Axel Kerinec

2 bis rue Fénelon
44000 Nantes, France
T +33 6 23 02 39 62
E axel.kerinec@yahoo.com
W https://axelkrnc.github.io/

Education

Master, *Ecole Normale Superieure de Lyon*, *Computer Science*. **2017-2018**
Automata, Coinduction, and Relational Algebra; Monadic Second Order Logic, Automata, Expressivity and Decidability; Implicit Computational Complexity; Models of Concurrency, Categories, and Games; Complex Networks; Lower Bound Methods; Graph Decompositions: From Tree-Width to Perfect Graphs

Master, *Ecole Normale Superieure de Lyon*, *Computer Science*. **2016-2017**
Parallel and Distributed Algorithms; Information Theory; Optimisation and Approximation; Performance Evaluation of Networks; Semantics and Verification; Programs and Proofs; Data Bases and Data Mining; Computational Complexity; Machine Learning

License, *Ecole Normale Superieure de Lyon*, *Computer Science*. **2015-2016**
Language theory; Turing Machines and Automatons; Mathematical Logic; Probability; Computer and Network Architecture; Algorithm Design/Complexity/Implementation

Preparatory School, *Centre International de Valbonne (Two years intensive courses preparing the competitive entrance exam to French 'Grandes Ecoles')*. **2013-2015**
Mathematic, Computer Science, Physic

Scientific Baccalaureat, *(French secondary school diploma)*, *Honors*. **2013**
Mathematic, Physic, Engineering

Research Experience

Post-Doctorate

Semantic Capture of Time and Space Costs for the Evaluation of Higher-Order Programming Languages, **09/2023-Present**
Inria and Laboratoire des Sciences du Numérique de Nantes,
A project led by Beniamino Accattoli (Inria Saclay) and Guilhem Jaber (Nantes Université).

The Operational Game Semantics (OGS), introduced in [Laird, 2007], interprets λ-calculus terms as states evolving through labeled transition systems (LTS), representing interactions with the environment. Unlike standard denotational game semantics, composition is not a primitive notion. Most OGS models consider a notion of closed composition between two states, where only actions with no interaction with the outside world can occur.
However, to build a compositional model and make it adequate, a notion of open composition is required. In this case, the merging of a pair of states no longer results in a closed term but another state that both performs internal synchronization actions and can interact with the environment through visible actions.
Our work aims to define such a composition and use it to introduce relevant categorical structures to model the λ-calculus in Call-by-Value.

PhD

A History of λ-Calculus and Approximation, **09/2019-06/2023**
Sorbonne Paris Nord University,
Supervisor: Giulio Manzonetto
Reviewers: Silvia Ghilezan (Novi Sad University), Tom Hirschowitz (CNRS, Savoie Mont Blanc University)
Committee: Pierre Clairambault (CNRS, Aix-Marseille University), Delia Kesner (Paris Cité University), Stefano Guerrini (Sorbonne Paris Nord University), Marie Kerjean (CNRS,Sorbonne Paris Nord University),
Manuscript: https://theses.hal.science/tel-04624826.
Slides: https://axelkrnc.github.io/includes/PhD.pdf.

This work combines type theory, linear logic, and categorical approaches to advance the understanding of λ-calculus models.
Böhm trees, introduced in [Barendregt, 1977], are historically the first notion of approximation for the λ-calculus. They are introduced in the Call-by-Name framework and have a strong connection with another notion of approximation: Taylor expansion [Ehrhard and Regnier, 2003]. The theory of program approximation is far less developed for the Call-by-Value λ-calculus [Plotkin, 1975], although this reduction strategy is closer to modern programming languages. We present the first notion of Böhm trees in this context and prove a connection between our Böhm trees and Taylor expansion similar to the Call-by-Name case.
Our Böhm trees allow, among other things, the characterization of observational equivalence and solvability, two essential notions that remain poorly understood in the Call-by-Value framework.
In the second part of this work, we focus on the Call-by-Name λ-calculus. Following the line of [Olimpieri, 2020, Olimpieri, 2021], we introduce categorized graph models in a bicategorical semantics based on distributors. They can be seen as a categorification of traditional relational models and, similarly, can be presented as intersection type systems.
In this framework, we prove an approximation theorem: the interpretation of a λ-term corresponds to the interpretation of its Böhm tree.
Unlike relational models, our models are also proof-dependent, meaning that the interpretation of a λ-term in our models contains not only its typings but also the set of type derivations. Thanks to the additional information carried by type derivations, we derive the characterization of the induced theory as a simple corollary of the approximation theorem: two λ-terms have isomorphic interpretations exactly when their Böhm trees coincide.

