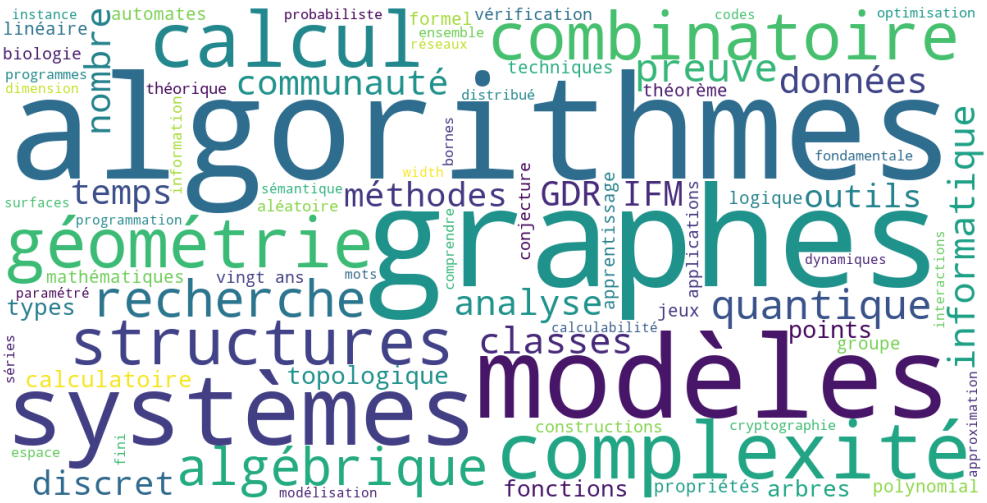
GDR Groupement de recherche
IFM Informatique Fondamentale et ses Mathématiques

Les vingt ans du GdR IFM « Informatique Fondamentale et ses Mathématiques »

17-19 mars 2026

https://20ansgdrifm.sciencesconf.org/panoramas



Edité par

Conseil Scientifique du GdR IFM

Disponible en ligne sur
https://20ansgdrifm.sciencesconf.org/panoramas

Date de publication
17 mars 2026

Le GdR IFM en quelques lignes

Les Groupements de Recherche (GdR) sont un outil du CNRS visant à animer, structurer et promouvoir les recherches scientifiques sur des thématiques spécifiques. Le GdR Informatique Fondamentale et ses Mathématiques (IFM) a été créé le 1er janvier 2024 par l'institut Sciences Informatiques du CNRS pour cinq ans. Il s'inscrit dans la suite directe du GdR Informatique Mathématique (IM) créé en 2006.

Les responsables successifs des GdR IM puis IFM ont été :

— Brigitte Vallée et Christiane Frougny (2006-2009)
— Brigitte Vallée et Arnaud Durand (2010-2013)
— Arnaud Durand et Jean-Michel Muller (2014-2018)
— Jean-Michel Muller et Guillaume Theyssier (2019-2023)
— Pierre Fraigniaud et Guillaume Theyssier (2024-2028)

Depuis sa création, le GdR couvre une très grande part des domaines fondamentaux sur lesquels repose l'informatique, et maintient à ce titre des liens très étroits avec les mathématiques. Il regroupe plus de 2500 scientifiques, majoritairement informaticien · nes ou mathématicien · nes, réparti · es dans plus d'une centaine d'unités en France (laboratoires universitaires, unités mixtes de recherche CNRS, centres de recherches Inria, etc.), voire même à l'étranger.

Les domaines de recherche couverts actuellement par le GdR IFM se structurent en cinq axes thématiques, non disjoints :

Axe 1 : Combinatoire, graphes et systèmes dynamiques

Axe 2 : Calculabilité, complexité, algorithmique et calcul quantique

Axe 3 : Calcul formel, arithmétique et cryptologie

Axe 4 : Programmes, vérification, preuve, automates et logique

Axe 5 : Géométries et image

Les activités du GdR IFM sont structurées en groupes de travail (GT) spécialisés, dont certains communs avec d'autres GdR. La vie scientifique du GdR se déroule au travers des réunions annuelles de ses GT, de ses journées nationales annuelles, de son école dédiée aux jeunes chercheurs et chercheuses, et de diverses actions spécifiques dont des journées transverses, les années thématiques, etc.

La direction du GdR est assurée par ses responsables s'appuyant d'une part sur un comité de direction (ComDir) composé des responsables du GdR, de la présidence du Conseil Scientifique du GdR, et de trois membres nommés, et d'autre part sur le comité exécutif (ComEx) du GdR réunissant l'ensemble des responsables de GT. Le conseil scientifique (CS) est entre autre en charge d'aider la direction du GdR dans la détermination de sa politique scientifique.

Les directions actuelles et passées du GdR s'associent pour profiter de l'anniversaire des 20 ans du GdR afin de remercier vivement l'ensemble des collègues qui, depuis 20 ans, se sont investis dans la vie du GdR. Sans leur investissement au service de notre communauté, rien n'aurait été possible !

Pierre Fraignaud et Guillaume Theyssier

Table des matières

Géométrie(s) et image

Les vingt ans du GdR IFM, vus du Conseil Scientifique

Algorithmes, logique informatique, complexité, graphes, aléa... les grandes questions et les objets au cœur du GdR IFM sont centraux dans les sciences informatiques. Motivée par des problèmes issus de divers domaines de l'informatique ou parfois d'autres disciplines, leur étude fait souvent appel à des outils et méthodes mathématiques spécifiques, qui demandent des développements propres et différents des approches mathématiques classiques. C'est cette combinaison d'un ancrage au cœur de l'informatique et d'une méthodologie mathématique rigoureuse procurant des garanties qui fédère les communautés scientifiques du GdR IFM.

Des sujets de recherche qui ont diffusé bien au-delà de l'informatique fondamentale.

Les sujets abordés au sein du GdR qui, il y a vingt ans, étaient principalement définis comme des points d'articulation entre informatique et mathématique, irriguent désormais pratiquement toutes les disciplines des sciences informatiques : calcul formel en robotique et en optimisation, géométrie et topologie algorithmique en sciences des données, par exemple.

Les frontières entre le GdR IFM et le reste des sciences informatiques se sont déplacées, et les sujets d'intérêt à l'interface avec d'autres communautés (combinatoire des mots et séquençage génétique, calcul quantique, vérification de programmes, imagerie et informatique graphique...) se sont multipliés, comme en témoigne le rattachement d'une partie des GTs du GdR IFM à un deuxième GdR, souvent plus tourné vers un domaine applicatif [1], ainsi que la participation des chercheurs à plusieurs GdR. Au cours des vingt dernières années, l'informatique fondamentale a réalisé des progrès algorithmiques et en complexité marquants, mais elle a aussi trouvé un retentissement important dans l'ensemble de la société à travers, par exemple, la popularisation du mot « algorithme » et la prise de conscience de l'impact des algorithmes dans la « vie quotidienne » et donc de l'importante problématique de leur transparence, de leur qualité et de leur équité.

Une singularité des liens entre le GdR IFM et les autres domaines de l'informatique, ou d'autres disciplines au premier rang desquelles on trouve les mathématiques, mais aussi la biologie ou la physique, est qu'ils ne se réduisent pas à l'apport de motivations nouvelles, mais sont caractérisés par de multiples « allers-retours ». Par exemple, le lien entre théorie des types et homotopie, introduit il y a vingt ans, a stimulé les interactions entre ces thématiques. Ces travaux ont à la fois constitué un fondement logique essentiel pour le développement d'assistants de preuve et ont aussi attiré des mathématiciens du domaine de la topologie algébrique vers la formalisation de leurs preuves avec un regard renouvelé sur celles-ci. Dans un registre différent, le langage Kappa, langage basé sur des règles de réécriture de graphes avec une syntaxe inspirée de la chimie, fut initialement introduit pour décrire des interactions entre protéines, mais il a depuis permis de modéliser et d'étudier bien d'autres phénomènes,

1. Les GTs Bioss (Biologie systémique symbolique) et Seq-BIM (Séquences en Bioinformatique, Informatique et Mathématiques), sont naturellement également affiliés au GdR BIMM – *Bioinformatique Moléculaire : Modélisation et Méthodologie*, le GT Bioss dépendant aussi du GdR RADIA – *Raisonnement, Apprentissage, et Décision en Intelligence Artificielle*. Les GTs GDMM (Géométrie discrète et morphologie mathématique) et MG (Modélisation géométrique) sont aussi rattachés au GdR IG-RV – *Informatique géométrique et graphique, réalité virtuelle et virtualisation*. Le GT C2 (Codage et Cryptographie) dépend également du GdR Sécurité, alors que le GT Arith (Arithmétique des ordinateurs) constitue l'un des groupes du tout récent GdR C4P – *Calcul : Paradigmes, parallélisme, performance, précision*. Le GT IQ (Information quantique), lui, est aussi impliqué dans le GdR *Technologies Quantiques* et le GdR C4P.

en dynamique des populations, en épidémiologie, et a conduit à la formulation de nouvelles hypothèses biologiques, comme l'identification de bio-marqueurs de la fibrose hépatique. Les liens étroits entretenus entre la combinatoire et la physique statistique, ou les systèmes dynamiques et la physique mathématique en sont également une preuve.

Des interactions fortes entre GTs.

Les GTs étant définis autour de méthodes et d'objets fondamentaux en informatique, leur dénomination a relativement peu varié au cours de ces vingt ans. Mais cette stabilité masque des évolutions importantes. On peut mentionner le développement considérable de certaines communautés, visible à partir de l'effectif des GTs correspondants. C'est le cas notamment du GT Informatique quantique et du GT Codage et Cryptographie. Ce développement est d'ailleurs révélateur de nouvelles interactions entre communautés (et entre GTs), symboles d'un enrichissement mutuel entre domaines au fil des années. Ainsi, la résolution de systèmes polynomiaux, sujet de prédilection du GT Calcul Formel, est devenu un outil de cryptanalyse, développé et adapté à ce contexte particulier au sein du GT Codage et Cryptographie. Les phénomènes aléatoires discrets, étudiés historiquement au sein du GT Aléa, intéressent désormais les GTs Complexité et Algorithmique, Graphes et Informatique Quantique, à travers les algorithmes probabilistes. Ces liens étroits apparaissent clairement sur le graphe des interactions entre GTs, défini par le nombre de membres en commun entre deux GTs, visualisable sur MyGDR.

L'objectif de la suite de ce texte est de mettre en avant, de manière non exhaustive, certaines évolutions scientifiques marquantes du GdR IFM, et notamment de montrer comment des progrès significatifs sur certains sujets, au sein même ou à la frontière de nos disciplines, ont considérablement influencé les questions de recherche abordées au cours des dernières années.

1 Automatisation du raisonnement, des démonstrations et de certains calculs : nouveaux outils, nouveaux sujets

Les thématiques du GdR IFM au cours des vingt dernières années ont été durablement marquées par le développement de divers outils d'automatisation des preuves, de certains calculs et du raisonnement, qu'il s'agisse des assistants de preuve, des solveurs (SAT, SMT, MILP...) ou de systèmes de calcul formel, mais aussi des LLMs dont l'essor a eu un retentissement bien au-delà des sciences informatiques. Ces techniques d'automatisation sont devenues centrales dans nos disciplines, à la fois comme outils et comme objets d'étude.

1.1 La multiplication des usages des assistants de preuve et solveurs

L'apparition la plus retentissante des assistants de preuve en informatique fondamentale remonte à la preuve du théorème des quatre couleurs. Mais, depuis les travaux de G. Gonthier et ses collaborateurs en 2005, les capacités des assistants de preuve ont considérablement augmenté, comme l'a montré le développement de CompCert, un compilateur C optimisant opérationnel et formellement prouvé. Cette maturité a transformé les assistants de preuve en outils essentiels en informatique fondamentale. Ainsi, le besoin de garanties sur l'exactitude et la précision des calculs a naturellement conduit des domaines comme le calcul formel et l'arithmétique des ordinateurs à fréquemment utiliser (et faire progresser) les assistants de preuve, et à développer des liens étroits avec les communautés de preuve formelle.

Les assistants de preuve sont aussi un élément important direct ou indirect de la recherche sur la preuve de programmes, sujet qui va au-delà du périmètre du GdR IFM. Il peut s'agir du développement de bibliothèques dédiées, comme le projet Iris pour la preuve de programmes Rust, ou bien d'un usage en back-end d'outils de vérification déductive de programmes, comme pour le système Why3, ou encore pour renforcer l'assurance de correction de résultats d'articles de recherche en théorie de la programmation. Un autre signe de maturité est l'arrivée sur la scène d'un nouvel assistant de preuve Lean « petit cousin » de Rocq qui, tout en s'appuyant sur des principes analogues, a su attirer des communautés nouvelles en particulier chez les mathématiciens.

Plus généralement, le traitement automatique de certaines parties des preuves par des outils « en boîte noire » s'est généralisé dans de nombreux domaines. C'est le cas par exemple de la méthode de déchargement, étape classique dans diverses preuves de théorie des graphes traitée de manière automatique par le biais de la programmation linéaire. On assiste à un mouvement similaire en cryptographie où l'absence et l'optimisation d'attaques différentielles ou linéaires en cryptographie symétrique est analysée grâce à des solveurs MILP (Mixed-Integer Linear Programming), celles d'attaques dites « algébriques » par des outils de résolution de systèmes polynomiaux issus du calcul formel. Dans toutes ces situations, la phase délicate est devenue la recherche d'une modélisation appropriée du problème considéré qui permette un traitement automatique efficace. Le domaine de l'implémentation, matérielle ou logicielle, des protocoles cryptographiques, repose aussi en grande partie sur des outils automatiques dédiés, tels EasyCrypt, CryptoVerif ou Tamarin.

Plus récemment, l'utilisation de LLM comme aide dans la résolution de certains problèmes de combinatoire, tels certains des célèbres problèmes de Erdös, souligne l'apport de l'IA générative pour identifier la littérature pertinente de manière efficace, ou pour aider à localiser des contre-exemples. Un sujet de recherche très actuel est donc l'utilisation des techniques d'apprentissage automatique dans les assistants de preuve pour guider la stratégie utilisée.

1.2 L'apprentissage, source de nouvelles problématiques en informatique fondamentale

L'apprentissage profond est également devenu un outil essentiel dans divers domaines de l'informatique fondamentale, notamment dans des tâches de modélisation. Il offre par exemple une méthode extrêmement efficace en remplacement de modèles statistiques explicites pour réaliser les attaques dites par canaux auxiliaires, qui exploitent l'analyse de traces d'exécution physique (par exemple la consommation de courant ou le rayonnement électromagnétique émis) pour retrouver des quantités secrètes. Une évolution similaire peut être observée dans les travaux de modélisation de systèmes biologiques, avec l'utilisation de modèles dits substituts (surrogate modeling) pour gagner en efficacité et en capacité de prédiction.

