

Travaux accomplis

Charles Grellois

Janvier 2017

Résumé. Dans ce document, je décris mes activités à ce jour :

- de **recherche** :
 - à la croisée de la sémantique et du model-checking d'ordre supérieur : durant ma thèse, nous avons repris avec Paul-André Melliès le résultat de décidabilité du model-checking d'ordre supérieur tel que présenté par Kobayashi et Ong [KO09] à la lumière de la logique linéaire et de ses modèles, introduisant ainsi de nouveaux outils sémantiques issus de ces considérations venant de la théorie des automates ; nous avons également apporté un éclairage nouveau sur le problème [GM15a, GM15b, GM15c, Gre16],
 - pour la terminaison des langages fonctionnels d'ordre supérieur probabilistes : durant mon post-doc avec Ugo dal Lago, nous avons introduit un système de types pour ces langages garantissant leur terminaison quasi-sûre (avec probabilité 1) [DG17],
 - en coinduction et combinatoire des mots : j'ai fait une remarque sur la nature coinductive de la solution d'une équation de langages permettant de reprendre et de comprendre un résultat d'indécidabilité de Kunc [Kun05] à la lumière de la notion de coinduction, remarque qui a donné lieu à une communication aux Journées Montoises 2014 [Gre14],
- d'**enseignement** : monitorat à Paris 7 (192h), TD au Master 2 LMFI (20h) sur la dualité en logique et en informatique, colles de mathématiques et cours d'informatique (en Maple) en classes préparatoires,
- d'**animation de la recherche** : animation de groupes de travail sur le model-checking d'ordre supérieur au LIAFA et à PPS (Université Paris 7) dans le cadre de la démarche de fusion de ces laboratoires, participation à un groupe de travail d'étudiants à Ulm, communications (44 depuis 2009), représentant des doctorants de mon ED, visites et invitations,
- et mes **publications** : 4 conférences internationales avec actes + 1 workshop international avec actes.

Table des matières

1	Curriculum Vitae	3
2	Recherche scientifique	7
2.1	Une vision sémantique du model-checking d'ordre supérieur . . .	7
2.1.1	Historique	7
2.1.2	Contributions	9
2.2	Terminaison des programmes probabilistes fonctionnels d'ordre supérieur	12
2.2.1	Historique	12
2.2.2	Contributions	12
2.3	Coinduction et combinatoire des mots	13
3	Enseignement	14
3.1	Dans le cadre du monitorat	14
3.2	Complément à l'Université	14
3.3	Hors de l'Université	14
4	Animation de la recherche et responsabilités collectives	16
4.1	Communication scientifique et animation de la recherche	16
4.2	Responsabilités collectives	16
4.3	Comités de programme	16
4.4	Activités de relecture d'articles	17
4.5	Visites et invitations	17
4.6	Autres évènements	17
5	Collaborations	19
5.1	Participation à des projets	19
5.2	Collaborations scientifiques	19
6	Publications	20
6.1	Conférences d'audience internationale avec actes et comité de sélection	20
6.2	Workshops d'audience internationale avec actes et comité de sélection	20
6.3	Thèses et mémoires	20

Note importante. Dans la partie sur les contributions scientifiques, des pans de mon programme de recherche sont repris, afin de rendre ce document autonome. Seules les sections 2.1.2 et 2.3 sont nouvelles.

1 Curriculum Vitae

Charles GRELLOIS
Né le 5 février 1988 à Bordeaux
Contact postal : 36 rue de Laseppe
33000 Bordeaux
+33 6 71 14 12 44
charles.grellois@inria.fr
<http://research.grellois.fr>

Intérêts scientifiques

J'ai travaillé durant ma thèse à l'interface de la **sémantique** et de la **vérification** : automates d'arbres, langages et reconnaissance, logique sur les arbres, logique linéaire et ses modèles, sémantique des jeux, théorie des types, coinduction.

Je travaille actuellement sur les aspects **probabilistes** du calcul, et en particulier sur la terminaison probabiliste, avec Ugo Dal Lago à Bologne.

J'ai travaillé un peu sur la combinatoire des mots, et ai montré l'importance de la nature coinductive des centralisateurs de langages.

J'ai un goût prononcé pour la synthèse et l'**unification** de domaines.

Situation actuelle

Post-doctorant, équipe FOCUS (INRIA Sophia-Antipolis et Université de Bologne), depuis le 1er janvier 2016 et jusqu'au 31 août 2017. Responsable : Ugo DAL LAGO.

Emplois précédents

- **Research Assistant**, University of Dundee, UK, du 1er septembre 2015 au 31 décembre 2015.
- **Allocataire Moniteur Normalien**, Université Paris 7, du 1er septembre 2012 au 31 août 2015.
- **Fonctionnaire-stagiaire (élève de l'ENS Cachan)**, du 1er septembre 2008 au 31 août 2012.

Formation post-baccalauréat

- 2012 - 2016 **Thèse de doctorat** d’informatique, laboratoires PPS et LIAFA (Université Paris 7), mention *très honorable*.
Titre : *Sémantique de la logique linéaire et model-checking d’ordre supérieur*.
Directeurs : Paul-André MELLIÈS et Oliver SERRE.
Relecteurs : Kazushige TERUI and Igor WALUKIEWICZ.
Soutenue le 8 avril 2016.
- 2011 - 2012 **Master II** d’informatique “MPRI”, Paris 7/Cachan/Ulm/X, mention *bien*.
Stage à PPS (Paris 7) sous la direction de Paul-André MELLIÈS.
Titre : *Interprétation catégorique des arbres rationnels*.
<http://research.grellois.fr/doc/grellois-interpretation-categorique-des-arbres-rationnels.pdf>
- 2010 - 2011 **Master II** de Mathématiques Fondamentales, Université Paris 6, mention *bien*.
Mémoire à PPS (Paris 7) sous la direction de Paul-André MELLIÈS.
Titre : *Algebraic theories, monads, and arities*.
<https://arxiv.org/abs/1110.3294>
- 2009 - 2010 **Master I** d’informatique “MPRI”, Paris 7/Cachan/Ulm/X, mention *bien*.
Stage à Oxford sous la direction de Luke ONG.
Titre : *Game semantics and higher-order model-checking*
<http://research.grellois.fr/doc/grellois-game-semantics-and-ho-model-checking.pdf>

Recherche

Publications

Conférences d’audience internationale avec actes et comité de sélection

1. Ugo Dal Lago et Charles Grellois. *Probabilistic Termination by Monadic Affine Sized Typing*, European Symposium on Programming (ESOP) 2017.
2. Charles Grellois et Paul-André Melliès. *Relational semantics of linear logic and higher-order model-checking*. EACSL Annual Conference on Computer Science Logic (CSL) 2015.
3. Charles Grellois et Paul-André Melliès. *Finitary semantics of linear logic and higher-order model-checking*. Mathematical Foundations of Computer Science (MFCS) 2015.
4. Charles Grellois et Paul-André Melliès. *An infinitary model of linear logic*. Foundations of Software Science and Computation Structures (FoSSaCS) 2015.

