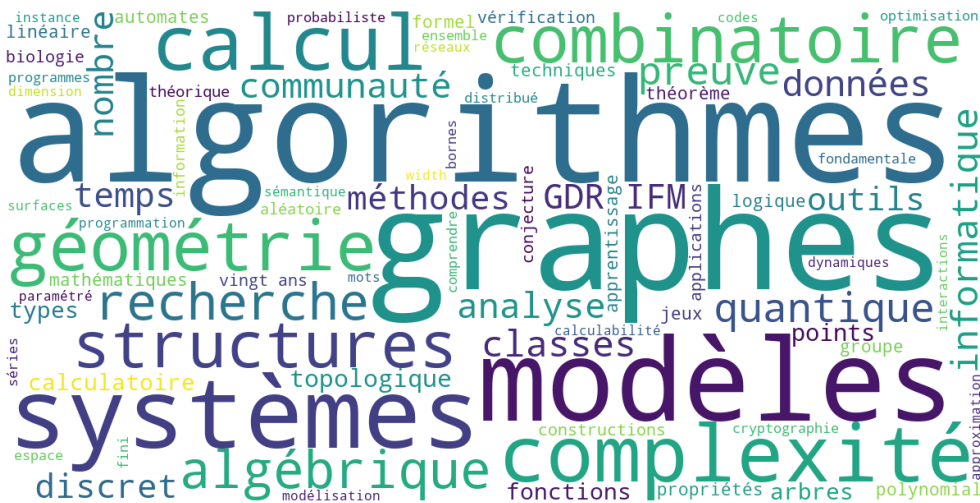




Les vingt ans du GdR IFM



Programmes, vérification, preuve, automates et logique

GT Daal (Data, Automata, Algebra, & Logic)	3
GT Vérif (Vérification)	13
GT Scalp (Structures formelles pour le calcul et les preuves)	23
GT LHC (Logic, Homotopy, Categories)	35
GT Bioss (Biologie systémique symbolique)	41



Twenty Years of GdR IFM, seen from GT Data, Automata, Algebra & Logic

The Working Group on Data, Automata, Algebra & Logic (GT DAAL) of the GDR Fundamental Computer Science and its Mathematics brings together researchers working in automata theory, logic, games, algebra, and topology applied to fundamental computer science, as well as in database theory. It continues the activities of GT ALGA (2015–2021) and GT Jeux (2008–2015), while expanding their scientific scope, in particular through the explicit integration of data-related issues. The interplay between automata, logic, games, and algebra has long been central to many advances in theoretical computer science, driven by major motivations and applications in formal verification, system synthesis, performance analysis, security, and data processing. Over the past twenty years, these areas have undergone several major paradigm shifts, prompted by the increasing complexity of models, the consideration of structured and unstructured data, and the need to reason about quantitative, timed, probabilistic, or AI-driven systems.

1 Algorithmic game theory

One of the major breakthroughs of the last two decades is the development of more efficient algorithms for solving parity games, which play a central role in the model checking of the μ -calculus and MSO logic. In these graph-based games, two players interact indefinitely, and one player aims to ensure that the smallest color occurring infinitely often is even. In 2017, Calude et al. [28] introduced a quasi-polynomial-time algorithm, breaking a long-standing sequence of exponential-time approaches. This result triggered an intense period of research, leading to the introduction of new analytical structures—notably succinct progress measures [66]—and to the use of universal trees to derive lower bounds [37], later generalized to universal graphs [36]. The existence of a polynomial-time algorithm for parity games remains open and constitutes one of the central challenges of algorithmic game theory, an area that is actively investigated within GT DAAL.

Beyond these algorithmic questions, graph games have witnessed renewed interest over the past twenty years around the notion of history-determinism. This property of automata allows for nondeterminism as long as it does not compromise the subsequent use of two-player games for system verification. Introduced by Henzinger and Piterman [63], this notion has been refined and generalized in numerous works, with applications to the synthesis of reactive systems [70, 60, 29], and continues to generate substantial research activity.

Another concept that has attracted considerable attention over the last two decades concerns the amount of memory required in winning strategies. In 2005, Gimbert and Zielonka [59] characterized winning conditions for which, whenever a player can win, memory-less strategies suffice. More recently, these results have been extended to characterize winning conditions under which strategies only require finite chromatic memory [22]. The complexity of computing the minimal required memory for a given game has also been investigated and shown to lie in NP in [30].

The connections between games and the synthesis of reactive systems have been further strengthened through the study of multi-player graph games, where the objective is no longer to compute a winning strategy for a single player, but to identify equilibria, such as Nash equilibria [77]. From a synthesis perspective, this has led to the study of subgame-perfect equilibria that are more realistic in practice [26], as well as to a detailed analysis of imperfect information in this setting [15].

Finally, distributed games have also seen significant progress, both through decidability results for acyclic communication architectures [56, 76], and through a general undecidability result [57], which resolved a question that had remained open for over two decades. At the same time, decidability results for distributed control under lock-based synchronization [58] have shown that it is possible to identify models where distributed control becomes decidable using local invariants, thereby opening new research directions.

2 Transducers and transformation theory

Transductions are functions or relations between words that were first introduced and studied in the 1970s. While transducers played a particularly prominent role in linguistics until the 1990s, more recent developments have been strongly motivated by programming language formalisms. As a result, the past twenty years have witnessed a genuine revival of transducer theory, driven by the emergence of new, more expressive models capable of representing richer classes of programs (see, for instance, the surveys [53, 75]).

A wide variety of new transducer models has been proposed. Some provide alternatives to existing formalisms, such as word transducers with registers introduced by Alur [2]. Others, by contrast, define new classes of transformations, such as token transducers, which characterize polyregular functions [17]. This diversification has prompted substantial work on the expressive power of these models [48, 20, 51, 35]. In parallel, more declarative formalisms for specifying word-to-word functions have been studied [39]. Transducers equipped with richer memory structures, organized as visible stacks to model XML document transformations, have also been investigated [52]. Considerable effort has been devoted to simplifying transduction models, in the spirit of automata minimization, for example by minimizing memory usage in register-based models [40], or by restricting or allowing bidirectional access to the input [49, 11]. A connection with synthesis has also emerged: given a specification defined as an input–output relation, the goal is to construct a machine that realizes a function with the same domain and whose image is included in that of the relation [50].

3 Extensions of models with quantities or over richer structures

Over the past twenty years, there has been a significant expansion of quantitative automata and logics, as well as of their extension to structures richer than words, typically trees (or nested words [3], introduced by Alur and Madhusudan to represent trees as enriched words and which have since generated a long line of fruitful research) or graphs.

The work of Droste, Kuich, Vogler, Gastin, Colcombet *et al.* on weighted automata and their associated logics [45, 34] has led to major advances in the understanding of these models. This was complemented by more algorithmic work on infinite words, notably through the notion of quantitative languages introduced by Chatterjee, Doyen, and Henzinger [31]. Extensions to two-player games have given rise to fundamental algorithmic problems that have seen substantial progress over the past twenty years, in particular for mean-payoff and energy games [27].

Probabilistic models also play a central role, especially for modeling uncertainty in distributed or cyber-physical systems. The reference books by Baier and Katoen [8], together with the work of Kwiatkowska *et al.* [71], have provided a solid foundation for Markov chains, MDPs, and probabilistic logics, and have enabled the development of efficient algorithms for reasoning about these models, with applications to the verification of the

aforementioned systems. The community has paid particular attention to providing guarantees for approximation algorithms—previously used without guarantees—for computing optimal values in probabilistic systems [61], even in learning settings where the probabilistic system is not fully known [25]. Significant progress has also been made on probabilistic automata, both over finite words [47] and infinite words [7], sometimes relying on new algebraic (profinite) techniques [46].

A persistent challenge is the development of unified algebraic foundations that can coherently account for weights, probabilities, and resources, a question that remains largely open.

4 Integration of structured data

Database theory is naturally shaped by real-world requirements. Over the past decades, alternative data models have emerged and gained prominence to address specific needs for which the traditional relational model is ill-suited, such as trees, labeled graphs, and more recently property graphs. The study of these models has led the research community both to revisit and adapt established techniques and to develop new ones.

Tree automata, closely connected to MSO and modal logics, have provided a foundation for studying tree-structured data models such as XML. However, understanding the interaction between structure and data has revealed a new need: manipulating objects labeled with values drawn from infinite domains. This has given rise to register automata [86] and nominal automata [18], which extend classical automata with mechanisms for storing and comparing data. Once again, the connection with logic has been central to characterizing the expressive power and limitations of these models [19]. Links between description languages such as XPath and tree automata have also been established [85].

Automata-based approaches, such as regular path queries (RPQs) [72], as well as Datalog and its extensions [88, 14], have proved effective for studying data graphs modeled as labeled graphs. However, the rise of property graphs (Cypher, GQL) [65, 42] has demonstrated that traditional approaches are insufficient to capture real-world expressiveness. This expressiveness manifests itself both in data schemas (multi-labeled multigraphs) [4] and in query languages [54], which may return paths of unbounded length and potentially infinite cardinality. These recent developments have pushed the community to revisit fundamental questions long considered settled [41].

5 Algebraic methods for formal languages

The relationship between formal languages and algebra has always been central to automata and language theory, as illustrated by the seminal results of Schützenberger, McNaughton, and Papert [82, 73], and later by Simon [84], for deciding subclasses of formal languages. The past twenty years have seen sustained activity in this area, particularly around the problem of language separability: given two languages, determining whether there exists a third language in a restricted class that contains the first language while being disjoint from the second. This problem, which is strictly harder than testing emptiness of the intersection but provides more refined information, has led to a series of results across several hierarchies of languages and logics [79, 78, 80]. The past two decades have also seen progress on the star height problem for regular languages, originally introduced by Hashiguchi [62], with simpler proofs emerging since [81, 68, 16]. Further results concern heights induced by the subword order, related to the hierarchy of piecewise testable languages [67].

Algebraic techniques have also been generalized to richer structures, making it possible, for instance, to obtain Krohn–Rhodes-style decompositions of languages by simple automata [69] in the distributed setting of Mazurkiewicz traces [1].

6 Logics for strategic reasoning, epistemic dynamics, and hyperproperties

Motivated in particular by issues related to imperfect information in strategic reasoning, control under imperfect information, and software security, substantial effort has been devoted to extending temporal logics. For strategic reasoning, the *Strategy Logic* of Chatterjee, Henzinger, and Piterman [32] is a canonical example of a first-order temporal logic in which quantification ranges over players’ strategies. Extending this framework to games with incomplete information has raised significant challenges, since the required semantics for quantifiers go beyond what is captured by classical Skolem functions [55, 12]. Other approaches have addressed this issue by introducing logical formalisms tailored to the analysis of systems in the presence of binary relations between their executions. It is well known that undecidability arises very quickly for such logics. Nevertheless, a variety of logics have been introduced and are still actively studied for which decidability and complexity results have been obtained—covering classical problems such as satisfiability and model checking—as well as connections with enriched automata models: hyperlogics where quantification ranges over executions rather than strategies [33, 24], restrictions to recognizable relations between executions [23, 43], and specific cases relevant to epistemic planning [44, 21].

7 Emerging links with artificial intelligence

Beyond these logical aspects related to “deterministic” artificial intelligence, a new research area has emerged over the past decade at the intersection of “statistical” AI and formal methods.

Formal models such as automata and logics can be used to extract explanations from AI systems, for example neural networks [38]. Other approaches to learning formal models, more firmly grounded in language theory, have also been developed. Some build on Angluin’s L^* algorithm [5], whose many extensions have been studied over the past twenty years, including for infinite words [6], transductions [83], quantitative automata [10], and nominal automata over infinite alphabets [74]. Others apply theoretical tools from artificial intelligence in new settings, such as spectral analysis for quantitative automata [9], or the use of GPUs to learn logical formulas [87]. In particular, the results mentioned above on probabilistic automata have enabled new breakthroughs in the study of partially observable Markov decision processes [13], which received the Best Paper Award at AAAI 2025, the largest conference in Artificial Intelligence.

Finally, research has also addressed the security and robustness of neural networks, typically by leveraging logical tools to guide the analysis [64].

This emerging and still rapidly evolving area now shapes part of the research challenges of the community.

8 Complexity of Petri nets

Petri nets are a fundamental model for the modeling and analysis of concurrent systems: their study is shared with the GT VERIF where we give a detailed overview of the last 20

years on the subject. The study of the complexity of problems on Petri nets was particularly intense in the last few years, both in GT VERIF and in GT DAAL.

Contributors.

Benjamin Monmege and Marie Van Den Bogaard, with the help of Patricia Bouyer-Decitre, Nathanaël Fijalkow, Nadime Francis, Anca Muscholl, Sophie Pinchinat, Jean-Marc Talbot.

References

- 1 Bharat Adsul, Paul Gastin, Saptarshi Sarkar, and Pascal Weil. Asynchronous wreath product and cascade decompositions for concurrent behaviours. *Log. Methods Comput. Sci.*, 18(2), 2022. URL: [https://doi.org/10.46298/lmcs-18\(2:22\)2022](https://doi.org/10.46298/lmcs-18(2:22)2022), doi:10.46298/LMCS-18(2:22)2022.
- 2 Rajeev Alur and Pavol Černý. Expressiveness of streaming string transducers. In *Proceedings of the 30th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010, Chennai, Tamil Nadu, India, December 15-18, 2010*, volume 8 of *LIPICs*, pages 1–12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.
- 3 Rajeev Alur and P. Madhusudan. Adding nesting structure to words. *Journal of the ACM*, 56(3):16:1–16:43, 2009. doi:10.1145/1516512.1516518.
- 4 Renzo Angles, Angela Bonifati, Stefania Dumbrava, George Fletcher, Alastair Green, Jan Hidders, Bei Li, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Stefan Plantikow, Ognjen Savković, Michael Schmidt, Juan Sequeda, Sławek Staworko, Dominik Tomaszuk, Hannes Voigt, Domagoj Vrgoč, Mingxi Wu, and Dušan Živković. Pg-schema: Schemas for property graphs. In *SIGMOD'23*. ACM, 6 2023.
- 5 Dana Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.
- 6 Dana Angluin and Dana Fisman. Learning regular Omega languages. In *Proceedings of the 25th International Conference on Algorithmic Learning Theory (ALT'14)*, LNCS, pages 125–139. Springer, 2014.
- 7 Christel Baier, Nathalie Bertrand, and Marcus Größer. On decision problems for probabilistic büchi automata. In Roberto Amadio, editor, *Foundations of Software Science and Computational Structures*, pages 287–301, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- 8 Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
- 9 Borja Balle, Xavier Carreras, Franco M. Luque, and Ariadna Quattoni. Spectral learning of weighted automata - A forward-backward perspective. *Mach. Learn.*, 96(1-2):33–63, 2014. URL: <https://doi.org/10.1007/s10994-013-5416-x>, doi:10.1007/S10994-013-5416-X.
- 10 Borja Balle and Mehryar Mohri. Learning weighted automata. In Andreas Maletti, editor, *Algebraic Informatics*, pages 1–21, Cham, 2015. Springer International Publishing.
- 11 Félix Baschenis, Olivier Gauwin, Anca Muscholl, and Gabriele Puppis. One-way definability of two-way word transducers. *Log. Methods Comput. Sci.*, 14(4), 2018. URL: [https://doi.org/10.23638/LMCS-14\(4:22\)2018](https://doi.org/10.23638/LMCS-14(4:22)2018), doi:10.23638/LMCS-14(4:22)2018.
- 12 Dylan Bellier, Massimo Benerecetti, Fabio Mogavero, and Sophie Pinchinat. Plan logic. In Siddharth Barman and Slawomir Lasota, editors, *Proceedings of the 44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2024, Gandhinagar, Gujarat, India, December 16-18, 2024*, volume 323 of *LIPICs*, pages 9:1–9:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.FSTTCS.2024.9.
- 13 Marius Belly, Nathanaël Fijalkow, Hugo Gimbert, Florian Horn, Guillermo A. Pérez, and Pierre Vandenholte. Revelations: A decidable class of pomdps with omega-regular objectives. In *AAAI Conference on Artificial Intelligence, AAAI (outstanding paper award)*, 2025. URL: <https://arxiv.org/abs/2412.12063>.