Internships

Conservativity in the Algebraic λ-Calculus, **03/2019-08/2019**
Institut de Mathématiques de Marseille,
Supervised by Lionel Vaux Auclair.

The algebraic λ-calculus was introduced as a general framework for studying the rewriting theory of λ-terms in the presence of weighted superpositions [Vaux, 2007, Vaux, 2009]. A long-standing result is that of conservativity: two classical λ-terms are equivalent in the algebraic framework if and only if they are β-equal. However, previous proofs of this result were flawed. During this internship, we studied why and developed a new proof using an original technique.

Probabilistic λ-Calculus and Monte Carlo Methods, **10/2018-03/2019**
University of Bologna,
Supervised by Ugo Dal Lago.

We focused on the probabilistic λ-calculus. In particular, we considered the typed functional programming language "Programming Computable Functions," which can be seen as an extended version of the λ-calculus [Plotkin, 1977] and modified it to become probabilistic.
In this context, observing the possible results of the reduction of a λ-term yields a distribution rather than a single λ-term. These distributions cannot be computed exactly, necessitating inference algorithms to approximate them. We explored the Sequential Monte Carlo algorithm [Del Moral, 1997, Del Moral et al., 2006] as well as a specific case of Markov Chain Monte Carlo: Metropolis-Hastings [Keith Hastings, 1970, Metropolis et al., 1953] and the Particle Markov Chain Monte Carlo method [Christophe Andrieu, 2010]. The first two methods are classical and have been used for studying probabilistic programming languages, while the latter is a more original combination of the previous two.

Approximation of λ-Terms in Call-by-Value, **02/2018-06/2018**
IRIF, Université Paris Diderot,
Supervised by Michele Pagani and Giulio Manzonetto.

The Call-by-Value λ-calculus [Plotkin, 1975] has a much less developed theory than the Call-by-Name λ-calculus. Taylor expansion is a classical approximation method known in both frameworks [Ehrhard and Regnier, 2003] but has been much more studied in the Call-by-Name setting, where it notably has a strong link with Böhm trees [Barendregt, 1977], which do not exist in the Call-by-Value setting. During this internship, we worked on developing and studying similar tools in Call-by-Value.

Effectiveness of Multi-Task Learning for Neural Networks, **05/2017-09/2017**
University of Copenhagen,
Supervised by Anders Søgaard.

This internship explored the benefits of multi-task learning in the context of deep neural networks compared to traditional single-task learning.
Previous studies empirically investigated when multi-task learning was beneficial [Bingel and Søgaard, 2017, Alonso and Plank, 2016]. However, we were the first to study this question in the context of text classification, an essential topic.

Detection of Spatio-Temporal Patterns in Multivariate Signals, **06/2016-08/2016**
LabSTICC, École Telecom Bretagne,
Supervised by Vincent Gripon.

Detecting recurring patterns in images or videos is a major challenge. By considering images as graphs, we can compute their pseudo-Fourier transforms. The question during this internship was whether using this transform on the data improved detection with the Support Vector Machines learning method.