Workshops d’audience internationale avec actes et comité de sélection

1. Charles Grellois et Paul-André Melliès. *Indexed linear logic and higher-order model checking*. Intersection Types and Related Systems (ITRS) 2014.

Communications

44 exposés dont :

- 3 conférences internationales avec actes et comité de sélection,
- 2 conférences internationales sans actes et avec comité de sélection,
- 1 conférence nationale sans actes et avec comité de sélection,
- 1 workshop international avec actes et comité de sélection,
- 4 workshops internationaux sans actes et avec comité de sélection,
- 4 workshops internationaux sans actes et sur invitation,
- 2 workshops internationaux sans actes ni comité de sélection,
- 5 workshops nationaux sans actes ni comité de sélection,
- 19 séminaires,
- 3 exposés lors de journées de rentrée de laboratoire ou d'équipe.

Liste complète et slides : http://research.grellois.fr/talks_date.htm

Evénements

- Août-septembre 2016 : programme “Automata, Logic and Games” de l’*Institute for Mathematical Science* de Singapour. Invité sur un financement pour jeunes chercheurs.
- Mars 2016 : workshop NII Shonan sur la vérification d’ordre supérieur (sur invitation).
- Mai 2015 : école “EPIT” sur Coq organisée par Yann Régis-Gianas
- Février 2013 : école d’hiver du réseau ESF Games sur la logique et les jeux
- Printemps 2011 : école sur la géométrie de l’interaction du projet LOGOI
- Juillet 2010 : séminaire Dagstuhl 10252 “Game semantics and verification” (sur invitation)

Organisation de la recherche

Animation de groupes de travail

- Organisateur et fondateur du groupe de travail bimensuel *Sémantique et Vérification* de novembre 2014 à septembre 2015. Audience usuelle : 15-20 personnes. Ce groupe, commun au LIAFA et à PPS, était le premier du genre et s’inscrivait dans la démarche de fusion de ces deux laboratoires qui ont donné l’IRIF en janvier 2016.
Page du groupe : <http://research.grellois.fr/gdt.htm>
- Organisateur et fondateur du groupe de lecture *Schémas, Automates et Sémantique*, plus restreint (audience usuelle 6-7 personnes) de mars à mai 2013. Site : <http://sas.grellois.fr>
- Membre organisateur du groupe de travail des étudiants en logique de Paris qui se tenait chaque semaine à Ulm de 2010 à 2012. Programme de la dernière année : <http://gdt.ludics.eu/index.php/S%C3%A9ances2011-2012>

Responsabilités collectives

Représentant élu des doctorants, école doctorale 386 *Sciences Mathématiques de Paris Centre* (2014-2016).

Relectures et comités de programme

- Comité de programme : DICE-FOPARA 2017 (membre).
- Relecteur pour des conférences : 12 articles pour les conférences TCS (2014), ICFP (2015), MFPS (2015), CSL (2015, 2016), FoSSaCS (2016, 2017), LICS (2016), FSCD (2016), et LCC (2016).
- Relecteur pour *Mathematical Reviews*, qui fait partie de l'American Mathematical Society (AMS).

Enseignement

- Monitorat (2012-2015) : 192h en licence à l'UFR d'Informatique de Paris 7.
- Complément (2013-2014) : 20h de TDs au Master 2 LMFI pour le cours de Mai Gehrke sur la dualité en mathématiques et en informatique.
- Classes préparatoires (2010-2012) : colles de mathématiques (Lycée Buffon, Paris) et TPs d'informatique/Maple (Lycée Carnot, Paris).

2 Recherche scientifique

2.1 Une vision sémantique du model-checking d'ordre supérieur

2.1.1 Historique

Les systèmes automatisés occupent aujourd'hui une place prépondérante – et toujours croissante – dans nos sociétés. La qualité des algorithmes et des logiciels, notamment lorsqu'ils gèrent des systèmes critiques, est ainsi une problématique primordiale des sciences du logiciel. Parmi les techniques utilisées pour garantir la qualité du code, citons :

- la *sémantique*, qui étudie la structure mathématique et la modélisation de la programmation logicielle,
- la *théorie des types*, qui permet notamment de garantir certaines propriétés des programmes de façon modulaire, en garantissant que la composition de diverses instructions préserve la propriété,
- et le *model-checking*, une technique de vérification de programmes visant à vérifier de façon automatisée qu'une approximation du programme (le *modèle*) satisfait une *propriété* logique donnée.

Mes contributions autour du model-checking d'ordre supérieur [GM15a, GM15b, GM15c, Gre16] s'inscrivent à l'intersection de ces trois courants.

Model-checking d'ordre supérieur. Le *model-checking* est une technique clef en vérification, introduite dans les années quatre-vingt par Clarke, Emerson et Sifakis [CE81, QS82]. Le raisonnement sur les programmes se heurtant à l'indécidabilité induite par le théorème de Rice, l'idée du model-checking est de remplacer le programme par une approximation, son *modèle*. On peut par exemple abstraire les instructions conditionnelles *if* et les remplacer par un symbole binaire *if*, remplacer les données par une unique constante *data*,... obtenant ainsi un modèle du programme sous la forme d'un arbre dont les branches représentent des abstractions des traces d'exécution potentielles du programme. Un exemple est donné en Figure 1, et représente les traces d'exécution d'un serveur qui attend des données et, à chaque étape, clôt la connection et renvoie les données accumulées, ou continue à écouter. En formulant des propriétés sur les arbres de traces dans une logique adaptée, on peut obtenir des résultats de décidabilité, permettant de garantir algorithmiquement que l'approximation du programme qu'est son modèle satisfait ou non une propriété donnée. Pour permettre un raisonnement algorithmiquement décidable, le modèle du programme doit admettre une représentation finitaire. Dans un premier temps, des modèles tels que les graphes finis, permettant de représenter des arbres infinis dits *réguliers*, furent considérés. Notons que l'arbre \mathcal{T} de la Figure 1 n'est pas régulier. Dans la seconde partie des années quatre-vingt-dix, des arbres infinis plus complexes mais finiment présentables furent introduits, comme ceux générés par des processus à piles [Wal96, BEM97].