- 14 Michael Benedikt, Pierre Bourhis, Georg Gottlob, and Pierre Senellart. Monadic datalog, tree validity, and limited access containment. *ACM Trans. Comput. Logic*, 21(1), October 2019. URL: <https://doi.org/10.1145/3344514>, doi:10.1145/3344514.
- 15 Dietmar Berwanger, Anup Basil Mathew, and Marie van den Bogaard. Hierarchical information and the synthesis of distributed strategies. *Acta Informatica*, 55(8):669–701, 2018. URL: <https://doi.org/10.1007/s00236-017-0306-5>, doi:10.1007/s00236-017-0306-5.
- 16 Mikolaj Bojanczyk. Star height via games. In *2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 214–219, 2015. doi:10.1109/LICS.2015.29.
- 17 Mikolaj Bojanczyk. Polyregular functions. *CoRR*, abs/1810.08760, 2018. URL: <http://arxiv.org/abs/1810.08760>, arXiv:1810.08760.
- 18 Mikołaj Bojańczyk, Laurent Braud, Bartek Klin, and Sławomir Lasota. Towards nominal computation. In *Proceedings of POPL'12*. ACM, 2012.
- 19 Mikolaj Bojanczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27:1–27:26, 2011. URL: <https://doi.org/10.1145/1970398.1970403>, doi:10.1145/1970398.1970403.
- 20 Mikolaj Bojanczyk, Sandra Kiefer, and Nathan Lhote. String-to-string interpretations with polynomial-size output. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 106:1–106:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: <https://doi.org/10.4230/LIPICs.ICALP.2019.106>, doi:10.4230/LIPICs.ICALP.2019.106.
- 21 Thomas Bolander, Tristan Charrier, Sophie Pinchinat, and François Schwarzentruber. Del-based epistemic planning: Decidability and complexity. *Artif. Intell.*, 287:103304, 2020. doi:10.1016/j.artint.2020.103304.
- 22 Patricia Bouyer, Mickael Randour, and Pierre Vandenhove. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *TheoretCS*, Volume 2, Jan 2023. URL: <https://theoretics.episciences.org/9608>, doi:10.46298/theoretics.23.1.
- 23 Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Uniform strategies, rational relations and jumping automata. *Inf. Comput.*, 242:80–107, 2015. doi:10.1016/j.ic.2015.03.012.
- 24 Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. Unifying hyper and epistemic temporal logics. In Andrew Pitts, editor, *Foundations of Software Science and Computation Structures*, pages 167–182, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.
- 25 Tomáš Brázdil, Krishnendu Chatterjee, Martin Chmelík, Vojtěch Forejt, Jan Křetínský, Marta Kwiatkowska, David Parker, and Mateusz Ujma. Verification of markov decision processes using learning algorithms. In *Automated Technology for Verification and Analysis*, pages 98–114. Springer, 2014.
- 26 Thomas Brihaye, Véronique Bruyère, Aline Goeminne, Jean-François Raskin, and Marie van den Bogaard. The complexity of subgame perfect equilibria in quantitative reachability games. *Log. Methods Comput. Sci.*, 16(4), 2020. URL: <https://lmcs.episciences.org/6883>.
- 27 Luboš Brim, Jakub Chaloupka, Laurent Doyen, Rafaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods for System Design*, 38(2):97–118, 2011.
- 28 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM Symposium on the Theory of Computing (STOC'17)*, pages 252–263. ACM Press, 2017.
- 29 Antonio Casares, Olivier Idir, Denis Kuperberg, Corto Mascle, and Aditya Prakash. On the Minimisation of Deterministic and History-Deterministic Generalised (Co)Büchi Automata. In *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*, volume 326 of

- LIPICs*, pages 22:1–22:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.CSL.2025.22.
- 30 Antonio Casares and Pierre Ohlmann. The Memory of ω -Regular and $\text{BC}(\Sigma_2^0)$ Objectives. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 149:1–149:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.ICALP.2025.149.
 - 31 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Transactions on Computational Logic*, 11(4), 2010.
 - 32 Krishnendu Chatterjee, Thomas A Henzinger, and Nir Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.
 - 33 M. R. Clarkson, Bernd Finkbeiner, M. Koleini, K. K. Micinski, Markus N. Rabe, and César Sánchez. Temporal logics for hyperproperties. In *POST 2014*, volume 8414 of *LNCS*, pages 265–284. Springer, 2014. doi:10.1007/978-3-642-54792-8_15.
 - 34 Thomas Colcombet. Regular cost functions, part I: logic and algebra over words. *Log. Methods Comput. Sci.*, 9(3), 2013. URL: [https://doi.org/10.2168/LMCS-9\(3:3\)2013](https://doi.org/10.2168/LMCS-9(3:3)2013), doi:10.2168/LMCS-9(3:3)2013.
 - 35 Thomas Colcombet, Gaëtan Douéneau-Tabot, and Aliaume Lopez. \mathbb{Z} -polyregular functions. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. URL: <https://doi.org/10.1109/LICS56636.2023.10175685>, doi:10.1109/LICS56636.2023.10175685.
 - 36 Thomas Colcombet, Nathanaël Fijalkow, Pawel Gawrychowski, and Pierre Ohlmann. The theory of universal graphs for infinite duration games. *Logical Methods in Computer Science*, 18(3), 2022. URL: [https://doi.org/10.46298/lmcs-18\(3:29\)2022](https://doi.org/10.46298/lmcs-18(3:29)2022), doi:10.46298/LMCS-18(3:29)2022.
 - 37 Wojciech Czerwinski, Laure Daviaud, Nathanaël Fijalkow, Marcin Jurdzinski, Ranko Lazic, and Pawel Parys. Universal trees grow inside separating automata: Quasi-polynomial lower bounds for parity games. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2333–2349. SIAM, 2019. URL: <https://doi.org/10.1137/1.9781611975482.142>, doi:10.1137/1.9781611975482.142.
 - 38 Adnan Darwiche. Logic for Explainable AI . In *38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–11. IEEE Computer Society, 2023. doi:10.1109/LICS56636.2023.10175757.
 - 39 Vrunda Dave, Paul Gastin, and Shankara Narayanan Krishna. Regular transducer expressions for regular transformations. *Inf. Comput.*, 282:104655, 2022. URL: <https://doi.org/10.1016/j.ic.2020.104655>, doi:10.1016/J.IC.2020.104655.
 - 40 Laure Daviaud, Pierre-Alain Reynier, and Jean-Marc Talbot. A generalised twinning property for minimisation of cost register automata. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 857–866. ACM, 2016. doi:10.1145/2933575.2934549.
 - 41 Claire David, Nadime Francis, and Victor Marsault. Distinct shortest walk enumeration for rpgs. *Proc. ACM Manag. Data*, 2(2), May 2024. URL: <https://doi.org/10.1145/3651601>, doi:10.1145/3651601.
 - 42 Alin Deutsch, Nadime Francis, Alastair Green, Keith Hare, Bei Li, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Wim Martens, Jan Michels, Filip Murlak, Stefan Plantikow, Petra Selmer, Hannes Voigt, Oskar van Rest, Domagoj Vrgoč, Mingxi Wu, and Fred Zemke. Graph pattern matching in GQL and SQL/PGQ. In *SIGMOD'22*. ACM, 2022.

- 43 Catalin Dima, Bastien Maubert, and Sophie Pinchinat. Relating paths in transition systems: the fall of the modal μ -calculus. In Vittorio Bilò and Antonio Caruso, editors, *Proceedings of the 17th Italian Conference on Theoretical Computer Science, Lecce, Italy, September 7-9, 2016*, volume 1720 of *CEUR Workshop Proceedings*, pages 240–244. CEUR-WS.org, 2016. URL: <https://ceur-ws.org/Vol-1720/short4.pdf>.
- 44 Gaëtan Douéneau-Tabot, Sophie Pinchinat, and François Schwarzentruher. Chain-monadic second order logic over regular automatic trees and epistemic planning synthesis. In Guram Bezhanishvili, Giovanna D’Agostino, George Metcalfe, and Thomas Studer, editors, *Advances in Modal Logic 12, proceedings of the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, August 27-31, 2018*, pages 237–256. College Publications, 2018. URL: <http://www.aiml.net/volumes/volume12/DoueneauTabot-Pinchinat-Schwarzentruher.pdf>.
- 45 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.
- 46 Nathanaël Fijalkow. Profinite techniques for probabilistic automata. *Bull. EATCS*, 122, 2017. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/497>.
- 47 Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Log. Methods Comput. Sci.*, 11(2), 2015. URL: [https://doi.org/10.2168/LMCS-11\(2:12\)2015](https://doi.org/10.2168/LMCS-11(2:12)2015), doi:10.2168/LMCS-11(2:12)2015.
- 48 Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. Logical and algebraic characterizations of rational transductions. *Log. Methods Comput. Sci.*, 15(4), 2019. URL: [https://doi.org/10.23638/LMCS-15\(4:16\)2019](https://doi.org/10.23638/LMCS-15(4:16)2019), doi:10.23638/LMCS-15(4:16)2019.
- 49 Emmanuel Filiot, Olivier Gauwin, Pierre-Alain Reynier, and Frédéric Servais. From two-way to one-way finite state transducers. In *Proceedings of the 28th Annual IEEE Symposium on Logic in Computer Science (LICS’13)*, pages 468–477. IEEE Computer Society Press, 2013.
- 50 Emmanuel Filiot, Ismaël Jecker, Christof Löding, and Sarah Winter. On Equivalence and Uniformisation Problems for Finite Transducers. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPICs*, pages 125:1–125:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.125.
- 51 Emmanuel Filiot, Ismaël Jecker, Gabriele Puppis, Christof Löding, Anca Muscholl, and Sarah Winter. Finite-valued streaming string transducers. *TheoretCS*, 4, 2025. URL: <https://doi.org/10.46298/theoretics.25.1>, doi:10.46298/THEORETICS.25.1.
- 52 Emmanuel Filiot, Jean-François Raskin, Pierre-Alain Reynier, Frédéric Servais, and Jean-Marc Talbot. Visibly pushdown transducers. *J. Comput. Syst. Sci.*, 97:147–181, 2018. URL: <https://doi.org/10.1016/j.jcss.2018.05.002>, doi:10.1016/J.JCSS.2018.05.002.
- 53 Emmanuel Filiot and Pierre-Alain Reynier. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19, 2016.
- 54 Nadime Francis, Am’elie Gheerbrant, Paolo Guagliardo, Leonid Libkin, Victor Marsault, Wim Martens, Filip Murlak, Liat Peterfreund, Alexandra Rogova, and Domagoj Vrgoc. Gpc: A pattern calculus for property graphs. In *Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS ’23*, page 241–250, New York, NY, USA, 2023. Association for Computing Machinery. URL: <https://doi.org/10.1145/3584372.3588662>, doi:10.1145/3584372.3588662.
- 55 Patrick Gardy, Patricia Bouyer, and Nicolas Markey. Dependences in strategy logic. *Theory Comput. Syst.*, 64(3):467–507, 2020. doi:10.1007/S00224-019-09926-Y.
- 56 Blaise Genest, Hugo Gimbert, Anca Muscholl, and Igor Walukiewicz. Asynchronous games over tree architectures. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP 2013)*, volume 7966 of *LNCS*, pages 275–286. Springer, 2013. doi:10.1007/978-3-642-39212-2_26.

- 57 Hugo Gimbert. Distributed asynchronous games with causal memory are undecidable. *Log. Methods Comput. Sci.*, 18(3), 2022. URL: [https://doi.org/10.46298/lmcs-18\(3:30\)2022](https://doi.org/10.46298/lmcs-18(3:30)2022), doi:10.46298/LMCS-18(3:30)2022.
- 58 Hugo Gimbert, Corto Mascle, Anca Muscholl, and Igor Walukiewicz. Distributed controller synthesis for deadlock avoidance. *Log. Methods Comput. Sci.*, 21(3), 2025. URL: [https://doi.org/10.46298/lmcs-21\(3:24\)2025](https://doi.org/10.46298/lmcs-21(3:24)2025), doi:10.46298/LMCS-21(3:24)2025.
- 59 Hugo Gimbert and Wiesław Zielonka. Games where you can play optimally without any memory. In *CONCUR 2005*, pages 428–442. Springer Berlin Heidelberg, 2005.
- 60 Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. A Bit of Nondeterminism Makes Pushdown Automata Expressive and Succinct. *Logical Methods in Computer Science*, 2023.
- 61 Serge Haddad and Benjamin Monmege. Interval iteration algorithm for MDPs and IMDPs. *Theoretical Computer Science*, 735:111–131, July 2018. doi:10.1016/j.tcs.2016.12.003.
- 62 Kosaburo Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988. URL: <https://www.sciencedirect.com/science/article/pii/0890540188900338>, doi:[https://doi.org/10.1016/0890-5401\(88\)90033-8](https://doi.org/10.1016/0890-5401(88)90033-8).
- 63 Thomas A. Henzinger and Nir Piterman. Solving games without determinization. In Zoltán Ésik, editor, *Computer Science Logic*, pages 395–410, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- 64 Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In - *Proceedings of the 29th International Conference on Computer Aided Verification (CAV 2017)*, volume 10426 of *LNCS*, pages 3–29. Springer, 2017. doi:10.1007/978-3-319-63387-9_1.
- 65 International Organization for Standardization. GQL. <https://www.iso.org/standard/76120.html>, 2024. Standard ISO/IEC CD 39075.
- 66 Marcin Jurdziński and Ranko Lazić. Succinct progress measures for solving parity games. In *LICS*, 2017.
- 67 Prateek Karandikar and Philippe Schnoebelen. The height of piecewise-testable languages and the complexity of the logic of subwords. *Log. Methods Comput. Sci.*, 15(2), 2019. URL: [https://doi.org/10.23638/LMCS-15\(2:6\)2019](https://doi.org/10.23638/LMCS-15(2:6)2019), doi:10.23638/LMCS-15(2:6)2019.
- 68 Kirsten, Daniel. Distance desert automata and the star height problem. *RAIRO-Theor. Inf. Appl.*, 39(3):455–509, 2005. URL: <https://doi.org/10.1051/ita:2005027>, doi:10.1051/ita:2005027.
- 69 Kenneth Krohn and John Rhodes. Algebraic theory of machines. i. prime decomposition theorem for finite semigroups and machines. *Transactions of The American Mathematical Society - TRANS AMER MATH SOC*, 116, 04 1965. doi:10.2307/1994127.
- 70 Denis Kuperberg and Michał Skrzypczak. On determinisation of good-for-games automata. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP'15)*, volume 9135 of *LNCS*, pages 299–310. Springer, 2015.
- 71 Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV 2011)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011. doi:10.1007/978-3-642-22110-1_47.
- 72 Leonid Libkin and Domagoj Vrgoč. Regular path queries on graphs with data. In *Proceedings of the 15th International Conference on Database Theory, ICDT '12*, page 74–85, New York, NY, USA, 2012. Association for Computing Machinery. URL: <https://doi.org/10.1145/2274576.2274585>, doi:10.1145/2274576.2274585.

- 73 Robert F. McNaughton and Seymour A. Papert. *Counter-Free Automata*. MIT Press, 1971. MIT Research Monograph 65.
- 74 Joshua Moerman, Matteo Sammartino, Alexandra Silva, Bartek Klin, and Michał Szynwelski. Learning nominal automata. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL'17)*, volume 52, pages 613–625. ACM, 2017.
- 75 Anca Muscholl and Gabriele Puppis. The many facets of string transducers (invited talk). In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, Berlin, Germany, March 13-16, 2019*, volume 126 of *LIPICs*, pages 2:1–2:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: <https://doi.org/10.4230/LIPICs.STACS.2019.2>, doi:10.4230/LIPICs.STACS.2019.2.
- 76 Anca Muscholl and Igor Walukiewicz. Distributed synthesis for acyclic architectures. In Venkatesh Raman and S. P. Suresh, editors, *Proceedings of the 34th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 201, New Delhi, India, December 15-17, 2014*, volume 29 of *LIPICs*, pages 639–651. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014. URL: <https://doi.org/10.4230/LIPICs.FSTTCS.2014.639>, doi:10.4230/LIPICs.FSTTCS.2014.639.
- 77 John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.
- 78 Thomas Place, Varun Ramanathan, and Pascal Weil. Covering and separation for logical fragments with modular predicates. *Log. Methods Comput. Sci.*, 15(2), 2019. URL: [https://doi.org/10.23638/LMCS-15\(2:11\)2019](https://doi.org/10.23638/LMCS-15(2:11)2019), doi:10.23638/LMCS-15(2:11)2019.
- 79 Thomas Place and Marc Zeitoun. Separating regular languages with first-order logic. *Log. Methods Comput. Sci.*, 12(1), 2016. URL: [https://doi.org/10.2168/LMCS-12\(1:5\)2016](https://doi.org/10.2168/LMCS-12(1:5)2016), doi:10.2168/LMCS-12(1:5)2016.
- 80 Thomas Place and Marc Zeitoun. Separation for dot-depth two. *Log. Methods Comput. Sci.*, 17(3), 2021. URL: [https://doi.org/10.46298/lmcs-17\(3:24\)2021](https://doi.org/10.46298/lmcs-17(3:24)2021), doi:10.46298/LMCS-17(3:24)2021.
- 81 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.
- 82 Marcel-Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8:190–194, 1965.
- 83 Muzammil Shahbaz and Roland Groz. Inferring mealy machines. In Ana Cavalcanti and Dennis R. Dams, editors, *FM 2009: Formal Methods*, pages 207–222, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- 84 Imre Simon. Piecewise testable events. In *Proceedings of the 2nd GI Conference on Automata Theory and Formal Languages*, pages 214–222, Berlin, Heidelberg, 1975. Springer-Verlag.
- 85 Balder ten Cate and Luc Segoufin. Transitive closure logic, nested tree walking automata, and xpath. *J. ACM*, 57(3):18:1–18:41, 2010. URL: <https://doi.org/10.1145/1706591.1706598>, doi:10.1145/1706591.1706598.
- 86 Nikos Tzevelekos. Fresh-register automata. In *Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages (POPL'11)*, pages 295–306. ACM, 2011.
- 87 Mojtaba Valizadeh, Nathanaël Fijalkow, and Martin Berger. LTL learning on gpus. In *Proceedings of the 36th International Conference on Computer Aided Verification (CAV 2024)*, volume 14683 of *LNCS*, pages 209–231. Springer, 2024. doi:10.1007/978-3-031-65633-0\ _10.
- 88 Victor Vianu. Datalog unchained. In *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, pages 57–69. ACM, 2021. URL: <https://doi.org/10.1145/3452021.3458815>, doi:10.1145/3452021.3458815.

