

Aurore Alcolei

C.V. *analytique*

Centre Inria de l'Université de Rennes
Avenue du Général Leclerc
35042 Rennes Cedex

✉ aurore.alcolei@ens-lyon.org

🌐 <https://perso.ens-lyon.fr/aurore.alcolei/>

Candidature au poste de Maître de conférences

Poste 4230 – Campagne 2024

Parcours

POSITION ACTUELLE

Oct. 2023 – **Ingénieure de Recherche**, équipe *EPICURE*, Centre *INRIA* de l'Université de Rennes, action exploratoire *Back to the Trees* dirigée par S. Castellan
Sept. 2025 — Vacation pour l'ISTIC - dep. Info. de l'Université de Rennes (jan.–juin. 2024)

POSITIONS PASSÉES

Jan. 2022 – **Post-doctorante**, *Laboratoire de d'Algorithmique, Complexité et Logique*,
Mai 2023 *Université Paris Est Créteil (UPEC)*, collaboration avec L. Pellissier et A. Saurin.

— Vacation pour le département Informatique de l'UPEC (sept.–dec. 2022)

Nov. 2019 - **Post-doctorante**, *Università degli Studi di Bologna*, projet ERC DIAPASoN
Nov. 2021 dirigé par Ugo Dal Lago

— Vacation pour la Bologna Business School (BBS) (mars–juin 2021)

Sept. 2016 – **Doctorante**, équipe *PLUME*, *Laboratoire de l'Informatique et du Parallélisme*, *ENS de Lyon*, thèse encadrée par P. Clairambault, O. Laurent, G. Winskel

— Activité Complémentaire d'Enseignement (ACE) (2016–2018)

— Activité Complémentaire de Diffusion (ACD) (2018–2019)

Sept. 2012 – **Élève fonctionnaire**, *École Normale Supérieure de Lyon*

Août 2016 — dont 4 stages de recherches (de 1, 2 et 2x5 mois) au LAMA (Chambéry), LIS (Marseille), Loria (Nancy), Computer Laboratory (Cambridge, Royaume-Uni)

FORMATION

2016–2019 **Doctorat Informatique**, *Université de Lyon opéré par ÉNS de Lyon*, Lyon
Thèse en codirection internationale (Computer Laboratory, Cambridge, RU).

Titre : Jeux concurrents enrichis : témoins pour les preuves et les ressources.

Jury : P. Clairambault (co-directeur), C. Faggian (examinatrice), D. Ghica (rapporteur), D. Kesner (présidente), O. Laurent (directeur), D. Miller (rapporteur), G. Winskel (co-directeur).

2013–2015 **Licence et Master Informatique Fondamentale**, *École Normale Supérieure de Lyon*, Lyon, Mention Bien.

— M1 en échange à l'Université de Calgary, Canada

2010–2012 **DEUG Mathématiques et Informatique**, *Université Joseph Fourier*, Grenoble, Mention Très Bien.

— stage d'excellence de 6 semaines au sein du laboratoire VERIMAG (Grenoble)

2009–2010 **Baccalauréat Série Scientifique**, Spécialité Mathématiques, Grenoble, Mention Très Bien.

Recherche

J'établis ici une liste synthétique de mes productions scientifiques, les titres sont cliquables. Plus de détails sont donnés en Annexes p.14 et dans mon Projet de Recherche p.6.

Mots clés Sémantique dénotationnelle, sémantique interactive, jeux concurrents, logique linéaire, réseaux de preuves, programmation concurrente, théorie des langages. (GDR IM, GT SCALP)

PUBLICATIONS AVEC COMITÉ DE SÉLECTION

- MFPS 2023 *The Exponential Logic of Sequentialization*, A. Alcolei, L. Pellissier, A. Saurin
(Conf. B) 39th Conference on Mathematical Foundations of Programming Semantics.
- FoSSACS 2019 *Resource-Tracking Concurrent Games*, A. Alcolei, P. Clairambault et O. Laurent
(Conf. A) Foundations of Software Science and Computation Structures 2019. Lecture Notes in Computer Science, vol 11425.
- CSL 2018 *The True Concurrency of Herbrand's Theorem*, A. Alcolei, P. Clairambault, M. Hyland et G. Winskel
(Conf. B) 27th EACSL Annual Conference on Computer Science Logic. Leibniz International Proceedings in Informatics (LIPIcs), vol. 119.
- SASB 2015 *On the Flora of Asynchronous Locally Non-Monotonic Boolean Automata Networks*, A. Alcolei, K. Perrot et S. Sené
(Workshop) The 6th International Workshop on Static Analysis and Systems Biology. Electronic Notes in Theoretical Computer Science, vol. 326, 2016.

MANUSCRITS

- Thèse *Enriched Concurrent Games Witnesses for Proofs and Resource Analysis*, A. Alcolei.
- Rapports d'activité — *Impact de la non-monotonie locale dans les RAB asynchrones* (M2),
— *Concurrent program as proofs: compacting message passing logic* (M1),
— *Sémantique 2-catégorique des langages de programmation* (L3).

RÉALISATIONS LOGICIELLES

Back to the trees. Contributrice depuis octobre 2023 (projet démarré en avril 2022).

COMMUNICATIONS SCIENTIFIQUES

J'ai présenté mes travaux ou des exposés introductifs dans les rencontres scientifiques suivantes :

- Conférences CSL 2018 ; FoSSaCS 2019 ; MFPS 2023.
- Workshops FMCS 2014 ; Galop 2017 ; Structural Deduction 2017 ; Game Semantics 25 (2018) ; TLLA 2023.
- Séminaires LIMD (Chambéry, 2017) ; PPS (Paris, 2019) ; CHoCoLa (Lyon, 2019) ; LDP invités (Marseille, 2019) ; LoVe (Paris, 2021).
- Groupes de travail Rencontre GT (Scalp 2023), séminaire d'équipe (Plume 2017, Diapason 2020, LACL 2023), séminaire des doctorants (Lyon 2017, Cambridge 2018).

RESPONSABILITÉS

- Membre Comité de programme — Conférences : *Mathematics of Program Construction* (MPC 2022)
— Workshops : *Higher-Order Programming with Effects* (HOPE 2024) ; *Games for Logic and Programming Languages* (GALOP 2023)
- Relectrice — Conférences : *Logic in Computer Science* (LICS 2020) ; *Foundations of Software Science and Computation Structures* (FoSSaCS 2021)

Enseignement

Le tableau ci-dessous synthétise mes différentes expériences d'enseignements universitaires, principalement axées sur la programmation et de l'informatique fondamentale. Une liste détaillée de ces enseignements est fournie en Annexes. Mon projet d'enseignement est développé en p.5.

