

Raphaël Monat

CHARGÉ DE RECHERCHE À L'INRIA, LILLE
MOTS-CLÉS : SCIENCES DU LOGICIEL, MÉTHODES FORMELLES, SÛRETÉ DU LOGICIEL

✉ raphael.monat@inria.fr | 🏠 rmonat.fr | 📅 28/07/1995 | 📁 gitlab.com/rmonat

Formations

Thèse sous la direction d'Antoine Miné

Analyse statique de type et de valeur, par interprétation abstraite, de programmes Python avec des bibliothèques C.

LIP6, Sorbonne Université
Septembre 2018 – Novembre 2021

	Antoine Miné	Sorbonne Université, France	Directeur
	Isabella Mastroeni	Università di Verona, Italie	Rapportrice
	Anders Møller	Aarhus Universitet, Danemark	Rapporteur
Jury :	Emmanuel Chailloux	Sorbonne Université, France	Président du jury
	Francesco Logozzo	Facebook Seattle, USA	Examineur
	Peter Müller	ETH Zürich, Suisse	Examineur
	Alan Schmitt	Inria Rennes, France	Examineur

Master 2 Master Parisien de Recherche en Informatique

Moyenne 16.81/20

Université Paris Diderot
2017-2018

Master 2 Informatique Fondamentale

Moyenne 16.49/20

École Normale Supérieure de Lyon
2016-2017

Master 1 Informatique Fondamentale

Moyenne 16.36/20

École Normale Supérieure de Lyon
2015-2016

Licence 3 Informatique Fondamentale

Moyenne 16.43/20

École Normale Supérieure de Lyon
2014-2015

Entrée à l'École Normale Supérieure de Lyon sur le concours informatique

École Normale Supérieure de Lyon
2014

Classes préparatoires MPSI/MP* option informatique

Lycée Louis-le-Grand, Paris
2012-2014

Baccalauréat scientifique, mention très bien

Lycée Aristide Bergès, Isère
2012

Expériences professionnelles

Chargé de recherche

Équipe SyCoMoRES

Inria @ Université de Lille
Depuis Septembre 2022

Attaché temporaire d'enseignement et de recherche

Recherche dans l'équipe APR (dirigée par Antoine Miné). Enseignements dans l'UFR d'ingénierie (192h).

LIP6, Sorbonne Université
Septembre 2021 - Août 2022

Doctorant contractuel avec mission complémentaire d'enseignement

Recherche sous la direction d'Antoine Miné. Enseignements dans l'UFR d'ingénierie (64h/an).

LIP6, Sorbonne Université
Septembre 2018 - Août 2021

Stage de recherche de M2 sous la direction d'Antoine Miné

Création d'une analyse relationnelle de typage pour Python.

LIP6, Sorbonne Université
Mars – Juin 2018

J'ai défini de manière théorique une analyse de programmes Python proche d'un système de type avec du polymorphisme paramétrique. Cette analyse utilisait un unique domaine abstrait monolithique, qui pouvait exprimer que certaines variables avaient un même type, celui-ci variant dans un ensemble. L'implémentation permettait d'obtenir des résultats sur des petits programmes de moins de 50 lignes de code. Ma première année de thèse a consisté à reprendre ces travaux pour séparer le domaine abstrait en plusieurs domaines indépendants, et améliorer significativement le passage à l'échelle de l'analyse.

Stage de recherche de M2 sous la direction d'Eva Darulova

Extension d'une formalisation en Coq et HOL4 pour un synthétiseur de programmes sur les flottants.