Les techniques d'apprentissage fournissent donc des outils précieux dans certains domaines du GdR. Mais elles sont également devenues des objets d'étude, conduisant la communauté d'informatique théorique à s'intéresser aux algorithmes sous-jacents sous l'angle de la complexité, de la fiabilité ou des modèles de calcul, mais aussi de la confidentialité et de la sobriété.

L'informatique fondamentale a pu apporter des réponses à ces questions dans le cas des heuristiques d'optimisation randomisées (algorithmes évolutionnaires, recuit simulé...), sujet sur lequel les outils théoriques développés depuis la fin des années 1990 comme l'analyse de drift, les techniques d'analyse lissée, ont amené une quinzaine d'années plus tard des résultats-clefs sur la complexité de ces algorithmes et ont influencé directement leur conception. Dans le cas de l'apprentissage profond, les outils de la théorie PAC (Probably Approximately Correct),

les notions de complexité de communication, la théorie de l'information apparaissent comme des éléments essentiels dans les nombreux travaux visant à une meilleure compréhension des performances de ces techniques. Ainsi, la question des limites de l'expressivité des réseaux de neurones profonds peut être abordée en revisitant les résultats classiques de théorie de la complexité des circuits puisque les réseaux de neurones peuvent être vus comme une classe particulière de circuits opérant, non sur des entrées booléennes mais sur des réels.

Garantir certaines propriétés, fonctionnelles ou géométriques, en apprentissage profond est aussi un axe de recherche développé au sein du GdR IFM. De nouveaux composants non-linéaires fondés sur des algèbres $(\max, +)$ sont proposés pour remplacer les circuits convolutionnels, afin d'intégrer du traitement d'image non-linéaire. D'autres composants intègrent les opérateurs d'algèbres géométriques, pour rendre les réseaux invariants sous certaines transformations géométriques. De nouveaux algorithmes d'optimisation permettent enfin d'imposer une fonction implicite lipschitzienne en sortie, et rendent donc les traitements géométriques beaucoup plus stables et prédictibles.

Pouvoir expliquer les résultats produits par certains systèmes d'IA est une autre question essentielle pour laquelle les modèles formels comme la logique ou les automates fournissent des outils pertinents. La théorie algorithmique des jeux est également un domaine fondamental pour comprendre l'IA multi-agent.

La sécurité de l'IA est un autre sujet extrêmement vaste, abordé depuis quelques années par la communauté d'informatique fondamentale, couvrant des questions allant de la certification des logiciels d'IA à la protection de la confidentialité des données manipulées via la cryptographie homomorphe — dont la possibilité, démontrée pour la première fois en 2009, a ouvert de nombreuses perspectives et est désormais explorée par un tissu industriel dense dans lequel le GdR est impliqué. Les besoins en apprentissage automatique, et en HPC, et les évolutions matérielles des fabricants de puces ont également des répercussions dans le domaine de l'arithmétique des ordinateurs. L'utilisation de calculs sur des nombres de 16 bits ou moins soulève des questions liées à l'algorithmique à très petite précision et à la maîtrise de cette précision.

2 Le GdR IFM, moteur dans la mise à disposition de nouveaux logiciels

Les membres du GdR IFM ont joué au cours de ces vingt ans un rôle très important dans la conception et la mise à disposition de la communauté scientifique de nombreux logiciels et librairies spécialisées, devenus d'usage courant dans diverses disciplines. Ces logiciels résultent généralement d'un effort collaboratif soutenu, qui bénéficie à de multiples utilisateurs, au sein de la communauté informatique fondamentale et à l'extérieur. Nombre d'entre eux ont d'ailleurs été récompensés par des prix prestigieux [2]. Cette capacité à développer des logiciels de grande ampleur sur le long terme est une spécificité et une grande force mondialement reconnue de la communauté française d'informatique fondamentale.

On pense bien sûr au logiciel Rocq (anciennement Coq) dont le développement a débuté il y a 40 ans. Comme déjà évoqué, ces vingt dernières années ont été marquées par l'appropriation de cet outil par plusieurs communautés qui l'utilisent pour aller plus loin dans leur discipline,

2. ACM Software System award pour Coq/Rocq, Prix Science Ouverte du logiciel libre de recherche du MESR pour MPFR et Pari/GP, également ACM SIGSAM Richard Dimick Jenks Memorial Prize pour Pari/GP, Prix Levchin pour CADO-NFS, Test of Time award du Symposium on Computational Geometry pour CGAL, Software award du Symposium on Geometry Processing pour DGtal...

par des travaux théoriques pour faire évoluer le langage autour de la théorie homotopique des types qui a permis de mieux comprendre la structure des preuves d'égalité en les interprétant comme des chemins transformant un objet en un autre qui lui est égal et qui a de nombreuses répercussions en terme théorique et en terme d'outils.

Il est impossible de citer toutes les autres réalisations logicielles notables impliquant des membres du GdR. Parmi ces travaux, on peut notamment mentionner l'avènement de SageMath, un système de calcul mathématique open source et collaboratif, qui offre une interface de programmation simple en Python et facilite l'accès à de très nombreux composants logiciels autour du calcul numérique et symbolique, comme le système Pari/GP ou la bibliothèque spécialisée efficace MPFR. Des membres du GT Calcul Formel ont activement participé au développement du système et de ses composants (FLINT, fplll, FFLAS-FFPACK, msolve, etc.), ainsi qu'à sa diffusion dans le monde de la recherche et de l'enseignement.

Le logiciel CADO-NFS, seule implémentation disponible au monde de l'intégralité de l'algorithme du crible algébrique pour le calcul du logarithme discret et la factorisation, a permis de battre les records (toujours en vigueur) de taille des entiers pour lesquels ces deux problèmes ont pu être résolus. Mais il est aussi utilisé en boîte noire dans des attaques sur des protocoles cryptographiques ou dans des travaux mathématiques.

En géométrie, la bibliothèque CGAL, dont le développement a été initié à la fin des années 1990, est une contribution majeure de la communauté française de géométrie algorithmique, désormais référence mondiale du domaine, qui ne cesse d'évoluer et possède plus de 200 utilisateurs commerciaux dans le monde, et compte plus de 10 000 téléchargements par an. La bibliothèque DGtal, initiée par la communauté française de géométrie discrète, s'est, elle, imposée comme la bibliothèque de référence dans la communauté académique internationale depuis 10 ans. Ces bibliothèques d'informatique géométrique ont accompagné l'essor du traitement géométrique des données numériques, avec des applications en CFAO, impression 3D, architecture, maillage pour le calcul scientifique, analyse d'images bio-médicales et matériaux. Les recherches autour de l'analyse topologique des données, sujet introduit il y a une vingtaine d'années, ont fourni un nouveau cadre robuste, alternatif aux algorithmes d'apprentissage pour analyser les données. Ces travaux ont également donné lieu à des applications dans de multiples domaines (visualisation des données, sciences des matériaux, et même à l'optimisation du recrutement des joueurs en NBA [3]). Ils se sont notamment matérialisés à travers le logiciel Gudhi.

3 L'informatique fondamentale à l'épreuve du réel ?

Analyser les performances des algorithmes indépendamment de propriétés spécifiques vérifiées par les données auxquelles ils s'appliquent est un objectif historique de l'informatique théorique. Cette approche fournit des garanties universelles, atteste du caractère généraliste d'un algorithme et dispense ses utilisateurs d'avoir à identifier les types d'entrées qui sont pertinents pour leurs applications. Toutefois, cette analyse dans le pire cas ne reflète pas toujours le comportement de ces algorithmes et ces systèmes dans la « vraie vie », car elle fait parfois la part belle à des exemples pathologiques qui ne sont pas pertinents en pratique.

La référence illustrant le mieux cette situation est l'algorithme du simplexe en programmation linéaire dont la complexité dans le pire cas est exponentielle en le nombre de variables, alors qu'en pratique, la croissance de son temps de calcul reste modeste, au point qu'il est plus efficace que la méthode de l'ellipsoïde qui, elle, est pourtant polynomiale dans le pire

3. Cf. article du New-York Times de mars 2012.

cas. Ce fossé entre le pire cas et la pratique a été comblé par Spielman et Teng grâce à l'introduction d'un nouveau cadre d'analyse, l'analyse lissée, montrant que si l'on perturbe une instance quelconque (même pathologique) par un petit bruit aléatoire gaussien, alors on obtient un temps d'exécution polynomial. Autrement dit, les instances du pire cas sont fragiles, et les données du monde réel, qui contiennent toujours de légères imprécisions ou perturbations, conduisent à de bonnes performances.

Ce type d'analyse, introduit pour la première fois il y a 25 ans, a ouvert une perspective nouvelle et fructueuse en informatique théorique. L'idée générale consiste à identifier des propriétés des données du monde réel et à les exploiter pour apporter des garanties rigoureuses sur l'algorithme quand il opère sur des entrées ayant ces propriétés particulières. L'identification des structures pertinentes (parcimonie, symétrie..) a parfois lieu en interaction avec d'autres disciplines, par exemple la robotique ou la cryptographie dans le cas de la résolution de systèmes polynomiaux. Diverses techniques d'analyse de complexité, qui reflètent mieux la pratique, ont donc fleuri au cours des vingt dernières années : la complexité lissée mentionnée précédemment, la complexité paramétrée qui différencie les instances en fonction d'un ou plusieurs paramètres caractéristiques, typiquement la largeur arborescente d'un graphe ou la twin-width, quantité récemment identifiée par des membres du GT Graphes.

Comparer les performances des algorithmes rapides est également devenu un sujet de recherche important, ceux-ci étant désormais confrontés à des données de très grande taille. Dans ce but, la complexité à grain fin fournit une information plus précise sur la complexité des problèmes polynomiaux en reliant leur difficulté de manière à préserver le degré des polynômes en jeu. Ce cadre permet donc de distinguer plusieurs grandes catégories au sein des problèmes polynomiaux. La complexité à grain fin fournit également des bornes inférieures précieuses sur la complexité d'un grand nombre de problèmes (calcul du diamètre d'un graphe pondéré ou non, du nombre chromatique...) en se ramenant à un petit nombre d'hypothèses communément admises. De tels résultats démontrent ainsi l'optimalité de certains algorithmes.

L'essor du Big Data a conduit à l'analyse de données de plus en plus massives et disparates. L'approche standard est de les placer en très grande dimension, puis de leur trouver une cohérence interne en les approchant par des variétés de dimension intermédiaire. Afin de mieux comprendre leurs propriétés, de nombreuses techniques d'analyse topologique ou géométrique des données efficaces ont dû être développées : réduction de dimension, construction de complexes « nerf », analyse robuste de la topologie. Ces travaux sont aussi utilisés pour caractériser les espaces latents de certains réseaux de neurones profonds, afin d'expliquer le rôle des paramètres essentiels.

Dans ce même souci de mieux prendre en compte les contraintes du monde réel dans la conception et l'analyse des algorithmes, divers modèles de calcul plus pertinents dans certains contextes ont été définis, par exemple quand les données arrivent de façon séquentielle et doivent être traitées à la volée, sans pouvoir être stockées. Les modèles distribués ont également été sources de nombreuses avancées dans le domaine de la complexité et de l'algorithmique.

La prise en compte de nouvelles structures de données, mieux adaptées au monde réel, a aussi fortement marqué la théorie des bases de données : elle a été amenée à définir et à manipuler de nouveaux modèles de données sous forme d'arbres (comme dans XML) ou de graphes (comme dans RDF), suscitant alors de nouveaux travaux en algorithmique, en complexité, mais aussi sur les langages de requête.

4 De la fiction aux perspectives concrètes du calcul quantique

Fiction introduite au début des années 1980, le calcul quantique a initialement éveillé l'intérêt grâce aux algorithmes de Shor et de Grover à la fin du siècle dernier, puis plus récemment avec l'algorithme de Harrow-Hassidim-Lloyd et la méthode QSVT [4] pour la résolution de systèmes linéaires. Suite à la conjonction de ces progrès théoriques et de nouvelles perspectives technologiques, un énorme effort financier public et privé a été consenti sur ce sujet dans de nombreux pays depuis une dizaine d'années, incarné en France par la stratégie nationale quantique [5], ce qui a considérablement développé et transformé la communauté scientifique, académique et industrielle. L'émergence de programmes de R&D sur le sujet dans de nombreuses grandes entreprises et l'apparition de multiples startups ont été rendues possibles grâce au rôle moteur d'un grand nombre de docteurs mais aussi de chercheurs confirmés issus de la recherche publique. La croissance fulgurante de la communauté internationale peut se mesurer par le nombre de soumissions à la conférence annuelle QIP, passé de 160 en 2006 à 700 vingt ans plus tard.

Les perspectives ouvertes par le calcul quantique, attisées à la fois par la promesse ultime d'un ordinateur quantique et par l'arrivée de prototypes très bruités (NISQ), touchent un très grand nombre de domaines de l'informatique fondamentale : nouveaux modèles de calcul, nouvelle hiérarchie de complexité, nouveaux algorithmes, nouvelles attaques en cryptographie, nouveaux langages de programmation... Certaines de ces questions sont au cœur du GT Informatique Quantique, d'autres sont à l'interface avec d'autres GTs.

De nombreux sujets nouveaux sont naturellement apparus en complexité et algorithmique, incluant par exemple la classification des problèmes selon leur complexité en requêtes ou encore de communication. Des résultats de séparation ont ainsi permis de clarifier les frontières entre calcul classique et calcul quantique, offrant des bases solides pour identifier des tâches pour lesquelles on pourrait apporter une preuve rigoureuse que les algorithmes quantiques offrent un avantage. Mais ce sont aussi de nouvelles méthodes qui ont été développées et qui, peu à peu, ont influencé des domaines non quantiques : en algorithmique (avec la déquantisation d'algorithmes quantiques ou la mise en évidence de l'absence de solutions par programmation linéaire pour le problème du voyageur de commerce), en théorie des codes correcteurs d'erreurs (par exemple les codes localement décodables), ainsi qu'en complexité classique, en particulier la réduction quantique de Regev qui relie la difficulté moyenne de Learning With Errors (LWE) à la difficulté dans le pire des cas de problèmes sur les réseaux euclidiens.