Année	Statut	Matière	Public	CM	TD	TP
2024	Vacataire	Programmation Distribuée	M1 – Rennes 1			42
2022	Vacataire	Algorithmique et Structure des Données	L2 – UPEC		18	
		Programmation fonctionnelle	L2 – UPEC			24
		Application Mobile (Android)	L3 – UPEC			18
2021	Vacataire	Intro.to Python Programming	L2 – BBS	10		25
2019	Vacataire	Théorie des langages réguliers	L2 – EPITA	10	10	6
2019	Bénévole	Intro. à la vérification de programmes	L1/L2 – École d'été		4	6
2016 – 2018	ACE	Intro. aux réseaux et au web	L1 – Lyon 1			12
		Projet de programmation	L3 – ENSL			32
		Compilers and Program Analysis	M1 – ENSL			28
		Preuves et Programmes	M1 – ENSL		2x20	
2016	Vacataire	Types	L3 – Cambridge		8	

Total : 293 HTD

Stages : J'encadrerai en Juin-Juillet un stage de L2 sur des problèmes de compilation pour le projet Back to the trees. En 2018, j'ai été membre du jury de stages de L3 de l'ENSL.

Je m'intéresse également à la pédagogie. J'ai notamment pris part à des groupes de travail sur la classe inversée (2016), l'informatique débranchée (2017-2019) et l'enseignement à l'anthropocène (2022).

En 2020 j'ai obtenu ma **Qualification Maître de Conférence section 27**.

Médiation scientifique

La diffusion de l'esprit et des connaissances scientifiques au-delà des frontières universitaires me semble fondamentale. Je présente ici une liste synthétique d'activités de médiation réalisées en tant que bénévole ou dans le cadre de mon ACD (activité complémentaire de diffusion, 2018 – 2019) à la MMI (Maison des Mathématiques et de l'Informatique de Lyon).

- Ateliers**
 - *Déranchés* : Base de données à tricoter (Fête de la Science 2018), Hasard et probabilités (Pop science 2019), Jeux de graphe (stage de 3^{ème})
 - *Branchés* : Initiation à la programmation (Ladies Learning Code 2015), Codage et objets connectés (pop science 2019)
 - *Causerie* : La recherche automatique de preuve (Fête de la science 2018)
- Événements**
 - Organisatrice de projections et atelier de go autour du documentaire *α-go* (2018, 2019)
 - Intervenante pour des rencontres lycéennes-chercheuses (2018, 2022)
- Exposition**
 - Membre du comité de pilotage de l'exposition *Entrez dans le monde l'IA* (MMI 2022).

— Vie de laboratoire

- Élu(e) étudiante suppléante au conseil scientifique de l'ENS (2016–2017).
- Organisatrice des ateliers de lecture mensuel Science en Société à INRIA Rennes (depuis janv. 2024)
- Organisatrice des Journées Des Doctorants du LIP 2018. <http://www.ens-lyon.fr/LIP/index.php/phd-students/448-phd-days-2018>.
- Administratrice du site d'équipe DIAPASoN (2019–2021)
- Gestion du café et des matchs de foot ;-)

— Compétences transverses

Programmation	— expérience forte : OCaml, C/C++, Python — base de : Coq, HTML, Ada, Java et MATLAB
Outils	Git, L ^A T _E X, VIM, suite Office, Unix, Apps Script
Langues	Français (maternelle), Anglais (courant), Italien (conversationnel)
Autre	Permis de conduire, Formation aux premiers secours

Activités et Projet d'Enseignement

L'enseignement est pour moi une activité importante et complémentaire au travail de recherche. Elle permet de réfléchir à la discipline d'un point de vue plus général et pédagogique. J'apprécie également le contact avec les étudiants et leur potentiel d'apprentissage.

Depuis 2016 j'ai enseigné l'informatique tant sous des aspects théoriques qu'appliqués dans différents formats de cours allant du mentorat ou cours magistral en passant par l'encadrement de TD et TP.

Par ma formation et mon expérience d'enseignement je maîtrise notamment les thématiques suivantes :

Programmation	Python, HTML-CSS, Android, Java (web), Ocaml, C/C++, Coq
Informatique théorique	Théorie des automates, complexité, algorithmie, logique et modèle compilation et analyse de programme, théorie de la preuve
Encadrement	Projet de programmation

J'apprécie d'enseigner tant dans des domaines qui me sont familiers (informatique théorique, programmation fonctionnelle) que dans des domaines plus éloignés de ma pratique (programmation web, programmation distribuée, programmation mobile, ...). Dans le second cas, il me faut cependant prendre en compte un plus grand temps de préparation et d'appropriation du sujet.

En tant qu'enseignante, je souhaite développer des cours basés sur le principe de la pédagogie centrée apprenant, c'est à dire avec un scénario pédagogique cherchant à rendre l'étudiant le plus actif possible tout en lui proposant un contenu digéré et adapté à son niveau. Je pense par exemple que l'approche par projet guidé et modulable est efficace dans l'enseignement de la programmation.

Depuis 2022, je m'intéresse également aux dispositifs pédagogiques développés par d'autres enseignant-chercheurs pour la prise en compte des enjeux sociaux et environnementaux dans les cursus du supérieur. J'ai par exemple pu observer la mise en place de séances d'étude et d'analyse d'articles issus des recherches en science et société, ou encore le développement d'une carte interactive reliant des exercices de recherche opérationnelle à leur utilisation dans les infrastructures sociales [21]. D'autres exemples dont j'ai pu avoir des retours concernent la programmation du modèle World3, son utilisation et la discussion autour de ses hypothèses de conception, ou encore des initiations à l'analyse de cycle de vie, par exemple en faisant une recherche documentaire sur les métaux utilisés dans la fabrication des écrans de téléphone.

Cette démarche semble coller parfaitement avec la philosophie de l'école Grenoble INP Phelma qui vise à former des ingénieurs·e-s dans un contexte de transition écologique. Je suis donc enthousiaste à l'idée de pouvoir mobiliser mes connaissances sur le sujet pour faire évoluer les enseignements proposés au sein de l'école.

Intégration à Grenoble INP Phelma

Par anticipation de la transition numérique à venir, l'école Grenoble INP Phelma souhaite renforcer les enseignements informatiques dispensés au sein de ses formations, le but étant d'acquérir une culture générale en informatique allant des bases en systèmes et réseaux, aux grands principes algorithmiques en passant par, les bases de données ou la programmation impérative en C.

De par mon cursus et mon expérience d'enseignement, je devrais déjà être compétente dans la majeure partie des enseignements relevant de la programmation, de l'algorithmie, ou de la compilation. Je suis en grès au temps nécessaire pour organiser un cours selon un certain curriculum. Je serais moins à l'aise dans le cas de cours portant sur l'intelligence artificielle ou les bases de données, cependant je pense être capable de m'en approprier le matériel en fonction des besoins au sein de l'équipe pédagogique.

