Tom Cornebize

Graduate student
Theoretical computer science

35 rue des Allobroges
38180 Seyssins – France
☎ *+33(0)6 74 69 59 41*
✉ *tom.cornebize@ens-lyon.fr*
🖱 *cornebize.net*

Education

École Normale Supérieure de Lyon — **Lyon, France**
Master of science in theoretical computer science — *current*

École Normale Supérieure de Lyon — **Lyon, France**
Bachelor of science in theoretical computer science, with great honor — *2013–2014*

Université Joseph Fourier — **Grenoble, France**
Undergraduate program in computer science and mathematics — *2011–2013*

Experience

Research internships

Inria, AOSTE team — **Sophia-Antipolis, France**
Modelisation and verification of concurrent systems. — *June 2014–July 2014*
Under the supervision of Robert de Simone.

- Studied classical models (Büchi automata, Petri nets, temporal logic, synchronous languages).

Verimag laboratory — **Grenoble, France**
Monitoring of distributed systems — *June 2013–July 2013*
Under the supervision of Yliès Falcone.

- Designed an algorithm for decentralized monitoring of distributed systems.
- Implemented a benchmark to obtain experimental results.
- Wrote a report, "Efficient and Generalized Decentralized Monitoring of Regular Languages", published at FORTE 2014.

Laboratoire d'informatique de Grenoble (LIG) — **Grenoble, France**
Monitoring of distributed systems — *June 2012*
Under the supervision of Yliès Falcone.

- Performed experimentations to assess the efficiency of an algorithm for decentralized monitoring.
- Proposed several optimizations.

School projects

Projet Pensées Profondes

- A modular and open source question answering framework. Team of seven students.
- Developed a question parsing module in Python, with a grammatical approach (Stanford CoreNLP and NLTK libraries).

SAT solver

- Developed a program implementing the DPLL algorithm to solve the SAT problem, in C++ language.
- Added *watched literals* and *clause learning* heuristics.
- Added a SMT (satisfiability modulo theories) solver.

Cellular automata, in C. Used *MPI*.
Simulation of distributed search, in Erlang.
P2P client, in C. Used *pthread* and *socket*.

Languages

French: Mother tongue

English: Fluent

German: Basic

Computer skills

Programming languages: Python, C, C++, OCaml, Erlang, assembly languages (ARM and MIPS)

Presentation languages: LaTeX, Markdown

Distributed systems / parallel programming: MPI, pthread, socket

Miscellaneous: GNU/Linux, Git, unit testing