Des algorithmes quantiques sont proposés dans de nombreux domaines, pour la simulation de systèmes quantiques, pour la chimie quantique... mais aussi en cryptanalyse pour attaquer d'autres problèmes que ceux résolus par l'algorithme de Shor, par exemple des constructions classiques de codes d'authentification de messages. La recherche d'algorithmes quantiques en cryptanalyse est tout particulièrement importante pour évaluer la sécurité des systèmes *post-quantiques* en cours de standardisation. Censés remplacer les systèmes à clef publique classiques reposant sur la difficulté du logarithme discret et de la factorisation, ces alternatives n'ont évidemment d'intérêt que si on acquiert la conviction qu'elles résistent à un adversaire quantique. Les technologies quantiques ouvrent également des nouvelles possibilités en cryptographie en permettant de réaliser des fonctionnalités inatteignables classiquement, grâce à l'impossibilité de cloner l'information. Auparavant limitées à la distribution de clefs

4. Quantum Singular Value Transformation.
5. `https://quantique.france2030.gouv.fr/`

inconditionnellement sûre, les fonctionnalités visées se sont considérablement enrichies au cours des dernières années, apportant par exemple la possibilité d'évaluer un programme tout en empêchant sa copie.

Parmi les briques essentielles à la construction d'un ordinateur quantique, la correction d'erreurs joue un rôle important car l'information quantique est rapidement corrompue par le bruit. Il faut donc corriger les erreurs plus vite qu'elles se créent. La recherche de bons codes quantiques permettant de réaliser des circuits quantiques tolérants aux fautes avec un surcoût constant a suscité de nombreux travaux au cours des dix dernières années, et a également introduit des concepts qui se sont avérés centraux en complexité quantique.

L'impossibilité de cloner l'information quantique rend l'écriture même de programmes quantiques délicate, et complique également leur vérification par des techniques classiques de débuggage. La recherche de langages de programmation adaptés, et d'outils de vérification suscite donc depuis quelques années de nombreux travaux, en particulier sur la sémantique des langages de programmation quantiques et la représentation des circuits quantiques, notamment par des formulations du calcul quantique dérivées de la théorie des catégories.

5 L'omniprésence des probabilités

Les probabilités sont par essence au cœur de constructions d'objets en combinatoire, de l'analyse d'algorithmes en moyenne, des méthodes de recuit simulé, d'apprentissage, et bien entendu du calcul quantique. Mais elles sont aussi, au cours de ces vingt ans, devenues centrales dans la plupart des autres domaines de l'informatique fondamentale, au point que le GdR IFM a organisé une année thématique sur le sujet en 2023-24 [6]. Ainsi, toutes les thématiques où la notion d'aléa est essentielle manipulent des probabilités, comme la calculabilité, les algorithmes randomisés, les systèmes dynamiques... C'est également le cas de la correction d'erreurs et de la théorie de l'information qui modélise du bruit affectant les données.

Plus généralement, la gestion de l'incertitude est devenue centrale pour tous les GTs du GdR IFM. Il s'agit de quantifier l'incertitude sur les données, du fait par exemple de données incomplètes ou bruitées, situations classiques en bases de données, ou pour l'inférence géométrique, mais aussi l'incertitude intrinsèque à beaucoup de systèmes distribués ou cyber-physiques étudiés en vérification. L'étude de ces systèmes a nécessité le développement de modèles probabilistes, et a suscité nombre d'analyses pour évaluer la robustesse des algorithmes : comprendre ce qui demeure calculable malgré les perturbations, apporter des garanties sur le résultat dans un contexte bruité. Cette même approche probabiliste apparaît naturellement dans l'étude des algorithmes randomisés, mais aussi dans le contexte de l'analyse lissée qui perturbe artificiellement les instances pour analyser le comportement d'un algorithme hors de certains cas pathologiques. La théorie des probabilités est également devenue un outil essentiel dans la démonstration de résultats importants, comme celle de la conjecture d'Erdös-Faber-Lovász [7], cinquante ans après sa formulation.

La théorie des probabilités a également fait son apparition en vérification de programmes, de systèmes informatiques, de multiples façons. On y vérifie des systèmes informatiques ayant un comportement probabiliste — chaînes de Markov, processus de Markov partiellement observables, jeux stochastiques — et dans ce cadre, au-delà de la question de savoir si une

6. https://gdr-ifm.fr/thematic-years/probabilities
7. Selon laquelle tout graphe correspondant à l'union de n cliques de taille n dont l'intersection deux-à-deux contient au plus un sommet, a un nombre chromatique inférieur ou égal à n.

propriété donnée d'un système est vérifiée ou non, on calcule ou on estime la probabilité que cette propriété soit vraie. Les méthodes de vérification statistiques sont apparues, dans lesquelles on échantillonne des traces d'exécution pour en déduire des garanties approchées, rapidement. La recherche en sémantique des langages de programmation s'est développée, elle, d'une part pour traiter des langages de programmation statistiques, dont la création répond aux besoins de description de distributions et d'algorithmes d'échantillonnage complexes, et d'autre part pour établir les bases de futurs langages de programmation quantique.

6 Quand discret et continu se rejoignent

Historiquement, l'informatique fondamentale s'est construite autour d'objets discrets (mots, graphes, automates, circuits...) et de modèles finis. Les outils fournissant des garanties d'exactitude sont donc longtemps restés de nature discrète, même si l'importance croissante de la théorie des probabilités a fait entrer des outils de nature continue dans plusieurs disciplines, via des modèles probabilistes, des approximations ou des métriques continues, comme l'entropie en théorie de l'information.

Mais cette interaction avec le continu est devenue de plus en plus riche au cours des vingt dernières années, à travers de nouveaux outils et de nouveaux objets d'étude. Ainsi de nouveaux modèles de calcul manipulant des quantités continues, parfois mêlées avec du discret, ont vu le jour, visant à explorer ce qu'il est possible de calculer en utilisant l'information quantique, des composants analogiques, des modèles biologiques... Le continu est par exemple au cœur du calcul quantique, qui manipule des états dans des espaces de Hilbert où la notion d'erreur est elle-même continue et non plus discrète comme en classique. Il est aussi au cœur des réseaux neuronaux, qui calculent et optimisent certaines fonctions sur $\mathbb{R}^n$, pour de très grandes valeurs de n.

De plus, l'essor de l'apprentissage statistique et des algorithmes opérant sur les réels a conduit à formuler pour le calcul continu des questions de pouvoir de calcul et de complexité identiques à celles du cadre discret classique. C'est ainsi que des travaux récents ont par exemple montré comment on pouvait simuler des calculs sur des réels par des équations différentielles polynomiales, de manière à pouvoir exprimer des ressources classiques, telles que le temps de calcul, par des propriétés de ces équations, comme la longueur de la courbe.

Ce glissement vers le continu apparaît également en cryptographie, où les systèmes à clef publique reposant sur l'arithmétique modulaire sont peu à peu supplantés par des systèmes reposant sur les réseaux euclidiens, en cryptographie post-quantique ou pour le chiffrement homomorphe : si les objets manipulés restent discrets, c'est leur plongement dans $\mathbb{R}^n$ et donc des propriétés géométriques dans un espace continu de grande dimension qui sont à l'origine de la difficulté des problèmes sous-jacents.

Le problème de représentation d'un objet continu par un objet discret est au cœur des problèmes étudiés en géométrie informatique. L'approximation convergente des données, souvent simple échantillonnage d'objets réels continus, se fait assez naturellement par des primitives finies (mailles, surfaces splines). Ces dernières années ont vu l'émergence de nombreuses techniques pour définir une géométrie différentielle convergente sur ces objets finis. De façon plus profonde encore, la théorie géométrique de la mesure (cycle normal, courants normaux corrigés), adaptée aux données réelles, offre maintenant un cadre unifié pour les représentations finies et continues, garantissant stabilité des mesures et calcul efficace.

La combinatoire est riche d'exemples de ce mouvement d'aller-retour entre discret et continu. Lorsque la taille d'un système aléatoire atteint l'infini, un objet discret peut se transformer en un objet continu. Les phénomènes limites continus sont souvent plus parlants

que les phénomènes discrets et apportent des réponses universelles ; le mouvement brownien en est l'illustration, et ce en particulier dans l'étude des cartes planaires (ce sont des graphes dessinés sur une surface orientée). Celles-ci sont en particulier source de modèles pour la gravitation quantique, dont l'objet est de définir un espace-temps relativiste et quantique : les limites de cartes de grande taille décrivent la théorie quantique de Liouville.

Les langages de programmation étudiés classiquement en sémantique, et même les langages de programmation probabilistes, ne considéraient traditionnellement que des distributions discrètes. Au vu des besoins croissants de formalismes de description de distributions statistiques complexes, un glissement s'est opéré vers le continu, avec l'émergence de langages de programmation dits statistiques : on peut y tirer des objets au hasard dans des domaines continus, typiquement $\mathbb{R}^n$, mais aussi dans des espaces de fonctions d'ordre supérieur ou de distributions, qui seront donc elles-mêmes aléatoires ; un exemple typique en est donné par les processus de Dirichlet.

L'hybridation entre approche numérique et approche symbolique est aussi devenu un axe important, à la croisée des problématiques d'optimisation, d'approximation et d'algorithmique numérique. Elle repose sur l'idée d'exploiter la rapidité des méthodes numériques pour accélérer certains calculs, tout en préservant la capacité à manipuler des objets symboliques de manière exacte. Cette démarche s'accompagne de techniques de certification, qui garantissent la validité mathématique des résultats obtenus. Cette stratégie se révèle efficace lorsqu'elle est combinée à la méthode homotopique pour le calcul d'approximations de racines d'équations polynomiales. Elle trouve également des applications dans l'étude et la manipulation d'opérateurs différentiels. Elle est mobilisée pour le calcul d'intégrales intervenant dans l'estimation des probabilités de collision entre satellites et débris en orbite. Un algorithme fondé sur ces principes a notamment été embarqué sur un mini-satellite expérimental de l'Agence spatiale européenne.

Conclusion

Au-delà de ces quelques évolutions marquantes, le recueil des documents rédigés par chaque GT donne une vision kaléidoscopique de ces deux décennies de recherche en informatique théorique, vision qui témoigne à la fois de la richesse et du foisonnement de sujets abordés, et d'une approche commune où aspects fondamentaux, implémentation et mise en œuvre dans de nombreux domaines sont toujours étroitement mêlés. En particulier, il apparaît clairement que les concepts, les formalismes et les outils de l'informatique fondamentale jouent un rôle structurant dans l'ensemble des sciences informatiques, et plus généralement dans tous les domaines recherchant une analyse rigoureuse et des garanties sur l'optimalité, la qualité des résultats d'un algorithme et sur sa mise en œuvre. Il est donc essentiel que la mise en avant de quelques sujets (IA, informatique quantique, cryptographie) qui bénéficient actuellement d'une forte couverture médiatique et de financements dédiés conséquents ne se fasse pas au détriment d'autres aspects, et qu'elle ne compromette pas l'équilibre entre fondements théoriques et avancées technologiques, entre recherche pilotée et recherche guidée par la curiosité, dont l'alliance est à l'origine des succès scientifiques et technologiques de ces vingt dernières années.

Les membres du Conseil Scientifique du GdR IFM.

Valérie Berthé, Anne Canteaut, Jérémie Chalopin, Jean Goubault-Larrecq, Emmanuel Jeandel, Jacques-Olivier Lachaud, Frédéric Magniez, Anca Muscholl, Christine Paulin-Mohring, Sophie Tison, Ioan Todinca, Gilles Villard.

Algorithmes, complexité, calculabilité et calcul quantique

Twenty years of GdR IFM, seen from GT CoA

1 Central Topics of GT CoA: Algorithms and Complexity

The aim of the CoA working group of GdR IFM is to bring together all researchers in computer science and mathematics interested in methods and tools for:

- designing and analyzing efficient algorithms,
- establishing lower bounds on properties of algorithms (e.g., computation time, circuit size, approximation factors, quantity of bits exchanged, etc.).

The CoA working group focuses on all forms of algorithms including sequential, parallel or distributed algorithms, online algorithms, streaming algorithms, approximation algorithms, parameterized algorithms, probabilistic algorithms, quantum algorithms, and other new paradigms. We focus on the design and analysis of algorithms, approached from the joint point of view of upper and lower bounds regarding complexity and other bounded resources. The CoA working group is also interested in algorithms motivated by and applied to all types of environments: graphs, networks, biological systems, images, combinatorial objects, etc.

We list below some examples of areas of particular interest to CoA researchers in France and discuss the evolution of these domains over the last twenty years. This list is far from being exhaustive. In fact, algorithmic and complexity-theoretic questions can be found across (almost) all working groups of GDR IFM.

2 Algorithms

Approximation Algorithms

The field of approximation algorithms focuses on the design of provably good polynomial-time solutions typically for NP-hard problems, and has been studied extensively for several decades. The last 20 years have again seen major improvements on longstanding questions thanks to new techniques related for example to linear programming relaxations or randomization over specific distributions. We can cite for illustration the Traveling Salesperson Problem, for which the approximation factor of the metric version has been improved for the first time since the 1970s [51], and the first constant approximation algorithm for the asymmetric case has been designed [73], also following decades of research.

Besides such progress, there is still a significant gap between lower and upper bounds on approximation factors for many problems. A new direction has therefore been focusing on conditional lower bounds, assuming stronger hypotheses than $\mathsf{P} \neq \mathsf{NP}$ to exhibit inapproximability results, thus unifying and making explicit the core open problems at the origin of such gaps. For example, assuming *Unique Games Conjecture* variants has been proven to be sufficient to get tight lower bounds in several scheduling problems [72, 12].

As approximation algorithms focus on guaranteeing the quality of the solution for *all* instances, they may therefore be pessimistic on some real-world instances. This observation has motivated several directions aiming to go "beyond the worst-case", a perspective that has flourished in the last 20 years, leading to new research domains such as smooth analysis, robust optimization, advice complexity and learning-augmented algorithms [69].

Distributed Algorithms

The last 20 years have seen a considerable expansion in the range of applications for distributed computing, originally driven mainly by computer networks (multi-core processors, sensor networks, data centers, peer-to-peer networks, blockchains, etc.), which now extends to biological systems and nanotechnologies, and even sociology (social networks) and physics (complex systems). This evolution has led to a wide diversification of the computational models considered in the context of distributed computing.