Activités de Recherche

Au cours de mes stages d'étude j'ai abordé différents domaines de recherche en informatique théorique allant des algorithmes de monitoring décentralisé à la théorie des graphes, en passant par les réseaux d'automates booléens et la sémantique (catégorique) des preuves et des programmes. Depuis ma thèse, je me suis focalisée sur ce dernier point et j'ai notamment développé une expertise dans le domaine de la sémantique des jeux, qui représente les programmes comme des structures réagissant à leur environnement, c'est à dire comme des stratégies pour un jeu à deux joueurs, le joueur opposant étant l'environnement du programme. Par la suite, mes différents post-doctorats m'ont permis d'explorer d'autres branches de la théorie de la programmation : réseaux de preuve, π -calcul, programmation probabiliste et programmation approximée.

Je présente ici trois des projets de recherches les plus importants dans lesquels je me suis ou suis encore impliquée.

1 - Modèles de jeux concurrents annotés

Ce projet correspond à mes travaux de thèse, encadrée, en codirection internationale, par Pierre Clairambault, Olivier Laurent (LIP, Lyon) et Glynn Winskel (Computer lab, Cambridge), de septembre 2017 à octobre 2019.

La sémantique dénotationnelle est une branche de l'informatique théorique qui vise à élaborer des structures mathématiques compositionnelles permettant de décrire le contenu calculatoire des preuves et des programmes indépendamment de leur syntaxe. Traditionnellement, les modèles sémantiques pour les langages de programmation cherchent à capturer des aspects qualitatifs du calcul tels que la terminaison (le programme termine-t-il ou non ?) ou l'équivalence observationnelle de programmes (étant donnés deux programmes, y a-t-il un contexte d'évaluation capable de les distinguer ?). A l'heure actuelle ils forment un socle théorique solide pour de nombreuses primitives de calcul telles que les appels de fonction, l'état, les exceptions, le non-déterminisme, la récursion ou encore la concurrence. Mais, face à l'émergence de questions sémantique plus quantitatives telles que la consommation de ressource ou la durée du calcul pour l'optimisation de programmes, un enjeu est de comprendre comment ces nouvelles informations peuvent être reflétée au sein des modèles.

Les modèles de jeux sont des modèles sémantiques centré sur l'interaction : preuves et programmes y sont représentés par des stratégies modélisant, par le flot d'exécution, leur manière de réagir à leur environnement. Il existe deux grandes familles de modèles de jeux : la famille des jeux séquentiel et la famille des jeux concurrents. Ces derniers se distinguent des premiers par leur représentation causale et non chronologique du flot d'exécution. Ce changement de représentation à notamment pour but de donner une représentation plus fidèle des primitives de calcul parallèle en intégrant une notion de parallélisme dans la structure même des stratégies représentant les programmes.

Le but de ma thèse était de comprendre comment de tels modèles pouvait être enrichis par des informations quantitatives. Dans une approche Curry-Howard (correspondance preuve/programme), j'ai montré que la structure causale sous-jacente au modèle de jeux concurrent pouvait être exploitée pour garder trace de certaines méta-données – c'est à dire d'informations pouvant variées au cours de l'exécution mais n'influençant par le flot de contrôle – telles que :

- la consommation de ressources telles que le temps pour des programmes concurrents [4].
- la construction de témoins dans les preuves de la logique classique du premier ordre [3],

J'ai ensuite généralisée ces deux constructions dans un même cadre en définissant un enrichissement paramétrique du modèle de jeux concurrent à base de structures d'événements[1, 2], permettant de capturer toute méta-donnée d'exécution représentable par une théorie (in)équationnelle donnée (le paramètre du modèle) [6].

Ces travaux ont fait l'objet de deux publications[3, 4] à FoSSaCS et CSL (conférences de rang A et B), et ont été présentés dans divers séminaires scientifiques.

2 - Syntaxe parallèle pour la programmation logique

Ce projet a été initié lors de mon post-doctorat avec Luc Pellissier (LACL, Créteil) et Alexis Saurin (IRIF, Paris) de janvier 2022 à mai 2023.

La programmation logique est un paradigme de programmation dans lequel un programme correspond à une série de clauses logiques et un but à prouver à partir de ces clauses. L'exécution du programme (pouvant échouer) revient ensuite à la construction d'une preuve du but à l'aide des clauses dans un système de déductif donné. L'état d'un programme durant son exécution correspond donc à une preuve en construction. La recherche de cette preuve peut-être guidée par la structure de la formule, mais aussi par d'autres éléments : des heuristiques, des informations apprises pendant des phases précédentes de la recherche de preuve, ou encore des instructions données par l'utilisateur. Ces derniers artefacts sont mal compris d'un point de vue logique : ils n'ont pas de pendant dans les systèmes déductifs classiques.

La ludique est une sémantique interactive pour les système déductif, c'est un cadre théorique dans lequel la validité d'une preuve est exprimée par sa capacité à "bien réagir" à toute contre-preuves de la formule à prouver. Les objets de la ludique sont plus génériques que les preuves, en particulier ils permettent de représenter la notion de preuve inachevée... ou en construction : dans des travaux préliminaires [13], Saurin a montré que la ludique était un cadre adapté à la programmation logique.

D'un point de vue effectif cependant, cette approche souffre de la trop forte séquentialité syntaxique de la ludique. En effet, celle-ci est utilisée pour décrire des tests servant à guider la recherche de preuve, mais les différents tests ne s'accordent pas forcément sur l'ordre dans lequel cette recherche doit être menée. Pour contourner ce problème, il est possible d'ajouter des contraintes artificielles sur les tests qui garantissent alors l'homogénéité de l'ordre de la recherche. Cependant cet ordre est arbitraire et peut rendre la recherche particulièrement inefficace. Le but de ce projet est donc de construire une structure logique non-séquentielle, inspirée de la ludique, pour la recherche de preuve.

Des syntaxes parallèles pour la ludique existent, ce sont les L-nets (réseaux ludiques) développés par Faggian et al. [16]. Malheureusement, tout comme la ludique dont ils sont inspirés, les L-nets sont difficile à manipuler. Inversement, il existe une version moderne de la ludique, la C-ludique, définie par Terui [14], avec une syntaxe plus facilement manipulable, mais qui reste un calcul séquentiel. C'est pourquoi, une première étape de recherche vise à la construction d'une syntaxe parallèle (ou syntaxe de réseaux) pour la C-ludique. Une deuxième étape vise ensuite à généraliser la recherche de preuves interactives proposée dans [13] à ce cadre déséquentialisée en vue de lever les contraintes contre-productives à une recherche optimale.

Ce projet est encore en cours et a dû être mis en pause suite à la fin des financements et au commencement de mon nouveau post-doctorat à Rennes qui a requis toute mon attention.

La première étape du travail a cependant bien avancée : afin d'avoir un cadre sémantique commun à toutes les constructions envisagées, j'ai montré que la ludique, la c-ludique et les L-nets, admettaient une sémantique commune dans le modèle jeux concurrents [1].