Teaching Experience

My teaching activities were carried out as part of a doctoral teaching assignment during the first three years of my PhD, followed by part-time teaching during my post-doctorate.
My main responsibilities included leading practical sessions and grading exams. However, in 2021-2022, I supervised groups of students in their first-year of BUT for SAÉ projects (Learning and Evaluation Situations). These are practical projects spanning several weeks in small groups: students are given a realistic problem, which they must analyze, identify the needs, plan an appropriate solution, and implement it. That same year, I was also a jury member for the internship defenses of second-year DUT students. Currently, in 2024-2025, I have been responsible for certain lectures in logic for computer science in the second year of a bachelor's program: I had the opportunity to design a chapter of the course, create the associated presentation slides, and prepare the exercises for the practical sessions.

During the 2019-2020 and 2020-2021 academic years, some of the teaching was conducted remotely due to the COVID-19 pandemic.

Computer Science Department Nantes University. **2024-2025**
- Logic for computer scientists,
  the presentation I prepared for the course: https://axelkrnc.github.io/includes/cours_logique.pdf.

Computer Science Department Nantes University. **2023-2024**
- Mathematics for computer scientists.

Computer Science Department BUT Villetaneuse. **2021-2022**
- Supervision of SAÉ Projects.
- Jury for Internship Defenses.
- Introduction to Operating Systems.
- Advance Databases.

Computer Science Department DUT Villetaneuse. **2020-2021**
- Architecture and Programming.
- Introduction to Algorithm and Programming.
- Design of Digital Documents and Interfaces.

Computer Science Department DUT Villetaneuse. **2019-2020**
- Data Communication Network.
- Data Structures and Fundamental Algorithms.
- Architecture and Programming.

Author, H&K publisher. **05/2018**
Redaction of a corrected version of 'grandes ecoles' entrance exam for H&K publisher.

Publications

Abbreviations

HOR : International Workshop on Higher-Order Rewriting

POPL : Symposium on Principles of Programming Languages

FSCD : International Conference on Formal Structures for Computation and Deduction

LMCS : Logical Methods in Computer Science

Why Are Proofs Relevant in Proof-Relevant Models? [Kerinec et al., 2023], **POPL 2023**
Authors : A. Kerinec, G. Manzonetto et F. Olimpieri,
Link : https://dl.acm.org/doi/pdf/10.1145/3571201.

Call-By-Value, Again! [Kerinec et al., 2021], **FSCD 2021**
Authors : A. Kerinec, G. Manzonetto et S. Ronchi Della Rocca,
Link : https://drops.dagstuhl.de/storage/00lipics/lipics-vol195-fscd2021/LIPIcs.FSCD.2021.7/LIPIcs.FSCD.2021.7.pdf.

Revisiting Call-by-value Bohm trees in light of their Taylor expansion [Kerinec et al., 2020], **LMCS 2020**
Authors : A. Kerinec, G. Manzonetto et M. Pagani,
Link : https://arxiv.org/abs/1809.02659.

Workshops

The algebraic λ-calculus is a conservative extension of the ordinary λ-calculus [Kerinec and Vaux Auclair, 2023], **HOR 2023**
Authors : A. Kerinec et L. Vaux Auclair,
Link : https://hal.science/hal-04759250.

When does deep multi-task learning work for loosely related document classification tasks? [Kerinec et al., 2018], **BlackboxNLP 2018**
Authors : A. Kerinec, C. Braud et A. Søgaard,
Link : https://hal.science/hal-02374086v1.

Presentations

Conferences

FSCD 2021, **07/20/2021**
Call-By-Value, Again!.
Slides: https://axelkrnc.github.io/includes/fscd.pdf
Video: https://www.youtube.com/watch?v=K06e-xIcH-c&t=356s

Team Seminars

Team Logique et Intéractions (I2M Aix-Marseille University), **08/09/2022**
Bicategorical Models.
Slides: https://axelkrnc.github.io/includes/i2m.pdf

Team LoVe (LIPN Sorbonne Paris Nord University), **30/09/2021**
Call-By-Value, Again!.
Slides: https://axelkrnc.github.io/includes/lipn.pdf

Other

LHC (Logic, Homotopy, Categories), **06/05/2024**
Towards Categorical Structures for Operational Game Semantics.
Slides: https://axelkrnc.github.io/includes/LHC.pdf