In terms of fundamental research, recent algorithmic developments aim to circumvent the numerous obstacles that make it impossible to solve certain tasks in environments prone to failures or attacks (e.g., consensus). Studies focus in particular on the use of sophisticated communication mechanisms and cryptographic primitives. In the context of developing network algorithms for solving graph problems, distributed graph decomposition techniques have seen enormous growth, as well as derandomization techniques. It is also worth mentioning recent advances in distributed quantum computing, and the emergence of a line of research dedicated to limiting the energy consumption of distributed algorithms.

Finally, the identification of lower bounds has also seen enormous progress through the use of tools from algebraic topology, graph theory (e.g., round reduction) and communication complexity, coupled with a better understanding of the power and limitation of using random resources.

Linear Programming Algorithms

Linear programming is one of the landmark achievements of modern mathematics and computing. Developed in the 1940s to formalize questions of resource allocation, it soon became a powerful language for modeling and solving problems across science, industry, and economics. From logistics and scheduling to energy planning and finance, its influence has been enormous, while at the same time it has driven some of the most important advances in algorithms.

The earliest breakthrough was the *simplex method,* designed by Dantzig in 1947. The simplex method is an algorithm that can solve linear programming problems efficiently in practice, and it is a cornerstone of modern optimization software. Unlike its efficiency in practice, in the 1970s the simplex method was proven to require exponential time in the theoretical worst case. Similar to the field of approximation algorithms, this has inspired many researchers to innovate new approaches for algorithm analysis [69]. For the past 25 years, the leading approach has been *smoothed analysis*, introduced by Spielman and Teng. Smoothed analysis shows that if the input is perturbed by tiny random noise, then simplex runs in polynomial time in expectation [71]. This result has launched a wave of refinements, leading to variants with provably much fewer pivot steps [23, 46, 11]. Today, nearly matching upper and lower bounds on the smoothed complexity are known [11]. This has prompted new analytical approaches, basing mathematical assumptions on close observation of algorithm implementations and user manual specifications [10].

A very different approach appeared in the 1980s with *interior point methods,* which follow a continuous path through the interior of the feasible region [52]. These algorithms have a theoretical guarantee of about the square root of the problem dimension in the number of iterations, yet in practice they converge much faster, making them a cornerstone of modern solvers [80]. Their algebraic structure also makes it possible to combine them with fast linear system solvers, which has powered breakthroughs in theory for fundamental algorithmic problems such as maximum flow and minimum cost flow [64, 60, 65, 22, 8, 53, 39, 9, 78, 77,

76, 21, 79].

Recent work has even started to connect the two algorithmic lines. Many analyses of simplex methods are based on a geometric quantity called the shadow size, which measures the number of segments of a two-dimensional projection of the feasible polyhedron. Recent work has shown that the running time of certain interior-point methods is no greater than this number [3]. This led to strongly polynomial algorithms for new classes of problems [24], and opened a promising path towards the solution to Smale's 9th problem, whether all linear programs can be solved in strongly polynomial-time.

Fixed-Parameter Algorithms

Over the past two decades, the parameterized complexity community has developed an extensive toolkit for tackling hard problems, notably the techniques of color coding, iterative compression, bidimensionality, important separators, Cut and Count, and more recently, flow augmentation. The latter four techniques have influenced other areas, especially those concerned with cut and flow problems. In the last 10 years, new width parameters, such as mim-width, twin-width, and merge-width (the successors of treewidth and clique-width), have emerged and yielded logic-based meta-theorems, providing a clearer and more unified landscape of tractability. This is particularly true for the first-order logic model checking problem, where intensive efforts led to the characterization of the FPT tractability barrier for monotone classes (nowhere dense graphs), and significant progress for hereditary classes. Kernelization, a subarea of parameterized complexity, has obtained similar meta-theorems, via a technique called *protrusion replacement.* In parallel, lower-bound techniques have enabled one to match the running time of the fastest known algorithms for most of the parameterized problems.

While historically the focus of parameterized complexity has been on NP-hard graph problems, recent years have seen an expansion to handling problems from computational geometry, computational social choice, bioinformatics, etc., as well as polynomial-time-solvable graph problems, in the so-called FPT in P program. The latter has gradually merged with the field of fine-grained complexity. Another fruitful development of parameterized complexity is its interface with approximation algorithms. For problems that are both hard to approximate and fixed-parameter intractable, can we get improved approximation factors in parameterized time that is not attainable by exact computation? This line of work has brought a wealth of positive and negative results, building upon the toolboxes of (hardness of) approximation and of parameterized complexity.

Online Algorithms

The online algorithm paradigm models situations where the problem instance arrives in form of a request sequence, and the algorithm needs to serve each request immediately, without having knowledge about future requests. Algorithms' performance is then compared with that of an ideal *offline* algorithm having full information about the problem instance in advance. The resulting *competitive ratio* thus measures the price of not knowing the future.

The paging problem, the secretary problem, the prophet inequality problem, the "rent or buy" problem, and the cowpath problem are well-studied examples of an online problem. Many combinatorial optimization problems have their online counterpart.

The online setting differs from that of streaming algorithms, where the algorithms need to store information in small memory to answer a final query. It also differs from robust optimization, which is typically a two stage setting, where a decision has to be made in

the first phase, in such a way that changes in the instance during the second phase can be handled without large cost.

In 2011 [Emek, Fraigniaud, Korman, Rosén] introduced "with advice complexity", asking questions like, how many bits of "additional information" about the whole sequence are needed per request so as to guarantee a particular competitive ratio? In 2018, [Purohit, Svitkina, Kumar] and shortly later [Angelopoulos, Dürr, Jin, Kamali, Renault] studied a model where the algorithm is provided with a prediction on future requests, which could be learned from past instances, and which can have errors. The performance of an ideal online algorithm should degrade smoothly with the error. Such algorithms and matching lower bounds have been developed for many the important online problems.

Randomized Optimization Heuristics

Randomized optimization heuristics (ROHs) such as simple randomized local search algorithms, simulated annealing, evolutionary algorithms, or ant colony optimizers can be traced back to Turing's "learning machine" from 1950. They gained popularity in the 1970s, where in particular genetic algorithms were introduced as a means to reliably optimize settings with very limited problem-specific knowledge. Since then, ROHs thrive as general-purpose optimizers in such black-box settings, making use of more and more complex operators [27, 41, 75].

The majority of the research conducted on ROHs being empirical, their theoretical analysis was pioneered by Ingo Wegener and his students in the 1990s [36, 37, 48]. Initial theoretical results considered very simple algorithms that could be analyzed via elementary tools from probability theory, such as the coupon collector theorem [35]. Soon it was observed that already for the analysis of simple heuristics, the classic toolbox from randomized algorithms does not suffice. A prominent example is the analysis of the runtime of the (1+1) evolutionary algorithms on pseudo-Boolean linear functions, which led to the invention of a set of methods now known as *drift analysis* [29, 30, 44], which allow to translate information on the one-step progress into estimates for the runtime.

Over the years, the mathematical tools for analyzing ROHs became considerably more refined, allowing for the analysis of increasingly complex ROHs and developing complexity-theoretic models. Key results gave very precise estimates for algorithms' expected runtimes [47], they proved guarantees for entire generic classes of operators [61], and they studied dynamically adapting ROHs [59]. This increased body of knowledge also led to the design of new ROHs that were provably better than their predecessors [31, 33, 57]. In particular, in the evolutionary computation community, theory has become a cornerstone of the field [34]. However, theoretical work on ROHs is now also regularly present at the large AI conferences [15, 25, 28, 32, 63, 82].

Quantum Algorithms

In 2006, the 9th edition of the largest international conference devoted to the study of Quantum Information Processing (QIP) was held in Paris. The conference gathered an audience of about 200 participants and featured 40 talks selected from 160 submissions. Over the past decade, attendance has grown to around 1,000 participants annually, with the program now comprising 3 parallel sessions and approximately 130 presentations selected by an international committee. For the 29th edition, in 2026, the program committee consists of 130 members and received about 700 submissions. What has changed?

Since 2006, theoretical progress in quantum algorithms has accelerated, broadening both

research directions and applications. The Harrow–Hassidim–Lloyd (HHL) algorithm [43], introduced in 2009, provided the first quantum method for solving linear systems with exponential improvement in the system dimension, leading to developments in quantum machine learning, including clustering [54], classification [68], and recommendation systems [56]. In addition, generic algorithmic primitives such as quantum walks [74] and quantum Monte Carlo techniques [42] have been refined and applied to problems in optimization [5], sampling, backtracking [67], search [66], and simulation [14]. At the same time, results on oracle separations have clarified the boundaries between classical and quantum computation, providing a rigorous foundation for identifying tasks where quantum algorithms offer provable advantages, for instance in terms of query complexity [2], space complexity [49] and time complexity [1].

These advances have influenced both national research agendas and industrial initiatives. The NIST post-quantum cryptography standardization process has emphasized the practical relevance of quantum algorithms for information security, while national and industrial programs worldwide have supported research in quantum computing. As a result, the current algorithms community includes both researchers focused on fundamental theoretical developments and those working on applied use cases, bridging physics, computer science, and potential industrial applications. Area of applications include simulation of quantum physics for quantum chemistry and materials simulation, post-quantum cryptography and cryptanalysis, and more generally optimization through quantum heuristics such as quantum variational approaches [38]. Some of the proposed solution even include hybrid quantum–hpc approaches [50].

Algorithmic Game Theory

Algorithmic Game Theory lies at the intersection of computer science, economics, and mathematics, focusing on the computational aspects of strategic decision-making and equilibrium analysis. Its foundations were laid in the early 2000s, marked by three seminal papers that were later recognized with the Gödel Prize: “Worst-case equilibria” by Koutsoupias and Papadimitriou, “How bad is selfish routing?” by Roughgarden and Tardos, and “Algorithmic Mechanism Design” by Nisan and Ronen.

Recent advances have deepened our understanding of the algorithmic complexity of finding equilibria, particularly Nash equilibria in large-scale and dynamic games. Researchers have developed efficient approximation algorithms and equilibrium computation techniques for specific game classes, such as congestion, auction, and network games. The rise of online platforms and decentralized systems (Adwords, etc) has further motivated the study of algorithmic mechanisms that ensure efficiency, fairness, and incentive compatibility under computational constraints.

Modern research in Algorithmic Game Theory increasingly integrates machine learning, data-driven modeling, and mechanism design to address real-world challenges in markets, networks, and multi-agent systems. Deep learning-based approaches now allow agents to learn strategies and equilibria in environments too complex for explicit analytical solutions. Meanwhile, algorithmic mechanism design continues to evolve toward robust and adaptive frameworks that account for uncertainty, partial information, and dynamic participation. This convergence of learning, optimization, and strategic reasoning positions Algorithmic Game Theory as a foundational field for understanding and engineering intelligent, self-organizing systems in economics, artificial intelligence, and beyond.

Combinatorial Reconfiguration

Combinatorial reconfiguration is an emerging discipline in theoretical computer science that, since the late 2000s, has led to a systematic study of algorithmic and structural questions about so-called *solution graphs*. For an instance of some combinatorial problem, such as the k-coloring problem, the solution graph has as vertices all solutions to the instance (e.g., k-colorings of a given graph) and two solutions are adjacent whenever they are sufficiently similar. Typically, two solutions are similar, if one can be obtained from the other by a simple modification; for k-colorings, a popular modification is to change the color of a single vertex. Solution graphs are relevant in many applications, such as random generation of combinatorial structures, enumeration, combinatorial games, motion planning, statistical physics, and bioinformatics. However, they are are usually too large to be constructed explicitly. A typical question in combinatorial reconfiguration is whether two combinatorial structures are equivalent in the sense that one can be transformed into the other by applying a sequence of small modifications. Such a sequence of modifications corresponds to a path in a solution graph. We may ask under which conditions the solution graph is connected or what is diameter is, that is, how many modifications suffice in order to transform one k-coloring into any other k-coloring.

The number of combinatorial structures that have been studied from the reconfiguration point of view in the last fifteen years or so is vast, so let us focus the aforementioned k-colorings. The general goal is to classify coloring problems based on the complexity of determining whether two colorings are equivalent. For example, deciding the equivalence of two 3-colorings is polynomial, even though deciding whether a graph admits a 3-coloring is NP-complete [20]. Meanwhile, determining if two 4-colorings of a graph are equivalent is PSPACE-complete [17]. An important insight is that there are certain topological reasons for whether two colorings are equivalent [81]. This insight has lead to a much better understanding of the complexity of testing the equivalence of graph homomorphisms (a generalization of k-colorings), but a complete classification currently seems out of reach [62, 81]. Besides such a classification, a major open problem in the area is the conjecture of Cereceda, which states that for $k \geq d+2$, the diameter of the graph of k-colorings of a d-degenerate graph G is quadratic in the number of vertices of G. The best current bound is polynomial [18].

3 Complexity

Classical Complexity Theory

Classical complexity theory has witnessed several landmark results. One standout achievement is Harvey and van der Hoeven's algorithm for integer multiplication in $O(n \log n)$ time, achieving the theoretically optimal bound for this fundamental operation. Another major breakthrough is Ryan Williams' work on simulating time with square-root space, which established a surprising relationship between time and space complexity, showing that certain time-bounded computations can be simulated using much less space than previously thought.

Fine-grained Complexity

Fine-grained complexity focus on the precise running times of algorithms for fundamental problems. The methodology of fine-grained complexity mimics the approach of NP-hardness: researchers select key problems conjectured to require a certain amount of time (such as k-SAT under the Strong Exponential Time Hypothesis (SETH)) and then use fine-grained reductions to transfer these hardness assumptions to other problems. This has led to a web

of tight conditional lower bounds. At its core the hardness of most of the studied problems in fine-grained complexity can be based on the presumed hardness of three key problems: the 3SUM problem, the All-Pairs-Shortest Paths (APSP) problem and k-SAT. The first problem is assumed to not have subquadratic algorithm, for the second one, the best algorithm is believed to be only cubic.The Strong Exponential Time Hypothesis states that for every $\varepsilon > 0$, there exists a constant k such that k-SAT (the satisfiability problem for conjunctive normal form formulas with k literals per clause) cannot be solved in time $O(2^{(1-\varepsilon)n})$, where n is the number of variables. In other words, SETH asserts that exhaustive search is essentially optimal for SAT, even for large k. Williams showed that this hypothesis implies that the Orthogonal Vectors problem (find in a set of vectors, two which are orthogonal) can not be solved by a subquadratic algorithm. Under these hypotheses, it is shown that the best algorithms for a large class of classical problems (edit distance, graph diameter, shortest cycle, planar motion planning, sequence local alignment, ...) can not be substantially improved.