Par ailleurs, il est apparu qu'une syntaxe de réseau pour la C-ludique ferait sans nul doute intervenir un artefact graphique propre aux réseaux, les "jumps", permettant d'exprimer certaines contraintes de séquentialisation au sein d'un réseau. Ces structures sont souvent introduite de manière ad hoc dans les réseaux et n'avaient jusqu'à présent pas de pendant logique. Dans un travail parallèle à notre objectif de départ, nous avons montré qu'il était possible de donner un contenu logique à ces artefacts graphiques en les encodant dans les réseaux eux mêmes.

Ces résultats ont fait l'objet d'une publication à MFPS [7] (conférence de rang B) et ont été présentés dans divers séminaires scientifiques.

3 - Sémantique probabiliste pour l'identification botanique

Le projet de recherche *Back to the trees* (ci-après BTT) est un projet interdisciplinaire qui a été initié par Simon Castellan (Inria Rennes), Jos Käfer (ISEM, Montpellier) et Eric Tannier (LLBE, Lyon) en 2022. J'ai rejoint le projet en octobre 2023 en tant qu'ingénieure de recherche sur l'action exploratoire affiliée au projet.

Traditionnellement, l'apprentissage de la botanique (entendu ici comme la capacité à identifier les plantes et à en connaître leur propriété) passe par des sorties de terrain et l'usage de *flores*, gros livres recensant l'ensemble des espèces végétales connues pour une géographie donnée et contenant également une *clé de détermination*, c'est à dire un arbre de décision utilisant des critères morphologiques pour pouvoir identifier ces espèces.

Les clés de déterminations présentes dans les flores sont souvent difficiles à utiliser : dans un souci d'économie de place les questions sont choisies pour être les plus effectives en terme de discrimination mais peuvent demander une expertise hors d'atteinte pour le néophyte ; il n'est pas non plus possible de passer une question et toute erreur de réponse est fatale dans la réussite de la détermination. Par ailleurs, l'élaboration d'une clé de détermination est une tâche coûteuse qui nécessite au préalable un grand nombre de connaissances botaniques, il est donc difficile pour de petits espaces naturels à visée pédagogique d'éditer une flore qui leur serait propre, même si, sur un nombre d'espèces plus restreint les clé de détermination, moins longues, sont plus simples à utiliser.

Pour faire face à ces différentes problématiques, le projet BTT a pour objectif de créer un outil numérique permettant la génération automatique de clés de détermination numériques propre à un espace donné. Celle-ci s'appuie sur une méthodologie en trois points :

- inventaire botanique des espèces présentes dans l'espace donné ;
- récupération des fiches espèces correspondant à l'inventaire dans la base de données ouverte du projet, et édition de nouvelles fiches si nécessaire ;
- génération d'une clé de détermination à l'aide du logiciel BTT, à partir d'un ensemble de questions choisies pour être accessible au public visé

Principe de fonctionnement de l'algorithme Afin de refléter le savoir botanique, chaque espèce est décrite en suivant un schéma morphologique commun S . S est construit comme une combinaison de types produits et sommes afin de mettre en avant la dépendance entre certains traits morphologiques (par exemple, une plante ne peut avoir de pétales roses que si elle a des fleurs). Pour exprimer la variabilité morphologique des individus au sein d'une espèce (et parfois au sein même d'un individu), la description d'une espèce correspond à une distribution probabiliste sur S (vu comme ensemble d'instances morphologiques).

L'ensemble des questions pouvant être posées à l'utilisatrice est également organisé dans un type Q . Un langage de programmation probabiliste permet de décrire la manière dont doit être effectuée la traduction des descriptions de S vers Q . Une fois compilé et exécuté chaque espèce correspond donc à une distribution probabiliste sur les réponses aux questions de Q . Cette structure à l'avantage de séparer la représentation du savoir botanique des questions peut-être plus vernaculaires pouvant être posées à l'utilisateur.

L'algorithme de détermination sous-jacent à la clé procède ensuite par inférence Bayésienne : partant d'une distribution de probabilité uniformes sur les espèces référencées, chaque réponse de l'utilisatrice aux questions de Q va permettre une mise à jour de la distribution distinguant progressivement l'espèce à identifier. Pour faire face au niveau de l'utilisateur il est possible de pondérer ses réponses par un facteur de confusion. Enfin, l'ordre des questions est déterminé selon une heuristique visant à maximiser l'information (au sens entropique) gagnée en moyenne en répondant à la question.

Travail sémantique Mon premier travail a consisté en la formalisation de la sémantique probabiliste des descriptions présentées ci-dessus. En particulier j'ai montré que l'ensemble des descriptions morphologiques (syntaxique) sur S était une rétraction de l'ensemble des distributions sur S (la sémantique des descriptions). À partir de cette sémantique nous travaillons à prouver les propriétés de terminaison et de correction de l'algorithme. Nous cherchons en particulier à formaliser sous quelles conditions de distinguabilité vis à vis d'un ensemble de questions, un ensemble de description est satisfaisant pour assurer ces propriétés. Nous travaillons en effet avec des données parfois incomplètes et avec des questions éditables pour faciliter l'utilisation, il est donc important de pouvoir analyser notre corpus pour assurer le bon fonctionnement de l'algorithme de détermination.

L'outil de traduction entre schéma est un travail mené depuis octobre par Virgil Marionneau, stagiaire masterant à l'ENS Rennes, et principalement supervisé par Simon Castellan. Nous travaillons ensemble au développement de l'expressivité de ce langage et à sa formalisation sémantique pour assurer que les descriptions calculées par les traductions respectent leur sémantique probabiliste. Je serai amenée en mai à poursuivre ce travail avec une stagiaire de L2 de l'Université de Rennes.

Parallèlement à la formalisation de l'algorithme déjà produit, je travaille également au développement d'une logique de description qui permettra de combiner entre elles les questions de la clé via des opérateurs de la logique propositionnelle (et à terme du premier ordre). Ce travail théorique fera ensuite l'objet d'une implémentation dans l'outil.

Interdisciplinarité et diffusion Le projet BTT est l'occasion d'un travail interdisciplinaire

avec des botanistes. Cette collaboration a été renforcé par l'obtention récente d'une ANR, *Flore*, ayant pour but de fusionner le projet de BTT avec celui de *Botascopia*, une base de donnée morphologique visant à l'édition automatique de fiches pédagogiques pour la description des plantes. Outre Jos Käfer, nous travaillons donc également avec Sophie Nadot, Xavier Aubriot (ESE, Saclay) et Agnes Scherman (Ecobio, Rennes) dans le but d'améliorer notre schéma de description botanique *S* et de le faire fusionner avec celui de Botascopia. La compréhension du savoir botanique d'un côté et l'appropriation des outils informatique de l'autre est cruciale dans la réalisation du projet, je participe donc régulièrement à des réunions de coordination avec ces différents chercheurs.