“CHoCoLa” Meeting (Curry-Howard : Calcul et Logique), **05/11/2023**
Why Are Proofs Relevant in Proof-Relevant Models?.
Slides: https://axelkrnc.github.io/includes/chocola.pdf

Responsabilities

I helped organize the conference Microservices 2022, *https://www.conf-micro.services/2022/*. **10 to 12 may 2022**

IT Skills

Programmation : OCaml, C, C++, Python
Web : HTML5, CSS3, SQL
Autres : Git, Unity, Latex

Languages

French: Native
English: Fluent (CAE level C1)
German: Intermediate (a few years ago level C2)

Bibliography

[Alonso and Plank, 2016] Alonso, H. M. and Plank, B. (2016). Multitask learning for semantic sequence prediction under varying data conditions. *CoRR*, abs/1612.02251.

[Barendregt, 1977] Barendregt, H. P. (1977). The type free lambda calculus. In Barwise, J., editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, pages 1091–1132. North-Holland, Amsterdam.

[Bingel and Søgaard, 2017] Bingel, J. and Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks.

[Christophe Andrieu, 2010] Christophe Andrieu, A. D. (2010). Particle markov chain monte carlo methods. In *journal of the royal statistical society*.

[Del Moral, 1997] Del Moral, P. (1997). Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*.

[Del Moral et al., 2006] Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436.

[Ehrhard and Regnier, 2003] Ehrhard, T. and Regnier, L. (2003). The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3):1–41.

[Keith Hastings, 1970] Keith Hastings, W. (1970). Monte carlo sampling methods using markov chains and their application. *Biometrika*.

[Kerinec et al., 2023] Kerinec, A., Manzonetto, G., and Olimpieri, F. (2023). Why are proofs relevant in proof-relevant models? *Proc. ACM Program. Lang.*, 7(POPL):218–248.

[Kerinec et al., 2020] Kerinec, A., Manzonetto, G., and Pagani, M. (2020). Revisiting call-by-value Böhm trees in light of their Taylor expansion. *Log. Methods Comput. Sci.*, 16(3).

[Kerinec et al., 2021] Kerinec, A., Manzonetto, G., and Ronchi Della Rocca, S. (2021). Call-by-value, again! In Kobayashi, N., editor, *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021, July 17-24, 2021, Buenos Aires, Argentina (Virtual Conference)*, volume 195 of *LIPIcs*, pages 7:1–7:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[Kerinec and Vaux Auclair, 2023] Kerinec, A. and Vaux Auclair, L. (2023). The algebraic λ-calculus is a conservative extension of the ordinary λ-calculus. *arXiv e-prints*, page arXiv:2305.01067.

[Kerinec et al., 2018] Kerinec, E., Søgaard, A., and Braud, C. (2018). When does deep multi-task learning work for loosely related document classification tasks? In *Proceedings of the 2018 EMNLP*

Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 1 – 8, Brussels, Belgium. Association for Computational Linguistics.

[Laird, 2007] Laird, J. (2007). A fully abstract trace semantics for general references. In Arge, L., Cachin, C., Jurdzinski, T., and Tarlecki, A., editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 667–679. Springer.

[Metropolis et al., 1953] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*.

[Olimpieri, 2020] Olimpieri, F. (2020). Intersection type distributors. *CoRR*, abs/2002.01287.

[Olimpieri, 2021] Olimpieri, F. (2021). Intersection type distributors. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–15. IEEE.

[Plotkin, 1977] Plotkin, G. (1977). Lcf considered as a programming language. *Theoretical Computer Science*.

[Plotkin, 1975] Plotkin, G. D. (1975). Call-by-name, call-by-value and the lambda-calculus. *Theor. Comput. Sci.*, 1(2):125–159.

[Vaux, 2007] Vaux, L. (2007). On linear combinations of λ-terms. In Baader, F., editor, *Term Rewriting and Applications*, pages 374–388, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Vaux, 2009] Vaux, L. (2009). The algebraic lambda-calculus. *Mathematical Structures in Computer Science*, page accepted for publication. 29 pages.