Circuits Complexity

Circuit complexity has also seen significant progress over the past decade. Rossman, Servedio, and Tan established an average-case depth hierarchy theorem for Boolean circuits, demonstrating that circuits of increasing depth can solve more problems on average, even when size is restricted. Monotone circuits (i.e., without NOT gates) are a fundamental model where exponential lower bounds are known. Pitassi and Robere improved these results by obtaining strongly exponential lower bounds for them, by developing a refined version of the lifting theorems. In the arithmetic realm, Limaye, Srinivasan, and Tavenas showed superpolynomial lower bounds against low-depth algebraic circuits, the similar question has been known for Boolean circuits for 50 years, but this question remained unresolved in the arithmetic case. In fact arithmetic circuits of constant depth seem more powerful. This idea has been strengthened by Andrews and Wigderson's work, arithmetic circuits of constant-depth are able to efficiently compute the gcd of two polynomials, as well as the related problems of the discriminant, resultant, Bézout coefficients, squarefree decomposition.

Meta-complexity

The field of meta-complexity refers to the complexity of problems that are themselves about computational complexity. Central examples are the Minimum Circuit Size Problem (MCSP), which asks for the smallest circuit computing a given Boolean function the problem of determining the Kolmogorov complexity of a string. Research has revealed deep connections between MCSP and other fundamental areas, such as cryptography and learning theory. Notably, quasi-polynomial time algorithms for PAC-learning constant-depth circuits with parity gates were developed by Carmosino, Impagliazzo, Kabanets, and Kolokolova, and new worst-case to average-case reductions for NP problems were established by Hirara.

Query complexity

Query complexity measures the number of queries to an input required to solve a task when access is restricted. For example, sorting a size-n array requires at least $n \log n$ comparisons (queries) to the elements, though it can be faster with direct access. It has also been extensively studied for its deep connections to combinatorial structures. It also has potential impact on data queries in the cloud. Prior to 2006, the relationships among measures of query complexity for total Boolean functions were only partially understood. Deterministic query complexity $D(f)$ could exceed randomized complexity $R(f)$, with the best explicit separation

realized for the recursive NAND tree: $R(f) = \Theta(D(f)^{0.753\cdots})$ [70]. Sensitivity $s(f)$ was known to relate to block sensitivity $bs(f)$, but the *sensitivity conjecture*, asserting a polynomial relationship between $s(f)$ and $bs(f)$ and thus $D(f)$, remained unresolved. Since 2006, Huang's proof established this relationship [45], while pointer functions provide the strongest known explicit separations for total functions: $D(f) = \Omega(n/\log n)$, $R_0(f) = \tilde{O}(\sqrt{n})$, and $Q(f) = \tilde{O}(n^{1/4})$ [4]. These results both clarify the structural role of sensitivity and furnish explicit constructions demonstrating near-optimal gaps among deterministic, randomized, and quantum query complexities, forming a foundational toolkit for lower-bound techniques.

(Quantum) Communication complexity

Communication complexity is a fundamental framework for proving lower bounds across a variety of computational models. Early applications include circuit depth, formula size, VLSI complexity, and time–space tradeoffs. Over the past two decades, it has found new applications in establishing lower bounds and tradeoffs for computational models designed for massive data, while also incorporating techniques from information theory. Some information-theoretic methods were also inspired by quantum information, in particular by non-local games [55]. In a different vein, quantum communication complexity separations developed to distinguish quantum from classical models, such as the *hidden matching problem* [13], have led to further unexpected results. This separation was use to prove space lower bounds for streaming algorithms solving graph problems such as *maximum matching* [6]. More generally, the reduction of communication complexity, and more generally of information complexity [19], to other models, has provided explicit lower bounds for *streaming algorithms* (e.g., statistics estimation [58]), for *distributed computing* tasks(e.g. minimum spanning tree, shortest paths, and minimum cut [26]) and for *property testing* problems (e.g., monotone functions and the class of sparse polynomials [16]). These methods unify lower-bound arguments across deterministic, randomized, and quantum models.

Applications to Learning

Information-theoretic and communication complexity techniques have been instrumental in establishing space-sample tradeoffs for PAC learning. In memory-bounded settings, any one-pass algorithm that learns a class of functions from m labeled examples must use memory proportional to the communication complexity of an associated hard problem. This implies, for instance, that learning parity functions or certain conjunctions with sublinear memory requires an exponential number of samples. These tradeoffs have direct ramifications in algorithm design, limiting what can be learned efficiently in streaming or online settings, and in privacy-preserving learning, since the memory constraints effectively restrict the amount of information retained about individual data points, providing inherent privacy guarantees.

Proof Complexity

Proof complexity studies proofs as computational objects, asking how hard it is to certify that a statement is true. By analyzing the size and structure of proofs in formal systems, it sheds light on the limits of efficient reasoning and their connection to algorithmic complexity. Since 2006, proof complexity has developed into a rich and cross-disciplinary research program linking propositional proofs, circuit complexity, algebraic methods, optimization, and communication-based techniques. Grochow and Pitassi's Ideal Proof System [40] marked an important step toward unifying algebraic proof systems with circuit complexity, connecting algebraic-circuit lower bounds, polynomial identity testing, and the hardness of proving

tautologies. A major milestone established by Atserias and Müller showed that Resolution, a fundamental propositional proof system underlying many SAT solvers, is not automatizable unless $\mathsf{P} = \mathsf{NP}$ [7], resolving a decades-old question about whether short proofs can be found efficiently. This breakthrough spurred a robust line of work extending NP-hardness of automatizability to other systems. In parallel, the Sum-of-Squares/Lasserre proof system emerged as a central bridge between optimization, algorithms, and proof complexity, yielding deep results on both algorithmic power and proof-size lower bounds, in particular for constraint satisfaction and planted problems. Finally, a family of lifting and query-to-communication theorems has provided a versatile framework for transferring query-complexity lower bounds into communication and proof-complexity separations, producing strong lower bounds across multiple proof systems.

Coordination

Carola Doerr (LIP6, Paris) and Alantha Newman (LIP, Lyon).

Contributors

Benjamin Bergougnoux (LIS, Marseille), Marthe Bonamy (LaBRI, Bordeaux), Edouard Bonnet (LIP, Lyon), Monika Csikos (IRIF, Paris), Benjamin Doerr (LIX, Ecole Polytechnique, IP Paris), Christoph Dürr (LIP6, Paris), Laurent Feuilloley (LIRIS, Lyon), Joanna Fijalkow (LaBRI, Bordeaux), Pierre Fraigniaud (IRIF, Paris), George Giakkoupis (IRISA, Rennes), Bruno Grenet (LJK, Grenoble), Sophie Huiberts (LIMOS, Clermont-Ferrand), Martin S. Krejca (LIX, Ecole Polytechnique, IP Paris), Sophie Laplante (IRIF, Paris), William Lochet (LIRMM, Montpellier), Frédéric Magniez (IRIF, Paris), Mathieu Mari (LIRMM, Montpellier), Simon Mauras (INRIA, Paris-Saclay), Moritz Muhlenthaler (G-SCOP, Grenoble), Nicolas Nisse (Inria et I3S, Sophia Antipolis), Adi Rosén (IRIF, Paris), Bertrand Simon (LIG, Grenoble), Tatiana Starikovskaya (DIENS, Paris), Sebastien Tavenas (LAMA, Chambery), Nguyen Kim Thang (LIG, Grenoble), Adrian Vladu (IRIF, Paris).

References

1. Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 141–150, New York, NY, USA, 2010. Association for Computing Machinery.
2. Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 307–316, New York, NY, USA, 2015. Association for Computing Machinery.
3. Xavier Allamigeon, Daniel Dadush, Georg Loho, Bento Natura, and László A Végh. Interior point methods are not worse than simplex. *SIAM Journal on Computing*, 54(5):FOCS22–178, 2025.
4. Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *J. ACM*, 64(5), September 2017.
5. Simon Apers and Ronald de Wolf. Quantum speedup for graph sparsification, cut approximation, and laplacian solving. *SIAM Journal on Computing*, 51(6):1703–1742, 2022.
6. Sepehr Assadi and Janani Sundaresan. Hidden permutations to the rescue: Multi-pass streaming lower bounds for approximate matchings. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 909–932, 2023.

7 Albert Atserias and Moritz Müller. Automating resolution is np-hard. *J. ACM*, 67(5):31:1–31:17, 2020.

8 Kyriakos Axiotis, Aleksander Mądry, and Adrian Vladu. Circulation control for faster minimum cost flow in unit-capacity graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 93–104. IEEE, 2020.

9 Kyriakos Axiotis, Aleksander Mądry, and Adrian Vladu. Faster sparse minimum cost flow by electrical flow localization. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 528–539. IEEE, 2022.

10 Eleon Bach, Alexander E. Black, Sophie Huiberts, and Sean Kafer. Beyond smoothed analysis: Analyzing the simplex method by the book, 2026. To appear in STOC.

11 Eleon Bach and Sophie Huiberts. Optimal smoothed analysis of the simplex method. In *Proceedings of the 66th Annual Symposium on Foundations of Computer Science (FOCS)*, 2025.

12 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 453–462. IEEE, 2009.

13 Ziv Bar-Yossef, T. S. Jayram, and Iordanis Kerenidis. Exponential separation of quantum and classical one-way communication complexity. *SIAM Journal on Computing*, 38(1):366–384, 2008.

14 Dominic W. Berry and Andrew M. Childs. Black-box hamiltonian simulation and unitary implementation. *Quantum Info. Comput.*, 12(1–2):29–62, January 2012.

15 Chao Bian, Shengjie Ren, Miqing Li, and Chao Qian. An archive can bring provable speed-ups in multi-objective evolutionary algorithms. In *International Joint Conference on Artificial Intelligence, IJCAI 2024*, pages 6905–6913. ijcai.org, 2024.

16 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Comput. Complex.*, 21(2):311–358, June 2012.

17 Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009.

18 Nicolas Bousquet and Marc Heinrich. A polynomial version of cereceda's conjecture. *J. Comb. Theory B*, 155:1–16, 2022.

19 Mark Braverman. Interactive information complexity. *SIAM Journal on Computing*, 44(6):1698–1739, 2015.

20 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of Graph Theory*, 67(1):69–82, 2011.

21 Li Chen, Rasmus Kyng, Yang Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. *Journal of the ACM*, 72(3):1–103, 2025.

22 Michael B Cohen, Aleksander Mądry, Piotr Sankowski, and Adrian Vladu. Negative-weight shortest paths and unit capacity minimum cost flow in $\tilde{O}(m^{10/7} \log w)$ time. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 752–771. SIAM, 2017.

23 Daniel Dadush and Sophie Huiberts. A friendly smoothed analysis of the simplex method. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 390–403, 2018.

24 Daniel Dadush, Zhuan Khye Koh, Bento Natura, Neil Olver, and László A Végh. A strongly polynomial algorithm for linear programs with at most two nonzero entries per row or column. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1561–1572, 2024.

25 Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Conference on Artificial Intelligence, AAAI 2023*, pages 12390–12398. AAAI Press, 2023.

26 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 363–372, New York, NY, USA, 2011. Association for Computing Machinery.

27 Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

28 Anh Viet Do, Aneta Neumann, Frank Neumann, and Andrew M. Sutton. Rigorous runtime analysis of MOEA/D for solving multi-objective minimum weight base problems. In *Advances in Neural Information Processing Systems, NeurIPS 2023*, pages 36434–36448, 2023.

29 Benjamin Doerr and Leslie A. Goldberg. Adaptive drift analysis. *Algorithmica*, 65:224–250, 2013.

30 Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64:673–697, 2012.

31 Benjamin Doerr and Martin S. Krejca. Significance-based estimation-of-distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 24:1025–1034, 2020.

32 Benjamin Doerr, Martin S. Krejca, and Andre Opris. Tight runtime guarantees from understanding the population dynamics of the GSEMO multi-objective evolutionary algorithm. In *International Joint Conference on Artificial Intelligence, IJCAI 2025*, pages 8876–8884. ijcai.org, 2025.

33 Benjamin Doerr, Huu Phuoc Le, Régis Makhmara, and Ta Duy Nguyen. Fast genetic algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2017*, pages 777–784. ACM, 2017.

34 Benjamin Doerr and Frank Neumann, editors. *Theory of Evolutionary Computation—Recent Developments in Discrete Optimization*. Springer, 2020. Also available at `http://www.lix.polytechnique.fr/Labo/Benjamin.Doerr/doerr_neumann_book.html`.

35 Stefan Droste, Thomas Jansen, and Ingo Wegener. On the optimization of unimodal functions with the $(1 + 1)$ evolutionary algorithm. In *Parallel Problem Solving from Nature, PPSN 1998*, pages 13–22. Springer, 1998.

36 Stefan Droste, Thomas Jansen, and Ingo Wegener. A rigorous complexity analysis of the (1+1) evolutionary algorithm for linear functions with Boolean inputs. In *International Conference on Evolutionary Computation, ICEC 1998*, pages 499–504. IEEE, 1998.

37 Stefan Droste, Thomas Jansen, and Ingo Wegener. A rigorous complexity analysis of the $(1 + 1)$ evolutionary algorithm for separable functions with Boolean inputs. *Evolutionary Computation*, 6:185–196, 1998.

38 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size. *Quantum*, 6:759, July 2022.

39 Yu Gao, Yang Liu, and Richard Peng. Fully dynamic electrical flows: Sparse maxflow faster than goldberg–rao. *SIAM Journal on Computing*, (0):FOCS21–85, 2023.

40 Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018.

41 Ryoki Hamano, Shota Saito, Masahiro Nomura, and Shinichi Shirakawa. CMA-ES with margin: lower-bounding marginal probability for mixed-integer black-box optimization. In *Genetic and Evolutionary Computation Conference, GECCO 2022*, pages 639—647. ACM, 2022.

42 Yassine Hamoudi and Frédéric Magniez. Quantum Chebyshev's Inequality and Applications. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 69:1–69:16, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

43 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

44 Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127:51–81, 2001.

45 Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3), November 2019.

46 Sophie Huiberts, Yin Tat Lee, and Xinzhi Zhang. Upper and lower bounds on the smoothed complexity of the simplex method. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1904–1917, 2023.

47 Hsien-Kuei Hwang, Alois Panholzer, Nicolas Rolin, Tsung-Hsi Tsai, and Wei-Mei Chen. Probabilistic analysis of the (1+1)-evolutionary algorithm. *Evolutionary Computation*, 26:299–345, 2018.

48 Thomas Jansen and Ingo Wegener. On the analysis of evolutionary algorithms – a proof that crossover really can help. In *European Symposium on Algorithms, ESA 1999*, pages 184–193. Springer, 1999.