MPI-SWS, Sarrebruck, Allemagne
Février – Juin 2017

Le groupe de vérification automatique et d'approximation (AVA) du Max Planck Institute for Software Systems, dirigé par Eva Darulova, développe un outil appelé Daisy, permettant de compiler des programmes numériques idéalisés, définis sur les réels, en des programmes calculant sur les flottants, avec une marge d'erreur donnée en sortie par le compilateur. Ce compilateur était capable d'optimiser l'équilibre entre précision et performance des programmes grâce à l'utilisation de plusieurs précisions pour les flottants. Un doctorant du

groupe, Heiko Becker, avait établi une formalisation (en Coq et HOL) permettant de vérifier via des certificats que la compilation utilisant une analyse d'intervalles et une précision fixée était correcte. J'ai généralisé cette formalisation pour prouver la compilation correcte dans le cas d'une précision mixte. J'ai aussi travaillé à formaliser une analyse basée sur l'arithmétique affine. Une publication de groupe sur ce travail a été acceptée à FMCAD 2018.

Stage de recherche de M1 sous la direction de Hongseok Yang

Inférence variationnelle pour les langages de programmation probabilistes.

Oxford, Royaume-Uni

Mai – Juillet 2016

Les modèles probabilistes sont utilisés dans de nombreux domaines pour résoudre différents problèmes, allant de la reconnaissance d'images au diagnostic de maladies. L'utilisation de modèles permet de séparer l'encodage du problème (dans un modèle probabiliste) de sa résolution, et d'avoir des algorithmes d'inférence spécifiques à chaque classe de modèle. Une approche à l'intersection des langages de programmation et de l'apprentissage automatique est la programmation probabiliste. Cette approche permet de généraliser la notion de modèle probabiliste, et a pour but de pouvoir réaliser automatiquement l'inférence à l'exécution des programmes probabilistes. J'ai travaillé avec Hongseok Yang à formaliser les programmes probabilistes sous une forme de système de transition probabiliste. Le problème d'inférence peut être vu comme un problème d'optimisation, que nous avons simplifié pour dériver un nouvel algorithme d'inférence variationnel général pour les programmes probabilistes.

Stage de recherche de L3 sous la direction d'Antoine Miné

Développement d'une analyse de programmes concurrents avec une mémoire à cohérence séquentielle.

Antique, ENS Ulm

Juin – Juillet 2015

J'ai utilisé le cadre de l'interprétation abstraite pour développer, implémenter et tester de nouvelles méthodes pour analyser plus précisément des programmes concurrents. Le modèle mémoire considéré est celui de la cohérence séquentielle. J'ai étendu une approche d'analyse modulaire sur les threads, développée précédemment par Antoine et basée sur la logique "Rely-Guarantee", pour améliorer la précision de l'analyse dans la prise en compte des interférences entre les threads. J'ai développé à partir de zéro un prototype, appelé Batman, permettant d'analyser des programmes concurrents jouets. Ce prototype est écrit en OCaml et utilise la bibliothèque de domaines abstraits numériques Apron. J'ai comparé les résultats obtenus à ceux d'un autre outil appelé ConcurInterproc. J'ai étendu ces travaux en dehors de mon stage et publié ces résultats avec Antoine Miné à VMCAI 2017.

Enseignements

Sauf mention contraire, les enseignements détaillés ci-dessous ont été effectués à Sorbonne Université, dans l'UFR d'ingénierie, auprès d'étudiants en informatique, et les enseignements en master ont été effectués dans le cadre du master STL "Sciences et Techniques du Logiciel".

Acronymes :

- TME : travaux sur machine encadrés
- CM : cours magistral

Type d'enseignement effectué par niveau

	L1	L2	L3	M1	M2
CM					4h
TD	60h	52h	30h	60.5h	14h
TME	58h	71h	29.5h		24h
Total	118h	104h	59.5h	60.5h	42h

Type d'enseignement effectué par thème

	L1	L2	L3	M1	M2
Info. théorique	20h	64h		52h	
Programmation	98h	40h	31.5h		
Spécialité			28h	8.5h	42h
Total	118h	104h	59.5h	60.5h	42h

Info. théorique : mathématiques discrètes, logique, algorithmique
Spécialité : compilation, analyse et preuve de programmes

