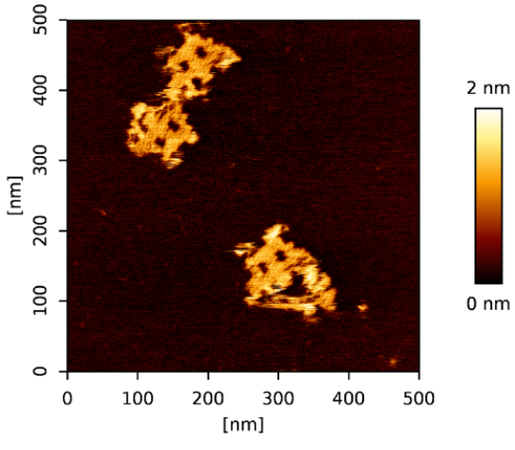
- **Evaluation: FINAL (20 pts) = Homework (10 pts) + Exam (10pts)**
  - **Homework = HW1 + HW2 + HW3 + HW4**
    HW_i = a tiny exercice to solve from one week to the next (< 30mins each)
  - **Exam** = **Mini-project (4pts)** + **Written Exam (6pts)**

- **Mini-project =** design a DNA origami from scratch using ENSnano; by groups of 3-5 people over about a month
- **Written exam** = 1h30/2h-written exam at the end of the lecture



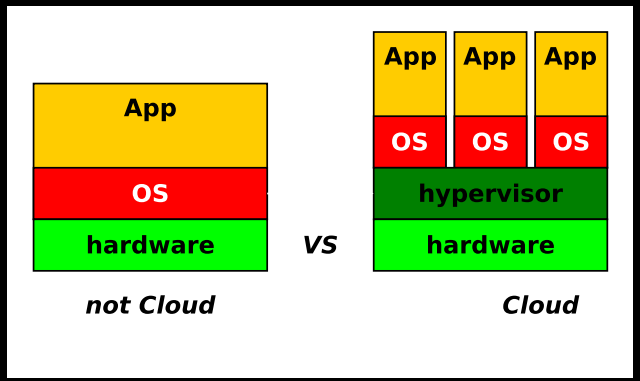






CR17-Virtualization technologies

- Virtualization is everywhere (data centers, phones, etc.)
- Allows quick deployment, optimal resource utilization, etc.
- The virtualization system is as critical as the operating system
  - both are *system software*
  - essential to understand performance issues



Class objectives

- Para-, hardware assisted and OS level virtualization
- Type-1 (Xen/VMware/Hyper-V) and Type-2 (KVM) virtualization
- Cloud models (IaaS/FaaS)
- Advanced research contributions

Prof. Alain Tchana - Ensimag (Grenoble) - LIG