

Curriculum Vitæ

Fabien GIVORS

Nationalité française, né le 04/04/1986 à Aubenas (07)

Portable : (+33) 6 70 00 45 87

Courriel : fabien.givors@ens-lyon.org / Page personnelle : <http://chezlefab.net/>

Cursus

- 2011–2013** : 3e et 4e années de doctorat sous la direction de Bruno Durand et Gregory Lafitte, LIRMM, Université Montpellier 2, Montpellier, France : *Vers une structure fine des calculabilités (Towards a Fine Structure of Computabilities)*.
- 2009–2011** : 1ère et 2e années de doctorat sous la direction de Bruno Durand et Gregory Lafitte, LIF, CMI, Université de Provence, Marseille, France : *Vers une structure fine des calculabilités (Towards a Fine Structure of Computabilities)*.
- 2008–2009** : Master 2 d'Informatique – parcours Système complexes (mention Très Bien, classement : 2/6), Université de Nice Sophia-Antipolis (en coopération avec l'ÉNS Lyon), Nice, France
- 2007–2008** : Master 1 d'Informatique Fondamentale, Université de Nice Sophia-Antipolis (en coopération avec l'ÉNS Lyon), Nice, France
- 2006–2007** : Licence d'Informatique Fondamentale (mention Bien), École Normale Supérieure de Lyon, Lyon, France
- 2004–2006** : Classe préparatoire MPSI / MP, Lycée la Martinière Monplaisir, Lyon, France

Compétences métier

- Travail en équipe (notamment avec des pairs non francophones et géographiquement éloignés) ou seul en autonomie
- Présentation de travaux
- Transmission des connaissances
- Apprentissage rapide (langages, frameworks, technologies, science)
- Résolution de problèmes (recherche de documentation et de nouvelles solutions techniques, débogage)
- Veille (technologique, scientifique)
- Encadrement de stage et de projets
- **Langues parlées :**
 - Français (langue maternelle)
 - Anglais (courant)
 - Italien (à réactiver)
 - Japonais (débutant)

Compétences techniques

Programmation, langages de scripts, langages Web et autres.

Langage	Expérience	Langage	Expérience
C	élevée	HTML	élevée
C++	élevée	XML	élevée
OCaml	élevée	CSS	élevée
Haskell	faible	JavaScript	faible
Java	faible	PHP	élevée
Python	élevée	SQL (MySQL)	élevée
Perl	faible	Git	élevée
Ruby	faible	Subversion	élevée
Shell POSIX	élevée	darcs	faible
Makefile	élevée	mercurial	faible
L ^A T _E X/ Beamer	élevée	Coq	faible

Administration système

Outils	Expérience	Détails
Debian Linux	très élevée	Mainteneur Debian, multiples postes et serveurs (10 ans)
Systemd	élevée	
Apache2/Nginx	élevée	Administration de multiples sites web, vhosts, reverse-proxy, SSL
MySQL	élevée	Gestion des utilisateurs, sauvegarde, réplication Master-{Slave,Master}
Munin/Nagios/NRPE	bonne	Monitoring de cluster de serveurs
OpenVPN	bonne	
Bind9	bonne	Zones DNS, DNSSEC
SSH	élevée	chroots des utilisateurs, sftp, clés et automatisation de commandes
Iptables/Netfilter	bonne	pare-feu, routage
LVM2	bonne	
Xen	bonne	Administration d'un hyperviseur et de ses VM
LXC/cgroups/OpenVZ	élevée	chroots cloisonnés / virtualisation légère
VirtualBox/KVM	bonne	
Postfix/Dovecot	bonne	

Enseignement (546h)

- 2014–2015** : Attaché Temporaire d'Enseignement et de Recherche, 192h, Université de Nice–Sophia-Antipolis
2012–2013 : Attaché Temporaire d'Enseignement et de Recherche, 176h, Université de Montpellier II
2011–2012 : Mission complémentaire d'enseignement, 64,5h, Université de Montpellier II
2008–2009 : Vacances, 20h, Université de Nice Sophia-Antipolis

Intitulé	Niveau
Outils Formels pour l'Informatique	L2
Calculabilité	M1
Programmation Web Serveur	L1
Introduction Web	L1
Réseaux	L3
Systèmes d'exploitation	L2
Simulation et modèles discrets	L2
Systèmes informatiques (UNIX)	L1
Histoire de l'informatique	L1

Intitulé	Niveau
Concepts de base en informatique (C2I)	L1
Systèmes d'exploitation	L3
Théorie algorithmique de l'information	M1/M2
Programmation impérative (C)	L1
Analyse d'algorithmes	L2

Responsabilités collectives

- Organisation de groupes de travail pendant deux ans ;
- Mise en place d'une Plateforme collaborative ;
- Aide à l'installation de machines au CINES pour le projet HPC@LR ;
- Aide à l'encadrement d'étudiants dans le cadre de projets TER.