Twenty Years of GdR IFM, seen from GT Verification

Over the past two decades, the field of formal verification has experienced a series of significant scientific advances, driven by the growing complexity of software, the heterogeneity of architectures, and the introduction of probabilistic components or artificial intelligence systems. The French research community has made notable contributions to several of these breakthroughs, particularly in the study of infinite-state systems, the analysis of the complexity of the reachability problem in Petri nets, game-theoretic approaches, interactions between verification and AI, as well as quantitative verification and the formal analysis of distributed or parameterized systems, and also for the development of formal methods for security.

1 Infinite-State Systems, Parameterized Systems, and Distributed Architectures

The gradual extension of models beyond finite-state systems constitutes one of the structuring trends of this period. Research on infinite-state systems, incorporating stacks, counters, channels, data, or unbounded resources, has benefited from major contributions from the French community [26, 53]. Techniques based on well-founded orders, symbolic representations, acceleration, and invariant construction have led to general decidability characterizations as well as effective procedures for many classes of systems, and have been internationally recognized with prestigious awards (CAV award, 2017).

A line of works has studied separation logic as a powerful tool to model and analyze programs manipulating pointers. This has led to tool development and successful verification results [58, 74, 51].

The verification of parameterized systems, where the size of the system is unknown or arbitrary, has been structured around induction methods, cut-offs, and symbolic abstraction. Influential results have been obtained for distributed protocols, and have been further consolidated by more recent French contributions studying various models of communicating systems [2, 20, 67, 35].

Regarding distributed systems, a line of works has studied the problem of distributed synthesis: it consists in determining whether, and how, a program can be realized in a distributed manner. Being undecidable in general, this problem has been studied for a long time, in order to identify sufficient conditions to recover decidability, and refine the border with undecidability [54, 55, 56].

2 Petri Nets: Complexity of Reachability

Over the past twenty years, a major advance has been the refined understanding of the complexity of reachability in Petri nets—a problem whose decidability has been known since Mayr (1984), but whose precise complexity bounds have only been progressively clarified much more recently. French contributions have played a decisive role in this clarification, particularly with respect to non-primitive recursive complexity and structurally restricted classes [63, 64, 65, 47].

These works have enabled a more nuanced characterization of decidability boundaries for various submodels, as well as the development of algorithmic approaches grounded in

well-founded orders and structural analysis, directly influencing the verification of concurrent and symbolic systems.

3 Open Systems, Games, and Synthesis

The modelling of open systems as two-player games represents a major conceptual shift. The transition from system verification to system synthesis has made it possible to reformulate fundamental problems as questions of the existence of winning strategies in graphs, giving rise to a rich algorithmic theory. As a consequence, this has significantly increased the interest of the community (together with that of GT DAAL) for studying games. The integration of these tools into verification pipelines has enabled reactive synthesis algorithms, techniques to automatically build controllers that are correct by construction, and robustness analysis with respect to partially observable environments. For instance, tools developed by members of the GT Vérification obtain outstanding results in the annual SyntComp competition [73].

A topic that has attracted a lot of interest is the study of the memory used by strategies, aiming at characterizing the amount of memory needed to win [34, 40, 70], as managing the memory required for strategies is essential in the applications of synthesis.

There has also been strong interest for extensions of games to systems involving more quantitative aspects, such as stochastic games, timed games and weighted games. Numerous works have tackled such classes of games, in order to identify decidable classes, obtain more precise complexity bounds, and develop efficient algorithms for computing winning strategies, thus extending synthesis algorithms to quantitative settings [29, 52, 31, 14, 69].

Finally, another important setting that has been actively investigated during the last twenty years is that of history-deterministic automata (aka good-for-games), which constitute a fruitful compromise between determinism and non-determinism, and are useful to describe specifications in the context of games [39, 62].

4 Quantitative and Probabilistic Verification

Quantitative verification has emerged in response to the needs of autonomous, embedded, and cyber-physical systems. Probabilistic models such as Markov chains, MDPs, and stochastic games have become central objects of study [7, 17]. When considering real-time systems, timed automata are the most widely used model.

Timed automata [3] have generated a substantial body of literature, particularly active in France over the past twenty years. These automata are equipped with clocks that model the continuous passage of time, yielding infinite-state systems that nonetheless retain enough structure to admit meaningful algorithmic results. Contributions over the last two decades have led, among other advances, to a better understanding of region and zone abstractions enabling efficient algorithms, and to precise decidability boundaries when extending timed automata with stacks [1], weights [28], games [48], or combinations of these extensions [41, 32, 36]. Connections with real-time logics such as MTL have also been studied [72], particularly in the context of model checking.

Probabilistic and real-time model-checking have seen strong improvements, with the development of mature tools. The french community has been strongly involved in these works, in particular for the development of tools such as UppAal [27, 15, 30], UppAal TiGa [16], TChecker [57], Roméo [66], and IMITATOR [4].

The notion of robustness has emerged as an important criterion to discard irrelevant models. It has been particularly considered in the context of timed systems, leading to a

new line of research addressing both model checking and synthesis [33, 38].

Probabilistic approaches have also been considered to develop new model checking algorithms in order to address the state space explosion problem. These techniques are known as statistical model checking [61, 9, 71].

5 Verification and Artificial Intelligence

The last two decades have seen the strong development of machine learning techniques. Interactions between verification and artificial intelligence have taken two main forms.

The use of learning techniques to guide invariant construction, strategy synthesis, or state-space reduction has been explored in several recent works [37, 75, 50].

A second research direction concerns the verification of AI systems, and more specifically neural networks. Research focuses on robustness, formal explainability, verifiable abstractions for deep networks, and the integration of learned components into critical architectures [43, 59]. This area remains largely open, but it has now become a structuring domain within formal verification.

6 Formal Methods for Security

Security is a domain where formal verification is of great interest. In particular, many research activities in the past decades have focused on the verification of cryptographic protocols, that are used to secure communications over untrusted networks (Internet, wireless communications, etc.). In the last few years, since the creation of GdR Sécurité Informatique, these activities have mostly migrated to the newly created GT Méthodes Formelles pour la Sécurité, but they were originally part of GT Vérif.

A line of work that has been very active and fruitful in the past twenty years is the design of methods to automate formal proofs of security protocols. The idea is to propose tools that can automate the verification of security properties for protocols, as well as the search for attacks on them, and help produce machine-checked security proofs. Such approaches were historically based on techniques related to the verification of transition systems, rewriting, and constraint systems. Many members of GT Vérif have contributed to this field, by creating and developing tools, designing their theoretical foundations, and improving their automation capabilities over the years. This has led to the production of many tools such as ProVerif [22, 24, 25], Tamarin [68, 13, 44], Deepsec [42], EasyCrypt [11, 10], CryptoVerif [23], Squirrel [8, 6], which have been successfully used to find bugs and verify security properties for many widely deployed protocols, e.g. TLS, Signal, EMV or 5G AKA [60, 21, 12], demonstrating their impact and generality.

Another notable activity of the security-focused side of GT Vérif is the study of electronic voting systems, which is a domain where formally verifying the resistance to various attacks against vote privacy or verifiability is of vital importance for our society. Members of GT Vérif have largely contributed to this subject, with the goal of establishing definitions, general results on them (reduction to small attacks, hierarchies of properties, etc.), and theoretical frameworks to analyse the security of e-voting schemes [49, 46, 19, 5, 18]. They have also proposed e-voting protocols that bring strong security guarantees such as Belenios [45], which is now used in many small-scale elections.

Contributors

Nathalie Bertrand, Joseph Lallemand and Pierre-Alain Reynier.

References

- 1 Parosh Aziz Abdulla, Mohamed Faouzi Atig, and Jari Stenman. Dense-timed pushdown automata. In *2012 27th Annual IEEE Symposium on Logic in Computer Science*, pages 35–44. IEEE, 2012.
- 2 Parosh Aziz Abdulla, Giorgio Delzanno, Othmane Rezine, Arnaud Sangnier, and Riccardo Traverso. Parameterized verification of time-sensitive models of ad hoc network protocols. *Theor. Comput. Sci.*, 612:1–22, 2016. doi:10.1016/J.TCS.2015.07.048.
- 3 Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- 4 Étienne André. IMITATOR 3: Synthesis of timing parameters beyond decidability. In Alexandra Silva and K. Rustan M. Leino, editors, *Computer Aided Verification - 33rd International Conference, CAV 2021, Virtual Event, July 20-23, 2021, Proceedings, Part I*, volume 12759 of *Lecture Notes in Computer Science*, pages 552–565. Springer, 2021. doi:10.1007/978-3-030-81685-8_26.
- 5 Myrto Arapinis, Véronique Cortier, and Steve Kremer. When are three voters enough for privacy properties? In Ioannis G. Askoxylakis, Sotiris Ioannidis, Sokratis K. Katsikas, and Catherine Meadows, editors, *Computer Security - ESORICS 2016 - 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 26-30, 2016, Proceedings, Part II*, volume 9879 of *Lecture Notes in Computer Science*, pages 241–260. Springer, 2016. doi:10.1007/978-3-319-45741-3_13.
- 6 David Baelde, Stéphanie Delaune, Charlie Jacomme, Adrien Koutsos, and Solène Moreau. An interactive prover for protocol verification in the computational model. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 537–554. IEEE, 2021. doi:10.1109/SP40001.2021.00078.
- 7 Christel Baier, Marcus Größer, and Nathalie Bertrand. Probabilistic ω -automata. *J. ACM*, 59(1):1:1–1:52, 2012. doi:10.1145/2108242.2108243.
- 8 Gergei Bana and Hubert Comon-Lundh. A computationally complete symbolic attacker for equivalence properties. In Gail-Joon Ahn, Moti Yung, and Ninghui Li, editors, *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 609–620. ACM, 2014. doi:10.1145/2660267.2660276.
- 9 Benoît Barbot, Serge Haddad, and Claudine Picaronny. Coupling and importance sampling for statistical model checking. In Cormac Flanagan and Barbara König, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 18th International Conference, TACAS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7214 of *Lecture Notes in Computer Science*, pages 331–346. Springer, 2012. doi:10.1007/978-3-642-28756-5_23.
- 10 Gilles Barthe, Cédric Fournet, Benjamin Grégoire, Pierre-Yves Strub, Nikhil Swamy, and Santiago Zanella-Béguelin. Probabilistic relational verification for cryptographic implementations. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 193–206. ACM, 2014. doi:10.1145/2535838.2535847.
- 11 Gilles Barthe, Benjamin Grégoire, Sylvain Heraud, and Santiago Zanella-Béguelin. Computer-aided security proofs for the working cryptographer. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 71–90. Springer, 2011. doi:10.1007/978-3-642-22792-9_5.
- 12 David A. Basin, Jannik Dreier, Lucca Hirschi, Sasa Radomirovic, Ralf Sasse, and Vincent Stettler. A formal analysis of 5g authentication. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 1383–1396. ACM, 2018. doi:10.1145/3243734.3243846.

- 13 David A. Basin, Jannik Dreier, and Ralf Sasse. Automated symbolic proofs of observational equivalence. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 1144–1155. ACM, 2015. doi:10.1145/2810103.2813662.
- 14 Nicolas Basset, Marta Z. Kwiatkowska, and Clemens Wiltsche. Compositional strategy synthesis for stochastic games with multiple objectives. *Inf. Comput.*, 261:536–587, 2018. doi:10.1016/J.IC.2017.09.010.
- 15 Gerd Behrmann, Patricia Bouyer, Kim Guldstrand Larsen, and Radek Pelánek. Lower and upper bounds in zone-based abstractions of timed automata. *Int. J. Softw. Tools Technol. Transf.*, 8(3):204–215, 2006. doi:10.1007/S10009-005-0190-0.
- 16 Gerd Behrmann, Agnès Cougnard, Alexandre David, Emmanuel Fleury, Kim Guldstrand Larsen, and Didier Lime. Uppaal-tiga: Time for playing games! In Werner Damm and Holger Hermanns, editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 121–125. Springer, 2007. doi:10.1007/978-3-540-73368-3_14.
- 17 Marius Belly, Nathanaël Fijalkow, Hugo Gimbert, Florian Horn, Guillermo A. Pérez, and Pierre Vandenhover. Revelations: A decidable class of pomdps with omega-regular objectives. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 26454–26462. AAAI Press, 2025. doi:10.1609/AAAI.V39I25.34845.
- 18 David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. Sok: A comprehensive analysis of game-based ballot privacy definitions. In *2015 IEEE Symposium on Security and Privacy, SP 2015, San Jose, CA, USA, May 17-21, 2015*, pages 499–516. IEEE Computer Society, 2015. doi:10.1109/SP.2015.37.
- 19 David Bernhard, Véronique Cortier, Olivier Pereira, and Bogdan Warinschi. Measuring vote privacy, revisited. In Ting Yu, George Danezis, and Virgil D. Gligor, editors, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012*, pages 941–952. ACM, 2012. doi:10.1145/2382196.2382295.
- 20 Nathalie Bertrand, Nicolas Markey, Ocan Sankur, and Nicolas Waldburger. Parameterized safety verification of round-based shared-memory systems. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, Paris, France, July 4-8, 2022*, volume 229 of *LIPICs*, pages 113:1–113:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.ICALP.2022.113.
- 21 Karthikeyan Bhargavan, Vincent Cheval, and Christopher A. Wood. A symbolic analysis of privacy for TLS 1.3 with encrypted client hello. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 365–379. ACM, 2022. doi:10.1145/3548606.3559360.
- 22 Bruno Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), 11-13 June 2001, Cape Breton, Nova Scotia, Canada*, pages 82–96. IEEE Computer Society, 2001. doi:10.1109/CSFW.2001.930138.
- 23 Bruno Blanchet. A computationally sound mechanized prover for security protocols. In *2006 IEEE Symposium on Security and Privacy (S&P 2006), 21-24 May 2006, Berkeley, California, USA*, pages 140–154. IEEE Computer Society, 2006. doi:10.1109/SP.2006.1.
- 24 Bruno Blanchet, Martín Abadi, and Cédric Fournet. Automated verification of selected equivalences for security protocols. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 331–340. IEEE Computer Society, 2005. doi:10.1109/LICS.2005.8.
- 25 Bruno Blanchet, Vincent Cheval, and Véronique Cortier. Proverif with lemmas, induction, fast subsumption, and much more. In *43rd IEEE Symposium on Security and Privacy, SP*

- 2022, San Francisco, CA, USA, May 22-26, 2022, pages 69–86. IEEE, 2022. doi:10.1109/SP46214.2022.9833653.
- 26 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: Concurrency Theory, 8th International Conference, Warsaw, Poland, July 1-4, 1997, Proceedings*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997. doi:10.1007/3-540-63141-0_10.
 - 27 Patricia Bouyer. Untameable timed automata! In Helmut Alt and Michel Habib, editors, *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, volume 2607 of *Lecture Notes in Computer Science*, pages 620–631. Springer, 2003. doi:10.1007/3-540-36494-3_54.
 - 28 Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.
 - 29 Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim Guldstrand Larsen. Optimal strategies in priced timed game automata. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science, 24th International Conference, Chennai, India, December 16-18, 2004, Proceedings*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004. doi:10.1007/978-3-540-30538-5_13.
 - 30 Patricia Bouyer, Maximilien Colange, and Nicolas Markey. Symbolic optimal reachability in weighted timed automata. In Swarat Chaudhuri and Azadeh Farzan, editors, *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*, volume 9779 of *Lecture Notes in Computer Science*, pages 513–530. Springer, 2016. doi:10.1007/978-3-319-41528-4_28.
 - 31 Patricia Bouyer, Samy Jaziri, and Nicolas Markey. On the value problem in weighted timed games. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1-4, 2015*, volume 42 of *LIPICs*, pages 311–324. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICs.CONCUR.2015.311.
 - 32 Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. Almost optimal strategies in one-clock priced timed games. In *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06)*, volume 4337 of *LNCS*, pages 345–356. Springer, 2006.
 - 33 Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robust reachability in timed automata: A game-based approach. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 128–140. Springer, 2012. doi:10.1007/978-3-642-31585-5_15.
 - 34 Patricia Bouyer, Youssouf Oualhadj, Mickael Randour, and Pierre Vandenhove. Arena-independent finite-memory determinacy in stochastic games. *Log. Methods Comput. Sci.*, 19(4), 2023. doi:10.46298/LMCS-19(4:18)2023.
 - 35 Marius Bozga, Radu Iosif, Arnaud Sangnier, and Neven Villani. Counting abstraction and decidability for the verification of structured parameterized networks. In Ruzica Piskac and Zvonimir Rakamaric, editors, *Computer Aided Verification - 37th International Conference, CAV 2025, Zagreb, Croatia, July 23-25, 2025, Proceedings, Part III*, volume 15933 of *Lecture Notes in Computer Science*, pages 238–262. Springer, 2025. doi:10.1007/978-3-031-98682-6_13.
 - 36 Thomas Brihaye, Gilles Geeraerts, Axel Haddad, Engel Lefaucheux, and Benjamin Monmege. One-clock priced timed games with negative weights. *Logical Methods in Computer Science*, 18(3), August 2022. doi:10.46298/lmcs-18(3:17)2022.

