

Tom Cornebize

Étudiant en première année de master
Informatique fondamentale

+33(0)6 74 69 59 41
tom.cornebize@ens-lyon.fr
cornebize.net

Études

École Normale Supérieure de Lyon

Première année de master d'informatique fondamentale

Lyon, France

actuel

École Normale Supérieure de Lyon

Licence d'informatique fondamentale, mention bien

Lyon, France

2013–2014

Université Joseph Fourier

Première et deuxième année de licence de mathématiques et d'informatique

Grenoble, France

2011–2013

Expérience

Stages de recherche

Bull

Calcul haute performances : isolation de jobs et optimisation distribuée du routage.

Encadré par Matthieu Perotin.

Grenoble, France

Mai 2015–Août 2015

Inria, équipe AOSTE

Modélisation et vérification des systèmes concurrents

Encadré par Robert de Simone.

Sophia-Antipolis, France

Juin 2014–Juillet 2014

- Étude des modèles classiques : automates de Büchi, réseaux de Petri, logique temporelle, langages synchrones.

Verimag

Surveillance de systèmes distribués

Encadré par Yliès Falcone.

Grenoble, France

Juin 2013–Juillet 2013

- Développement d'un algorithme pour la surveillance décentralisée de systèmes distribués.
- Implémentation de l'algorithme et étude expérimentale.
- Rédaction d'un rapport, "Efficient and Generalized Decentralized Monitoring of Regular Languages", publié à FORTE 2014.

Laboratoire d'informatique de Grenoble (LIG)

Surveillance de systèmes distribués

Encadré par Yliès Falcone.

Grenoble, France

Juin 2012

- Étude expérimentale d'un algorithme de surveillance décentralisée de systèmes distribués.
- Réalisation de plusieurs optimisations.

Projets logiciels

Projet Pensées Profondes

- Un outil libre et modulaire de réponse automatique aux questions de connaissances générales.
- Développement d'un module d'analyse de question en Python, en utilisant une approche grammaticale (bibliothèques Stanford CoreNLP et NLTK).

Solveur SAT

- Développement d'un programme de résolution du problème SAT, basé sur l'algorithme DPLL, en C++.
- Ajout des heuristiques des *littéraux surveillés* et de l'*apprentissage de clauses*.
- Ajout d'un solveur SMT (satisfiabilité modulo théories).

Automate cellulaire, en C. Utilisation de *MPI*.

Simulation de recherche distribuée, en Erlang.

Client P2P, en C. Utilisation de *pthread* et *socket*.

Langues

Français: Langue maternelle

Allemand: Scolaire

Anglais: Courant

Connaissances en informatique

Langages de programmation: Python, C, C++, OCaml, Erlang, langages d'assemblage (ARM et MIPS)

Langages de présentation: \LaTeX , Markdown

Systèmes distribués et programmation parallèle: MPI, pthread, socket

Divers: GNU/Linux, Git, tests unitaires