L1	Éléments de Programmation 2 (C) , Chargé de TD/TME (20 étudiant-e-s, 38.5h) Correction des examens.	2021–2022
L1	Éléments de Programmation 1 (Python) , chargé de TD/TME (26 étudiant-e-s, 38.5h) Création et correction de 4 interrogations de TD, et correction des examens.	2019–2020
L1	Éléments de Programmation 1 (Python) , chargé de TME (29 étudiant-e-s, 19.25h) Correction d'une épreuve machine. Amélioration des supports.	2018–2019
L1	Éléments de Programmation 1 (Python) , remplacement en TD (30 étudiant-e-s, 1.75h)	2018–2019
L1	Ateliers de Recherche Encadrée , chargé de TD/TME (4 binômes, 20h) Création des sujets proposés (arbres bicolores et enveloppes convexes).	2018–2019
L2	Logique , Chargé de TD/TME (33 étudiant-e-s, 19.25h) Correction des examens.	2021–2022
L2	Mathématiques Discrètes , chargé de TD/TME (33 étudiant-e-s, 44.5h)	2021–2022

	Deux interrogations de TD à préparer et corriger. Correction des examens et des projets. Amélioration des supports.	
L2	Programmation Fonctionnelle (OCaml) , chargé de TD/TME (26 étudiant.e-s, 19.25h) Création de 2 sujets de TME et intégration dans la plateforme LearnOcaml. Correction des examens. Amélioration des supports.	2021–2022
L2	Programmation Fonctionnelle (OCaml) , remplacement ponctuel en TME (27 étudiant.e-s, 1.75h)	2019–2020
L2	Fonctions et Procédures de Calcul (OCaml) , chargé de TME (10 étudiant.e-s, 19.25h) Co-création d'un sujet de projet, et correction du projet.	2018–2019
L3	Programmation Objet Avancée (Java) , chargé de TD/TME (27 étudiant.e-s, 31.5h) Correction des examens. Amélioration des supports.	2020–2021
L3	Compilation , chargé de TD/TME (12 étudiant.e-s, 28h) Correction des examens. Amélioration des supports.	2018–2019
M1	Algorithmique Avancée , chargé de TD (31 étudiant.e-s, 32h) Correction des examens. Amélioration des supports. Correction d'un projet de programmation.	2021–2022
M1	Algorithmique Avancée , chargé de TD (31 étudiant.e-s, 20h) Correction des examens. Amélioration des supports.	2019–2020
M1	Projet STL , co-encadrant de binômes sur des projets (8.5h)	2019–2020
M2	MPRI¹ : Interprétation abstraite , intervenant CM (15 étudiant.e-s, 2h) Création du support pour mon intervention (résultats obtenus durant ma thèse). Cours donné en anglais.	2021–2022
M2	Typage et Analyse Statique , chargé de TD/TME (31 étudiant.e-s, 28h) Refonte des sujets de TD. Correction d'un projet, et oral d'évaluation des étudiants sur des articles de recherche.	2021–2022
M2	Spécification et Validation de Programmes (Coq) , chargé de TD/TME (10 étudiant.e-s, 10h) Création des quatre sujets de TME.	2021–2022
M2	Groupe de Recherche en Algorithmique et en Programmation , intervenant CM (12 étudiant.e-s, 2h) Création du support pour mon intervention (présentation de mon sujet de thèse et de ce qu'est une thèse).	2019–2020

Publications

Dans le domaine des méthodes formelles dont je fais partie, *les articles sont majoritairement publiés dans des conférences internationales avec comité de sélection*. Les articles dans les journaux avec comités de sélection sont plus rares. *Les auteur·rice·s sont usuellement ordonné·e·s par contribution décroissante*. Pour les publications parlant de travaux de groupe sur Mopsa (VSTTE 2019 et JFLA 2021), nous avons choisi d'utiliser un ordre alphabétique. Mes publications sont ordonnées par ordre chronologique décroissant. SAS est la conférence spécialisée dans le domaine de l'analyse statique par interprétation abstraite.