- 37 Damien Busatto-Gaston, Debraj Chakraborty, and Jean-François Raskin. Monte carlo tree search guided by symbolic advice for mdps. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria (Virtual Conference), September 1-4, 2020*, volume 171 of *LIPICs*, pages 40:1–40:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CONCUR.2020.40.
- 38 Damien Busatto-Gaston, Benjamin Monmege, Pierre-Alain Reynier, and Ocan Sankur. Robust controller synthesis in timed büchi automata: A symbolic approach. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2019. doi:10.1007/978-3-030-25540-4_33.
- 39 Antonio Casares, Thomas Colcombet, and Karoliina Lehtinen. On the size of good-for-games rabin automata and its link with the memory in muller games. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, Paris, France, July 4-8, 2022*, volume 229 of *LIPICs*, pages 117:1–117:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.ICALP.2022.117.
- 40 Antonio Casares and Pierre Ohlmann. Characterising memory in infinite games. *Log. Methods Comput. Sci.*, 21(1), 2025. doi:10.46298/LMCS-21(1:28)2025.
- 41 Franck Cassez, Alexandre David, Emmanuel Fleury, Kim G. Larsen, and Didier Lime. Efficient on-the-fly algorithms for the analysis of timed games. In *Proceedings of the 16th International Conference on Concurrency Theory (CONCUR'05)*, volume 3653 of *LNCS*, pages 66–80. Springer, 2005.
- 42 Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. The DEEPSEC prover. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part II*, volume 10982 of *Lecture Notes in Computer Science*, pages 28–36. Springer, 2018. doi:10.1007/978-3-319-96142-2_4.
- 43 Judith Clymo, Haik Manukian, Nathanael Fijalkow, Adria Gascon, and Brooks Paige. Data generation for neural programming by example. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3450–3459. PMLR, 26–28 Aug 2020. URL: <https://proceedings.mlr.press/v108/clymo20a.html>.
- 44 Véronique Cortier, Stéphanie Delaune, and Jannik Dreier. Automatic generation of sources lemmas in tamarin: Towards automatic proofs of security protocols. In Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider, editors, *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14-18, 2020, Proceedings, Part II*, volume 12309 of *Lecture Notes in Computer Science*, pages 3–22. Springer, 2020. doi:10.1007/978-3-030-59013-0_1.
- 45 Véronique Cortier, David Galindo, Stéphane Glondu, and Malika Izabachène. Election verifiability for helios under weaker trust assumptions. In Mirosław Kutylowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wrocław, Poland, September 7-11, 2014. Proceedings, Part II*, volume 8713 of *Lecture Notes in Computer Science*, pages 327–344. Springer, 2014. doi:10.1007/978-3-319-11212-1_19.
- 46 Véronique Cortier and Ben Smyth. Attacking and fixing helios: An analysis of ballot secrecy. In *Proceedings of the 24th IEEE Computer Security Foundations Symposium, CSF 2011, Cernay-la-Ville, France, 27-29 June, 2011*, pages 297–311. IEEE Computer Society, 2011. doi:10.1109/CSF.2011.27.
- 47 Wojciech Czerwinski, Sławomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The reachability problem for petri nets is not elementary. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of*

- Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 24–33. ACM, 2019. doi:10.1145/3313276.3316369.
- 48 Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar, and Mariëlle Stoelinga. The element of surprise in timed games. In *Proceedings of the 14th International Conference on Concurrency Theory (CONCUR'03)*, volume 2761 of *LNCS*, pages 144–158. Springer, 2003.
- 49 Stéphanie Delaune, Steve Kremer, and Mark Ryan. Coercion-resistance and receipt-freeness in electronic voting. In *19th IEEE Computer Security Foundations Workshop, (CSFW-19 2006), 5-7 July 2006, Venice, Italy*, pages 28–42. IEEE Computer Society, 2006. doi:10.1109/CSFW.2006.8.
- 50 Florent Delgrange, Guy Avni, Anna Lukina, Christian Schilling, Ann Nowé, and Guillermo A. Pérez. Composing reinforcement learning policies, with formal guarantees. In Sanmay Das, Ann Nowé, and Yevgeniy Vorobeychik, editors, *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, pages 574–583. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2025. doi:10.5555/3709347.3743573.
- 51 Stéphane Demri, Étienne Lozes, and Alessio Mansutti. A complete axiomatisation for quantifier-free separation logic. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/LMCS-17(3:17)2021.
- 52 Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Log. Methods Comput. Sci.*, 11(2), 2015. doi:10.2168/LMCS-11(2:12)2015.
- 53 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.
- 54 Paul Gastin and Nathalie Sznajder. Fair synthesis for asynchronous distributed systems. *ACM Trans. Comput. Log.*, 14(2):9:1–9:31, 2013. doi:10.1145/2480759.2480761.
- 55 Hugo Gimbert. Distributed asynchronous games with causal memory are undecidable. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/LMCS-18(3:30)2022.
- 56 Hugo Gimbert, Corto Mascle, Anca Muscholl, and Igor Walukiewicz. Distributed controller synthesis for deadlock avoidance. *Log. Methods Comput. Sci.*, 21(3), 2025. doi:10.46298/LMCS-21(3:24)2025.
- 57 Frédéric Herbreteau, B. Srivathsan, and Igor Walukiewicz. Better abstractions for timed automata. *Inf. Comput.*, 251:67–90, 2016. doi:10.1016/J.IC.2016.07.004.
- 58 Radu Iosif, Adam Rogalewicz, and Jirí Simáček. The tree width of separation logic with recursive definitions. In Maria Paola Bonacina, editor, *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, volume 7898 of *Lecture Notes in Computer Science*, pages 21–38. Springer, 2013. doi:10.1007/978-3-642-38574-2_2.
- 59 Igor Khmelnitsky, Daniel Neider, Rajarshi Roy, Xuan Xie, Benoît Barbot, Benedikt Bollig, Alain Finkel, Serge Haddad, Martin Leucker, and Lina Ye. Property-directed verification and robustness certification of recurrent neural networks. In Zhe Hou and Vijay Ganesh, editors, *Automated Technology for Verification and Analysis - 19th International Symposium, ATVA 2021, Gold Coast, QLD, Australia, October 18-22, 2021, Proceedings*, volume 12971 of *Lecture Notes in Computer Science*, pages 364–380. Springer, 2021. doi:10.1007/978-3-030-88885-5_24.
- 60 Nadim Kobeissi, Karthikeyan Bhargavan, and Bruno Blanchet. Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach. In *2017 IEEE European Symposium on Security and Privacy, EuroS&P 2017, Paris, France, April 26-28, 2017*, pages 435–450. IEEE, 2017. doi:10.1109/EUROSP.2017.38.
- 61 Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking: An overview. In Howard Barringer, Yliès Falcone, Bernd Finkbeiner, Klaus Havelund, Insup Lee, Gordon J. Pace, Grigore Rosu, Oleg Sokolsky, and Nikolai Tillmann, editors, *Runtime Verification - First International Conference, RV 2010, St. Julians, Malta, November 1-4, 2010. Proceedings*,

- volume 6418 of *Lecture Notes in Computer Science*, pages 122–135. Springer, 2010. doi:10.1007/978-3-642-16612-9_11.
- 62 Karoliina Lehtinen and Aditya Prakash. The 2-token theorem: Recognising history-deterministic parity automata efficiently. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 1839–1850. ACM, 2025. doi:10.1145/3717823.3718310.
 - 63 Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021. doi:10.1109/FOCS52979.2021.00121.
 - 64 Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 56–67. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.16.
 - 65 Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785796.
 - 66 Didier Lime, Olivier H. Roux, Charlotte Seidner, and Louis-Marie Traonouez. Romeo: A parametric model-checker for petri nets with stopwatches. In Stefan Kowalewski and Anna Philippou, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5505 of *Lecture Notes in Computer Science*, pages 54–57. Springer, 2009. doi:10.1007/978-3-642-00768-2_6.
 - 67 Corto Mascle, Anca Muscholl, and Igor Walukiewicz. Model-checking parametric lock-sharing systems against regular constraints. In Guillermo A. Pérez and Jean-François Raskin, editors, *34th International Conference on Concurrency Theory, CONCUR 2023, Antwerp, Belgium, September 18-23, 2023*, volume 279 of *LIPICs*, pages 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICs.CONCUR.2023.24.
 - 68 Simon Meier, Benedikt Schmidt, Cas Cremers, and David A. Basin. The TAMARIN prover for the symbolic analysis of security protocols. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 696–701. Springer, 2013. doi:10.1007/978-3-642-39799-8_48.
 - 69 Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. Decidability of one-clock weighted timed games with arbitrary weights. *Log. Methods Comput. Sci.*, 21(1), 2025. doi:10.46298/LMCS-21(1:8)2025.
 - 70 Benjamin Monmege, Julie Parreaux, and Pierre-Alain Reynier. Playing stochastically in weighted timed games to emulate memory. *Log. Methods Comput. Sci.*, 21(1), 2025. doi:10.46298/LMCS-21(1:19)2025.
 - 71 Ayoub Nouri, Saddek Bensalem, Marius Bozga, Benoît Delahaye, Cyrille Jégourel, and Axel Legay. Statistical model checking qos properties of systems with SBIP. *Int. J. Softw. Tools Technol. Transf.*, 17(2):171–185, 2015. doi:10.1007/S10009-014-0313-6.
 - 72 Joël Ouaknine and James Worrell. On the decidability and complexity of metric temporal logic over finite words. *Log. Methods Comput. Sci.*, 3(1), 2007. URL: [https://doi.org/10.2168/LMCS-3\(1:8\)2007](https://doi.org/10.2168/LMCS-3(1:8)2007), doi:10.2168/LMCS-3(1:8)2007.
 - 73 Florian Renkin, Philipp Schlehuber-Caissier, Alexandre Duret-Lutz, and Adrien Pommellet. Dissecting ltsynt. *Formal Methods Syst. Des.*, 61(2):248–289, 2022. doi:10.1007/S10703-022-00407-6.
 - 74 Mihaela Sighireanu, Juan Antonio Navarro Pérez, Andrey Rybalchenko, Nikos Gorogiannis, Radu Iosif, Andrew Reynolds, Cristina Serban, Jens Katelaan, Christoph Matheja, Thomas Noll, Florian Zuleger, Wei-Ngan Chin, Quang Loc Le, Quang-Trung Ta, Ton-Chanh Le, Thanh-Toan Nguyen, Siau-Cheng Khoo, Michal Cyprian, Adam Rogalewicz, Tomás Vojnar,

- Constantin Enea, Ondrej Lengál, Chong Gao, and Zhilin Wu. SL-COMP: competition of solvers for separation logic. In Dirk Beyer, Marieke Huisman, Fabrice Kordon, and Bernhard Steffen, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 25 Years of TACAS: TOOLympics, Held as Part of ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part III*, volume 11429 of *Lecture Notes in Computer Science*, pages 116–132. Springer, 2019. doi:10.1007/978-3-030-17502-3_8.
- 75 Mojtaba Valizadeh, Nathanaël Fijalkow, and Martin Berger. LTL learning on gpus. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part III*, volume 14683 of *Lecture Notes in Computer Science*, pages 209–231. Springer, 2024. doi:10.1007/978-3-031-65633-0_10.

Les vingt ans du GdR IFM, vus du GT « Structures formelles pour le calcul et les preuves »

Les thématiques du GT Scalp (Structures formelles pour le CALCul et les Preuves) du GdR IFM recouvrent l'étude des structures mathématiques permettant de modéliser preuves et programmes, avec une attention particulière (mais non exclusive) à l'interface entre les deux. Le spectre du groupe, qui s'insère dans un éventail de disciplines allant de la théorie de la démonstration à la théorie des langages de programmation, couvre notamment les systèmes de calcul d'origine logico-algébrique (lambda-calcul, réécriture de termes et de graphes, calculs de processus), les systèmes de déduction logique (classique, intuitionniste, linéaire), la théorie des types et les systèmes d'inférence, ainsi que les outils de preuve automatiques ou interactifs, avec des applications à l'analyse et la vérification de programmes (interprétation abstraite, logiques de programmes, automates d'arbres) et de leurs propriétés quantitatives (analyse de complexité, analyse de programmes probabilistes ou quantiques, complexité algorithmique implicite). Les méthodes employées sont à la fois de nature syntaxique (séquents, systèmes de types, machines abstraites, induction et coinduction, recherche de preuves) et sémantique (catégories, domaines, jeux, espaces vectoriels, réalisabilité).

Le GT Scalp s'appuie sur une longue tradition héritée de la logique mathématique du 20ème siècle, de la théorie de la calculabilité et des mathématiques constructives. Au cœur de notre culture commune, se trouvent des objets d'étude comme le lambda-calcul de Church, introduit dans les années 1930 qui est la source de nombreux formalismes essentiels de l'informatique actuelle. Nous présentons ici quelques-unes des évolutions majeures de ces domaines en terme d'outils, de méthodes et de thématiques.

Sur le plan méthodologique, une notion essentielle est la correspondance de Curry-Howard, qui, dans sa forme initiale [46], formalise la coïncidence entre deux objets : d'une part, un système de déduction pour la logique intuitionniste et d'autre part, un langage de programmation idéalisé (le lambda-calcul simplement typé). Elle permet des transferts de méthodes et résultats entre logique et langages de programmation, et a ainsi engendré un domaine à l'interface entre la théorie de la preuve et la théorie de la programmation.

La théorie des catégories [59] est un autre formalisme sur lequel s'appuient de nombreux travaux du groupe. Les catégories sont *la* structure algébrique reflétant les aspects de typage et de composition que l'on retrouve au niveau des preuves comme des programmes. Elles permettent de gérer la complexité combinatoire des objets en les encapsulant sous une couche d'abstraction compositionnelle. La structure qu'elles révèlent suggère de nouveaux principes de raisonnement (par exemple, la co-induction) ou méthodes d'abstraction en programmation (par exemple, les monades).

Le GT Scalp a de nombreuses thématiques et outils communs avec le GT LHC (Logique, Homotopie, Catégories). Mais les développements relevant de Scalp tendent à être plus attachés aux structures formelles (preuves, programmes, notamment) là où LHC se focalise davantage sur les structures mathématiques.

1 Théorie des types et assistants de preuve

1.1 Essor des assistants de preuve, et de leur adoption

La théorie des types et son implémentation dans les assistants de preuve ne sont pas récents. La théorie des types de Martin-Löf [60] a été publiée en 1972, et le système Automath de De Bruijn [19] est encore antérieur. Cependant, l'usage de ces systèmes a vu un essor

fulgurant depuis le début du 21^{ème} siècle, et en particulier ces 20 dernières années. Cet essor a été porté par le succès de projets de formalisation majeurs tels que le théorème des 4 couleurs [33], la preuve de la conjecture de Kepler [42], le théorème de Feit-Thompson [32] ou la conception d'un compilateur C optimisant certifié [58].