Dans les années soixante-dix, pour donner une sémantique à des langages fonctionnels, Nivat [Niv72] puis Damm [Dam77] introduirent les *schémas de récursion d'ordre supérieur*. Ceux-ci permettent de générer des arbres infinis représentant une approximation de l'ensemble des traces du programme d'origine, et correspondent aux arbres générés par une certaine classe d'automates :

les automates à piles de piles de piles... munis d'une opération d'effondrement [HMOS08]. L'arbre \mathcal{T} de la Figure 1 peut par exemple être généré par le schéma de récursion

$$\mathcal{G} = \begin{cases} S & = L \text{ Nil} \\ L x & = \text{if } x (L (\text{data } x)) \end{cases}$$

ou par un automate à piles de piles... effondrant équivalent. Les schémas de récursion ont connu un regain d'intérêt au cours de la dernière décennie car ils permettent de modéliser les langages de récursion fonctionnels d'ordre supérieur (tels que C++, Haskell, OCaml, Javascript, Python, ou Scala), dont l'importance est croissante dans la pratique de la programmation. Des méthodes d'abstraction automatique à partir du code vers les schémas de récursion existent, et ont notamment été développées par le groupe de Naoki Kobayashi à Tohoku puis à Tokyo.

Sur ces arbres infinis, il reste à exprimer logiquement les formules dont on veut tester la véracité. Une logique usuellement considérée en vérification est la logique monadique du second ordre (MSO), car elle est "proche de la frontière de la décidabilité" : elle est décidable dans diverses situations dignes d'intérêt, telles que l'arbre binaire complet [Rab69] ou les graphes infinis générés par des processus à piles [MS85]; cependant la plupart de ses extensions "suffisamment significatives" sont indécidables.

Le problème du model-checking d'ordre supérieur consiste alors, étant donné un schéma de récursion \mathcal{G} représentant un arbre infini $\langle \mathcal{G} \rangle$, et une formule φ exprimée en logique monadique du second ordre, à décider si φ est vraie à la racine de $\langle \mathcal{G} \rangle$. Après plusieurs résultats préliminaires sur des restrictions du problème (voir [Gre16] pour un historique), Ong prouva en 2006 la décidabilité du problème général [Ong06]. Sa preuve, particulièrement élaborée, fait intervenir la sémantique des jeux pour l'analyse du schéma de récursion, et est un premier point de rencontre entre sémantique et model-checking. Deux problèmes plus difficiles, celui du model-checking *global* et celui de la sélection, sont introduits et prouvés décidables dans [BCOS10] et [CS12].

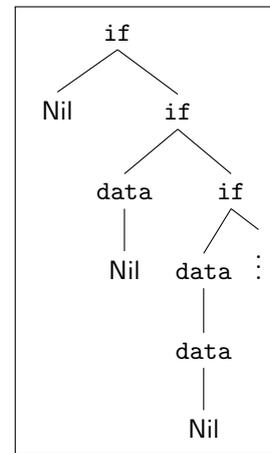


FIGURE 1 – L'arbre de traces \mathcal{T} , représenté finement par le schéma de récursion \mathcal{G} .

Types intersection et model-checking d'ordre supérieur. Le résultat d'Ong donna lieu à de multiples preuves alternatives, toutes particulièrement élaborées, et que nous ne listerons pas ici (voir [Gre16]). Le travail de Kobayashi [Kob09], puis de Kobayashi et Ong [KO09] donne une telle preuve alternative mais utilisant cette fois la *théorie des types*. Il est à noter que cette preuve est loin d'être triviale; la version publiée [KO09] est incomplète, et la version complétée [KO] n'est pas encore publiée¹. Les formules MSO peuvent être vérifiées au

1. Une preuve complète apparaît dans [Gre16]. Elle est basée sur [KO] mais est adaptée afin de révéler la véritable nature sémantique de l'opération de coloration des types par les automates, qui devient une comonade paramétrée.

moyen d’automates alternants à parité, automates d’arbres non-déterministes qui :

- sont *alternants* : ils ont la capacité, au cours de leur exécution, de dupliquer un sous-arbre pour l’explorer dans plusieurs états différents, ou de ne pas l’explorer,
- et sont régis par une condition de *parité* : chaque état se voit attribuer un entier, appelé *couleur*, et une condition sur l’apparition de celles-ci le long des branches infinies d’une exécution permet de discriminer les exécutions *gagnantes* des exécutions *perdantes*.

Une formule MSO φ peut être traduite en un automate alternant à parité \mathcal{A}_φ qui lui est équivalent : φ est vraie à la racine d’un arbre potentiellement infini si et seulement si \mathcal{A}_φ a une exécution gagnante sur celui-ci. Etant donné un automate alternant à parité, Kobayashi et Ong donnent un système de types pour les schémas de récursion, doté de trois particularités :

- ces types sont des types *intersection* : les dérivations de typage ont la capacité de dupliquer un terme pour prouver qu’il a plusieurs types, ou de ne pas l’explorer ; ces types étant *idempotents* : $\tau \wedge \tau \equiv \tau$,
- ces types sont *annotés* par une couleur, afin de refléter l’action de la coloration des automates dans le typage des schémas,
- et la récursion est gérée, de façon externe, par un jeu de parité — ce qui peut également se faire de façon interne au moyen de dérivations infinitaires colorées [Gre16].

Dans le jeu de parité induit par le typage, Eve admet une stratégie gagnante si et seulement si l’arbre infini représenté par le schéma de récursion est accepté par l’automate qui paramètre la construction du système. La décidabilité des jeux de parité et la finitude du typage, due à l’idempotence, permet à Kobayashi et Ong de donner une preuve de décidabilité du model-checking d’ordre supérieur.

2.1.2 Contributions

Types intersection et sémantique relationnelle de la logique linéaire.

La logique linéaire est une logique sous-structurelle qui fut introduite par Girard [Gir87] suite à l’étude de la décomposition des fonctions stables dans la sémantique cohérente. L’intuition majeure est qu’une fonction de A dans B peut se décomposer en deux étapes, selon l’équation

$$A \rightarrow B = !A \multimap B$$

où :

- $!A$ est l’*exponentielle* de l’objet A , et indique que A peut être dupliqué ou ne pas être utilisé, tandis qu’un objet qui n’est pas appliqué à une exponentielle doit être utilisé une unique fois : les règles structurelles usuelles ne s’appliquent qu’aux objets *promus* par l’exponentielle,
- et \multimap est l’*implication linéaire*, qui utilise une unique fois son argument $!A$ pour produire B .