Politique de publication. Je préfère soumettre mes travaux à des conférences qui ont des politiques de publication ouverte avec des frais raisonnables pour les auteurs (proches du prix coûtant, $\leq 100\text{€}$), ou qui sont atteignables sans prendre l'avion. ECOOP² est la conférence du domaine respectant le plus ces critères, puisqu'elle a lieu en Europe et qu'elle utilise les actes ouverts LIPIcs³ (les autres conférences du domaine utilisent l'ACM ou Springer comme éditeurs).

Légende :

- Titre en noir : auteur principal (gris sinon).
- Artefacts logiciels :    (ACM),  (autres).
-  article double colonne ( sinon).
-  article démonstration d'un outil.

Les titres sont cliquables et renvoient aux informations de publication sur mon site (avec des liens vers une version publique et la version de l'éditeur, au minimum); les acronymes des conférences pointent vers le programme de l'événement correspondant; les badges d'artefacts logiciels vers l'artefact validé par les pairs.

CONFÉRENCES INTERNATIONALES AVEC COMITÉ DE LECTURE

A Multilanguage Static Analysis of Python Programs with Native C Extensions

Raphaël Monat, Abdelraouf Ouadjaout, Antoine Miné

SAS 2021
23 pages   

A Modern Compiler for the French Tax Code

















Denis Merigoux, Raphaël Monat, Jonathan Protzenko

CC 2021
11 pages   

1. MPRI : master parisien de recherche en informatique, spécialisé en informatique fondamentale et à destination des étudiants poursuivant dans la recherche. Formation supervisée par l'université de Paris, ENS Ulm, ENS Paris-Saclay, École Polytechnique, Telecom Paris.

2. <https://ecoop.org/>

3. <https://www.dagstuhl.de/en/publications/lipics>

Static Type Analysis by Abstract Interpretation of Python Programs Raphaël Monath, Abdelraouf Ouadjaout, Antoine Miné	ECOOP 2020 27 pages  
Combinations of Reusable Abstract Domains for a Multilingual Static Analyzer (Invité) Matthieu Journault, Antoine Miné, Raphaël Monath, Abdelraouf Ouadjaout	VSTTE 2019 16 pages 
A Verified Certificate Checker for Finite-Precision Error Bounds in Coq and HOL4 Heiko Becker, Nikita Zyuzin, Raphaël Monath, Eva Darulova, Magnus O. Myreen, Anthony Fox	FMCAD 2018 8 pages 
Precise Thread-Modular Abstract Interpretation of Concurrent Programs using Relational Interference Abstractions Raphaël Monath, Antoine Miné	VMCAI 2017 17 pages 
WORKSHOP INTERNATIONAUX AVEC COMITÉ DE LECTURE	
Value and Allocation Sensitivity in Static Python Analyses Raphaël Monath, Abdelraouf Ouadjaout, Antoine Miné	SOAP 2020 6 pages 
CONFÉRENCES NATIONALES AVEC COMITÉ DE LECTURE	
Mlang : an Open-Source Toolchain for the Income Tax Computation Denis Merigoux, Raphaël Monath	JFLA 2021 2 pages  
Démonstration de la plateforme Mopsa d'analyse statique de programmes par interprétation abstraite Matthieu Journault, Antoine Miné, Raphaël Monath, Antoine Miné	JFLA 2021 2 pages  
Étude formelle de l'implémentation du code des impôts Denis Merigoux, Raphaël Monath, Christophe Gaie	JFLA 2020 16 pages 
MANUSCRITS ET RAPPORTS TECHNIQUES	
Static Type and Value Analysis by Abstract Interpretation of Python Programs with Native C Libraries Raphaël Monath	Manuscrit de thèse 275 pages 
Static Analysis by Abstract Interpretation Collecting Types of Python Programs Raphaël Monath	Rapport de stage (M2 2018) 20 pages 
Certificate Checking in Coq and HOL4 for Static Analyses of Mixed-Precision Floating-Point Arithmetic Raphaël Monath	Rapport de stage (M2 2017) 24 pages 
Variational Inference in Probabilistic Programs : Formal Derivation of a Black-box Approach Raphaël Monath	Rapport de stage (M1) 26 pages 
Thread-Modular Analysis Designing Relational Abstractions of Interferences Raphaël Monath	Rapport de stage (L3) 21 pages 

Exposés

Le temps indiqué est celui de l'exposé, sans les questions. Titres cliquables vers mon site.