Aujourd'hui, les assistants de preuve sont de plus en plus adoptés pour des projets de formalisation des mathématiques classiques. Ancrés dans les thématiques de Scalp, par leur rigueur et leur fonctionnalité d'automatisation, ils rendent possibles les preuves de correction pour des programmes réalistes, dont la combinatoire serait autrement rédhibitoire. Des bibliothèques spécialisées ont vu le jour pour la vérification de programmes : citons par exemple le projet Iris [53] (en Rocq) pour la vérification de propriétés de sûreté de programmes concurrents, basé sur la logique de séparation, et son utilisation pour la vérification du système de type du langage Rust [51]. Dans certaines branches de la vérification de programmes, la présence d'une formalisation dans un assistant de preuve accompagnant un article de recherche est devenue une attente et non plus un bonus.

Depuis un peu plus d'une dizaine d'années, les techniques d'apprentissage statistique commencent à être déployées aux assistants de preuve, pour (l'aide à) la recherche de preuve. Elles sont aujourd'hui utilisées quotidiennement dans certains assistants de preuve, par exemple pour l'interface avec des prouveurs automatiques. Leur développement pour des outils plus puissants est un sujet de recherche très actuel, notamment dans la communauté Scalp, afin de diminuer le coût d'accès des assistants en termes d'expertise et de temps.

La communauté française autour des thématiques de Scalp a joué un rôle central dans le développement, la conception et l'évolution des assistants de preuve, notamment en développant le logiciel Rocq (auparavant nommé Coq), un des principaux assistants de preuve.

1.2 Théorie homotopique des types

Ces 20 dernières années, la recherche en théorie des types a connu un renouveau suite à la découverte par Voevodsky de la nature homotopique du type égalité [71, 65] (suite à des travaux précurseurs de Hofmann et Streicher [44]).

D'une part, elle suggère la possibilité d'une nouvelle génération d'assistants de preuve permettant une gestion plus flexible de l'égalité. Avec sa bibliothèque "Foundations" pour le système Rocq (cf. Section 1.1), Voevodsky a lui-même contribué à un pilier de la formalisation sur ordinateur [72] de l'approche homotopique. D'autre part, la théorie des types homotopique peut servir de fondement pour mécaniser la théorie de l'homotopie elle-même.

Cette thématique développée conjointement par le GT LHC et le GT SCALP est détaillée dans la section concernant le GT LHC de ce document.

2 Théorie de la preuve / Fondements logiques

2.1 Extensions de la correspondance de Curry-Howard

Il est apparu dès les années 90 que la correspondance de Curry-Howard pouvait s'étendre au delà de la simple logique intuitionniste; et notamment que le raisonnement classique correspondait calculatoirement aux opérateurs de contrôle issus de la programmation fonctionnelle [39]. Cette découverte a bousculé les mathématiques constructives, en associant un sens calculatoire à la logique classique, l'exemple paradigmatique d'une logique non constructive.

Cette idée puissante a profondément marqué notre communauté ces dernières décennies, et a inspiré bon nombre de travaux. Certains cherchent à mieux comprendre et décrire le contenu calculatoire de la logique classique ; de cette ligne ont émergé des travaux fondamentaux associant l'ordre d'évaluation des programmes à la notion de polarisation en logique [18]. D'autres cherchent à cerner le contenu logique d'autres effets calculatoires ou primitives de calcul [27, 10, 17]. La réalisabilité classique, initiée en France par les résultats de Krivine [54], va jusqu'à donner un sens calculatoire aux axiomes de ZF [55, 26], établissant un pont entre lambda-calcul et théorie des ensembles.

2.2 Logique linéaire

En 1987, Girard a introduit la logique linéaire [30] : il s'agit d'un raffinement des logiques classiques et intuitionnistes permettant de contrôler l'usage des règles structurelles telles que la contraction et l'affaiblissement. Cela en fait une logique qui, au delà de la vérité des formules, est capable d'exprimer des contraintes sur l'usage des ressources de processus.

Ces 30 dernières années, la logique linéaire a révolutionné la branche de la théorie de la preuve marquée par la correspondance de Curry-Howard. Toute une littérature s'est développée sur la théorie de la preuve de la logique linéaire, se focalisant soit sur ses aspects structurels (canonicité, c'est à dire l'élimination des commutations entre règles dans le calcul des séquents), soit sur ses aspects calculatoires, c'est à dire liés à l'élimination des coupures. Les idées autour de la logique linéaire ont eu une très forte influence dans nombre de sujets connexes : complexité implicite [7], typage linéaire en programmation [9] ou calculs de processus [14], théorie des catégories ou encore sémantique dénotationnelle (voir ci-dessous).

2.3 Nouvelles structures formelles pour les programmes et les preuves

La vision traditionnelle d'un système de preuves est que les axiomes sont vrais, et les règles préservent la vérité. Une preuve est alors un arbre fini enchaînant ces règles, et témoignant de la validité d'une formule.

Ces dernières décennies, un certain nombre de travaux sont venus remettre cette vision traditionnelle en question. Les réseaux de preuve de la logique linéaire [40] présentent les preuves comme des graphes, munis d'un critère de correction global. Les systèmes de preuve non bien fondés autorisent des preuves infinies ou circulaires [21], sujettes à un critère de validité inspiré des jeux de parité, assurant que la preuve « progresse » globalement. Les systèmes à inférence profonde [41] autorisent l'application de règles d'inférence en profondeur dans les formules. Ils permettent ainsi une représentation des preuves plus compacte [13] ; par ailleurs certaines logiques nécessitent un traitement par inférence profonde [70].

Si le raisonnement diagrammatique a toujours accompagné les méthodes formelles (typiquement comme aide informelle au raisonnement), plusieurs lignes de travaux récentes traitent les diagrammes ou les graphes comme des objets syntaxiques formels [68], sur lesquels on peut par exemple développer une théorie de la réécriture adaptée, usuellement via des méthodes catégoriques.

3 Théorie de la programmation

3.1 Sémantiques quantitatives

La sémantique dénotationnelle propose de raisonner sur le comportement des programmes en les plongeant dans un univers mathématique adéquat [67] – historiquement, il s'agissait

d'interpréter les programmes fonctionnels par des fonctions mathématiques, souvent continues vis-à-vis d'une topologie appropriée, spécifiant le comportement entrée / sortie des programmes. Les 30 dernières années ont vu le développement de sémantiques dénotationnelles dépassant cette vision historique.

La sémantique quantitative revisite la tradition de la sémantique dénotationnelle à la lumière de la logique linéaire. Elle consiste en une famille de modèles et approches associées reflétant des aspects quantitatifs du calcul : plutôt que simplement l'information entrée / sortie ($f(1) = 2$), les modèles quantitatifs retiennent le nombre de fois qu'une fonction requiert son argument ($([1, 1], 2) \in f$). Cette sensibilité au nombre de copies fait que les modèles quantitatifs donnent des représentations plus proches de l'exécution [20], ce qui permet d'en déduire des techniques efficaces pour approximer finement le comportement infinitaire des programmes [8].

Là où les modèles dénotationnels historiques importent des notions topologiques, de nombreux modèles quantitatifs s'expriment en termes d'algèbre linéaire ou d'analyse fonctionnelle [69]. L'étude de modèles du lambda-calcul en termes de fonctions analytiques a donné lieu à la découverte du lambda-calcul différentiel [24], établissant un lien fascinant entre la linéarité au sens de l'algèbre linéaire et la linéarité au sens de la logique linéaire, c'est à dire le fait d'utiliser son argument exactement une fois. Cette connexion s'est élargie en tout un champ de recherche étudiant et liant de nombreux aspects de la différentiation, sous un jour mathématique, logique [52], catégorique [11] ou encore en termes de programmation. Certaines sémantiques quantitatives peuvent être décrites syntaxiquement, donnant lieu à de puissants systèmes de types (dits « à intersection ») capturant précisément la longueur de l'exécution [3].

3.2 Sémantique interactive et effets calculatoires

Les sémantiques interactives vont au delà des sémantiques quantitatives, en gardant également en mémoire une information sur la dynamique de l'évaluation. Il en existe plusieurs sortes, incluant la sémantique des jeux [2, 47] et la géométrie de l'interaction [31].

Ces sémantiques, qui datent de la fin du 20ème siècle, se sont fortement développées ces dernières décennies. Entre autres, elles ont permis de largement avancer l'état de l'art pour la sémantique d'effets dits *non commutatifs* (tels que l'état mutable), c'est à dire qu'ils dépendent de l'ordre d'évaluation [45]. Bien qu'originellement définie pour étudier le langage de programmation purement fonctionnel PCF, la sémantique des jeux s'est illustrée par sa capacité à modéliser de nombreux effets calculatoires dans un seul et même cadre [29].

3.3 Prise en compte des ressources en sémantique

Un point aveugle de la sémantique traditionnelle est que l'exécution des programmes implique des ressources, qui peuvent être de nature variée : il peut s'agir de temps, d'espace, d'énergie ou de bande passante, etc. Ces 15 dernières années, de nombreux travaux de recherche ont travaillé à prendre en compte ces aspects, que ce soit par le développement de sémantiques en rendant compte [28, 57, 4], ou bien par le biais de systèmes de types assurant des bornes sur l'utilisation de ressources [12, 56]. Les sémantiques quantitatives se sont révélées très adaptées pour les ressources, en apportant une sensibilité à la multiplicité des appels aux arguments.

3.4 Programmation probabiliste

Une dynamique de recherche significative s'est portée sur la sémantique des primitives probabilistes en programmation. Celles-ci appartiennent à deux catégories de natures très différentes. D'un côté, on a les primitives de probabilité discrète telles qu'un choix probabiliste booléen non biaisé, utile pour implémenter des algorithmes ou protocoles randomisés. De l'autre, on a des primitives pour les probabilités continues, comme des distributions de probabilités sur $[0, 1]$. Ces dernières sont utilisées dans le contexte du paradigme de la « programmation probabiliste », dont l'objet est de manipuler et calculer avec des modèles statistiques [34].

D'un point de vue sémantique, la communauté s'est heurtée à une difficulté historique : il est difficile de trouver des modèles à base de domaines (c'est-à-dire de dcpos continus, inventés dans les années 1970 [67]), qui bénéficient à la fois d'une monade de distributions probabilistes et qui permettent de traiter de l'ordre supérieur ; on peut consulter [50], qui est toujours d'actualité. Cette difficulté a été surmontée de plusieurs façons. Pour le cas des probabilités discrètes, une pondération par des scalaires des modèles quantitatifs issus de la logique linéaire a permis d'obtenir une surprenante caractérisation sémantique de l'équivalence contextuelle [22]. On obtient un résultat similaire d'abstraction complète dans un modèle ordinaire à base de domaines, à condition de traiter de langages plus expressifs, pouvant effectuer non seulement des choix probabilistes mais aussi non déterministes [35], ou bien avec une discipline de call-by-push-value [36]. Si l'on ne vise pas ce genre de résultats, de façon peut-être surprenante, les modèles à base de dcpos, non nécessairement continus, conviennent parfaitement [49, 37], et fournissent des modèles adéquats de langages d'ordre supérieur probabilistes. La communauté s'est également intéressée à des versions pondérées de nombre de notions classiques, telles que la logique [6, 5] ou la réécriture [25]. Pour le cas de probabilités continues, des notions généralisées d'espaces de probabilités ont été développées et appliquées à la sémantique de la programmation probabiliste [43, 23], et les dcpos eux-mêmes, non nécessairement continus, permettent d'atteindre le même but [37].

3.5 Calcul quantique

Les grandes évolutions de l'informatique quantique sont plus largement décrites dans la partie de ce document du GT IQ. Nous décrivons ici les directions à l'interface avec les objectifs et/ou outils du GT Scalp, avec deux directions particulièrement notables.

Tout d'abord, il y a 20 ans à Oxford a été initiée une ligne de travaux explorant les fondements du calcul quantique via des outils logiques et/ou catégoriques [1], relevant des thématiques Scalp. Cette ligne s'est épanouie en une large communauté fédérée par la conférence QPL (Quantum Physics and Logic), et s'intéressant à de nombreux aspects du quantique (causalité quantique, descriptions catégoriques ou logiques de la contextualité, etc). Une retombée particulière est l'étude de syntaxes diagrammatiques pour les transformations quantiques. Ces diagrammes, inspirés par la formulation catégorique du calcul quantique, supportent la représentation des circuits quantiques. Mais ils sont plus souples, et supportent une riche théorie équationnelle permettant de montrer des égalités entre circuits, utile par exemple pour l'optimisation de circuits quantiques. Ces 15 dernières années, cette école s'est largement développée y compris en France, et a proposé de nombreux calculs diagrammatiques dont l'exemple le plus connu est le calcul ZX (appuyé en particulier par un récent théorème de complétude marquant, donnant une théorie équationnelle complète pour le calcul ZX [48]).

Par ailleurs, des travaux ont porté sur la sémantique de langages de programmation munis de primitives pour le calcul quantique (initialisation, application d'unitaires, mesure).

Ici encore, il est apparu qu'une pondération adéquate des modèles quantitatifs issus de la logique linéaire permettait une représentation très précise des programmes quantiques [64], jusqu'à la pleine adéquation, ou abstraction complète [15], c'est à dire la caractérisation de l'équivalence contextuelle. L'effort de recherche se poursuit pour couvrir d'autres paradigmes quantiques (contrôle quantique, calcul quantique optique, etc).

3.6 Différentiation automatique en théorie de la programmation

La différentiation automatique regroupe des techniques pour calculer les dérivées partielles d'une fonction mathématique (disons de \mathbb{R} dans \mathbb{R}) calculée par un programme. Utile dans de nombreux contextes, la différentiation automatique est au cœur de l'explosion récente de l'IA puisqu'elle est utile pour l'apprentissage par descente de gradient, permettant de trouver un minimum local de la fonction calculée par un réseau de neurones.

Ces 10 dernières années, plusieurs travaux ont exploré des aspects logiques et sémantiques de la différentiation automatique [61], et en particulier de la rétropropagation. Les questions abordées incluent l'extension de la différentiation automatique à des langages de programmation plus expressifs ou sa présentation comme une transformation de programmes, mais aussi des liens avec la logique [52], ou encore la structure catégorique de la rétropropagation [16].

3.7 Convergences entre lambda-calcul et automates

Si la théorie des automates et le lambda-calcul / la sémantique dénotationnelle ont des origines communes, via par exemple la notion de « schémas de programmes » étudiée dans les années 70 à 80, elles ont divergé dans les années 90 pour donner lieu à des communautés essentiellement séparées, qu'on retrouve aujourd'hui dans différents GTs du GDR IFM. La théorie des automates se concentre sur des modèles de calcul à l'expressivité limitée, sur lesquels certains problèmes deviennent décidables, éventuellement efficacement. À l'inverse, en sémantique et lambda-calcul, on s'intéresse aux structures logiques et compositionnelles derrière des langages structurés et souvent Turing-complets.

Ces 20 dernières années, plusieurs travaux ont initié une convergence entre ces deux communautés. En 2006, Ong a montré la décidabilité de la logique monadique du second ordre sur les arbres infinis générés par schémas de récursion d'ordre supérieur [63], un problème majeur, et alors ouvert, en vérification. Sa preuve s'appuyait sur un modèle de sémantique des jeux. Si le théorème a depuis été redémontré purement par des outils d'automates, ce travail n'a pas moins engendré une ligne de recherche à l'interface entre automates et sémantique [38, 66], montrant que les outils sémantiques peuvent être mis à profit pour attaquer des problèmes algorithmiques difficiles en vérification où se cache l'ordre supérieur.

Dans une même veine, on peut aussi citer des travaux sur les « automates implicites » : dans des cadres typés, les calculs théoriques tels que le lambda-calcul ne sont pas Turing-complets. Avec un encodage des mots bien choisi, on peut alors les utiliser comme générateurs pour des langages, ou des transductions. On obtient alors des ponts avec la théorie des automates, exhibant en un certain sens des automates déjà présents implicitement en lambda-calcul [62].

Contributeurs et contributrices.

Le GT Scalp.