La proximité existant entre le comportement de l’exponentielle en logique linéaire et celui de l’opérateur d’intersection dans les systèmes de types précédemment cités a amené de Carvalho à introduire un système de types intersection capturant la sémantique relationnelle de la logique linéaire [dC09]. Ce travail procède d’intuitions venant de la logique linéaire indexée de Bucciarelli et Ehrhard [BE00, BE01].

Notre première contribution, avec Melliès [GM15b], a été :

- de relier le système idempotent de Kobayashi à une variante équivalente et non-idempotente, préparant ainsi la connexion avec la sémantique relationnelle de la logique linéaire,
- puis de relier le système de types intersection non-idempotents ainsi obtenu à la logique linéaire indexée de Bucciarelli et Ehrhahrd.

Une conséquence importante des travaux de Bucciarelli et Ehrhahrd [BE00, BE01] qui n’apparaît pas dans [GM15b] pour des raisons d’espace, mais apparaît dans [Gre16], est le théorème préliminaire de model-checking sémantique suivant : nous définissons un modèle dans lequel l’interprétation d’un terme sans récursion calculant un arbre est l’ensemble des états depuis lesquels un automate donné à *condition de parité triviale*, paramétrant le modèle, accepte l’arbre calculé par ce terme. Cette idée peut être comprise comme une extension de la notion de reconnaissance par monoïde à une reconnaissance par une structure adaptée aux termes : les domaines, suivant une idée d’Aehlig [Aeh06] et de Salvati [Sal09].

Un modèle relationnel infinitaire et coloré de la logique linéaire. Pour étendre cette approche aux schémas de récursion et aux automates avec condition de parité générale, la première étape que nous avons suivie avec Melliès a été d’introduire un modèle relationnel infinitaire de la logique linéaire, permettant de gérer le calcul d’arbres infinis et les annotations de couleur [GM15c]. Ce modèle de la logique linéaire propose une interprétation nouvelle de la modalité exponentielle, obtenue comme composition d’une exponentielle infinitaire et d’une comonade de coloration réalisée à l’aide d’une loi distributive entre comonades. L’intuition guidant ce travail est une extension de la connexion entre types intersection et sémantique relationnelle pour capturer une version non-idempotente et infinitaire du système de Kobayashi et Ong [KO09]. Nous introduisons aussi, pour la première fois à notre connaissance, un opérateur de point fixe qui n’est pas continu et qui se comporte de façon inductive ou coinductive selon la parité courante. Un tel opérateur a été défini de façon parallèle par Salvati et Walukiewicz [SW15].

La coloration de Kobayashi et Ong était une comonade ! L’introduction de ce modèle relationnel infinitaire est allée de pair avec une reprise du système de types de Kobayashi et Ong à la lumière d’outils sémantiques contemporains, et en particulier des comonades paramétrés de Melliès [Mel14]. Nous avons ainsi repris le système de types de Kobayashi et Ong dans [GM15d], obtenant une variante dans laquelle

- la coloration se comporte comme une modalité, ingrédient fondamental pour la reprise sémantique du résultat,
- les dérivations de typage sont de profondeur infinie et la condition de gain des automates alternants à parité est reflétée sur les branches infinies de ces dérivations : nous complétons en ce sens l’approche de Kobayashi et Ong qui utilisait un jeu de parité externe pour en extraire toute la portée en théorie de la preuve. A noter que de telles conditions de parité sur les branches infinies de dérivations apparaissent dans un travail en cours de Baelde, Doumane et Saurin sur la logique μ MALL.

Le modèle relationnel infinitaire du paragraphe précédent vient de ces deux

observations : la découverte de la nature modale de la coloration permet de la refléter comme une comonade (paramétrée) dans les modèles, et la reformulation de la condition de parité guide la définition du point fixe inductif-coinductif. Nous remarquons aussi dans [GM15d] la nature profondément duale en logique linéaire des automates et des schémas de récursion, qui motive la direction I.2 de mon projet de recherche, et nous introduisons une variante indexée et colorée de la logique tensorielle.

Décidabilité sémantique du model-checking d'ordre supérieur. Le modèle relationnel de la logique linéaire est relié au modèle dit de Scott de la logique linéaire par une relation d'effondrement extensionnel [Ehr12]. Dans ce modèle, l'interprétation d'un type est finie, et on y a donc espoir de retrouver des résultats de décidabilité. Ce modèle est cependant plus complexe que le modèle relationnel, ce qui a motivé notre approche préliminaire via la sémantique relationnelle. Nous avons finalement introduit une modalité de coloration et un opérateur de point fixe inductif-coinductif, obtenus comme des effondrements extensionnels de ceux de la sémantique relationnelle, dans [GM15a]. La finitude du modèle nous donne alors le théorème de reconnaissance par domaine désiré : l'interprétation d'un schéma de récursion d'ordre supérieur est l'ensemble des états depuis lesquels un automate à parité donné accepte l'arbre représenté par le schéma. L'aspect finitaire de la sémantique induit une nouvelle preuve de la décidabilité du problème de model-checking d'ordre supérieur.

Une connexion avec la réécriture infinitaire coinductive. Une dernière contribution, qui n'apparaît que dans ma thèse [Gre16], est la reformulation de la normalisation des schémas de récursion d'ordre supérieur à l'aide de la coinduction, et également la définition d'une réécriture coinductive faisant interagir schéma et automate pour calculer un arbre d'exécution. Cette seconde relation est dans l'esprit du travail de Salvati et Walukiewicz sur la machine de Krivine pour le model-checking d'ordre supérieur [SW14], mais est formulée sans cette machine et dans un cadre coinductif. Ainsi que je le détaille dans la direction I.1 de mon projet de recherche, je pense que cette reformulation pourra permettre une preuve plus élégante du théorème de pleine adéquation de Kobayashi et Ong.

Bilan de cet axe de recherche. L'étude sémantique de ce problème de model-checking d'ordre supérieur, menée durant ma thèse avec Paul-André Mellès, a fait intervenir une série de concepts de premier plan en sémantique contemporaine, tels que la logique linéaire et sa sémantique relationnelle, la logique linéaire indexée, les lois distributives entre comonades, les comonades paramétrées, la logique tensorielle, la coinduction. . . Ces concepts ont contribué de façon particulièrement naturelle à l'étude du model-checking d'ordre supérieur ; mais cette étude a également questionné la sémantique telle que nous la connaissons et a amené à l'introduction d'outils nouveaux : un modèle infintaire, un opérateur de point fixe inductif-coinductif qui n'est pas continu dans la sémantique relationnelle infinitaire, un système de coeffets modélisant la coloration induite par les automates au cours du calcul. . . Cette étude est préliminaire, et ouvre beaucoup de questions pertinentes en sémantique et en logique, ainsi que je le détaille dans mon projet de recherche. Se pose notamment la question

de l’extension de ces méthodes à d’autres propriétés et d’autres classes d’automates, par exemple pour vérifier des programmes d’ordre supérieur *probabilistes*.