49 John Kallaugher, Ojas Parekh, and Nadezhda Voronova. Exponential quantum space advantage for approximating maximum directed cut in the streaming model. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1805–1815, New York, NY, USA, 2024. Association for Computing Machinery.

50 Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, September 2017.

51 Anna R Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric tsp. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 32–45, 2021.

52 Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311, 1984.

53 Tarun Kathuria, Yang P Liu, and Aaron Sidford. Unit capacity maxflow in almost m^4/3 time. *SIAM Journal on Computing*, 53(6):FOCS20–175, 2022.

54 Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

55 Iordanis Kerenidis, Sophie Laplante, Virginie Lerays, Jérémie Roland, and David Xiao. Lower bounds on information complexity via zero-communication protocols and applications. *SIAM Journal on Computing*, 44(5):1550–1572, 2015.

56 Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

57 Martin S. Krejca and Carsten Witt. A flexible evolutionary algorithm with dynamic mutation rate archive. *Algorithmica*, 88:1–32, 2025.

58 Ravi Kumar, T. S. Jayram, Ziv Bar-Yossef, and D. Sivakumar. An Information Statistics Approach to Data Stream and Communication Complexity . In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, page 209, Los Alamitos, CA, USA, November 2002. IEEE Computer Society.

59 Jörg Lässig and Dirk Sudholt. Adaptive population models for offspring populations and parallel evolutionary algorithms. In *Foundations of Genetic Algorithms, FOGA 2011*, pages 181–192. ACM, 2011.

60 Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $O(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.

61 Per Kristian Lehre and Carsten Witt. Black-box search by unbiased variation. *Algorithmica*, 64:623–642, 2012.

62 Benjamin Lévêque, Moritz Mühlenthaler, and Thomas Suzan. Reconfiguration of digraph homomorphisms. *SIAM J. Discret. Math.*, 39(1):327–360, 2025.

63 Mingfeng Li, Qiang Zhang, Weijie Zheng, and Benjamin Doerr. Why popular MOEAs are popular: Proven advantages in approximating the Pareto front. In *Advances in Neural Information Processing Systems, NeurIPS 2025*, 2025. To appear.

64 Aleksander Madry. Navigating central path with electrical flows: From flows to matchings, and back. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 253–262. IEEE, 2013.

65 Aleksander Madry. Computing maximum flow with augmenting electrical flows. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 593–602. IEEE, 2016.

66 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011.

67 Ashley Montanaro. Quantum-walk speedup of backtracking algorithms. *Theory of Computing*, 14(15):1–24, 2018.

68 Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, Sep 2014.

69 Tim Roughgarden, editor. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, December 2020.

70 Michael Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 29–38, 1986.

71 Daniel Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 296–305, 2001.

72 Ola Svensson. Hardness of precedence constrained scheduling on identical machines. *SIAM Journal on Computing*, 40(5):1258–1274, 2011.

73 Ola Svensson, Jakub Tarnawski, and László A Végh. A constant-factor approximation algorithm for the asymmetric traveling salesman problem. *Journal of the ACM (JACM)*, 67(6):1–53, 2020.

74 M. Szegedy. Quantum speed-up of markov chain based algorithms. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.

75 Dirk Thierens and Peter A.N. Bosman. Optimal mixing evolutionary algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2011*, pages 617—-624. ACM, 2011.

76 Jan van den Brand, Yu Gao, Arun Jambulapati, Yin Tat Lee, Yang P Liu, Richard Peng, and Aaron Sidford. Faster maxflow via improved dynamic spectral vertex sparsifiers. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 543–556, 2022.

77 Jan Van Den Brand, Yin Tat Lee, Yang P Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 859–869, 2021.

78 Jan van den Brand, Yin-Tat Lee, Danupon Nanongkai, Richard Peng, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Bipartite matching in nearly-linear time on moderately dense graphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 919–930. IEEE, 2020.

79 Adrian Vladu. Breaking the barrier of self-concordant barriers: Faster interior point methods for m-matrices. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 2213–2224, 2025.

80 Stephen J. Wright. Optimization in theory and practice, 2025.

81 Marcin Wrochna. Homomorphism reconfiguration via homotopy. *SIAM J. Discret. Math.*, 34(1):328–350, 2020.

82 Weijie Zheng and Benjamin Doerr. Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II). *Artificial Intelligence*, 325:104016, 2023.

Twenty Years of GdR IFM, seen from GT Graphs

At the turn of the 2000s, graph theory experienced major advances. In structural graph theory, the strong perfect graph conjecture had just been proven, and the publication of the Graph Minors series was nearing its end. In graph algorithms, the first applications of Courcelle's Theorem were emerging, and the field of parameterized algorithms was also gaining momentum.

The *Strong Perfect Graph Theorem* [25] characterizes graphs G for which the chromatic number $\chi(G)$ equals the size of their largest clique $\omega(G)$, and for which all induced subgraphs preserve this property. The chromatic number $\chi(G)$ is the smallest number of parts in a partition of the vertices of G such that each edge has its endpoints in distinct parts. It is immediate that $\chi(G) \geq \omega(G)$ for any graph G. This theorem thus characterizes the graphs for which this bound is tight.

The *Graph Minors series* by Robertson and Seymour is an in-depth study of classes of graphs that are closed under the minor relation. This means that in such a class $\mathcal{C}$, for any graph $G \in \mathcal{C}$, deleting a vertex, deleting an edge, or contracting an edge produces a graph that also belongs to $\mathcal{C}$. This is, for example, the case for the class of planar graphs—those that can be drawn in the plane without any edge crossings. One of the highlights of this series is the proof of Wagner's Conjecture: the minor relation is a well-quasi-ordering. This means that every minor-closed class of graphs $\mathcal{C}$ can be characterized by a finite set of excluded minors.

Courcelle's Theorem [31] guarantees that any property Q that can be expressed in a certain logical language — the monadic second-order logic of graphs — and that concerns graphs of tree-width at most t, admits an algorithm running in time $f(Q,t) \cdot |G|$, where G is the input graph and f is some function. Tree-width is a graph invariant that, in a sense, measures how many vertices need to be removed to split a graph—or any of its subgraphs—into smaller pieces (each piece containing at most two-thirds of the original vertices).

Parameterized Complexity is a response to the proven difficulty of many algorithmic problems—the so-called NP-hard problems. For such problems, it is believed that no algorithm can solve them in polynomial time, that is, in time $O(|G|^c)$, where G is the input graph and $c > 0$ is a constant. This approach proposes, for any graph problem Π, to determine the graph parameters $p(G)$ for which there is an algorithm solving Π in time $f(p(G)) \cdot |G|^c$, for some function f and value $c > 0$. This allows characterizing sub-classes of graphs for which Π can be solved efficiently.

Since 2006, graph theory has witnessed an impressive number of works and advances. Given this abundance, it is impossible to provide an exhaustive overview. Likewise, it is difficult to single out the most significant developments. Nevertheless, we will venture to do so: starting from the four major milestones we have just discussed, we will explore in Section 1 how they have nourished our field and fostered the emergence of new theories, powerful tools, and remarkable results. Next, in Section 2, we will look at examples of new tools or new models originating from other fields, which have had a significant impact and that influenced the evolution of our research questions.

1 Continuation of major works

1.1 Perfect graphs and χ-boundedness

Understanding why a graph has large chromatic number is a central question in graph theory, and one that most likely does not admit a simple answer. It is clear that a graph containing a large clique (that is a large complete graph) must have a large chromatic number. But the converse is not always true. It has been known since Erdős [48] that there exist triangle-free graphs (and even graphs with no short cycles) with arbitrarily large chromatic number. These examples show that the difficulty of coloring cannot always be explained by local properties.

The strong perfect graph theorem [25] (solved exactly 20 years ago) tells us that a chromatic number strictly larger than the clique number forces the presence of an induced odd cycle in the graph or its complement. In his foundational paper [58], Gyárfás went further and asked which structures must necessarily appear when the gap between $\chi(G)$ and $\omega(G)$ becomes arbitrarily large. He introduced the notion of a χ-bounded class of graphs and formulated several conjectures that have guided subsequent research, including a number concerning the cycle lengths that must occur.

These conjectures remained out of reach for a long time, but the past twenty years have witnessed spectacular progress, starting with the resolution of a special case by Bonamy, Charbit and Thomassé in 2014 [11]. In 2016, Scott and Seymour proved that graphs with no odd hole are indeed χ-bounded [91], that is, the absence of odd holes suffices to bound the chromatic number as a function of the clique number. In 2019, with Chudnovsky and Spirkl, they subsequently established a stronger result: forbidding induced long cycles, or even only induced long odd cycles, also suffices [26, 27].

Among the famous conjectures of Gyárfás, one has resisted attempts for the past forty years: the *Gyárfás–Sumner conjecture* [58, 93], which asserts that for every tree T, the class of graphs with no induced copy of T is χ-bounded. The conjecture is now proved for many specific families of trees (paths, stars, trees of radius two, etc.), and the methods developed to attack it have spread much further, notably in algorithmics (the so-called *Gyárfás path argument*). But the conjecture remains open, and nothing guarantees that a counterexample will not eventually be found. The Gyárfás–Sumner conjecture is still actively studied, in particular by members of the GT Graphs group, both in its original form and in various extensions, such as to signed graphs [2] and oriented graphs [1].

Interestingly, the Gyárfás path argument has also led to new advances on the Erdős–Hajnal conjecture [49], another major open problem in structural graph theory. This conjecture states that forbidding a single graph as an induced subgraph should suffice to move far away from the standard behavior of random graphs, where the largest clique and stable set are only logarithmic in size, and instead guarantee the existence of a clique or a stable set of polynomial size. Perfect graphs, for instance, always contain a clique or a stable set of size $\Omega(\sqrt{n})$. An alternative direction that has proved fruitful is to forbid two (families of) graphs rather than just one, in order to gain additional structure: for example, forbidding a path and its complement [21], or forbidding long cycles and their complements or forbidding just one cycle of fixed length together with its complement [10], or a forest together with its complement [28], already ensures the existence of a very large clique or a very large stable set.

1.2 Sparse Graphs

Sparsity vs denseness is a natural dichotomy when studying the structure of mathematical objects. However, there is not a unique definition of what a sparse graph would look like. While there might be many notions of sparsity, they all agree on some monotonicity conditions. For instance, one can consider as sparse any graph G whose maximum average degree, that is $\max_{H\subseteq G} \frac{1}{|V(H)|}\sum_{v\in V(H)} \deg_H(v)$ where H ranges over the subgraphs of G, is small. However, this definition while can be useful for some partition problems such as coloring problems, is not convenient in other context as simply subdividing every edge of a dense graph gives rise to a graph of small maximum average degree.

In the seminal work of Robertson and Seymour on classifying graph classes with respect to the minor quasi-ordering, the family of non-trivial minor-closed graph classes appear to meet many desired properties of sparsity : (1) the number of edges is linear, (2) they are hereditary w.r.t. minor, and (3) cannot contain all graphs as minors. Nonetheless, graphs of small degree are not minor-closed and still share many properties with non-trivial minor-closed graph classes, indicating better notions of sparsity. In their seminal work [79, 78, 81], rooting from the study of graph homomorphism problems, Nešetřil and Ossona de Mendez proposed to substitute r-shallow minor[1] to subgraph in the definition of maximum average degree. While the definition seems as for maximum average degree to be arbitrary, it allows to define a sparsity parameter for each value $r \geq 0$, ranging from the maximum average degree ($r = 0$) to parameters characterizing minor-closed classes ($r = \infty$). It happens also that this definition is effective enough that an equivalent hierarchy is obtained if we replace minor by other quasi-orders such as topological minor or immersion [82]. Using these parameters, an infinite lattice of sparse graph classes can be defined [82], the most prominent one being the familly of *bounded expansion* graph classes where the parameters are bounded by a function f. For graph classes where these parameters are not bounded by any function $f(r)$, one can distinguish between *nowhere dense* and *somewhere dense* graph classes. Here, the dense graph classes are those that contain all graphs in their r-shallow minors, for some r. Conversely, the *nowhere dense* classes are such that the clique-number of each r-shallow minor is bounded, but they can have super-linear number of edges, a property that cannot be true for non-trivial minor-closed graphs. The links of this class trichotomy with clique number have triggered many questions regarding their links with coloring problems, and several other characterizations of these classes have been proposed, the main structural ones being those based on *generalized coloring* [97], those based on *low tree-width/tree-depth coloring* [80], those based on variants of cops and robber games [56] and those based on *neighbourhood complexity* [86].

Another way of studying a graph class $\mathcal{C}$ is to understand the structure of the set of graphs not belonging to $\mathcal{C}$, and in particular the minimal ones w.r.t. a quasi-order and called obstructions. These obstructions are for instance used as certificates for recognition algorithms. For instance, any minor-closed class of graphs is characterized by a finite list of obstructions, allowing a polynomial recognition algorithm for any minor-closed graph class. Another line of research in studying sparse graph classes is to identify sets of obstructions for important graph properties. One can cite among such results the characterization of NIP and stable graph classes w.r.t. induced subgraph quasi-order [72], and the constructive proof computing the set of obstructions for having the Erdős-Pósa property[2] for any minor-closed

[1] Roughly an r-shallow minor is a minor where contraction are done only on connected subgraphs of diameter at most r.

[2] Having the Erdős-Pósa property w.r.t. a class $\mathcal{H}$, is, for a function f, having either k pairwise

class of graphs [84].

This field of research had important consequence for graph algorithms. It allowed extending the classes of graphs for which FO model checking can be performed efficiently. This is explained in the next subsection.

1.3 Meta-theorems

Algorithmic meta-theorems are general results that identify broad classes of problems which can be solved efficiently when the input graph meets certain structural conditions. Informally, they are expressed as follows:

> *"Every problem expressible in some language $\mathcal{L}$ can be efficiently solved for the graphs of some class $\mathcal{G}$."*

Courcelle's theorem was the first major statement of this form, taking $\mathcal{L}$ as the MSOL-definable properties[3], and $\mathcal{G} = \{G \mid tw(G) \leq t\}$. These last two decades, a *proliferation of algorithmic meta-theorems is reported.* Let us divide them in three groups, the Courcelle-like ones that are using a tree-based measure of graph complexity, the ones doing FO-model checking for wide classes of graphs, and those working in less general classes but for a richer languages $\mathcal{L}$.