Références

- 1 Samson Abramsky and Bob Coecke. Categorical quantum mechanics. *Handbook of quantum logic and quantum structures*, 2 :261–325, 2009.
- 2 Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria. Full abstraction for PCF. *Inf. Comput.*, 163(2) :409–470, 2000. doi:10.1006/INCO.2000.2930.
- 3 Beniamino Accattoli, Stéphane Graham-Lengrand, and Delia Kesner. Tight typings and split bounds, fully developed. *J. Funct. Program.*, 30 :e14, 2020. doi:10.1017/S095679682000012X.
- 4 Aurore Alcolei, Pierre Clairambault, and Olivier Laurent. Resource-tracking concurrent games. In Mikolaj Bojanczyk and Alex Simpson, editors, *Foundations of Software Science and Computation Structures - 22nd International Conference, FOSSACS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings*, volume 11425 of *Lecture Notes in Computer Science*, pages 27–44. Springer, 2019. doi:10.1007/978-3-030-17127-8_2.
- 5 Melissa Antonelli, Ugo Dal Lago, and Paolo Pistone. Towards logical foundations for probabilistic computation. *Ann. Pure Appl. Log.*, 175(9) :103341, 2024. doi:10.1016/J.APAL.2023.103341.
- 6 Giorgio Bacci, Radu Mardare, Prakash Panangaden, and Gordon D. Plotkin. Quantitative equational reasoning. In Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva, editors, *Foundations of Probabilistic Programming*, pages 333–360. Cambridge University Press, 2020. doi:10.1017/9781108770750.011.
- 7 Patrick Baillot, Marco Gaboardi, and Virgile Mogbil. A polytime functional language from light linear logic. In Andrew D. Gordon, editor, *Programming Languages and Systems, 19th European Symposium on Programming, ESOP 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6012 of *Lecture Notes in Computer Science*, pages 104–124. Springer, 2010. doi:10.1007/978-3-642-11957-6_7.
- 8 Davide Barbarossa and Giulio Manzonetto. Taylor subsumes Scott, Berry, Kahn and Plotkin. *Proc. ACM Program. Lang.*, 4(POPL) :1 :1–1 :23, 2020. doi:10.1145/3371069.
- 9 Jean-Philippe Bernardy, Mathieu Boespflug, Ryan R. Newton, Simon Peyton Jones, and Arnaud Spiwack. Linear Haskell : practical linearity in a higher-order polymorphic language. *Proc. ACM Program. Lang.*, 2(POPL) :5 :1–5 :29, 2018. doi:10.1145/3158093.
- 10 Valentin Blot. A direct computational interpretation of second-order arithmetic via update recursion. In Christel Baier and Dana Fisman, editors, *LICS '22 : 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 62 :1–62 :11. ACM, 2022. doi:10.1145/3531130.3532458.
- 11 Richard Blute, J. Robin B. Cockett, Jean-Simon Pacaud Lemay, and Robert A. G. Seely. Differential categories revisited. *Appl. Categorical Struct.*, 28(2) :171–235, 2020. doi:10.1007/S10485-019-09572-Y.
- 12 Alois Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coefficient calculus. In Zhong Shao, editor, *Programming Languages and Systems - 23rd European Symposium on Programming, ESOP 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, volume 8410 of *Lecture Notes in Computer Science*, pages 351–370. Springer, 2014. doi:10.1007/978-3-642-54833-8_19.
- 13 Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Trans. Comput. Log.*, 10(2) :14 :1–14 :34, 2009. doi:10.1145/1462179.1462186.
- 14 Luís Caires, Frank Pfenning, and Bernardo Toninho. Linear logic propositions as session types. *Math. Struct. Comput. Sci.*, 26(3) :367–423, 2016. doi:10.1017/S0960129514000218.

- 15 Pierre Clairambault and Marc de Visme. Full abstraction for the quantum lambda-calculus. *Proc. ACM Program. Lang.*, 4(POPL) :63 :1–63 :28, 2020. doi:10.1145/3371131.
- 16 J. Robin B. Cockett, Geoff S. H. Cruttwell, Jonathan Gallagher, Jean-Simon Pacaud Lemay, Benjamin MacAdam, Gordon D. Plotkin, and Dorette Pronk. Reverse derivative categories. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13–16, 2020, Barcelona, Spain*, volume 152 of *LIPICs*, pages 18 :1–18 :16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CSL.2020.18.
- 17 Liron Cohen. Computation first : Rebuilding constructivism with effects (invited talk). In *10th International Conference on Formal Structures for Computation and Deduction (FSCD 2025)*, pages 1–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.
- 18 Pierre-Louis Curien, Marcelo P. Fiore, and Guillaume Munch-Maccagnoni. A theory of effects and resources : adjunction models and polarised calculi. In Rastislav Bodík and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 44–56. ACM, 2016. doi:10.1145/2837614.2837652.
- 19 Nicolaas Govert De Bruijn. The mathematical language AUTOMATH, its usage, and some of its extensions. In *Studies in Logic and the Foundations of Mathematics*, volume 133, pages 73–100. Elsevier, 1994.
- 20 Daniel de Carvalho, Michele Pagani, and Lorenzo Tortora de Falco. A semantic measure of the execution time in linear logic. *Theor. Comput. Sci.*, 412(20) :1884–1902, 2011. doi:10.1016/J.TCS.2010.12.017.
- 21 Thomas Ehrhard, Farzad Jafarrahmani, and Alexis Saurin. On the denotation of circular and non-wellfounded proofs in linear logic with fixed points. In *40th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2025, Singapore, June 23–26, 2025*, pages 84–97. IEEE, 2025. doi:10.1109/LICS65433.2025.00014.
- 22 Thomas Ehrhard, Michele Pagani, and Christine Tasson. Full abstraction for probabilistic PCF. *J. ACM*, 65(4) :23 :1–23 :44, 2018. doi:10.1145/3164540.
- 23 Thomas Ehrhard, Michele Pagani, and Christine Tasson. Measurable cones and stable, measurable functions : a model for probabilistic higher-order programming. *Proc. ACM Program. Lang.*, 2(POPL) :59 :1–59 :28, 2018. doi:10.1145/3158147.
- 24 Thomas Ehrhard and Laurent Regnier. The differential lambda-calculus. *Theor. Comput. Sci.*, 309(1-3) :1–41, 2003. doi:10.1016/S0304-3975(03)00392-X.
- 25 Claudia Faggian. Probabilistic rewriting and asymptotic behaviour : on termination and unique normal forms. *Log. Methods Comput. Sci.*, 18(2), 2022. doi:10.46298/LMCS-18(2:5)2022.
- 26 Laura Fontanella, Guillaume Geoffroy, and Richard Matthews. Realizability models for large cardinals. In Aniello Murano and Alexandra Silva, editors, *32nd EACSL Annual Conference on Computer Science Logic, CSL 2024, February 19–23, 2024, Naples, Italy*, volume 288 of *LIPICs*, pages 28 :1–28 :18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.CSL.2024.28.
- 27 Guillaume Geoffroy. Classical realizability as a classifier for nondeterminism. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09–12, 2018*, pages 462–471. ACM, 2018. doi:10.1145/3209108.3209140.
- 28 Dan R. Ghica. Slot games : a quantitative model of computation. In Jens Palsberg and Martín Abadi, editors, *Proceedings of the 32nd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2005, Long Beach, California, USA, January 12–14, 2005*, pages 85–97. ACM, 2005. doi:10.1145/1040305.1040313.

- 29 Dan R. Ghica. The far side of the cube. *CoRR*, abs/1908.04291, 2019. URL : <http://arxiv.org/abs/1908.04291>, arXiv:1908.04291.
- 30 Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50 :1–102, 1987. doi:10.1016/0304-3975(87)90045-4.
- 31 Jean-Yves Girard. Geometry of interaction 1 : Interpretation of system F. In *Studies in Logic and the Foundations of Mathematics*, volume 127, pages 221–260. Elsevier, 1989.
- 32 Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O’Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. A machine-checked proof of the odd order theorem. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 163–179. Springer, 2013. doi:10.1007/978-3-642-39634-2_14.
- 33 Georges Gonthier et al. Formal proof—the four-color theorem. *Notices of the AMS*, 55(11) :1382–1393, 2008.
- 34 Andrew D Gordon, Thomas A Henzinger, Aditya V Nori, and Sriram K Rajamani. Probabilistic programming. In *Future of software engineering proceedings*, pages 167–181. Association for Computing Machinery, 2014.
- 35 Jean Goubault-Larrecq. Full abstraction for non-deterministic and probabilistic extensions of PCF I : the angelic cases. *J. Log. Algebraic Methods Program.*, 84(1) :155–184, 2015. doi:10.1016/j.jlamp.2014.09.003.
- 36 Jean Goubault-Larrecq. A probabilistic and non-deterministic call-by-push-value language. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785809.
- 37 Jean Goubault-Larrecq, Xiaodong Jia, and Clément Théron. A domain-theoretic approach to statistical programming languages. *J. ACM*, 70(5) :35 :1–35 :63, 2023. doi:10.1145/3611660.
- 38 Charles Grellois and Paul-André Melliès. Relational semantics of linear logic and higher-order model checking. In Stephan Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, volume 41 of *LIPICs*, pages 260–276. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICs.CSL.2015.260.
- 39 Timothy Griffin. A formulae-as-types notion of control. In Frances E. Allen, editor, *Conference Record of the Seventeenth Annual ACM Symposium on Principles of Programming Languages, San Francisco, California, USA, January 1990*, pages 47–58. ACM Press, 1990. doi:10.1145/96709.96714.
- 40 Rémi Di Guardia. *Identity of Proofs and Formulas using Proof-Nets in Multiplicative-Additive Linear Logic. (Identité des preuves et formules par l’usage des réseaux de preuve en logique linéaire multiplicative-additive)*. PhD thesis, École normale supérieure de Lyon, France, 2024. URL : <https://tel.archives-ouvertes.fr/tel-04830060>.
- 41 Alessio Guglielmi. Deep inference. In *All About Proofs, Proofs for All*. College Publications, 2015.
- 42 Thomas Hales, Mark Adams, Gertrud Bauer, Tat Dat Dang, John Harrison, Le Truong Hoang, Cezary Kaliszyk, Victor Magron, Sean McLaughlin, Tat Thang Nguyen, et al. A formal proof of the Kepler conjecture. In *Forum of mathematics, Pi*, volume 5, page e2. Cambridge University Press, 2017.
- 43 Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. A convenient category for higher-order probability theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017. doi:10.1109/LICS.2017.8005137.

- 44 Martin Hofmann and Thomas Streicher. The groupoid model refutes uniqueness of identity proofs. In *Proceedings of the Ninth Annual Symposium on Logic in Computer Science (LICS '94), Paris, France, July 4-7, 1994*, pages 208–212. IEEE Computer Society, 1994. doi:10.1109/LICS.1994.316071.
- 45 Naohiko Hoshino, Koko Muroya, and Ichiro Hasuo. Memoryful geometry of interaction : from coalgebraic components to algebraic effects. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 52 :1–52 :10. ACM, 2014. doi:10.1145/2603088.2603124.
- 46 William A Howard et al. The formulae-as-types notion of construction. *To HB Curry : essays on combinatory logic, lambda calculus and formalism*, 44 :479–490, 1980.
- 47 J. M. E. Hyland and C.-H. Luke Ong. On full abstraction for PCF : I, II, and III. *Inf. Comput.*, 163(2) :285–408, 2000. doi:10.1006/INCO.2000.2917.
- 48 Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Completeness of the ZX-calculus. *Log. Methods Comput. Sci.*, 16(2), 2020. doi:10.23638/LMCS-16(2:11)2020.
- 49 Xiaodong Jia, Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhev. Commutative monads for probabilistic programming languages. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–14. IEEE, 2021. doi:10.1109/LICS52264.2021.9470611.
- 50 Achim Jung and Regina Tix. The troublesome probabilistic powerdomain. In Abbas Edalat, Achim Jung, Klaus Keimel, and Marta Z. Kwiatkowska, editors, *Third Workshop on Computation and Approximation, COMPROX 1997, Birmingham, UK, September 11-13, 1997*, volume 13 of *Electronic Notes in Theoretical Computer Science*, pages 70–91. Elsevier, 1997. doi:10.1016/S1571-0661(05)80216-6.
- 51 Ralf Jung, Jacques-Henri Jourdan, Robbert Krebbers, and Derek Dreyer. Rustbelt : securing the foundations of the Rust programming language. *Proc. ACM Program. Lang.*, 2(POPL) :66 :1–66 :34, 2018. doi:10.1145/3158154.
- 52 Marie Morgane Kerjean and Pierre-Marie Pédrot. δ is for dialectica. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 48 :1–48 :13. ACM, 2024. doi:10.1145/3661814.3662106.
- 53 Robbert Krebbers, Amin Timany, and Lars Birkedal. Interactive proofs in higher-order concurrent separation logic. In Giuseppe Castagna and Andrew D. Gordon, editors, *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 205–217. ACM, 2017. doi:10.1145/3009837.3009855.
- 54 Jean-Louis Krivine. Dependent choice, 'quote' and the clock. *Theor. Comput. Sci.*, 308(1-3) :259–276, 2003. doi:10.1016/S0304-3975(02)00776-4.
- 55 Jean-Louis Krivine. A program for the full axiom of choice. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/LMCS-17(3:21)2021.
- 56 Ugo Dal Lago and Marco Gaboardi. Linear dependent types and relative completeness. *Log. Methods Comput. Sci.*, 8(4), 2011. doi:10.2168/LMCS-8(4:11)2012.
- 57 Jim Laird, Giulio Manzonetto, Guy McCusker, and Michele Pagani. Weighted relational models of typed lambda-calculi. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 301–310. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.36.
- 58 Xavier Leroy. Formal verification of a realistic compiler. *Communications of the ACM*, 52(7) :107–115, 2009.

- 59 Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 1998.
- 60 Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.
- 61 Damiano Mazza and Michele Pagani. Automatic differentiation in PCF. *Proc. ACM Program. Lang.*, 5(POPL) :1–27, 2021. doi:10.1145/3434309.
- 62 Lê Thành Dung Nguyễn and Cécilia Pradic. Implicit automata in typed λ -calculi I : aperiodicity in a non-commutative logic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 135 :1–135 :20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.135.
- 63 C.-H. Luke Ong. On model-checking trees generated by higher-order recursion schemes. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 81–90. IEEE Computer Society, 2006. doi:10.1109/LICS.2006.38.
- 64 Michele Pagani, Peter Selinger, and Benoît Valiron. Applying quantitative semantics to higher-order quantum computing. In Suresh Jagannathan and Peter Sewell, editors, *The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 647–658. ACM, 2014. doi:10.1145/2535838.2535879.
- 65 The Univalent Foundations Program. Homotopy type theory : Univalent foundations of mathematics. *arXiv preprint arXiv :1308.0729*, 2013.
- 66 Sylvain Salvati and Igor Walukiewicz. Using models to model-check recursive schemes. In Masahito Hasegawa, editor, *Typed Lambda Calculi and Applications, 11th International Conference, TLCA 2013, Eindhoven, The Netherlands, June 26-28, 2013. Proceedings*, volume 7941 of *Lecture Notes in Computer Science*, pages 189–204. Springer, 2013. doi:10.1007/978-3-642-38946-7\15.
- 67 Dana Scott and Christopher Strachey. *Toward a mathematical semantics for computer languages*, volume 1. Oxford University Computing Laboratory, Programming Research Group Oxford, 1971.
- 68 Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010.
- 69 Christine Tasson. *Sémantiques et syntaxes vectorielles de la logique linéaire (Vectorial Semantics and Syntax of Linear Logic)*. PhD thesis, Paris Diderot University, France, 2009. URL : <https://tel.archives-ouvertes.fr/tel-00440752>.
- 70 Alwen Tiu. A system of interaction and structure II : The need for deep inference. *Logical Methods in Computer Science*, 2, 2006.
- 71 Vladimir Voevodsky. Univalent foundations. In *Mathematisches Forschungsinstitut Oberwolfach Mini-workshop : the Homotopy Interpretation of Constructive Type Theory*, pages 7–10, 2011. Organised by Steve Awodey, Richard Garner, Per Martin-Löf and Vladimir Voevodsky. Report number 11.
- 72 Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. Unimath — a computer-checked library of univalent mathematics. Available at <http://unimath.org>. doi:10.5281/zenodo.10849216.

Les vingt ans du GdR IFM, vus du GT « Logique, Homotopie, Catégories »

1 Introduction

Le groupe de travail *LHC* s'intéresse aux produits de la collision entre trois grandes particules de l'informatique mathématique moderne : la logique, la théorie de l'homotopie et les catégories. La *logique* fournit une syntaxe et des règles précises pour les énoncés mathématiques mais aussi, au travers de la correspondance de Curry-Howard [32, 43, 49], des programmes typés et de leur exécution. La théorie de l'*homotopie*, qui est une branche de la topologie algébrique [26] initiée par les travaux de Poincaré [46], permet de donner une sémantique à la logique qui généralise les modèles ensemblistes, en apportant une grande richesse supplémentaire : elle est fondée sur la volonté de caractériser des invariants des espaces topologiques à déformation continue – ou homotopie – près. Enfin, la théorie des *catégories* [39] propose les structures algébriques qui permettent d'étudier de façon abstraite les modèles de la logique et leurs propriétés, voire de les caractériser [38, 41]. Les relations entre ces trois disciplines ont été fortement renforcées ces dernières années par la mise au jour de liens ténus dont l'émergence est le produit d'une longue maturation scientifique, et qui ont donné lieu à des évolutions profondes que ce soit dans l'étude sémantique de la logique ou dans les implémentations des assistants de preuves modernes.