EXPOSÉ INVITÉ

01/12/21 **A Multilanguage Static Analysis of Python/C Programs with Mopsa**, 25 minutes

Facebook TAV

“Testing and Verification (TAV) Symposium brings together academia and industry in an open environment to exchange ideas and *showcase the top experts from testing and verification scientific research and practice.*” (source)

EXPOSÉS DANS DES CONFÉRENCES

18/10/21	A Multilanguage Static Analysis of Python Programs with Native C Extensions , 15 minutes	SAS
07/04/21	Mlang : an Open-Source Toolchain for the Income Tax Computation , 15 minutes	JFLA
03/03/21	A Modern Compiler for the French Tax Code , 12 minutes	CC
15/11/20	Static Type Analysis by Abstract Interpretation of Python Programs , 15 minutes	ECOOP
15/06/20	Value and Allocation Sensitivity in Static Python Analyses , Best Presentation Award . 20 minutes	SOAP
30/01/20	Étude formelle de l'implémentation du code des impôts , 20 minutes	JFLA
07/10/19	Static Type Analysis of Python Programs : A Type Abstract Domain for Python , 20 minutes	DS@FM

SÉMINAIRES

28/04/22	Formal methods for realistic systems : a study of two cases , 45 minutes	LIP, ENS de Lyon
17/03/22	A Multilanguage Static Analysis of Python/C Programs with Mopsa , 30 minutes	SRG, Imperial, London
31/01/22	Static Type and Value Analysis by Abstract Interpretation of Python Programs with Native C Libraries , 40 minutes	MTV, LaBRI, Bordeaux
31/01/22	A Multilanguage Static Analysis of Python/C Programs with Mopsa , 30 minutes	SyCoMoRES, CRISTAL, Lille
02/12/21	A Modern Compiler for the French Tax Code , 40 minutes	IRILL, Paris
29/11/21	Static Analysis of Python Programs with Native C Libraries , 30 minutes	CASH, LIP, Lyon
26/11/21	A Multilanguage Static Analysis of Python/C Programs with Mopsa , 30 minutes	Binsec, CEA, Saclay
19/11/21	A Multilanguage Static Analysis of Python/C Programs with Mopsa , 30 minutes	Celtique, IRISA, Rennes
05/07/21	A Multilanguage Static Analysis of Python Programs with Native C Extensions , 20 minutes	LIP6, Paris
12/03/21	Mopsa, a Multi-lingual Static Analysis Platform , 20 minutes	GT LVP, GDR GPL
04/11/19	Formal study of the French tax code's implementation , 60 minutes	INRIA Cambium, Paris
23/10/18	Static Analysis by Abstract Interpretation of Dynamic Programming Languages , 20 minutes	LIP6, Paris

POSTER

02/07/19	Semantics & Static Type Analysis of Python Programs	Journée doctorants SIF
----------	--	------------------------

Participation à la vie de la communauté

MEMBRE DU COLLÈGE CODE SOURCE ET LOGICIEL LIBRE DU COMITÉ NATIONAL POUR LA SCIENCE OUVERTE

Mars 2022-...

CONSEIL DU LABORATOIRE DU LIP6 (MEMBRE ÉLU, 1H/MOIS)

Avril 2021-...

CONSEIL DES DOCTORANTS DU LIP6 (1H/MOIS)

Avril 2021-...