Courcelle's theorem and the wide usage of tree-width and variants in several areas motivated the search of more structural parameters that could be used in structural graph theory, and for designing algorithms. This led to several such parameters, that are also defined through a tree-based structure. The most-well known ones are *clique-width* and its equivalent one *rank-width*, because they admit a Courcelle-like theorem, where the tractable problems are those MSO_1-definable, a fragment of MSOL. We should also mention *mim-width* [4], which was very successful with its many algorithmic properties and applications in SAT solvers [89, 22]. However, all these contributions arising from Courcelle's theorem have the disadvantage of being closely connected to tree-based measures. It has been a successful challenge to go beyond these classes of graphs.

Among the graph classes where the above mentioned measures can be arbitrary large, we have planar graphs, of all its generalisations: minor-closed graph classes, bounded expansion classes, and the nowhere dense ones (see Section 1.2). For those graph classes, the expressive power of $\mathcal{L}$ must be reduced, as many MSOL-definable problems are already NP-hard, when restricted to them. Considering sparse graph classes, researchers realized that for such classes, many problems can be solved efficiently, in particular those definable in *first-order logic* (FO for short). This was confirmed, as among graph classes closed under subgraphs, it was shown that FO model-checking is tractable if and only if the graph class is nowhere dense [56].

FO model-checking can also be performed in dense graph classes. The work initiated in [57] for studying parameterized problems in permutation graphs had been surprisingly generalized to all graphs and yielded the so-called *twin-width* parameter, defined through a degree notion based on linear order decomposition [18]. Small twin-width graphs include many studied graph classes, sparse and dense, including those being minor-closed, those with bounded clique-width and many geometric graph classes, being the first of this kind. It has been shown that FO model checking is also tractable for graphs of bounded twin-width, an impressive generalization towards dense graph classes [18].

---

vertex-disjoint subgraphs in $\mathcal{H}$, or having $f(k)$ vertices whose deletions gives rise to a graph in $\mathcal{H}$.

[3] *Monadic second-order logic* (MSOL for short) is a logical language that extends FO logic and allow to express many NP-complete problems such as k-coloring, for any fixed k, or Hamiltonicity.

There are attempts to generalize the graph classes for which FO model-checking is known tractable (nowhere dense classes and graphs with bounded twin-width) through new graph complexity measures, namely *flip-width* and *merge-width*. Another way to present these generalizations is an attempt to revisit the class trichotomy by Nesetril and Ossona de Mendez through the lens of FO model-checking. Here, the three considered families are (1) the structurally sparse ones which are those that are FO-transductions[4] of sparse graph classes (those are conjectured to be exactly the *stable* graph classes studied by Shelah, see for instance [15]), (2) the NIP graph classes which are those that cannot produce all graphs by FO-transductions (for those, the *FPT NIP conjecture* states they correspond to the hereditary graph classes with tractable FO model-checking), and (3) the graph classes that are not NIP. All these characterizations have connected the classification program of graph classes into sparse vs dense with the notions of *VC-dimension* appearing in learning theory [95, 96, 38] and of *neighborhood complexity* appearing in metric graph theory [16, 14].

1.4 Fine grained complexity and parameterized algorithms

Parameterized complexity is a framework that aims at analyzing the running times of algorithms in finer details than classical complexity theory: rather than expressing running times solely as a function of the input size, it also considers the dependence on one or more parameters of the input. These parameters may be related to the solutions (e.g., their size) or to the structure of the input (e.g., its tree-width). Hence, the meta-theorems cited above form a particular type of parameterized algorithms. Among the desired notions of tractability, three play a central role. To define these notions, let us assume we are dealing with a problem whose input size is n and parameter is k. (For example, the problem could be of deciding whether an input graph on n vertices and tree-width k contains a Hamiltonian path: here n is the input size, and k is the parameter.) The first notion is that of *slice-wise polynomial* (XP) algorithms whose running times are of the form $n^{f(k)}$ for some computable function f, which is polynomial for bounded values of k. The second notion is the one of *fixed-parameter tractable* (FPT) algorithms whose running times are of the form $f(k) \cdot n^{O(1)}$, which are expected to be more efficient than XP algorithms when the parameter gets large. For both notions, research has been conducted in obtaining the optimal such functions f, leading to the notion of fine grained complexity. Finally, the third notion is that of kernelization, where the goal is to reduce the instance into one whose size is bounded by a function of the parameter. Problems admitting kernels trivially admit FPT algorithms, and so obtaining kernels of small size is considered as a better degree of tractability.

While not being limited to graph theory, the framework of parameterized complexity has been extensively developed in the context of graphs, as these structures offer numerous parameters, structural properties, and decomposition features that can be exploited by parameterized algorithms. Among the most significant developments in graph theory over the past two decades, one can highlight the emergence of new algorithmic techniques, the development of meta-theorems, the improvement of complexity bounds, and the use of new structural parameters, that we briefly survey here.

Algorithmic techniques. Examples of recent successful algorithmic techniques include the *representative sets technique* [51] that led to efficient algorithms for "cycle-type"

[4] An FO-transduction is roughly a function from relational structures to relational structures, where the domains and relations of the target structures are defined by FO formulas on the input structures, after a non-deterministic k-coloring, for some fixed integer k, of the domains of the inputs.

problems, the *cut-and-count method* [32] that led to improved time bounds for a number of connectivity and acyclic problems, the rank-based approaches allowing deterministic algorithms for many cut-and-count based algorithms [6, 5] and *flow-augmentation techniques* for parametrized graph cut problems [66, 67, 68].

Kernelization. After a growing line of research consisting of characterizing parameterized problems admitting or not polynomial-size kernels, new techniques were found and have expanded the theoretical scope of kernelization. Examples of such successful techniques include the one of *cross-composition* introduced in [9], later applied in various context such as packing problems [36].

Meta-theorems. Algorithmic meta-theorems are general results that identify broad classes of problems which are fixed-parameter tractable (FPT) or admit small kernels when certain structural conditions are met. These last two decades, a proliferation of algorithmic meta-theorems is reported [37, 69], reaching the field of kernelization [8]. We already mentioned the large classes for which FO model checking can be performed efficiently. Further developments allow to cope with larger fragments of logic on fairly large graph classes [54].

Parameterized approximations. The topic of parameterized approximation aims at studying problems that cannot be solved exactly by an FPT algorithm, but that admit an approximation within the same time constraint. It has been growing since its initial stage [75], leading to a systematic study of classical problems such as knapsack problems [61] or domination-type problems [64].

Fine grained complexity. Since the work of Impagliazzo et al. on strong exponential time [60], fine grained complexity has brought a more nuanced view of computational hardness. Influential results include [71] related to tree-width, [24] related to domination-type problems, or [70] related to pathwidth. Fine grained complexity also considers determining for polynomial-time tractable problem such as computing the diameter of a graph, what is the optimal complexity of an algorithm solving this problem. A current trend in this domains aims at finding sub-quadratic time algorithms for the diameter problem [41].

2 New questions and new techniques

The last two decades have seen the emergence of many new questions arising from related domains, and of many new techniques. In the following we sample a few of them.

2.1 New models of algorithms

As we have already seen, graph theory naturally raises many algorithmic questions. Over the past twenty years, new algorithmic models have emerged, which the graph community has naturally embraced. In addition, the tools known in graph theory provide new elements for studying these new models. In this section, we briefly survey a few of them, from distributed computing and temporal graphs to enumeration problems through combinatorial games.

Distributed and local computing. Since a distributed network can be modeled as a graph, the distributed settings offer a new playground for graph algorithms, and the tools of graph theory can be used to improve the understanding of distributed computing. Let us take three concrete examples, among many.

1. Local algorithms have been designed for decades almost only in bounded-degree or general graphs, but thanks to the interaction with graph theorists, this has changed and

there are now distributed analogues of Courcelle's theorem for structured graphs [50] and efficient distributed approximation for minor-closed graphs [12].

2. A new point of view on graph classes has emerged through local certification, a notion stemming from the study of distributed fault-tolerance. It gives a new measure about how well one can check locally the structure of a graph. This has been done for example for minor-closed classes of graphs [20] and for geometric graph classes [35].
3. The blossoming notion of universal graph is at the intersection of graph theory, where it is a tool to understand a graph class, and of distributed computing, where it can serve for routing and for locally encoding adjacency. For example, in [13, 42], the authors uses the product structure theorem (see Section 2.3) to construct a universal graph that leads to good adjacency labelling schemes for planar graphs.

Temporal graphs. A temporal graph is a graph whose edges (and sometimes vertices) are present only at certain points in time. Temporal graphs occur naturally in transportation, communication networks, social networks, robotics, scheduling, and distributed computing. On the theoretical side, these graphs pose important challenges, as many classical concepts and techniques from standard graph theory do not carry on easily. For example, reachability based on temporal paths (paths crossing the edges chronologically) is neither symmetric nor transitive, with important algorithmic consequences. Two seminal papers documenting these effects are [7] and [65], showing respectively that computing a maximum temporal component (set of vertices that can reach each other) is NP-hard, and deciding if there exists two node-disjoint temporal paths between a given source and target is also NP-hard, both problems being polynomial-time solvable in static graphs. Another significant negative result is [3], showing that temporal graphs do not admit sparse spanners in general, i.e. there exist temporal graphs with $\Theta(n^2)$ edges, all of which are critical for connectivity. In recent years, a particular effort is devoted to understanding special cases where the above problems become tractable, for example, through the definition of temporal graph parameters that allows for FPT algorithms (see, e.g. [47]).

Combinatorial game theory. These games involve (generally) two players that take turns on a common board with perfect information. Typically, Alice and Bob alternately select vertices of an hypergraph, Alice wins if eventually she gets all vertices of an hyperedge and Bob wins otherwise [90]. This general definition of combinatorial games is intimately related to graphs and some of the most famous such games (Hex, Tic-tac-toe) are actually games played on grids. Recent progress have been done showing that such games are PSPACE-complete (resp., polynomial) when hyperedges have size 6 [85] (resp., 3 [52]). During the last decades, many combinatorial games defined through graphs have been studied [39, 40], leading to a better understanding of the complexity of these games and, in particular, of their many winning conventions (e.g., Client-Waiter [53]) and their link with reconfiguration [59]. The current research focuses on the limits of the tractability of these games, namely, given a game defined through graphs, in which graph classes it becomes polynomially solvable? Recently, these games have also been considered through the parameterized complexity point of view [17].

Algorithmic enumeration. Algorithmic enumeration is a cross-cutting theme that does not apply only to graphs. Many enumeration problems come from applications in BDD [73], bioinformatics [74], chemoinformatics, data mining, etc. The key issue remains whether it is possible to enumerate the minimal transversals of a hypergraph in output-polynomial time, the best time known to date being output-quasi-polynomial [46]. Nevertheless, there have been significant results in enumeration that have made it possible to solve algorithmic problems in graphs. For example, the Proximity Search technique [29] has

solved the enumeration of maximal subgraphs for a large number of classes admitting orders (chordal graphs, degenerate graphs, etc.), but is not restricted to graphs. Recently, new lines of research have emerged in enumeration: parameterized enumeration and approximate enumeration. We refer to the following survey presenting the computational complexity of enumeration problems and the status of many of them [92].

2.2 Probabilistic method and extremal graph theory

The probabilistic method is a central tool to prove results in (extremal) graph theory and prove the existence of some structures. Although these methods have been known for many years, recent advances have led to new results and proven certain conjectures. Here are a few examples illustrating activity in this area over the past twenty years.

- The constructive proof of the Lovasz's Local Lemma by Moser and Tardos in 2010 [77] – also known as the *entropy compression method* – has been widely used in graph theory, in particular to prove better bounds on colorings and, more generally, on the existence of certain structures in graphs. Recent refinements to this method [87] have yielded even more accurate results.
- In 2014, Bukh [23] presented a random algebraic method to obtain random constructions that are more rigid than with uniform random graphs. The analysis is more difficult and uses profound results from algebraic geometry. Another slightly different approach consists of starting with a rigid algebraic structure (a projective plane, for example), and then applying operations randomly. One of the most spectacular applications of this approach is the new lower bound on the Ramsey number $R(4,t)$ [76].
- The Kahn-Kalai conjecture, proposed in 2006, about the expectation threshold for random graphs has been proved in 2024 by Park and Pham [83]. The result and the technique introduced in the proof have already numerous consequences in probabilistic graph theory.
- Probabilities have also been used to make important progress on big conjectures of graph theory. For example, the Erdős–Faber–Lovász conjecture[5] has been proved in 2023 for large n using *Rödl's niddles* that are constructions of large matchings with an iterative probabilistic procedure [62].

2.3 Product structure theorem and applications

In 2019, Dujmović, Joret, Morin, Micek, Ueckerdt, and Wood proved an unexpected result, the so-called product structure theorem [44]. It asserts that every planar graph is a subgraph of a graph of the form $P \boxtimes H$, the strong product of a path and a graph of bounded tree-width. This structural theorem has been extended to several graph classes, such as graphs embedded on surfaces or graphs admitting an embedding with few crossings per edges.

This result allowed many advances on several longstanding open problems concerning planar graphs. Among others, it allowed proving that :

- Planar graphs have bounded queue-number [44], that is a vertex ordering and an edge partition into boundedly many queues (conjectured since 1992).
- Planar graphs can be colored non-repetitively with a constant number of colors [43]. In this type of coloring, for every even path P, the sequence of colors appearing along the

[5] This conjectures states that a graph made of n cliques of size at most n that intersect two-by-two on at most one vertex is n-colorable.

first half of P is distinct from the one of the second half. This problem was open since 2002.

Quantitatively, the product structure theorem allowed improvement on the number of colors for p-centered colorings [34], on the treedepth fragility [45], or on the size of the adjacency labelling schemes [42]. These results respectively improve the complexity of some parameterized algorithms, speed-up approximation algorithms, and reduce memory requirements in distributed algorithms.

2.4 Computer-assisted graph theory

One of the most famous results in graph theory, the Four Color Theorem, has a proof that contains several computer-assisted parts. With the increase of both memory capacity and processor efficiency (and parallelization), the magnitude of the possibilities for computer assistance has exploded in the last 20 years.

The computer assistance can be found in various contexts within graph theory. First, fast generators have been developed to generate all graphs from a given graph class exhaustively, taking the maximal size of a graph as an input. They are then used to test claims of the form "every graph with property P_1 also has property P_2", or to find the smallest counterexamples to such a claim. Secondly, graph databases (such as the House of Graphs) [30] are available for the researchers so that they can search for a graph with specific properties. Thirdly, various methods have been developed to test specific properties in graphs, which are usually computationally hard, for graphs of reasonable size. Those, are usually based on mixed integer linear program solvers, semi-definite programming solvers and SAT solvers.