2.2 Terminaison des programmes probabilistes fonctionnels d’ordre supérieur

2.2.1 Historique

Systèmes aléatoires en informatique. Les modèles probabilistes sont de plus en plus présents en informatique : notamment pour l’étude du langage naturel [MS01], la planification de mouvement en robotique [Thr02], la théorie de l’apprentissage [Pea89]. . . Ils sont considérés à deux titres :

- les algorithmes probabilistes, introduits dès les débuts de l’informatique théorique [dLMSS56], permettent souvent d’obtenir des résultats plus efficacement que leurs pendants déterministes,
- et les modèles probabilistes permettent plus généralement de modéliser les systèmes dont le comportement dépend d’un environnement extérieur dont l’action n’est connue qu’au moyen d’une distribution de probabilités.

Ceci a motivé l’étude de langages de programmation probabilistes, particulièrement au cours des vingt dernières années, à la suite d’avancées en théorie des algorithmes aléatoires [MR95], en vérification de protocoles cryptographiques [BGB09, BGHB11], et en machine learning [GMR⁺08]. Dans un langage de programmation probabiliste, des instructions particulières — par exemple, simulant le jet d’une pièce ou d’un dé — sont fournies au programmeur, et permettent de faire évoluer le calcul de façon probabiliste plutôt que déterministe.

Terminaison et aléa : l’apport de la théorie des types. Une des propriétés les plus cruciales attendues d’un programme est sa *terminaison*. Pour les programmes (non-)déterministes, la formulation est qualitative : un programme termine, ou ne termine pas. Dans le cas probabiliste, chaque exécution a une probabilité particulière, et la notion de terminaison devient *quantitative*. Il est ainsi naturel d’autoriser un ensemble négligeable d’exécutions à diverger. On obtient alors la terminaison quasi-sûre (*almost-sure termination*, AST [BK02]) : un programme termine si et seulement si l’ensemble de ses calculs terminants est de mesure 1. De nombreuses techniques de décision (semi-)automatiques pour AST ont été récemment introduites [EGK12, FH15, CFNH16, CFG16]. Cependant, elles concernent toutes des programmes *impératifs*, et non *fonctionnels*. Pourtant, les langages fonctionnels probabilistes tels que Church [GMR⁺08] sont actuellement parmi les plus prometteurs en matière de programmation probabiliste.

2.2.2 Contributions

Nous avons proposé récemment, avec Ugo dal Lago, un système de types pour les langages fonctionnels probabilistes, garantissant que les programmes typables sont AST [DG17].

Ce système adapte au cadre probabiliste les *sized types* [HPS96, BFG⁺04, Abe04], qui sont des annotations des types de base avec des informations sur la taille des entrées. Un entier de type $\widehat{\text{Nat}}^i$ sera par exemple 0 (formé d’un

constructeur) ou $1 = S\ 0$ (formé de deux constructeurs), la taille \widehat{i} exprimant qu’au plus deux constructeurs peuvent être utilisés pour former l’entier. Avec ces tailles, on peut contraindre la règle de récursion en s’assurant qu’une fonction gérant une entrée de taille $n + 1$ ne s’appelle récursivement que sur des tailles au plus n . Cette idée permet de garantir, dans les systèmes de *sized types* introduits dans le cas déterministe, que les termes typables sont fortement normalisants. Dans [DG17], nous montrons que ce système peut être adapté à un cadre probabiliste, en permettant à des distributions de types de figurer dans les types eux-mêmes (mais seulement à droite des flèches, suivant les idées de Moggi), et en reliant la règle de point fixe à un processus markovien : pour définir une fonction récursive f sur la taille $n + 1$, on doit utiliser des appels à f qui, après suffisamment d’itérations, appelleront une taille au plus n avec probabilité 1. Notre système est cependant limité par cette modélisation par un processus markovien de type “random walk” : dans chaque branche probabiliste, la fonction f à définir récursivement ne peut apparaître qu’au plus une fois. Ce système est donc *affine*. Comme les autres systèmes de types précédemment cités, le nôtre est adéquat, mais pas complet : un terme typable est AST, mais un terme non typable peut l’être aussi. On teste donc le programme lui-même, mais la “réponse” donnée par le typage n’est pas exacte.

2.3 Coinduction et combinatoire des mots

En 2009, alors que j’étais en stage de L3 à l’Université de Turku sous la direction de Juhani Karhumäki, j’ai travaillé sur le problème de la complexité des centralisateurs de langages de mots. Etant donné un langage L , on appelle la plus grande solution X de l’équation de langages $X \cdot L = L \cdot X$ son *centralisateur*, note $\mathcal{C}(L)$. En 1971, Conway a posé la question suivante : que peut-on dire de la complexité d’un centralisateur $\mathcal{C}(L)$ en fonction de celle du langage L ? Kunc a montré en 2005 [Kun05] qu’il existe des langages réguliers, et même des langages finis, dont le centralisateur n’est pas récursivement énumérable. J’ai travaillé durant mon stage sur sa preuve, qui utilisait des machines de Minsky, et l’ai refaite à l’aide de machines de Turing circulaires [NW09]. J’ai réalisé quelques années plus tard, en étudiant la coinduction, que l’idée fondamentale du résultat de Kunc était précisément coinductive : les centralisateurs sont de plus grandes solutions d’équations, et sont donc des objets coinductifs. Ils permettent ainsi d’encoder non pas le comportement (inductif) d’une machine, mais l’ensemble de ses comportements, dont on peut exclure les composantes connexes à un état initial par un artifice technique. J’ai présenté ces idées lors des Journées Montoises 2014 [Gre14] puis à Turku en janvier 2015 lors d’un séminaire [Gre15] ; je prévois d’écrire une version journal suite à l’intérêt de la communauté de la combinatoire des mots pour ma solution.

3 Enseignement

Je détaille ici les enseignements que j'ai assurés au cours de ma thèse et lors de mes 3e et 4e années d'études à l'ENS Cachan. Je n'ai pas enseigné au cours de mes expériences post-doctorales.