Présentations courtes, ateliers et publications

- Sous-calculabilités, Groupe de travail MC3, Nice, France, 2014 ;
- Vers une structure fine des calculabilités, Lirmm, Montpellier, France, 2013 ;
- Higher Sub-Computabilities (w/ G. Lafitte), FRAC d'hiver 2012, Créteil, France, 2012 ;
- Sous-Calculabilités, SeminDoc Lirmm, Montpellier, France, 2012 ;
- (Sub-)Computabilities (w/ G. Lafitte), CRP 2011, Paris, France, 2011 ;
- Forcing et degrés Turing (w/ G. Lafitte), EJC 2010, Chambéry, France, 2010.
- Sub-Computabilities (w/ G. Lafitte), FCT 2011, Oslo, Norway, 2011 ;
- Infinite time cellular automata : a real computation model (w/ G. Lafitte et N. Ollinger), JAC 2010.
- Holes punched computabilities (w/ G. Lafitte), LICS 2011, Toronto, Canada, 2011.

Activités annexes

- **janvier–août 2014** Bénévolat pour l'association de loi 1901 Debian-Facile.
- **janvier 2011**– Président de l'association de loi 1901 Debian-Facile qui a pour objet l'aide et la mise à disposition de plateformes pour l'entraide à l'utilisation du système d'exploitation Debian et de ses dérivées.
- **mai 2010**– Administrateur du serveur dédié de Debian-Facile, hébergeant divers services : Web (plusieurs applications web, HTTPS), Git, E-Mail, etc.
- **mai 2007–2012** Administration d'un serveur de virtualisation Xen pour un groupe d'utilisateurs privés.
- Contributions au système d'exploitation Debian : Mainteneur Debian du paquet *obsession*

Résumé des travaux de thèse

Mots clefs : *Calculabilité, Sous-récursion, Théorie de la récursion d'ordre supérieur, Ordinaux rékursifs, Logique Mathématique.*

Mon manuscrit de thèse est consultable ici : <http://chezlefab.net/pub/memoires-2013-phd.pdf>.

Mon activité de recherche porte principalement sur l'étude des classes de fonctions calculables. En particulier, j'exhibe au fil de mes travaux différentes hiérarchies de classes de fonctions rékursives, sur les entiers et sur les ensembles, permettant des liens entre la calculabilité, la sous-récursion et la théorie de la récursion d'ordre supérieur, le tout grâce à des outils empruntés à la logique mathématique.

Présentation : Calculabilité, puissance de calcul et théories mathématiques Les fonctions calculables, telles qu'identifiées et formalisées par Church, Turing, et d'autres sont un peu à l'informatique théorique ce que les ensembles sont aux mathématiques. Elles constituent les objets de la théorie de la récursion et sont soumises à des mécanismes similaires. La calculabilité se concentre sur la comparaison des difficultés de calcul entre ces objets, afin d'identifier une structure de degrés de difficultés du calcul, par exemple la structure des degrés Turing.

Ces fonctions travaillent sur les entiers naturels, qui permettent d'encoder toutes les descriptions d'objets formalisables, et donc en particulier, de décrire ces fonctions elles-mêmes. Cette imprédictivité à la base des questions liées à la logique, telle l'indécidabilité du problème de l'arrêt, et intrinsèquement liée à la notion de calcul.

Aussi, rien d'étonnant à ce que l'étude des difficultés du calcul soit liée à des puissances de théories mathématiques et de systèmes inductifs. Les ordinaux sont un outil de la logique mathématique nous permettant justement de mesurer et hiérarchiser la puissance des différentes théories mathématiques, et d'identifier (et même de construire) des classes de fonctions calculables totales, prouvablement calculables et totales dans ces théories.