Le GT LHC a naturellement de nombreuses thématiques et outils communs avec le GT Scalp. Il s'en distingue par une préoccupation particulière pour les structures mathématiques de dimension supérieure et l'établissement de connexions fortes avec la topologie algébrique.

2 Logique et homotopie

2.1 Théorie homotopique des types

Une révolution majeure récente est l'introduction, à partir de 2006, de la *théorie homotopique des types* ou *HoTT* par Voevodsky [53]. Alors même que la notion d'égalité est fondamentale en mathématiques, les règles associées en théorie des types intuitionniste [40] sont longtemps restées sujet à débat. Par exemple, faut-il accepter le principe d'extensionnalité (deux fonctions sont égales lorsqu'elles donnent les mêmes valeurs sur toutes leurs entrées)? Ou encore, toutes les preuves d'égalité sont-elles elles-mêmes égales? Sur ce dernier point, les travaux de Hofmann et Streicher ont montré dans les années 90 que ce n'était pas nécessairement le cas en construisant un modèle de la théorie des types dans les groupoïdes qui ne valide pas ce principe [30]. Ces travaux ont ouvert la voie à une interprétation constructive de l'égalité qui a donné lieu à leur interprétation homotopique : on peut penser aux types comme à des espaces et aux preuves d'égalité comme à des chemins dans ces espaces (les espaces peuvent aussi être considérés comme des ∞ -groupoïdes faibles, et ce modèle généralise ainsi celui des groupoïdes). De plus, si l'on ajoute un unique axiome appelé *univalence*, toutes les constructions que l'on peut faire sur les types sont nécessairement invariantes par homotopie, et le principe d'extensionnalité devient démontrable. Cette intuition géométrique, validée par le modèle de Voevodsky dans les ensembles simpliciaux [33] et depuis généralisée aux ∞ -topos [48] a motivé de nombreux travaux.

D'une part, la théorie homotopique des types a permis de prouver formellement, et de façon synthétique et invariante par homotopie, un grand nombre de résultats en topologie

algébrique, comme le calcul de groupe d'homotopie des sphères [53], le théorème de Blakers-Massey [2], ou encore l'existence de désuspensions de groupes [6]. D'autre part, elle a motivé le développement d'extensions d'assistants de preuves comme Rocq ou Agda, en y ajoutant les types inductifs supérieurs qui permettent de définir des constructions et des types intéressants d'un point de vue géométrique (le joint, les sphères, etc.), ainsi que de nouveaux principes permettant de calculer en présence de l'univalence tels que les variantes cubiques des assistants de preuve [9, 54].

2.2 Topologie algébrique dirigée

Un autre lien fort entre calcul et topologie est donné par les modèles topologiques dirigés de la concurrence [15]. Là encore, on voit un programme comme un espace : ici, les points correspondent aux états du programme et les chemins aux exécutions du programme. L'une des difficultés majeures dans ce cadre provient du fait qu'il faut prendre en compte la direction du temps lors de l'exécution, ce qui incite à considérer des variantes « dirigées » des espaces topologiques. Des invariants utiles classiques ont pu être généralisés à ce cadre, tels que l'homologie [12, 13], donnant ainsi accès à de nouveaux outils pour l'étude des programmes concurrents, qui présentent des difficultés qui leur sont propres (on veut par exemple s'assurer de l'absence de situations d'interblocage entre processus).

3 Logique et catégories

3.1 Modèles catégoriques

Les travaux de Lambek [38] ont révélé une correspondance parfaite entre les modèles dénotationnels de certaines logiques et certaines structures catégoriques (par exemple, les catégories cartésiennes fermées pour les types simples), qu'on a cherché à étendre à d'autres cadres logiques. En particulier, l'étude des modèles de la logique linéaire s'est progressivement unifiée [41] puis a été étendue aux variantes telles que la logique linéaire différentielle [14]. Des généralisations catégoriques de modèles traditionnels de la logique linéaire, comme les espèces de structures généralisées [18] qui peuvent être vues comme une « catégorification » du modèle relationnel, ou encore des raffinements des modèles de jeux [42], motivent aujourd'hui l'introduction d'axiomatisations de modèles dans les catégories supérieures comme les bicatégories [17] ou même les ∞ -catégories [25].

3.2 Sémantique opérationnelle structurelle

Au-delà des modèles dénotationnels, qui donnent une interprétation modulaire des preuves ou programmes, il est également légitime de chercher à mathématiser la notion même de système logique ou de langage de programmation, c'est-à-dire les règles d'inférence et les relations de réduction définies par des règles de réécriture. Cette approche structurelle de la sémantique opérationnelle initiée par Plotkin et Turi [52] peut être comprise et présentée de façon catégorique, en mobilisant des notions algébriques et de logique catégorique [50, 19, 20]. Cela permet de rendre compte abstraitement de notions opérationnelles (comme celle de bisimilarité) et d'obtenir des résultats (par exemple, le fait que la bisimilarité est une congruence) s'appliquant uniformément à des familles de systèmes logiques et de langages de programmation, plutôt qu'à des exemples particuliers. Un succès notable de ces dernières années est l'extension de ce cadre aux syntaxes d'ordre supérieur, comme celle du λ -calcul, qui mobilisent une opération de substitution sans capture de variables [1, 29, 27, 28].

3.3 Algébrisation de la réalisabilité classique

La réalisabilité est une méthodologie issue de la théorie de la démonstration qui formalise l'idée que les formules logiques spécifient les comportements de programmes. Ce cadre est très bien compris catégoriquement dans le cas intuitionniste avec la notion de tripos [45]. La réalisabilité classique [34], programme initié par Krivine au début des années 2000, a eu un retentissement important, entre autres parce qu'elle permet de révéler le contenu calculatoire d'axiomes de l'arithmétique du second ordre (la logique de l'analyse), voire de la théorie des ensembles [36]. Dans sa version initiale, la réalisabilité classique était présentée de manière très syntaxique, les formules étant réalisées par les termes d'un λ -calcul étendu avec des primitives ad-hoc. Depuis le début des années 2010, des efforts particuliers ont porté sur des présentations algébriques [35, 51, 44, 10], ce qui a été déterminant pour mobiliser la réalisabilité classique comme une généralisation du forcing de Cohen (le forcing en étant en quelque sorte la version commutative), mais aussi pour réintégrer la réalisabilité classique dans l'écosystème logico-catégorique usuel.

3.4 Logique pour les catégories supérieures

Les catégories supérieures faibles sont ardues à définir : leur structure est donnée par des familles d'opérations qui satisfont des lois de cohérence dont témoignent des cellules de dimension supérieure, qui elles-mêmes doivent satisfaire des lois de cohérence à cellules supérieures près, etc. Afin de pouvoir raisonner dans ce cadre, des outils informatiques ont été proposés permettant de s'assurer de la validité des constructions réalisées. On a ainsi vu l'émergence d'outils graphiques comme Globular [4] ou son successeur homotopy.io [11], ou des assistants de preuve fondés sur des théories des types dédiées comme CaTT [5, 16].

4 Homotopie et catégories

4.1 Réécriture de dimension supérieure

Les approches modernes de la théorie de l'homotopie se font au travers de l'approche abstraite fournie par les catégories de modèles [31], qui permettent de travailler aussi bien avec les espaces topologiques qu'avec des structures de nature a priori beaucoup plus algébriques comme les ensembles simpliciaux. En particulier, on peut définir sur la catégorie des ω -catégories une structure de catégorie de modèle dite « folk » [3, 37]. La raison pour laquelle celle-ci intéresse les informaticiens est que les objets « libres » (les objets cofibrants de la structure de modèle) correspondent aux *polygraphes*, qui ont été introduits comme un cadre généralisant la réécriture en dimension supérieure [3, 7] : on ne s'intéresse plus seulement aux chemins de réécriture, mais aussi aux chemins de réécriture entre chemins de réécriture, etc. Ces systèmes permettent d'obtenir des résultats de cohérence sur les structures algébriques en utilisant les outils traditionnels en informatique comme la complétion de Knuth-Bendix [3, 21, 24], ce qui peut s'interpréter de façon homotopique comme le calcul de résolutions.

5 Logique, homotopie et catégories

5.1 Vers une théorie des types dirigée

Un champ de recherche très actif qui mêle la plupart des points de vue mentionnés ci-dessus est la recherche d'une théorie homotopique des types dirigée. En théorie homotopique

des types, les types sont interprétés par des espaces, c'est-à-dire, en suivant l'hypothèse d'homotopie de Grothendieck [23], par des ∞ -groupoïdes : ces derniers sont des catégories supérieures faibles dans lesquelles tous les morphismes sont inversibles. Il est alors tentant d'essayer de définir une variante dirigée de la théorie des types qui correspondrait aux ∞ -catégories (où l'on ne suppose plus les morphismes inversibles). Des premiers succès ont été obtenus dans ce domaine pour les $(\infty, 1)$ -catégories (où seules les 1-cellules ne sont pas supposées réversibles) par l'introduction de la théorie des types simpliciaux [47, 22] ainsi que d'autres approches axiomatiques [8].

Contributeurs

Samuel Mimram et Lionel Vaux.

Références

- 1 Benedikt Ahrens, André Hirschowitz, Ambroise Lafont, and Marco Maggesi. Reduction monads and their signatures. *Proc. ACM Program. Lang.*, 4(POPL) :31 :1–31 :29, 2020.
- 2 Mathieu Anel, Georg Biedermann, Eric Finster, and André Joyal. A generalized Blakers-Massey theorem. *Journal of Topology*, 13(4) :1521–1553, 2020.
- 3 Dimitri Ara, Albert Burroni, Yves Guiraud, Philippe Malbos, François Métayer, and Samuel Mimram. *Polygraphs : From Rewriting to Higher Categories*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2025.
- 4 Krzysztof Bar, Aleks Kissinger, and Jamie Vicary. Globular : an online proof assistant for higher-dimensional rewriting. *Logical Methods in Computer Science*, 14, 2018.
- 5 Thibaut Benjamin, Eric Finster, and Samuel Mimram. Globular weak ω -categories as models of a type theory. *Higher Structures*, 8(2) :1–69, 2024.
- 6 Marc Bezem, Ulrik Buchholtz, Pierre Cagne, Bjørn Ian Dundas, and Daniel R. Grayson. Symmetry. URL : <https://github.com/UniMath/SymmetryBook>.
- 7 Albert Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical computer science*, 115(1) :43–62, 1993.
- 8 Denis-Charles Cisinski, Bastiaan Cnossen, Kim Nguyen, and Tashi Walde. Synthetic category theory, 2025. Book in progress. URL : <https://drive.google.com/file/d/11Kaq7watGG13xvjw9qHjm6SDPFJ2-0o/view>.
- 9 Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory : A constructive interpretation of the univalence axiom. In *21st International Conference on Types for Proofs and Programs (TYPES 2015)*, pages 5–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.
- 10 Liron Cohen, Étienne Miquey, and Ross Tate. Evidenced frames : A unifying framework broadening realizability models. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021.
- 11 Nathan Corbyn, Lukas Heidemann, Nick Hu, Chiara Sarti, Calin Tataru, and Jamie Vicary. homotopy.io : a proof assistant for finitely-presented globular n -categories. Preprint, 2024. [arXiv:2402.13179](https://arxiv.org/abs/2402.13179).
- 12 Jérémy Dubut, Eric Goubault, and Jean Goubault-Larrecq. Natural homology. In *International Colloquium on Automata, Languages, and Programming*, pages 171–183. Springer, 2015.
- 13 Jérémy Dubut, Eric Goubault, and Jean Goubault-Larrecq. Directed homology theories and eilenberg-steenrod axioms. *Applied Categorical Structures*, 25(5) :775–807, 2017.
- 14 Thomas Ehrhard. An introduction to differential linear logic : proof-nets, models and antiderivatives. *Mathematical Structures in Computer Science*, 28(7) :995–1060, 2018.

- 15 Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, and Martin Raussen. *Directed algebraic topology and concurrency*, volume 138. Springer, 2016.
- 16 Eric Finster and Samuel Mimram. A type-theoretical definition of weak ω -categories. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017.
- 17 Marcelo Fiore, Nicola Gambino, and Martin Hyland. Monoidal bicategories, differential linear logic, and analytic functors. Preprint, 2024. [arXiv:2405.05774](https://arxiv.org/abs/2405.05774).
- 18 Marcelo Fiore, Nicola Gambino, Martin Hyland, and Glynn Winskel. The cartesian closed bicategory of generalised species of structures. *Journal of the London Mathematical Society*, 77(1) :203–220, 2008.
- 19 Marcelo Fiore and Sam Staton. A congruence rule format for name-passing process calculi. *Information and Computation*, 207(2) :209–236, 2009. Special issue on Structural Operational Semantics (SOS).
- 20 M. Fiore and D. Turi. Semantics of name and value passing. In *Proceedings 16th Annual IEEE Symposium on Logic in Computer Science*, pages 93–104, 2001.
- 21 Simon Forest and Samuel Mimram. Rewriting in Gray categories with applications to coherence. *Mathematical Structures in Computer Science*, 32(5) :574–647, 2022.
- 22 Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. The Yoneda embedding in simplicial type theory. In *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 127–142, 2025.
- 23 Alexander Grothendieck. Pursuing stacks. Letter to Daniel Quillen, 1983. [arXiv:2111.01000](https://arxiv.org/abs/2111.01000).
- 24 Yves Guiraud and Philippe Malbos. Coherence in monoidal track categories. *Mathematical Structures in Computer Science*, 22(6) :931–969, 2012.
- 25 Eliès Harington and Samuel Mimram. ∞ -categorical models of linear logic. In *10th International Conference on Formal Structures for Computation and Deduction (FSCD)*, volume 337 of *LIPICs*, page 23 :1–23 :20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025.
- 26 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- 27 André Hirschowitz, Tom Hirschowitz, and Ambroise Lafont. Modules over monads and operational semantics (expanded version). *Logical Methods in Computer Science*, 18(3), 2022.
- 28 André Hirschowitz, Tom Hirschowitz, Ambroise Lafont, and Marco Maggesi. Variable binding and substitution for (nameless) dummies. *Logical Methods in Computer Science*, 20(1), 2024.
- 29 Tom Hirschowitz and Ambroise Lafont. A categorical framework for congruence of applicative bisimilarity in higher-order languages. *Logical Methods in Computer Science*, 18(3), 2022.
- 30 Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. *Twenty-five years of constructive type theory (Venice, 1995)*, 36 :83–111, 1998.
- 31 Mark Hovey. *Model categories*, volume 63 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2007.
- 32 William A Howard. The formulae-as-types notion of construction. *To HB Curry : essays on combinatory logic, lambda calculus and formalism*, 44 :479–490, 1980.
- 33 Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The simplicial model of univalent foundations (after Voevodsky). *Journal of the European Mathematical Society*, 23(6) :2071–2126, 2021.
- 34 Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27 :197–229, 2009.
- 35 Jean-Louis Krivine. Realizability algebras : a program to well order R. *Logical Methods in Computer Science*, 7(3), 2011.
- 36 Jean-Louis Krivine. Realizability algebras II : new models of ZF + DC. *Logical Methods in Computer Science*, 8(1), 2012.