3.1 Dans le cadre du monitorat

J'ai effectué de septembre 2012 à juin 2015 les 192h d'enseignement de mon monitorat à l'**UFR d'Informatique** de Paris 7.

- **Introduction à la programmation** (L1, Université Paris 7, 48h par an, 2014-2015) : TPs de programmation en Java qui prenaient la suite du cours d'Introduction à la programmation. Je me suis particulièrement investi dans la réalisation des fiches de TP, hébergées sur un serveur qui permettait de coder et de tester son code directement en ligne, suite à mon expérience dans la mouture précédente de ce cours. Effectifs d'environ 35 personnes.
- **Éléments d'algorithmique** (L2, Université Paris 7, 24h par an, 2014-2015) : TDs d'algorithmique et contribution à la réalisation des feuilles de TDs, effectifs d'environ 20 personnes. Cf. <http://research.grellois.fr/t/ea3.htm>
- **Introduction à la programmation** (L1, Université Paris 7, 39h par an, 2012-2013 et 2013-2014) : TPs de programmation en Java, effectifs d'environ 35 personnes
- **Mathématiques pour l'informatique** (L3, Ecole d'Ingénieur Denis Diderot, 25h par an, 2012-2013 et 2013-2014) : TDs de mathématiques discrètes (logique, automates, grammaires), effectifs de 15 à 20 personnes. Cf <http://research.grellois.fr/t/mathinfo.htm>

3.2 Complément à l'Université

En complément, j'ai effectué bénévolement durant le second semestre de l'année 2013-2014 les TDs du cours de Mai Gehrke *Dualité en logique et en informatique* au Master 2 LMFI (Logique Mathématique et Fondements de l'Informatique, UFR de Mathématiques de l'Université Paris 7), plus la partie du cours sur les catégories. Total estimé : 20 heures. Effectifs : 6-7 étudiants.

3.3 Hors de l'Université

Avant ma thèse, durant ma 3e et ma 4e année à l'ENS Cachan (années universitaires 2010-2012), j'ai fait des heures complémentaires d'enseignement :

- comme *colleur de mathématiques* au Lycée Buffon (Paris) : examen oral de Mathématiques d'une heure chaque semaine à un groupe de trois étudiants de Mathématiques Supérieures MPSI

— comme *colleur d'informatique* au Lycée Carnot (Paris) : cours-TD de programmation en Maple de deux heures hebdomadaires en classe de Mathématiques Spéciales PC (environ 35 élèves), cf <http://maple.grellois.fr/>. J'ai également fait passer des oraux blancs de Mathématiques à ces classes.

4 Animation de la recherche et responsabilités collectives

4.1 Communication scientifique et animation de la recherche

Animation de groupes de travail.

- Organisateur et fondateur du groupe de travail bimensuel *Sémantique et Vérification* de novembre 2014 à septembre 2015. Audience usuelle : 15-20 personnes. Ce groupe, commun au LIAFA et à PPS, était le premier du genre et s’inscrivait dans la démarche de fusion de ces deux laboratoires qui ont donné l’IRIF en janvier 2016.
Page du groupe : <http://research.grellois.fr/gdt.htm>
- Organisateur et fondateur du groupe de lecture *Schémas, Automates et Sémantique*, plus restreint (audience usuelle 6-7 personnes) de mars à mai 2013. Site : <http://sas.grellois.fr>
- Membre organisateur du groupe de travail des étudiants en logique de Paris qui se tenait chaque semaine à Ulm de 2010 à 2012. Programme de la dernière année : <http://gdt.ludics.eu/index.php/S%C3%A9ances2011-2012>

Communication scientifique. J’ai donné 44 exposés depuis 2009 (plus sept que je ne compte pas ici au groupe de travail des étudiants en logique de Paris). La liste complète est disponible à l’adresse http://research.grellois.fr/talks_date.htm (avec les liens vers les divers événements et jeux de “slides”). Parmi ces exposés, distinguons :

- 3 conférences internationales avec actes et comité de sélection,
- 2 conférences internationales sans actes et avec comité de sélection,
- 1 conférence nationale sans actes et avec comité de sélection,
- 1 workshop international avec actes et comité de sélection,
- 4 workshops internationaux sans actes et avec comité de sélection,
- 4 workshops internationaux sans actes et sur invitation,
- 2 workshops internationaux sans actes ni comité de sélection,
- 5 workshops nationaux sans actes ni comité de sélection,
- 19 séminaires,
- 3 exposés lors de journées de rentrée de laboratoire ou d’équipe.

4.2 Responsabilités collectives

Représentant élu des doctorants, école doctorale 386 *Sciences Mathématiques de Paris Centre* (2014-2016).

4.3 Comités de programme

DICE-FOPARA 2017 (membre).

4.4 Activités de relecture d'articles

- Relecteur pour des conférences : 12 articles pour les conférences TCS (2014), ICFP (2015), MFPS (2015), CSL (2015, 2016), FoSSaCS (2016, 2017), LICS (2016), FSCD (2016), et LCC (2016).
- Relecteur pour Mathematical Reviews, qui fait partie de l'American Mathematical Society (AMS).

4.5 Visites et invitations

- Août-Septembre 2016 : participation² au mois thématique *Automata, Logic and Games* de l'*Institute for Mathematical Science* de Singapour. Invité sur un financement pour jeunes chercheurs.
- Mars 2016 : participation (sur invitation) au séminaire NII Shonan (Tokyo) sur la vérification d'ordre supérieur.
- Octobre 2015 : invitation dans le groupe de Lars BIRKEDAL à Aarhus (2 semaines)
- Septembre 2015 : invitation dans le groupe d'Antonio DI NOLA à Salerne (1 semaine). J'y ai donné un mini-cours de 8 heures sur ma recherche de thèse.
- Janvier 2015 : invitation dans le groupe de Juhani KARHUMÄKI à Turku (2 semaines) suite à ma communication aux Journées Montoises 2016 reliant la coinduction à un problème de combinatoire des mots.
- Juillet 2010 : participation (sur invitation) au séminaire Dagstuhl sur la sémantique des jeux et la vérification.

Visites à Oxford, à Luke ONG et Takeshi TSUKADA : février 2015 (1 semaine), octobre 2013 (1 semaine), août 2013 (1 mois)

4.6 Autres évènements

- Août-septembre 2016 : programme "Automata, Logic and Games" de l'*Institute for Mathematical Science* de Singapour. Invité sur un financement pour jeunes chercheurs.
- Mars 2016 : workshop NII Shonan sur la vérification d'ordre supérieur (sur invitation).
- Mai 2015 : école "EPIT" sur Coq organisée par Yann Régis-Gianas.
- Février 2013 : école d'hiver du réseau ESF Games sur la logique et les jeux.