Enfin, ce projet est avant tout un projet tourné vers la société et la diffusion des connaissances botaniques au grand public. Nous avons un partenariat avec le Jardin des mille pas à Rennes avec qui nous avons réalisé notre méthodologie entièrement : après l'inventaire réalisé au printemps dernier, et l'agrégation des données morphologiques réalisée au cours de l'année via des séances de sciences participatives ou via l'utilisation de la base de donnée de *Botascopia* (traduite dans notre schéma), une première version de la clé pour ce jardin pédagogique a été testée le 20 mars dernier. Celle-ci a été accueillie avec enthousiasme, tout comme l'ont été les séances d'édition des fiches espèces avec les bénévoles de l'association Bretagne Vivante au cours de l'année.

Un nouveau partenariat est en cours de création avec des professeurs du collège-lycée Emile Zola à Rennes pour une zone d'exploration botanique située sur le long du canal de la Vilaine. Pour se faire nous encadrons Lysa Dahmani, masterante à l'ENS Saclay, avec qui nous souhaitons également réalisée une première étude socio-pédagogique de la clé, co-encadrée par la sociologue Anne Atlan (ESO, Rennes).

■ Projet de recherche

Outre les projets de recherche 2 et 3 dans lesquels je me suis impliquée et dans lesquels je souhaite continuer à prendre part, j'envisage deux autres axes de recherches pour le futur : passée :

SÉMANTIQUE DE JEU POUR LE MODÈLE CHECKING DE PROGRAMMES CONCURRENT

Le model checking est un outil phare des méthodes formelles pour la vérification automatique de systèmes de calcul. Elle consiste à vérifier certaines propriétés de sécurité (par exemple : un seul thread peut écrire dans le registre partagé à la fois) ou de vivacité (par exemple : le registre partagé est toujours remis à 0 après une lecture) d'un système en abstrayant ce derniers par un système de transition capable de refléter les informations importantes à vérifier dans le système. Les propriétés du systèmes, exprimées le plus souvent sous la forme d'une formule logique temporelle, sont ensuite vérifiées via différents algorithmes de model checking sur les traces d'exécutions du système.

Comme expliqué dans la présentation de mes travaux de recherche 1, la sémantique des jeux est une sémantique interactive en ce qu'elle capture les interactions d'un programme avec son environnement. C'est également une sémantique de trace : ces interactions sont décrites par des séquences de coups (sémantique des jeux séquentielle) ou bien par des ordres partiels voir des structures d'événements (sémantique de jeux causale ou concurrente)

Deux types de systèmes de transition sont principalement utilisés pour le model checking de système concurrent : les système push-down concurrents et les réseaux de Petri. Les systèmes push-down concurrents sont des abstractions simples de système concurrents en cela qu'ils abstraient le système comme la somme de ses composants abstraits (ie représentés par un push-down system). Les traces du système sont donc des entrelacements des trace des composants. La simplicité a donc un coup puisqu'elle induit des lourdeurs de traitement : pour n transitions indépendantes les unes des autres, le modèle induits $n!$ traces différentes. À l'opposé, les réseaux de petri sont des systèmes de transition intrinsèquement concurrents et peuvent être monitorés par des traces structurés : les préfixes de réseaux de Petri, ou structure d'évènement. À cause de leur complexité, peut-être, ces modèles semblent cependant moins utilisées dans la littérature que les systèmes push-down concurrent en tant qu'abstraction de programme concurrent.

De récents travaux [9] ont montré qu'il était possible de compiler n'importe quel programme

du langage de programmation concurrente Idealized Parallel Algol (IPA) en un réseau de Petri (colorés) dont les préfixes correspondent au déroulement d'une sémantique de jeux concurrente pour le langage. Cela ouvre la voie vers de nouvelle méthode d'abstraction de programmes concurrents dans les réseaux de Pétri.

Plusieurs pistes de recherche s'ouvrent dans cette direction :

- Tout d'abord, il faudrait s'assurer que la sémantique des jeux telles que développées à l'heure actuelle fournissent une représentation intéressante pour les propriétés de modèle checking. Un point qui semble par exemple crucial est que les sémantiques de jeux sont pour la plupart développée pour des langages de programmation en appel par nom. Hors si cette stratégie d'évaluation est conceptuellement intéressant, elle est peu utilisée en pratique. Un point de départ évident serait donc d'explorer la sémantique de jeux concurrent en appel par valeur pour IPA et de voir comment le résultat de compilation peut-être adaptée à cette nouvelle interprétation.

- Une seconde piste serait de voir si cette méthodologie peut s'étendre à la sémantique de jeux concurrents avec traçage du temps d'exécution que j'ai donné pour IPA affine [4]. Il faudra pour cela commencer par comparer les préfixes des réseaux de Pétri temporisé avec les structures d'évènement annotées. Il faudra ensuite étendre le modèle de jeux annoté à une version avec symétries, ce qui permet de sortir du cadre affine et d'exprimer la récursion. La même question se pose pour d'autres enrichissement du modèle de jeux concurrents telles que les extensions probabilistes [11] ou quantiques [12].

ÉTUDES VARIÉES AUTOUR DU π -CALCUL

Un des modèles emblématiques dans l'étude du calcul parallèle et distribué est le π -calcul. Comme le λ -calcul pour les langages fonctionnels, c'est langage de programmation élémentaire qui permet d'exprimer différentes notions propres au calcul distribué : mise en parallèle de processus, échange de messages, création de canaux. . .

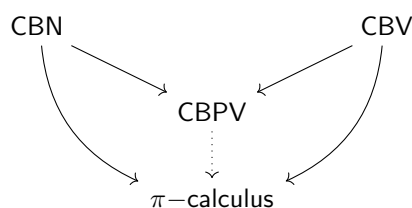
Ce calcul a été intensivement étudié, tant d'un point de vue sémantique que d'un point de vue opérationnel. Je propose ici trois pistes de recherches qui semblent cependant inexplorées :

1. Pi-calcul, jeux, réseaux : trois faces d'une même pièce

L'approche Curry-Howard pour le calcul concurrent a fait émerger trois modèles syntaxiques et sémantiques majeurs : le pi-calcul, les réseaux de preuve et les sémantiques de jeux. Il existe dans la littérature de nombreux travaux reliant pi-calcul et réseaux, réseaux et jeux, et, pi-calcul et jeux. Chacun de ces travaux exhibe pour ce faire des variantes particulières de ces trois modèles de calculs concurrents. Dans une optique de synthèse, je souhaite proposer un cadre sémantique et syntaxique dans lequel ces trois pôles se correspondent parfaitement. Cet axe de recherche est issu du groupe de travail initié à mon arrivée au sein du LACL avec Luc Pellissier (expert en réseau) et Daniele Varacca (expert en pi-calcul), avec qui je compte donc poursuivre ma collaboration.

