

# À Propos

Hector Suzanne

## 2020

Voir **stage de M2**

Deuxième année du master STL. Coloration voulue sur le typage, la compilation sûre, donc UEs SVP, FLST, TAS, GRAPA. Mais SVP est annulée pour manque d'étudiant, donc au final TAS, FLST, PPC, AAGA, GRAPA.

**TAS (1/2)** Typage, avec un point de vue appliqué au langage de programmation: Lambda-calcul polymorphique à la ML; sous-typage structurel (ML) et nominal (Java); Surcharge et résolution (Java & Haskell); Typage graduel. Projet: lecture et présentation de *Selective Applicative Functors (POPL'19)*.

**TAS (2/2)** Intéprétation abstraite. Abstractions de sémantiques; Calculs d'approximation de points fixes dans un treillis pour l'analyse de boucles; Produits réduits & domaines disjonctifs; domaine relationnel polyèdres; analyse de pointeurs et tableaux; Application à l'embarqué critique. Projet: lecture et présentation de *tatic analysis of binary code with memory indirections using polyhedra*.

**GRAPA** Groupe de recherche, avec pour thématique cette année l'unification d'ordre supérieure, dans le cadre d'une implémentation de -prolog en *Closure*. Travail de recherche en groupe d'une dizaine, bibliographie et présentation. Algorithme de recherche dans un espace semi-décidable (Huet). Programmation en groupe autour de Git partagé, Intégration continue.

**FLST** UE du MPRI, prise à Paris-Diderot. Principes et formalismes des langages fonctionnels typés modernes. Implémentation de la programmation fonctionnelle; méta-théorie des langages fonctionnels typés; Aspects avancés des systèmes de types; programmation avec types dépendants en Agda. Projet: optimisateur pour le système F explicitement typé.

**AAGA** Méthodes pour la complexité en moyenne et la génération aléatoire. Application aux structures arborescentes et l'algorithmique probabiliste. Introduction à la combinatoire analytique pour l'analyse de complexité en moyenne.

**PPC** Modèles de la concurrence: réseaux de Petri, CCS, -calcul, algèbre de processus temporels. Modèles de la programmation synchrone, passage de message, acteurs.

## 2019

- M1, second semestre  
**APS**  
**CPA** langages modernes.  
**CA** Compilation des structures de contrôle de haut niveau vers les machines virtuelles. Gestion automatique de la mémoire. Passage au code natif avec analyse et optimisation du code machine.  
**CPS** Concepts de l'implémentation de logiciels à base de composants. Techniques de spécification et de vérifications. Architectures à base de composants répartis.  
**PC2R** Modèles effectifs de concurrence: mémoire partagée/distincte/répartie. fils d'exécution coopératifs et préemptifs, programmation réactive. Modèle client-serveur, déploiement d'objets répartis.  
**PSTL**  
**Anglais**
- M1 STL, premier semestre  
**ALGAV** Structures de données efficaces. Preuves de complexité. Application à la gestion et compression de données massives.  
**COMPLEX**  
**DLP** Conception et implémentation des traits de haut-niveau (genre Python) au sein d'un interpréteur et compilateur vers C écrit en Java. Introduction aux grandes bases de code. Conception guidée par le test.  
**IL** Introduction au génie logiciel. Patrons de conception en UML. Mise en place de procédures de développement.  
**LRC**  
**MLDBA**
- Atelier de dessin

## 2018

**Gestion de parc informatique** école CLE

**Conseil** agence graphique *Au dessus des nuages*

**Aide au dév numérique** Groupe Prévantis

**Enseignement/Tutorat** Alice, douée mais en échec scolaire. Maths et Physique. Obtention de son bac avec mention.

## 2017

L3 de l'ENS Cachan

**Algorithmique 1 & 2** Théorie des preuves et analyses d'algorithmes pour la correction et la complexité. Applications aux algorithmes classiques et structures de données arborescentes.

**Calculabilité/Complexité** Machines de Turing; Théorème d'incomplétude; Problèmes indécidables; Réduction. Classes de complexité; Inclusions dans les classes; NP-complétudes; réduction aux problèmes NP-complets.

**-calcul**

**Langages formels** Théorie des langages et automates; Hiérarchie de Chomsky.

**Projet Logique COQ** Réduction à SAT de problèmes réels, Preuves formelles d'algorithmes sur les structures de données élémentaires.

**Logique**

**Maths Discrètes**

**Programmation 1 & 2**

**Architecture Systèmes** Outils UNIX; Automatisation; Linux; Principe de réseaux. Projet: Création depuis les sources d'une distribution Linux complète (bash + accès au réseau).

Anglais, validé *CAE*, niveau européen *C2*.

## 2015-2016

MPSI/MP\*, Lycée Descartes, Tours

**Maths** Analyse réelle et vectorielle; équations différentielles linéaires; algèbre générale et linéaire; théorie des groupes; dénombrabilité; théorie de la mesure; statistiques et probabilités

**Physique** Mécanique newtonienne; Electro-magnétisme; Relativité restreinte; Thermodynamique; Chimie élémentaire; Thermo-chimie; Électronique.

**Informatique** Programmation de base; Algorithmique; Simulation physique; programmation fonctionnelle.

**TIPE** Développement d'un système d'algèbre sur les corps finis dans un style purement fonctionnel en Haskell, implémentation du code correcteur de Reed-Solomon, étude de l'application au stockage sûr de Big Data chez Facebook.

## Avant

- Bac scientifique
- *Tournoi Français des Jeunes Mathématiciens et Mathématiciennes*
- Apprentissage de la programmation: C, Python, Haskell, Ocaml
- Electronique de base