2. http://www2.ims.nus.edu.sg/Programs/016auto/gen_visitorlist.php#overseas

- Printemps 2011 : école sur la géométrie de l'interaction du projet LOGOI.
- Juillet 2010 : séminaire Dagstuhl 10252 “Game semantics and verification” (sur invitation).

5 Collaborations

5.1 Participation à des projets

Je participe aux ANR PACE et ELICA qui m'emploient durant mon postdoc à Bologne (1 an sur PACE, 6 mois sur ELICA).

J'ai par ailleurs été invité, au titre de la proximité thématique, à parler lors des journées des ANR RAPIDO en 2015 et LOCALI en 2014.

Je suis membre du GDR Informatique Mathématique, et des groupes de travail Geocal (Géométrie du calcul), LAC (Logique, Algèbre et Calcul) et ALGA (Automata, Logic, Games and Algebra) de celui-ci. Je suis également membre du GDR Topologie algébrique et applications.

5.2 Collaborations scientifiques

Je collabore actuellement avec :

- Ugo dal Lago (Université de Bologne),
- Gilles Barthe (IMDEA, Madrid),
- Marco Gaboardi (University at Buffalo, State University of New York).

6 Publications

6.1 Conférences d'audience internationale avec actes et comité de sélection

1. Ugo Dal Lago et Charles Grellois. *Probabilistic Termination by Monadic Affine Sized Typing*, European Symposium on Programming (ESOP) 2017.
2. Charles Grellois et Paul-André Melliès. *Relational semantics of linear logic and higher-order model-checking*. 24th EACSL Annual Conference on Computer Science Logic (CSL) 2015 <http://drops.dagstuhl.de/opus/volltexte/2015/5419/>, 2015.
3. Charles Grellois et Paul-André Melliès. *Finitary semantics of linear logic and higher-order model-checking*. Mathematical Foundations of Computer Science (MFCS) 2015 http://link.springer.com/chapter/10.1007%2F978-3-662-48057-1_20, 2015.
4. Charles Grellois et Paul-André Melliès. *An infinitary model of linear logic*. Foundations of Software Science and Computation Structures (FoSSaCS) 2015 http://link.springer.com/chapter/10.1007%2F978-3-662-46678-0_3, 2015

6.2 Workshops d'audience internationale avec actes et comité de sélection

1. Charles Grellois et Paul-André Melliès. *Indexed linear logic and higher-order model checking*. Proceedings Seventh Workshop on Intersection Types and Related Systems, ITRS 2014, <https://arxiv.org/html/1503.04377v1>, 2014.

6.3 Thèses et mémoires

1. Charles Grellois. *Semantics of linear logic and higher-order model checking*. Thèse de doctorat, Université Paris 7, <http://research.grellois.fr/doc/these.pdf>, 2016.
2. Charles Grellois. *Interprétation catégorique des arbres rationnels*. Mémoire de Master, Université Paris 7, <http://research.grellois.fr/doc/grellois-interpretation-categorique-des-arbres-rationnels.pdf>, 2012.
3. Charles Grellois. *Algebraic theories, monads, and arities*. Mémoire de Master, Université Paris 6, <http://arxiv.org/abs/1110.3294>, 2011.
4. Charles Grellois. *Game semantics of higher-order recursion schemes establishes the decidability of MSO model-checking*. Mémoire de Master, Université Paris 7, <http://research.grellois.fr/doc/grellois-game-semantics-and-ho-model-checking.pdf>, 2010.

Références

- [Abe04] Andreas Abel. Termination checking with types. *ITA*, 38(4) :277–319, 2004.
- [Aeh06] Klaus Aehlig. A finite semantics of simply-typed lambda terms for infinite runs of automata. In Zoltán Ésik, editor, *CSL '06*, volume 4207 of *LNCS*, pages 104–118. Springer, 2006.
- [BCOS10] Christopher H. Broadbent, Arnaud Carayol, C.-H. Luke Ong, and Olivier Serre. Recursion schemes and logical reflection. In *LICS '10*, pages 120–129. IEEE Computer Society, 2010.
- [BE00] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics in multiplicative-additive linear logic. *Ann. Pure Appl. Logic*, 102(3) :247–282, 2000.
- [BE01] Antonio Bucciarelli and Thomas Ehrhard. On phase semantics and denotational semantics : the exponentials. *Ann. Pure Appl. Logic*, 109(3) :205–241, 2001.
- [BEM97] Ahmed Bouajjani, Javier Esparza, and Oded Maler. Reachability analysis of pushdown automata : Application to model-checking. In Antoni W. Mazurkiewicz and Józef Winkowski, editors, *CONCUR '97*, volume 1243 of *LNCS*, pages 135–150. Springer, 1997.
- [BFG⁺04] Gilles Barthe, Maria João Frade, Eduardo Giménez, Luis Pinto, and Tarmo Uustalu. Type-based termination of recursive definitions. *MSCS*, 14(1) :97–141, 2004.
- [BGB09] Gilles Barthe, Benjamin Grégoire, and Santiago Zanella Béguelin. Formal certification of code-based cryptographic proofs. In Shao and Pierce [SP09], pages 90–101.
- [BGHB11] Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella Béguelin. Computer-aided security proofs for the working cryptographer. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 71–90. Springer, 2011.
- [BK02] Olivier Bournez and Claude Kirchner. Probabilistic rewrite strategies. applications to ELAN. In Sophie Tison, editor, *RTA 2002*, volume 2378 of *LNCS*, pages 252–266. Springer, 2002.
- [CE81] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Dexter Kozen, editor, *Logics of Programs, Workshop*, volume 131 of *LNCS*, pages 52–71. Springer, 1981.
- [CFG16] Krishnendu Chatterjee, Hongfei Fu, and Amir Kafshdar Goharshady. Termination analysis of probabilistic programs through positivstellensatz’s. In Swarat Chaudhuri and Azadeh Farzan, editors, *CAV 2016*, volume 9779 of *LNCS*, pages 3–22. Springer, 2016.
- [CFNH16] Krishnendu Chatterjee, Hongfei Fu, Petr Novotný, and Rouzbeh Hasheminezhad. Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs. In Rastislav Bodík and Rupak Majumdar, editors, *POPL '16*, pages 327–342. ACM, 2016.