- 37 Yves Lafont, François Métayer, and Krzysztof Worytkiewicz. A folk model structure on omega-cat. *Advances in Mathematics*, 224(3) :1183–1231, 2010.
- 38 Joachim Lambek and Philip J Scott. *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press, 1988.
- 39 Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer, 1998.
- 40 Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.
- 41 Paul-André Mellies. Categorical semantics of linear logic. *Panoramas et synthèses*, 27 :15–215, 2009.
- 42 Paul-André Mellies. Template games and differential linear logic. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE, 2019.
- 43 Samuel Mimram. *Program = Proof*. Independently published, 2020. URL : <http://pp.mimram.fr>.
- 44 Alexandre Miquel. Implicative algebras : a new foundation for realizability and forcing. *Mathematical Structures in Computer Science*, 30(5) :458–510, 2020.
- 45 Andrew M. Pitts. Tripos theory in retrospect. *Mathematical Structures in Computer Science*, 12(3) :265–279, 2002.
- 46 Henri Poincaré. *Analysis situs*. Gauthier-Villars Paris, France, 1895.
- 47 Emily Riehl and Michael Shulman. A type theory for synthetic $(\infty, 1)$ -categories. *Higher Structures*, 1(1) :147–224, 2017.
- 48 Michael Shulman. All $(\infty, 1)$ -toposes have strict univalent universes. Preprint, 2019. [arXiv:1904.07004](https://arxiv.org/abs/1904.07004).
- 49 Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard isomorphism*, volume 149. Elsevier, 2006.
- 50 Sam Staton. General Structural Operational Semantics through Categorical Logic. In *Proceedings of the 23rd Annual IEEE Symposium on Logic in Computer Science*, pages 166–177. IEEE Computer Society, 2008.
- 51 Thomas Streicher. Krivine’s classical realisability from a categorical perspective. *Mathematical Structures in Computer Science*, 23(6) :1234–1256, 2013.
- 52 Daniele Turi and Gordon D. Plotkin. Towards a Mathematical Operational Semantics. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science*, pages 280–291. IEEE Computer Society, 1997.
- 53 The Univalent Foundations Program. *Homotopy Type Theory : Univalent Foundations of Mathematics*. <https://homotopytypetheory.org/book>, Institute for Advanced Study, 2013.
- 54 Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda : a dependently typed programming language with univalence and higher inductive types. *Proceedings of the ACM on Programming Languages*, 3(ICFP) :1–29, 2019.

Les vingt ans du GdR IFM, vus du GT « Biologie systémique symbolique »

1 Motivation générale

La biologie des systèmes est un domaine de recherche intrinsèquement multi-disciplinaire portant sur la compréhension des objets biologiques par leur modélisation en tant que systèmes, composés de plusieurs entités qui interagissent entre elles. Les objets biologiques étudiés sont à plusieurs échelles : cellules, organes, écosystèmes ; et les composés de différentes natures : protéines, gènes, métabolites, membranes, etc. La modélisation des systèmes biologiques fait appel à différentes disciplines, incluant l'informatique, les mathématiques, la physique, la biologie et la médecine.

Les méthodes formelles apportent des contributions importantes sur les méthodes informatiques et mathématiques facilitant la modélisation, l'analyse et la compréhension des systèmes biologiques dynamiques complexes. Ces développements méthodologiques particuliers, souvent utilisés en complément des méthodes de modélisation et d'analyse traditionnelles, sont motivés par le constat que les systèmes biologiques diffèrent des systèmes physiques par plusieurs aspects fondamentaux. En particulier, la modélisation, la spécification, le contrôle et la vérification de modèles qualitatifs, ainsi que l'étude de leurs invariants pour faire émerger des propriétés robustes y jouent un rôle central. Ces modèles incluent les réseaux booléens, les réseaux de réactions, les graphes et leurs règles de réécriture, et les modèles hybrides discret-continu et spatio-temporels.

Ces travaux fondamentaux sur les systèmes dynamiques discrets sont guidés par des applications concrètes en biologie et santé. Par exemple, pour prédire de nouvelles cibles et stratégies thérapeutiques, ou encore de nouvelles combinaisons de mutations génétiques pour modifier la fonction des cellules. En dehors d'une meilleure compréhension du fonctionnement des systèmes biologiques, des contributions plus théoriques sont également attendues. Parmi les principales questions étudiées, nous pouvons citer : En quoi les systèmes biologiques sont-ils formalisables ? Quels mécanismes de transmission d'information sont à l'œuvre en biologie moléculaire et cellulaire ? Qu'est-ce que l'informatique peut apporter à la biologie, au delà des approches de modélisation numérique ?

Les recherches en biologie des systèmes apportent des contributions à différents domaines de l'informatique, comme la combinatoire, la complexité, les modèles de calcul formel, la logique, le raisonnement par contraintes, par interprétation abstraite, par apprentissage, le contrôle des systèmes dynamiques et la simulation. Elles s'accompagnent également du développement de nombreux outils logiciels pour faciliter les applications en biologie.

2 Évolution de la thématique en 20 ans

Il y a 20 ans, les contributions dans le domaine de la biologie systémique symbolique portaient d'un côté sur l'étude d'invariants sur la structure des réseaux vis-à-vis de propriétés dynamiques essentielles pour la biologie, et d'un autre côté sur la formalisation de la dynamique des interactions moléculaires et la vérification de propriétés dynamiques.

L'étude du lien entre la structure d'un réseau était motivée par les conjectures de René Thomas des années 1970-1980, sur la nécessité de mécanismes de rétro-action pour faire émerger des comportements dynamiques complexes comme la différenciation cellulaire (le fait d'avoir plusieurs attracteurs ou comportements limites distincts) et des comportements

oscillatoires stables (comme le cycle circadien, ou le cycle cellulaire). Cela a abouti à des théorèmes fondamentaux sur les systèmes dynamiques [19, 18], en particulier sur la nécessité des cycles de rétro-action *positifs* pour l'existence d'attracteurs multiples.

La formalisation de la dynamique moléculaire est venue de nombreuses contributions inspirées par les recherches en informatique en vérification (*model checking*), algèbres de processus, systèmes de réécritures, et programmation logique [10]. Nous pouvons citer par exemple la vérification symbolique de réseaux biochimiques [6], le formalisme logique des réseaux de régulation [20], et le formalisme Kappa de réécriture de graphes à sites [7].

Ainsi, les questions portaient avant tout sur les outils formels de modélisation, leur adéquation pour représenter des mécanismes essentiels de la biologie, et les propriétés fondamentales qui découlent de ces représentations. Cette direction est toujours d'actualité, avec la problématique d'intégrer plusieurs échelles d'interactions moléculaires, et les dimensions spatiales et compartimentées des systèmes biologiques. Ceci aboutit à des formalismes hybrides, mêlant variables discrètes et continues. Progressivement, des recherches se sont également orientées sur des aspects plus algorithmiques pour rendre praticable la vérification de propriétés dynamiques, par exemple avec le développement de méthodes d'interprétation abstraite et de réduction de modèles [9, 15, 11].

Une évolution majeure dans les problématiques de recherches en biologie systémique symbolique est l'augmentation de l'importance des données dans les processus de recherche. Ceci est dû à la fois à la forte volonté d'appliquer ces méthodes formelles à des applications très concrètes en biologie, et à l'augmentation ininterrompue de la production de données biologiques (données patients, données ADN/ARN et métaboliques, suivis longitudinaux de populations, etc.) et de leur mise à disposition (entrepôts publics, agrégation de bases de données [5, 17]). À l'heure actuelle, de nombreuses recherches utilisent ces données pour créer ou enrichir de nouveaux modèles, ou pour proposer des plans d'expérience permettant d'en produire de nouvelles de façon ciblée [21]. On peut citer en résultats notables la production de modèles sur la différenciation des lymphocytes T [13] ou sur l'activation des cellules hépatiques étoilées [2].

La seconde évolution majeure, qui découle assez naturellement de la première, est l'utilisation croissante de méthodes d'apprentissage machine. En effet, face aux masses importantes de données à disposition, il peut devenir intéressant d'aller au-delà des études statistiques en vue d'automatiser au moins en partie les processus de modélisation. Les approches peuvent être symboliques, car la manipulation d'objets algébriques assure une bonne explicabilité des méthodes et permet plus facilement de construire les structures algébriques utilisées par les membres du GT pour créer des modèles causaux et mécanistes : graphes, automates, réseaux de réactions chimiques, etc. Cela peut se faire à travers l'utilisation d'outils dédiés à la représentation de connaissances [1]. Cependant, les méthodes statistiques (réseaux de neurones, apprentissage profond) sont aussi utilisées de façon croissante pour produire des modèles substituts ("surrogates") ayant une bonne capacité de prédiction et de diagnostic [12].

En plus de cela, les recherches théoriques restent actives et continuent à proposer de nouvelles approches d'analyse et de formalisation des modèles biologiques. On peut par exemple citer la démonstration de la Turing-complétude des modèles continus de réseaux de réactions chimiques et leur compilation en programmes analogique-numériques [8], ou encore la découverte d'un nouveau mode de mise à jour des réseaux booléens permettant de les lier formellement aux modèles quantitatifs et de les analyser à très grande échelle [16].

L'implémentation de telles méthodes théoriques dans des logiciels utilisables est également un sujet d'attention important pour la communauté. On peut citer par exemple les logiciels Biocham [4] et Kappa [3], ou encore le consortium CoLoMoto [14] dont le but est notamment

de fédérer les logiciels produits autour de formats d'échange communs.

3 Perspectives

L'utilisation de données réelles continuera de s'intensifier avec les progrès des technologies expérimentales (multi-omique unicellulaire, transcriptomique spatiale, imagerie haute résolution). La communauté sera amenée à développer des modèles multi-échelles, des méthodes robustes pour gérer l'hétérogénéité biologique, et des approches interactives permettant d'itérer entre données, modèles et expériences.

Pour les années à venir, il est probable que les membres du GT continuent sur cette même évolution consistant à intégrer davantage des données réelles à leurs recherches, et à utiliser davantage de procédés d'apprentissage automatique. Les enjeux incluent de garantir l'interprétabilité des modèles appris, d'intégrer des contraintes biologiques dans les processus d'apprentissage, et de combiner les garanties formelles offertes par les modèles symboliques avec la puissance prédictive des réseaux neuronaux.

Cependant, il est fort à parier que l'aspect théorique des travaux, historiquement très important, reste une des caractéristiques essentielles de cette communauté. Les prochaines années verront la poursuite des travaux sur la complexité, la formalisation des systèmes biologiques, les méthodes de vérification et de contrôle, ainsi que l'étude des mécanismes computationnels qui structurent le vivant. Au sein du GDR IFM, le GT Bioss s'inscrit dans cette continuité, avec des interactions entre informatique, mathématiques et biologie qui nourrissent une recherche rigoureuse et pleinement interdisciplinaire.

Contributeurs et contributrices.

Laurence Calzone, Jérôme Feret, Maxime Folschette, Loïc Paulevé, Sabine Pérès, Misbah Razzaq.

Références

- 1 Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge University Press, 2003.
- 2 Matthieu Bouguéon, Vincent Legagneux, Octave Hazard, Jérémy Bomo, Anne Siegel, Jérôme Feret, and Nathalie Thérêt. A rule-based multiscale model of hepatic stellate cell plasticity : Critical role of the inactivation loop in fibrosis progression. *PLoS Computational Biology*, 20(7) :1–29, 07 2024. URL : <https://doi.org/10.1371/journal.pcbi.1011858>, doi:10.1371/journal.pcbi.1011858.
- 3 Pierre Boutillier, Mutaamba Maasha, Xing Li, Héctor F Medina-Abarca, Jean Krivine, Jérôme Feret, Ioana Cristescu, Angus G Forbes, and Walter Fontana. The kappa platform for rule-based modeling. *Bioinformatics*, 34(13) :i583–i592, 06 2018. URL : <https://doi.org/10.1093/bioinformatics/bty272>, doi:10.1093/bioinformatics/bty272.
- 4 Laurence Calzone, François Fages, and Sylvain Soliman. Biocham : an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14) :1805–1807, 04 2006. URL : <https://doi.org/10.1093/bioinformatics/bt1172>, doi:10.1093/bioinformatics/bt1172.
- 5 Ethan G. Cerami, Benjamin E. Gross, Emek Demir, Igor Rodchenkov, Özgün Babur, Nadia Anwar, Nikolaus Schultz, Gary D. Bader, and Chris Sander. Pathway commons, a web resource for biological pathway data. *Nucleic Acids Research*, 39(suppl_1) :D685–D690, 11 2010. URL : <https://doi.org/10.1093/nar/gkq1039>, doi:10.1093/nar/gkq1039.

- 6 Nathalie Chabrier and François Fages. *Symbolic Model Checking of Biochemical Networks*, page 149–162. Springer Berlin Heidelberg, 2003. URL : http://dx.doi.org/10.1007/3-540-36481-1_13, doi:10.1007/3-540-36481-1_13.
- 7 Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1) :69–110, September 2004. URL : <http://dx.doi.org/10.1016/j.tcs.2004.03.065>, doi:10.1016/j.tcs.2004.03.065.
- 8 François Fages, Guillaume Le Guludec, Olivier Bournez, and Amaury Pouly. Strong turing completeness of continuous chemical reaction networks and compilation of mixed analog-digital programs. In Jérôme Feret and Heinz Koepl, editors, *Computational Methods in Systems Biology*, pages 108–127, Cham, 2017. Springer International Publishing.
- 9 François Fages and Sylvain Soliman. Abstract interpretation and types for systems biology. *Theoretical Computer Science*, 403(1) :52 – 70, 2008. doi:10.1016/j.tcs.2008.04.024.
- 10 François Fages. *Artificial Intelligence in Biological Modelling*, page 265–302. Springer International Publishing, 2020. URL : http://dx.doi.org/10.1007/978-3-030-06170-8_8, doi:10.1007/978-3-030-06170-8_8.
- 11 Jerome Feret, Heinz Koepl, and Tatjana Petrov. Stochastic fragments : A framework for the exact reduction of the stochastic semantics of rule-based models. *International Journal of Software and Informatics*, 7(4) :527 – 604, 2013.
- 12 Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljagic, Thomas Y. Hou, and Max Tegmark. KAN : Kolmogorov–arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL : <https://openreview.net/forum?id=0zo7qJ5vZi>.
- 13 Pedro T. Monteiro, Wassim Abou-Jaoudé, Denis Thieffry, and Claudine Chaouiya. Model checking logical regulatory networks. *IFAC Proceedings Volumes*, 47(2) :170–175, 2014. 12th IFAC International Workshop on Discrete Event Systems (2014). URL : <https://www.sciencedirect.com/science/article/pii/S1474667015373985>, doi:<https://doi.org/10.3182/20140514-3-FR-4046.00135>.
- 14 Aurélien Naldi, Pedro T. Monteiro, Christoph Müssel, the Consortium for Logical Models, Tools, Hans A. Kestler, Denis Thieffry, Ioannis Xenarios, Julio Saez-Rodriguez, Tomas Helikar, and Claudine Chaouiya. Cooperative development of logical modelling standards and tools with colomoto. *Bioinformatics*, 31(7) :1154–1159, 01 2015. URL : <https://doi.org/10.1093/bioinformatics/btv013>, doi:10.1093/bioinformatics/btv013.
- 15 Aurélien Naldi, Elisabeth Remy, Denis Thieffry, and Claudine Chaouiya. A reduction of logical regulatory graphs preserving essential dynamical properties. In Pierpaolo Degano and Roberto Gorrieri, editors, *Computational Methods in Systems Biology*, volume 5688 of *Lecture Notes in Computer Science*, pages 266–280. Springer Berlin / Heidelberg, 2009. doi:10.1007/978-3-642-03845-7_18.
- 16 Loïc Paulevé, Juri Kolčák, Thomas Chatain, and Stefan Haar. Reconciling qualitative, abstract, and scalable modeling of biological networks. *Nature Communications*, 11(4256), 2020. URL : <https://doi.org/10.1038/s41467-020-18112-5>, doi:10.1038/s41467-020-18112-5.
- 17 Alexander R Pico, Thomas Kelder, Martijn P van Iersel, Kristina Hanspers, Bruce R Conklin, and Chris Evelo. Wikipathways : Pathway editing for the people. *PLOS Biology*, 6(7) :1–4, 07 2008. URL : <https://doi.org/10.1371/journal.pbio.0060184>, doi:10.1371/journal.pbio.0060184.
- 18 Élisabeth Remy, Paul Ruet, and Denis Thieffry. Graphic requirements for multistability and attractive cycles in a boolean dynamical framework. *Advances in Applied Mathematics*, 41(3) :335 – 350, 2008. doi:10.1016/j.aam.2007.11.003.
- 19 Adrien Richard and Jean-Paul Comet. Necessary conditions for multistationarity in discrete dynamical systems. *Discrete Applied Mathematics*, 155(18) :2403 – 2413, 2007. doi:10.1016/j.dam.2007.04.019.

- 20 Adrien Richard, Jean-Paul Comet, and Gilles Bernot. *Modern Formal Methods and Applications*, chapter Formal Methods for Modeling Biological Regulatory Networks, pages 83–122. 2006. doi:10.1007/1-4020-4223-X_5.
- 21 Santiago Videla, Julio Saez-Rodriguez, Carito Guziolowski, and Anne Siegel. caspo : a toolbox for automated reasoning on the response of logical signaling networks families. *Bioinformatics*, 33(6) :947–950, 11 2016. URL : <https://doi.org/10.1093/bioinformatics/btw738>, doi: 10.1093/bioinformatics/btw738.