La grande inconnue des sous-calculabilités est la structure des classes de fonctions correspondant à la puissance de théories mathématiques données. Ces classes correspondent à des fragments de la calculabilité, qui doivent avoir des propriétés structurelles similaires. Pourtant, ils n'avaient jamais été étudiés sous le point de vue calculabiliste, et c'est ce que je m'emploie à faire.

Résultats : état de l'art

Des structures méconnues Mes travaux commencent par l'étude des classes de fonctions rékursives contenant les fonctions constantes, les fonctions de pairing et de projections, qui soient stables par composition et, qui disposent d'une fonction universelle réursive (hors de la classe). De telles classes sont appelées classes de fonctions énumérables, cette notion étant empruntée à Dexter Kozen qu'il a introduite dans un papier de TCS en 1980 alors qu'il tentait d'identifier des propriétés structurelles de certaines classes de complexité, en s'appuyant sur le procédé de diagonalisation.

Lorsque l'on parle de machine universelle, on sous-entend en général qu'il y a un encodage, un langage de programmation, qui soit interprétable par une machine : la machine universelle. Bien sûr, toute la difficulté est que si l'on demande à cette machine universelle d'être calculable, alors toute la calculabilité apparaît d'un coup, avec des objets extrêmement difficiles à appréhender ou à décrire.

Or, si l'interprétation d'une machine est quelque chose de difficile, il n'est pas nécessaire de savoir faire cela pour analyser ou modifier la description de la-dite machine. Plus formellement, la puissance de calcul requise pour assembler/composer des encodages de machines existantes est très faible.

Nous appelons classe fondamentale une classe de fonctions totales capables de faire cela, et ce sont ces classes qui sont à la base de notre étude. Les classes de fonctions partielles décrites par ces classes fondamentales sont alors des approximations de la calculabilité, mais composées uniquement de fonctions *facilement* descriptibles, en terme de puissance de calcul.

Une des sous-classes de fonctions calculables les plus connues est la classe des fonctions rékursives primitives. Nous avons tous appris qu'il n'était pas possible d'avoir une fonction universelle Turing parmi ces fonctions. Pourtant, cela ne nous empêche pas de pouvoir prouver une grande partie des théorèmes de calculabilité classique sur cette classe.

En fait, la manipulation de descriptions de fonction, et donc par exemple les opérations de compositions, sont à la portée de celles-ci. Une première partie de mes travaux a consisté à identifier les propriétés minimales que devait respecter une classe de fonctions afin de permettre une manipulation algébrique des fonctions calculables similaire à celle de la calculabilité.

Les travaux présentés ci-dessus permettent de remarquer une structure très proche de celle de la calculabilité sur ces classes qui a priori ne peuvent pas être des classes de même puissance expressive. Pourtant, les théorèmes classiques (Isomorphisme de Myhill, Rogers, s_n^m , point fixe de Kleene) se retrouvent tous, d'une certaine manière dans ces classes où les classes de fonctions partielles qu'elles définissent.

Les objets classiques de calculabilité (ensembles rékursivement énumérables, rékursifs, créatifs, productifs), etc. sont également définis naturellement, raffinant ainsi les notions associées, comme par exemple la notion de *rékursivité* d'un ensemble.

Un point de vue original L'étude des sous-calculabilités met en lumière tout un ensemble de classes de fonction qui n'avaient pas été étudiées notamment parce que ni leurs propriétés de clôture ni leurs définitions ne paraissaient pas naturelles.

La présentation par difficulté de preuve des fonctions (et donc de définition de celles-ci) permet d'amener naturellement ces classes de *complexité* et même de retrouver des propriétés structurelles mimant la calculabilité classique, tout en affinant le point de vue.

Un formalisme unifié Finalement, les notions de classes fondamentales sont définies suivant un formalisme suffisamment général pour que celles-ci ne se limitent pas aux fonctions calculables sur les entiers.

En fait, le cadre défini par les classes fondamentales et les sous-calculabilités permet d'englober également les *sur-calculabilités*, c'est à dire les calculabilités d'ordre supérieur comme la théorie de la récursion sur les ordinaux admissibles.

Derrière ce formalisme de calculabilité, nous avons réussi à réunir toute une hiérarchie de classes de fonctions, partant des fonctions sous-récurrentes (habituellement considérées comme un domaine à part) jusqu'aux fonctions récursives sur les ensembles constructibles.