- [CS12] Arnaud Carayol and Olivier Serre. Collapsible pushdown automata and labeled recursion schemes : Equivalence, safety and effective selection. In *LICS '12*, pages 165–174. IEEE Computer Society, 2012.
- [Dam77] Werner Damm. Higher type program schemes and their tree languages. In Hans Tzschach, H. Waldschmidt, and Hermann K.-G. Walter, editors, *Theoretical Computer Science, 3rd GI-Conference*, volume 48 of *LNCS*, pages 51–72. Springer, 1977.
- [dC09] Daniel de Carvalho. Execution time of lambda-terms via denotational semantics and intersection types. *CoRR*, abs/0905.4251, 2009.
- [DG17] Ugo Dal Lago and Charles Grellois. Probabilistic termination by monadic affine sized typing. In *ESOP 2017*, 2017.
- [dLMSS56] Karel de Leeuw, Edward F. Moore, Claude E. Shannon, and Norman Shapiro. Computability by probabilistic machines. *Automata Studies*, 34 :183–212, 1956.
- [EGK12] Javier Esparza, Andreas Gaiser, and Stefan Kiefer. Proving termination of probabilistic programs using patterns. In P. Madhusudan and Sanjit A. Seshia, editors, *CAV 2012*, volume 7358 of *LNCS*, pages 123–138. Springer, 2012.
- [Ehr12] Thomas Ehrhard. The scott model of linear logic is the extensional collapse of its relational model. *Theor. Comput. Sci.*, 424 :20–45, 2012.
- [FH15] Luis María Ferrer Fioriti and Holger Hermanns. Probabilistic termination : Soundness, completeness, and compositionality. In Sri-ram K. Rajamani and David Walker, editors, *POPL 2015*, pages 489–501. ACM, 2015.
- [Gir87] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50 :1–102, 1987.
- [GM15a] Charles Grellois and Paul-André Melliès. Finitary semantics of linear logic and higher-order model-checking. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *MFCS '15*, volume 9234 of *LNCS*, pages 256–268. Springer, 2015.
- [GM15b] Charles Grellois and Paul-André Melliès. Indexed linear logic and higher-order model checking. In Jakob Rehof, editor, *ITRS '14*, volume 177 of *EPTCS*, pages 43–52, 2015.
- [GM15c] Charles Grellois and Paul-André Melliès. An infinitary model of linear logic. In Andrew M. Pitts, editor, *FoSSaCS '15*, volume 9034 of *LNCS*, pages 41–55. Springer, 2015.
- [GM15d] Charles Grellois and Paul-André Melliès. Relational semantics of linear logic and higher-order model checking. In Kreutzer [Kre15], pages 260–276.
- [GMR⁺08] Noah D. Goodman, Vikash K. Mansinghka, Daniel M. Roy, Keith Bonawitz, and Joshua B. Tenenbaum. Church : a language for generative models. In David A. McAllester and Petri Myllymäki, editors, *UAI 2008*, pages 220–229. AUAI Press, 2008.

- [Gre14] Charles Grellois. On the coinductive nature of centralizers. Journées Montoises 2014, <https://jm2014.sciencesconf.org/44345>, September 2014.
- [Gre15] Charles Grellois. On the coinductive nature of centralizers. Seminar at the University of Turku <http://research.grellois.fr/doc/grellois-turku2015.pdf>, January 2015.
- [Gre16] Charles Grellois. *Semantics of linear logic and higher-order model-checking*. PhD thesis, Paris Diderot University, France, 2016.
- [HMOS08] Matthew Hague, Andrzej S. Murawski, C.-H. Luke Ong, and Olivier Serre. Collapsible pushdown automata and recursion schemes. In *LICS '08*, pages 452–461. IEEE Computer Society, 2008.
- [HPS96] John Hughes, Lars Pareto, and Amr Sabry. Proving the correctness of reactive systems using sized types. In Hans-Juergen Boehm and Guy L. Steele Jr., editors, *POPL'96*, pages 410–423. ACM Press, 1996.
- [KO] Naoki Kobayashi and C.-H. Luke Ong. A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes (journal version, in preparation).
- [KO09] Naoki Kobayashi and C.-H. Luke Ong. A type system equivalent to the modal mu-calculus model checking of higher-order recursion schemes. In *LICS '09*, 2009.
- [Kob09] Naoki Kobayashi. Types and higher-order recursion schemes for verification of higher-order programs. In Shao and Pierce [SP09], pages 416–428.
- [Kre15] Stephan Kreutzer, editor. *CSL '15*, volume 41 of *LIPICs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
- [Kun05] Michal Kunc. The power of commuting with finite sets of words. In Volker Diekert and Bruno Durand, editors, *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, volume 3404 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2005.
- [Mel14] Paul-André Melliès. The parametric continuation monad. *Math. Struct. Comp. Sci.*, 2014.
- [MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [MS85] David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theor. Comput. Sci.*, 37 :51–75, 1985.
- [MS01] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 2001.
- [Niv72] M. Nivat. On the interpretation of recursive program schemes. In *Symposia Matematica*, 1972.
- [NW09] Turlough Neary and Damien Woods. Four small universal turing machines. *Fundam. Inform.*, 91(1) :123–144, 2009.
- [Ong06] C.-H. Luke Ong. On model-checking trees generated by higher-order recursion schemes. In *LICS '06*, pages 81–90. IEEE Computer Society, 2006.

- [Pea89] Judea Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. 1989.
- [QS82] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In Mariangiola Dezani-Ciancaglini and Ugo Montanari, editors, *International Symposium on Programming '82*, volume 137 of *LNCS*, pages 337–351. Springer, 1982.
- [Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141 :pp. 1–35, 1969.
- [Sal09] Sylvain Salvati. Recognizability in the simply typed lambda-calculus. In Hiroakira Ono, Makoto Kanazawa, and Ruy J. G. B. de Queiroz, editors, *WoLLIC '09*, volume 5514 of *LNCS*, pages 48–60. Springer, 2009.
- [SP09] Zhong Shao and Benjamin C. Pierce, editors. *POPL 2009*. ACM, 2009.
- [SW14] Sylvain Salvati and Igor Walukiewicz. Krivine machines and higher-order schemes. *Inf. Comput.*, 239 :340–355, 2014.
- [SW15] Sylvain Salvati and Igor Walukiewicz. A model for behavioural properties of higher-order programs. In Kreutzer [Kre15], pages 229–243.
- [Thr02] Sebastian Thrun. Robotic mapping : A survey. In *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
- [Wal96] Igor Walukiewicz. Pushdown processes : Games and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *CAV '96*, volume 1102 of *LNCS*, pages 62–74. Springer, 1996.