COMITÉ DE PROGRAMME

SAS'22

COMITÉ D'ÉVALUATION D'ARTEFACTS LOGICIELS (30 À 40H PAR CONFÉRENCE)

SPLASH'22, PLDI'22, CAV'22, ECOOP'21, PLDI'21, POPL'21, SAS'20

COMITÉ DE REVUE EXTERNE

SPLASH'22

RELECTEUR EXTERNE

Science of Computer Programming, SAS'21, ACM TECS, SOAP'21, LOPSTR'19

ÉTUDIANT BÉNÉVOLE

POPL'17

Logiciels développés

Noms des logiciels cliquables.

Mopsa, plateforme open-source d'analyse statique de programmes

Depuis Septembre 2018

Contributeur principal. 60kLoc Ocaml. Développement et maintenance de l'analyse Python (13kLoc) et multilangage Python/C (2.7kLoc). Licence LGPL v3. Développement suite à l'octroi d'un financement

"Consolidator Grant" de l'ERC à Antoine Miné.

Autoévaluation INS2I : A3 S04 SM3 EM2 SDL4 DA4 CD4 MS4 TPM4.

Mlang, compilateur open-source pour le code des impôts

Depuis Mai 2019

Contributeur principal. 10kLoc OCaml. Mlang permet de répliquer le calcul de l'impôt sur le revenu.

Licence GPL v3.

Autoévaluation INS2I : A4 S02 SM3 EM4 SDL4 DA4 CD4 MS4 TPM4. Accompagnement de la DGFiP pour la migration d'un ancien compilateur vers Mlang (lien vers communiqué).

Daisy, compilateur de programmes numériques sur les réels vers des flottants.

Février – Juin 2017

J'ai participé au développement de Daisy. Daisy est écrit en 14kLoc Scala.

Autoévaluation INS2I : A3 S03 SM3 EM3 SDL4 DA1 CD2 MS1 TPM1.

FloVer, vérification mécanisée de certificats de correction générés par Daisy.

Février – Juin 2017

J'ai généralisé la formalisation de FloVer, qui contient 10kLoc HOL et 25kLoc Coq. Je faisais partie des deux développeurs principaux (avec Heiko Becker) durant ma période de stage. La généralisation effectuée avait augmenté la taille des formalisations de 30% pour Coq et 18% pour HOL. Daisy et FloVer ont été l'objet d'une publication de groupe à FMCAD'18.

Autoévaluation INS2I : A3 S03 SM3 EM3 SDL4 DA2 CD2 MS1 TPM1.

Batman, preuve de concept d'un analyseur statique capable d'analyser les différents threads de manière modulaire.

Juin 2015 – Janvier 2017

Seul contributeur. 5,5kLoc OCaml. Licence GPL v3. Ce prototype a été utilisé pour l'évaluation expérimentale de notre papier à VMCAI'17.

Autoévaluation INS2I : A1 S02 SM1 EM1 SDL4 DA4 CD4 MS4 TPM4.

Langues

- Français, langue maternelle
- Anglais, niveau B2 certifié par le CLES
- Allemand, niveau B1

Programmation & outils

- Pratique courante : OCaml, Python, C, \LaTeX , Beamer
- Pratique régulière : Java, Bash, GoHugo, OBS (cours à distance et enregistrement de présentations virtuelles), assistants de preuve Coq et HOL4

Engagements associatifs

Animation bénévole de cours d'escrime pour étudiants , 2h à 3h30 par semaine, 40 semaines par an	2018–2020
Suivi d'une formation fédérale à l'animation de cours d'escrime , 70 heures	2018–2019
Médaille de bronze lors de la finale régionale (SWERC) du concours de programmation ACM ICPC , Rang 12/75 (médailles jusqu'à la place 14), équipe ENSL2 avec Simon Mauras et Jean-Yves Franceschi.	2017
Délégué lors du M1 d'Informatique Fondamentale à l'ENS de Lyon	2015–2016