2. Modèles interactifs pour les stratégies d'évaluation du lambda calcul

Comme mentionné plus haut, le λ -calcul est un langage de programmation primitif permettant de refléter la dynamique de réécriture des systèmes d'ordre supérieur [19]. Deux stratégies d'évaluation sont particulièrement étudiées pour ce langage : la stratégie d'évaluation en appel par nom et la stratégie d'évaluation en appel par valeur. Ces deux stratégies peuvent être étudiées conjointement dans le métalangage Call-By-Push-Value [18]. Elles peuvent également être traduites par différents encodages dans le pi-calcul. Une question naturelle vient alors à l'esprit : existe-t-il une traduction de Call-By-Push-Value vers le pi-calcul qui fasse commuter le diagramme suivant ?



L'ensemble des calculs présentés ci-dessus admettent des sémantiques de jeux [5, 20]. Ce travail

sera donc une fois encore l'occasion de faire une synthèse de travaux passés avec l'espoir d'en tirer un modèle de jeux communs aux quatre langages et facilitant ainsi les preuves de correction des traductions.

3. Automates implicites dans le Pi-calcul

La théorie des automates implicites est un champs de recherche actif [22, 23] qui vise à caractériser les classes de langages issues de la théorie des automates par des classes de termes du λ -calcul simplement typé. Par exemple, pour tout langage régulier L sur un alphabet Σ on peut définir un terme $t : \text{Str}_\Sigma \rightarrow \mathbf{B}$ qui calcule la fonction indicatrice de L pour un certain encodage Str_Σ des chaîne de caractères sur Σ dans les types simples.

Tout comme le λ -calcul, le π -calcul admet un typage simple. Il serait intéressant de voir si des classes de langages peuvent être caractérisées par ce typage de manière directe, sans passer par une traduction du λ -calcul dans le π -calcul et exhibant ainsi des notions de parallélisme dans les langages réguliers.

Intégration au LIG, équipe SPADES ou SIGMA

Bien que mon profil de recherche soit plutôt théorique, avec une dominante sur la sémantique dénotationnelle des langages de programmation et des systèmes de preuves, je tends à me tourner vers des recherches plus appliquées, comme en témoigne mes activités récentes au sein du projet *Back to the trees* (BTT) p.7, ou le premier axe de mon projet de recherche p.6. Par ailleurs, mes recherches théoriques sont pour grandes parties tournées vers l'étude de la programmation concurrente et des modèles de calcul distribué. En cela, je m'intègre bien dans les thématiques de recherche associées au profil de ce poste.

Thématiquement, l'équipe SPADES me paraît plus proche de mon profil de recherche, en effet, elle utilise des méthodes de l'informatique théorique appliquée à la conception de systèmes embarqués fiables, notamment par l'analyse causale des événements de calcul et du flot de données. Mon expertise en modèle de jeux causaux pourrait servir pour faire le lien entre des langages de programmation évolués et des analyses de flot de donnée développées dans l'équipe (Alexandre Honorat) ou encore pour donner une sémantique dénotationnelle au langage ForeC (Alain Girault) afin de faciliter ses preuves de correction.

Une autre possibilité d'intégration serait l'équipe SIGMA. Plus éloignée de mes thématiques de recherche originales, elle étudie les systèmes d'information dans une philosophie de co-conception proche de celle menée dans le projet BTT et qui pourrait être sur le long terme un glissement thématique inspirant pour moi.

Vie de laboratoire. De manière plus générale, en tant que membre d'un laboratoire et d'une université, je compte m'impliquer dans au moins une de ses instances collégiales, ou bien participer, via l'organisation d'ateliers ou de rencontres, à l'émergence d'une pensée collective sur des problématiques transversales au monde académique telles que le rapport entre science et société, la décarbonation des pratiques de recherche ou encore la mise en pratique de l'égalité femmes-hommes.

Je participe déjà à des événements ou groupes de travail inter et intra communauté sur ces sujets : conférence Archipel (2022, 2023, 2024), conférence sur la science non faite en informatique (2024), mouvement pour des savoirs engagés et reliés, séminaire éthique, groupe de lecture Science en Société et sur la sociologie du genre en informatique.

Références

- S. Castellan, P. Clairambault, S. Rideau, G. Winskel, « Games and Strategies as Event Structures », In : Logical Methods in Computer Science, 13, 3, 2017.
- S. Castellan, P. Clairambault, G. Winskel, « Thin Games with Symmetry and Concurrent Hyland-Ong Games », In : Logical Methods in Computer Science, 2019.
- A. Alcolei, P. Clairambault, M. Hyland, G. Winskel, « The True Concurrency of Herbrand's Theorem », In : CSL'18.
- A. Alcolei, P. Clairambault, O. Laurent, « Resource-Tracking Concurrent Games » In : FSCS'19.
- M. Hyland, L. Ong, « On Full Abstraction for PCF : I, II, and III », In : Inf. Comput., 163, 2, 285–408, 2000.
- A. Alcolei, « Enriched concurrent games : witnesses for proofs and resource analysis », PhD thesis, 2019.
- A. Alcolei, L. Pellissier, A. Saurin, « The Exponential Logic of Sequentialization », In : MFPS'23.
- P. Clairambault, M. de Visme, G. Winskel, « Game semantics for quantum programming », In : POPL'19.
- C. Castellan, P. Clairambault, The geometry of causality : Multi-token geometry of interaction and its causal unfolding, POPL 2023.
- P. Clairambault, F. Olimpieri, H. Paquet, « From Thin Concurrent Games to Generalized Species of Structures », LICS2023
- H. Paquet, « Probabilistic Concurrent Game Semantics », PhD thesis, 2019.
- M. de Visme, « Sémantique des jeux quantique », PhD thesis, 2020.
- A. Saurin. « Towards Ludics Programming : Interactive Proof Search » In : ICLP'08.
- K. Terui. « Computational ludics ». In : Theor. Comput. Sci. 412.20 (2011), p. 2048–2071.
- J.-Y. Girard. « Locus solum ». In : MSCS'01.
- P. Di Giamberardino, C. Faggian. « Jump from parallel to sequential proofs : Multiplicatives ». In : CSL'06.
- C. Faggian, M. Hyland. « Designs, disputes and strategies ». In : CSL'02.
- P. B. Levy. « Call-by-push-value : A subsuming paradigm ». In Typed Lambda Calculi and Applications : 4th International Conference», TLCA'99.
- T. Nipkow. « Higher-order critical pairs ». University of Cambridge, Computer Laboratory, 1991.
- S. Castellan, Y. Nobuko Yoshida. « Two sides of the same coin : session types and game semantics : a synchronous side and an asynchronous side ». In : POPL'19. i
- H. Cambazard, P. Marin, S-L. Phouratsamay. Impacts sociétaux de la pensée industrielle et rôle de la Recherche Opérationnelle, 2023, <https://tools.caseine.org/pensee-industrielle>.
- Le Thanh Dung Nguyen, Pradic, Implicit automata in typed λ -calculi I : aperiodicity in a non-commutative logic, ICALP 2020.
- Le Thanh Dung Nguyen, Pradic, Implicit automata in typed λ -calculi II : streaming transducers vs categorical semantics, preprint.

Annexes

Liste détaillée de mon expérience d'enseignement

Dans le cadre de mon activité complémentaire d'enseignement de 2016 à 2018, j'ai encadré en TD ou en TP des groupes allant de la L1 au M1 principalement dans le département d'informatique de l'ENS de Lyon mais également à l'université Claude Bernard. Ci-dessous un résumé de ces activités, des liens vers les différents supports de cours sont disponibles depuis ma page perso.ens-lyon.fr/aurore.alcolei/teaching.html :

- 12 HTD **TP pour LIFASR2 : Introduction aux Réseaux et au Web, niveau L1**, organisé par O. Glück. Département d'informatique de l'UCB Lyon 1
Contenu : Construction de sites internet simples en HTML, CSS et PHP, mettant en scène les notions de client et serveur web.
Compétences : Ce cours m'a en particulier permis de comprendre comment gérer un cours à grande échelle (plus de 100 étudiants et 15 groupes de TP différents). Le matériel de ce cours était entièrement fourni.
- 32 HTD **Encadrement de Projet 2 (OCaml), niveau L3**, organisé par D. Hirschhoff. Département d'informatique de l'ENS de Lyon
Contenu : Programmation d'une machine abstraite, d'un interpréteur et d'un compilateur pour un langage fonctionnel.
Compétences : Ce cours m'a permis de comprendre les points clés pour guider des étudiants dans un projet de programmation : questionner sur ce qui est fait/compris et sur les étapes à courts et longs termes, rappeler les règles de structuration du code et de tests, avoir une démarche incrémentale. Le matériel de ce cours était préparé par son responsable, j'ai contribué aux relectures.
- 2 x 20 HTD **TD pour Preuve et Programme, niveau M1**, organisé par P. Audebaud. Département d'informatique de l'ENS de Lyon
Contenu : Cours d'introduction aux systèmes de types (simple, polymorphe, de Martin-Löf) et à la correspondance du Curry-Howard.
Compétences : J'ai préparé les feuilles de TD, TP, DM et corrigés pour ce cours. J'ai pu m'inspirer d'exercices ayant été donnés les années précédentes pour des cours similaires.
- 28 HTD **TP pour Compilers and Programm Analysis, niveau M1**, organisé par L. Gonnord. Département d'informatique de l'ENS de Lyon
Contenu : Principes de compilation et d'analyse de programmes, implémentation de ces principes (lexing, parsing, abstract syntax tree, abstract interpretation) au travers la compilation de différents langages vers la machine abstraite LEAI, en utilisant l'outil ANTLR4 (Python 3).
Compétences : Le matériel de ce cours était préparé par sa responsable, j'ai contribué aux relectures et aux tests.

Ces différents cours m'ont permis d'être confrontée à des situations d'enseignement variées : petits groupes d'étudiants ou classe entière, parcours avancés ou débutants, préparation d'exercices ou appropriation de matériel existant, correction de devoirs papier ou de relecture de programmes (html, php, python, ocaml, coq).

J'ai également eu d'autres expériences d'enseignement en dehors de mon ACD :

- 35 H **Cours (10h), TP (25h) pour Introduction to Python Programming, niveau L2**, BBS, Bologna
Contenu : Une introduction à la programmation (en python) passant en revue les opérateurs de contrôle et structures de données classique, des notions de complexité et quelques pythoneries.
Compétences : J'ai monté ce cours de toute pièce car c'était la première fois qu'il était enseigné dans cette école de business. J'ai été confrontée aux difficultés de l'enseignement à distance envers un public peu sensibilisé à l'informatique.

- 42 H **TP (42h) pour Introduction à la Programmation Distribuée, niveau M1, ISTIC, Université de Rennes**
Contenu : Une introduction à la programmation distribuée en java passant en revue différentes API plus ou moins bas niveau (java Socket, Message oriented middleware, remote procedure call, serveur http) par la programmation de petits serveur et clients type echo ou chat.
Compétences : Ce cours m'a permis d'apprendre la programmation distribuée. Les contenus étaient fournis. Au delà de l'aide en séance, j'ai participé à l'évaluation des travaux des étudiants.
- 24 H **TP (24h) pour Programmation fonctionnelle, niveau L2, UPEC, Créteil**
Contenu : Une introduction à la programmation fonctionnelle (en Caml) introduisant la notion de typage, de constructeur de type, de type et fonction récursive (list, arbre), et de pattern matching.
Compétences : Ce cours m'a permis de découvrir la plateforme d'exercice en ligne Caséine. Les contenus étaient fournis. J'ai contribué aux questions d'examen.
- 18 H **TD (18h) pour Algorithmie et structure de donnée, niveau L2, UPEC, Créteil**
Contenu : Algorithmes et manipulation classique sur les listes chaînées et les arbres.
Compétences : Organiser un TD à géométrie variable (de 50 à 10 étudiants). Corriger des examens. L'ensemble du matériel de ce cours était fourni.
- 35 H **Cours (10h), TP (25h) pour Introduction to Python Programming, niveau L2, UPEC, Créteil**
Contenu : Une introduction à la programmation (en python) passant en revue les opérateurs de contrôle et structures de données classique, des notions de complexité et quelques pythoneries.
Compétences : J'ai monté ce cours de toute pièce car c'était la première fois qu'il était enseigné dans cette école de business. J'ai été confrontée aux difficultés de l'enseignement à distance envers un public peu sensibilisé à l'informatique.
- 18 H **TP (18h), pour Application Mobile (Androida), niveau L2, organisé par J. Fabrizio. EPITA, Lyon**
Contenu : Introduction à la programmation mobile sur Android et des classes et objets spécifiques à ses librairies.
Compétences : J'ai appris la programmation mobile grâce à ce cours. Le matériel pour ces cours était entièrement fourni. J'ai contribué à l'écriture et à la correction des examens.
- 8 HTD **Tutorat pour Types, niveau L3, organisé par A. Pitts. The University of Cambridge Computer Laboratory**
Contenu : Introduction aux systèmes de types et à leur application dans la conception de langages de programmation.
Compétences : Ce cours m'a permis de voir un autre système d'éducation, par groupe d'étudiants extrêmement réduits (3 étudiants), je ne suis pas convaincue que cela soit nécessaire. Le matériel de ce cours était entièrement fourni, certains corrigés mis à part.
- 10 HTD **Tutorat pour Vérification de programmes, niveau L1–L3, organisé par P. Lafourcade et P. Mironescu, dans le cadre de École d'été MathInfoly 2019 (Lyon)**
Contenu : Introduction à la logique formelle et à la vérification de programmes au travers un projet de mise sous-forme SAT d'un jeu combinatoire et de TP Coq.
Compétences : Je retire de cette expérience la bonne dynamique que peuvent apporter l'apprentissage par projets adaptables aux niveau des étudiants. Le matériel pour ces cours était entièrement fourni.
- 4 HTD **Jury de stage de L3, Département d'informatique de l'ENS de Lyon**
Contenu : Relectures de deux rapports de stage et notations de quatre soutenances.

Liste détaillée de mon expérience de recherche

Au cours de mes années d'études j'ai exploré différents domaines de la recherche fondamentale en informatique, allant de la vérification de programmes à la théorie des graphes, en passant par la sémantique pour la logique et les langages de programmation qui est aussi mon domaine de recherche actuel :

- Oct. 2023 – **Back to the trees, système intelligible pour l’identification des plantes**,
 Oct 2025 *Post-doctorat, Centre Inria de l’Université de Rennes, Rennes, Chef de projet :*
 S. Castellan
 base de donnée morphologique et générateur de clé de détermination botanique écrit
 en OCaml. Clicable : version 1.0 de la clé avec 230 espèces.
- Jan. 2022 – **Syntaxe parallèle et sémantique interactive de la recherche de preuve**,
 Mai 2023 *Post-doctorat, Laboratoire d’Algorithmique, Complexité et Logique, Université*
 Paris Est Créteil, Chefs de projet : L. Pellissier, A. Saurin
Résumé : Etablissement d’une syntaxe parallèle et d’une sémantique de jeux concu-
 rrente pour la C-ludique.
- Nov. 2019 – **Sémantique différentielle pour les programmes (projet DIAPASoN)**,
 Oct. 2021 *Post-doctorat, Dipartimento Informatica Scienza e Ingeneria, Università di*
 Bologna, Chef de projet : U. Dal Lago
Résumé : Explorer l’expressivité des modèles de jeux pour refléter des notions de
 distances entre programmes ou d’approximation de programmes.

Face à l’apparition de nouveaux paradigmes de programmation approchée (données bruitées, optimisation avec perte de précision, calcul sur du matériel instable) les notions usuelles d’équivalence de programme doivent être repensées pour rendre compte plutôt de notions de *distance* de programme.

Dans ce cadre, le but du projet ERC Diapason porté par Ugo Dal Lago, et auquel je me suis jointe en novembre 2019, était d’investiguer les différents modèles sémantiques établis pour les langages de programmation afin de comprendre s’ils étaient capables de rendre compte de telles notions. En tant que sémanticienne des jeux, j’étais donc chargée d’analyser les modèles de jeux.

En suivant l’idée que les espaces métriques peuvent être vu comme des catégories enrichies sur des quantales, j’ai tout d’abord montré que certains modèles de jeux standard (les modèle de jeux séquentiel non-déterministe, les modèles de jeux causaux déterministes et disjonctifs) définissaient des catégories faiblement enrichies sur des semi-lattices (donc des quantales). En particulier, cela signifie que la distance entre deux programmes peut être interprétée comme le sup de leur interprétation dans le modèle, vu comme la stratégie approximée de ces deux stratégies.

Malheureusement, pour les programmes d’ordre supérieur, cette notion de distance comme stratégie approchée est beaucoup trop descriptive pour définir une bonne abstraction. Je n’ai pas trouvé de moyen d’alléger cette interprétation qui ne cassent pas la compositionnalité ou ne reviennent pas à simplement mesurer l’inéquivalence entre deux programmes.

Une autre limitation à cette approche était son manque de modularité : en particulier, en rajoutant la divergence dans le modèle sans changer le reste de la construction, on pouvait obtenir une distance non divergente à partir de deux termes divergents.

Ces recherches peu fructueuses ont également été rendues difficiles par la crise sanitaires du COVID en 2020, suivi l’année d’après par plusieurs décès familiaux.

- Sept. 2016 – **Jeux concurrents enrichis : témoins pour les preuves et les pro-**
 Oct. 2019 **grammes**, *Doctorat, Laboratoire de l’Informatique du Parallélisme, Lyon,*
 Encadrants : P. Clairambault, O. Laurent et G.Winskel
Résumé : Enrichissement du modèle des jeux concurrents par des annotations
 paramétrisables permettant de refléter certains aspects quantitatifs du calcul tel
 que le temps d’exécution de programmes concurrents ou la construction de témoins
 existentiels en logique du premier ordre.
- Mars–Août
 2016 **Jeux concurrents pour la logique du premier ordre**, *The Computer*
Laboratory, Université de Cambridge, Encadrant : G.Winskel
Résumé : Construction d’un modèle de jeux concurrents permettant l’interprétation
 des preuves de la logique du premier ordre avec coupure.
- Septembre
 2015– Février
 2016 **Problèmes de coloration dans les graphes cubiques et comptage de**
sous-ensembles denses, *Laboratoire Lorrain de Recherche en Informatique*
et ses Applications, Nancy, Encadrants : E. Jeandel et J-S. Sereni
Résumé : Travail autour de la conjecture d’existence d’une 2-coloration 1-défectueuse
 équitable dans tout graphe cubique sans pont autre que le graphe de Peterson.
 Recherche de bornes min et max au nombre de sous-graphes denses dans certaines
 famille de graphes.

- Février–Août 2015 **Impact de la non-monotonie locale dans les RAB asynchrones**, *Laboratoire d'Informatique Fondamentale*, Marseille, Encadrants : K. Perrot et S. Sené
Résumé : Caractérisation de la dynamique asynchrone des réseaux xor fortement connexes par des méthodes algorithmiques et de réécriture.
- Juin–Août 2014 **Concurrent program as proofs : compacting message passing logic**, *Département d'Informatique, Université de Calgary*, Canada, Encadrant : R. Cockett
Résumé : Étude des effets de la compactification d'une logique pour le passage de messages sur la sémantique catégorique et la sémantique de programmation associées.
- Juin–Juillet 2013 **Sémantique 2-catégorique des langages de programmation**, *Laboratoire de Mathématiques de l'Université de Savoie*, Chambéry, Encadrant : T. Hirschowitz
Résumé : Etude des modèles catégoriques des langages de programmation par la construction d'une 2-signature pour le lambda calcul en appel par valeur.
- Juin–Juillet 2012 **Monitoring d'automates décentralisé**, *Laboratoire d'Informatique de Grenoble*, Grenoble, Encadrant : Y. Falcone
Résumé : Recherche d'un modèle d'automate LTL performant pour la surveillance de système décentralisé.
- Juin 2011 **Débruitage de données d'un spectromètre de masse**, *Institut de Planétologie et d'Astrophysique de Grenoble*, Encadrants : V. Vuitton et Q. Mériqot
Résumé : Algorithme d'extraction de points pertinents à partir des données bruitées d'un spectromètre de masse.