Here are a few notable examples where computer assistance was essential to check a simple property many times.

- A famous problem of Hadwiger and Nelson asks for a minimum number of colors needed to color the plane such that no two points at distance 1 receive the same color. Colorings with seven colors are known, and examples of finite unit-distance graphs that need four colors have been known since the 60s. The lower bound was raised to five in 2018, when Aubrey de Grey found a 1581-vertex, non-4-colorable unit-distance graph, heavily relying on computer assistance to check the properties of the gadgets used in his construction [33].
- For the proof of the Barnette-Goodey conjecture [63] (stating that every 3-connected cubic planar graph with faces of size at most 6 is hamiltonian), the computer assistance was succesfully used to rule out a large number of rather small graphs.

There are also cases where the computer assistance lies more in the core of the proofs. For example, in [19] the authors used a routine based on linear programming to construct a set of "discharging rules" leading to progress on Wegner's conjecture about the distance 2-chromatic number of planar graphs with bounded maximum degree. In [88], the author uses the MSOL description of some graph properties, and multilinear algebra, to obtain tight asymptotic bounds on the number of many natural substructures of trees, with respect to their number of vertices n.

Proof assistants also have a history in our field. A little more than twenty years ago, a certified proof of the Four Color Theorem was obtained using Coq (now called Rocq). Today, LLMs enable better navigation in the literature, thereby enhancing researchers' work [55]. There are also a few examples of certified proofs that combine an LLM with a proof assistant [94]. It seems likely that these developments will have a major impact over the next twenty years.

Contributors.

Pierre Aboulker, Caroline Brosse, Arnaud Casteigts, Oscar Defrain, Louis Esperet, Daniel Gonçalves, Laurent Feuilloley, Mamadou Moustapha Kanté, Frantisek Kardos, Aurélie Lagoutte, Vincent Limouzy, Leandro Montero, Nicolas Nisse, Aline Parreau, Christophe Picouleau, Cléophée Robi, Ioan Todinca, Olivier Togni.

References

1 Pierre Aboulker, Pierre Charbit, and Reza Naserasr. Extension of Gyárfás-Sumner conjecture to digraphs. *The Electronic Journal of Combinatorics*, 28(2), May 2021.

2 Guillaume Aubian, Allen Ibiapina, Luis Kuffner, Reza Naserasr, Cyril Pujol, Cléophée Robin, and Huan Zhou. Extension of the gyárfás-sumner conjecture to signed graphs. 2025.

3 Kyriakos Axiotis and Dimitris Fotakis. On the size and the approximability of minimum temporally connected subgraphs. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, volume 55 of *LIPIcs*, pages 149:1–149:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. URL: https://doi.org/10.4230/LIPIcs.ICALP.2016.149, doi:10.4230/LIPICS.ICALP.2016.149.

4 Rémy Belmonte and Martin Vatshelle. Graph classes with structured neighborhoods and algorithmic applications. *Theor. Comput. Sci.*, 511:54–65, 2013. URL: https://doi.org/10.1016/j.tcs.2013.01.011, doi:10.1016/J.TCS.2013.01.011.

5 Benjamin Bergougnoux, Jan Dreier, and Lars Jaffke. A logic-based algorithmic meta-theorem for mim-width. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3282–3304. SIAM, 2023. URL: https://doi.org/10.1137/1.9781611977554.ch125, doi:10.1137/1.9781611977554.CH125.

6 Benjamin Bergougnoux and Mamadou Moustapha Kanté. More applications of the d-neighbor equivalence: Acyclicity and connectivity constraints. *SIAM J. Discret. Math.*, 35(3):1881–1926, 2021. URL: https://doi.org/10.1137/20M1350571, doi:10.1137/20M1350571.

7 Sandeep Bhadra and Afonso Ferreira. Computing multicast trees in dynamic networks and the complexity of connected components in evolving graphs. *J. Internet Serv. Appl.*, 3(3):269–275, 2012. URL: https://doi.org/10.1007/s13174-012-0073-z, doi:10.1007/S13174-012-0073-Z.

8 Hans L Bodlaender, Fedor V Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M Thilikos. (meta) kernelization. *Journal of the ACM (JACM)*, 63(5):1–69, 2016.

9 Hans L Bodlaender, Bart MP Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.

10 Marthe Bonamy, Nicolas Bousquet, and Stéphan Thomassé. The Erdős–hajnal conjecture for long holes and antiholes. *SIAM Journal on Discrete Mathematics*, 30(2):1159–1164, 2016.

11 Marthe Bonamy, Pierre Charbit, and Stéphan Thomassé. Graphs with large chromatic number induce $3k$-cycles. *arXiv preprint arXiv:1408.2172*, 2014.

12 Marthe Bonamy, Cyril Gavoille, Timothé Picavet, and Alexandra Wesolek. Local constant approximation for dominating set on graphs excluding large minors. In Alkida Balliu and Fabian Kuhn, editors, *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2025, Hotel Las Brisas Huatulco, Huatulco, Mexico, June 16-20, 2025*, pages 77–87. ACM, 2025. URL: https://doi.org/10.1145/3732772.3733531, doi:10.1145/3732772.3733531.

13 Marthe Bonamy, Cyril Gavoille, and Michał Pilipczuk. *Shorter Labeling Schemes for Planar Graphs*, pages 446–462. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.27, arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611975994.27, doi:10.1137/1.9781611975994.27.

14 Marthe Bonamy and Colin Geniet. χ-boundedness and neighbourhood complexity of bounded merge-width graphs. *CoRR*, abs/2504.08266, 2025. URL: `https://doi.org/10.48550/arXiv.2504.08266`, `arXiv:2504.08266`, `doi:10.48550/ARXIV.2504.08266`.

15 Édouard Bonnet, Samuel Braunfeld, Ioannis Eleftheriadis, Colin Geniet, Nikolas Mählmann, Michal Pilipczuk, Wojciech Przybyszewski, and Szymon Torunczyk. Separability properties of monadically dependent graph classes. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, July 8-11, 2025, Aarhus, Denmark*, volume 334 of *LIPIcs*, pages 147:1–147:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2025.147`, `doi:10.4230/LIPICS.ICALP.2025.147`.

16 Édouard Bonnet, Florent Foucaud, Tuomo Lehtilä, and Aline Parreau. Neighbourhood complexity of graphs of bounded twin-width. *Eur. J. Comb.*, 115:103772, 2024. URL: `https://doi.org/10.1016/j.ejc.2023.103772`, `doi:10.1016/J.EJC.2023.103772`.

17 Édouard Bonnet, Serge Gaspers, Antonin Lambilliotte, Stefan Rümmele, and Abdallah Saffidine. The parameterized complexity of positional games. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 90:1–90:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2017.90`, `doi:10.4230/LIPICS.ICALP.2017.90`.

18 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022. URL: `https://doi.org/10.1145/3486655`, `doi:10.1145/3486655`.

19 Nicolas Bousquet, Quentin Deschamps, Lucas de Meyer, and Théo Pierron. Square coloring planar graphs with automatic discharging. *SIAM Journal on Discrete Mathematics*, 38(1):504–528, April 2022. URL: `https://hal.science/hal-04960763`, `doi:10.1137/22M1492623`.

20 Nicolas Bousquet, Laurent Feuilloley, and Théo Pierron. Local certification of graph decompositions and applications to minor-free classes. *J. Parallel Distributed Comput.*, 193:104954, 2024. URL: `https://doi.org/10.1016/j.jpdc.2024.104954`, `doi:10.1016/J.JPDC.2024.104954`.

21 Nicolas Bousquet, Aurélie Lagoutte, and Stéphan Thomassé. The Erdős–hajnal conjecture for paths and antipaths. *Journal of Combinatorial Theory, Series B*, 113:261–264, 2015.

22 Simone Bova, Florent Capelli, Stefan Mengel, and Friedrich Slivovsky. On compiling cnfs into structured deterministic dnnfs. In Marijn Heule and Sean A. Weaver, editors, *Theory and Applications of Satisfiability Testing - SAT 2015 - 18th International Conference, Austin, TX, USA, September 24-27, 2015, Proceedings*, volume 9340 of *Lecture Notes in Computer Science*, pages 199–214. Springer, 2015. URL: `https://doi.org/10.1007/978-3-319-24318-4_15`, `doi:10.1007/978-3-319-24318-4\_15`.

23 Boris Bukh. Random algebraic construction of extremal graphs. *Bulletin of the London Mathematical Society*, 47, 2014. URL: `https://api.semanticscholar.org/CorpusID:12387209`.

24 Jianer Chen, Xiuzhen Huang, Iyad A Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8):1346–1367, 2006.

25 Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.

26 Maria Chudnovsky, Alex Scott, and Paul Seymour. Induced subgraphs of graphs with large chromatic number. iii. long holes. *Combinatorica*, 37(6):1057–1072, 2017.

27 Maria Chudnovsky, Alex Scott, Paul Seymour, and Sophie Spirkl. Induced subgraphs of graphs with large chromatic number. viii. long odd holes. *Journal of Combinatorial Theory, Series B*, 140:84–97, 2020.

28 Maria Chudnovsky, Alex Scott, Paul Seymour, and Sophie Spirkl. Pure pairs. i. trees and linear anticomplete pairs. *Advances in Mathematics*, 375:107396, 2020.

29 Alessio Conte and Takeaki Uno. New polynomial delay bounds for maximal subgraph enumeration by proximity search. In *Proceedings of the 51st Annual ACM SIGACT Symposium*

on Theory of Computing, STOC 2019, pages 1179–1190, New York, NY, USA, 2019. Association for Computing Machinery. URL: https://doi.org/10.1145/3313276.3316402, doi:10.1145/3313276.3316402.

30 Kris Coolsaet, Sven D'hondt, and Jan Goedgebeur. House of graphs 2.0: A database of interesting graphs and more. *Discrete Applied Mathematics*, 325:97–107, 2023. URL: https://www.sciencedirect.com/science/article/pii/S0166218X22004036, doi:https://doi.org/10.1016/j.dam.2022.10.013.

31 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Inf. Comput.*, 85(1):12–75, 1990. URL: https://doi.org/10.1016/0890-5401(90)90043-H, doi:10.1016/0890-5401(90)90043-H.

32 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan MM Van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Transactions on Algorithms (TALG)*, 18(2):1–31, 2022.

33 Aubrey D. N. J. de Grey. The chromatic number of the plane is at least 5, 2018. URL: https://arxiv.org/abs/1804.02385, arXiv:1804.02385.

34 Michał Dębski, Stefan Felsner, Piotr Micek, and Felix Schröder. Improved bounds for centered colorings. *Advances in Combinatorics*, 2021.

35 Oscar Defrain, Louis Esperet, Aurélie Lagoutte, Pat Morin, and Jean-Florent Raymond. Local certification of geometric graph classes. In Rastislav Královic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, August 26-30, 2024, Bratislava, Slovakia*, volume 306 of *LIPIcs*, pages 48:1–48:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.MFCS.2024.48, doi:10.4230/LIPICS.MFCS.2024.48.

36 Holger Dell and Dániel Marx. Kernelization of packing problems. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 68–81. SIAM, 2012.

37 Erik D Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3):292–302, 2008.

38 Jan Dreier and Szymon Torunczyk. Merge-width and first-order model checking. 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 1944–1955. ACM, 2025. URL: https://doi.org/10.1145/3717823.3718259, doi:10.1145/3717823.3718259.

39 Éric Duchêne, Valentin Gledel, Fionn Mc Inerney, Nicolas Nisse, Nacim Oijid, Aline Parreau, and Milos Stojakovic. Complexity of maker-breaker games on edge sets of graphs. *Discret. Appl. Math.*, 361:502–522, 2025. URL: https://doi.org/10.1016/j.dam.2024.11.012, doi:10.1016/J.DAM.2024.11.012.

40 Éric Duchêne, Valentin Gledel, Aline Parreau, and Gabriel Renault. Maker-breaker domination game. *Discret. Math.*, 343(9):111955, 2020. URL: https://doi.org/10.1016/j.disc.2020.111955, doi:10.1016/J.DISC.2020.111955.

41 Guillaume Ducoffe, Michel Habib, and Laurent Viennot. *Diameter computation on <italic>H</italic>-minor free graphs and graphs of bounded (distance) VC-dimension*, pages 1905–1922. 2020. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611975994.117, arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611975994.117, doi:10.1137/1.9781611975994.117.

42 Vida Dujmović, Louis Esperet, Cyril Gavoille, Gwenaël Joret, Piotr Micek, and Pat Morin. Adjacency labelling for planar graphs (and beyond). *Journal of the ACM (JACM)*, 68(6):1–33, 2021.

43 Vida Dujmović, Louis Esperet, Gwenaël Joret, Bartosz Walczak, and David R. Wood. Planar graphs have bounded nonrepetitive chromatic number. *Advances in Combinatorics*, 5:11, March 2020. URL: https://hal.science/hal-02165018, doi:10.19086/aic.12100.

44 Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R Wood. Planar graphs have bounded queue-number. *Journal of the ACM (JACM)*, 67(4):1–38, 2020.

45 Zdeněk Dvořák and Jean-Sébastien Sereni. On fractional fragility rates of graph classes. *The Electronic Journal of Combinatorics*, pages P4–9, 2020.

46 Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.

47 Jessica A. Enright, Samuel D. Hand, Laura Larios-Jones, and Kitty Meeks. Structural parameters for dense temporal graphs. In Rastislav Královic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, August 26-30, 2024, Bratislava, Slovakia*, volume 306 of *LIPIcs*, pages 52:1–52:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: `https://doi.org/10.4230/LIPIcs.MFCS.2024.52`, `doi:10.4230/LIPICS.MFCS.2024.52`.

48 Paul Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959. `doi:10.4153/CJM-1959-003-9`.

49 Paul Erdős and András Hajnal. Ramsey-type theorems. *Discrete Applied Mathematics*, 25(1-2):37–52, 1989.

50 Fedor V. Fomin, Pierre Fraigniaud, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. Distributed model checking on graphs of bounded treedepth. In Dan Alistarh, editor, *38th International Symposium on Distributed Computing, DISC 2024, October 28 to November 1, 2024, Madrid, Spain*, volume 319 of *LIPIcs*, pages 25:1–25:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: `https://doi.org/10.4230/LIPIcs.DISC.2024.25`, `doi:10.4230/LIPICS.DISC.2024.25`.

51 Fedor V Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 142–151. SIAM, 2014.

52 Florian Galliot. *Hypergraphs and the Maker-Breaker game : a structural approach. (Hypergraphes et jeu Maker-Breaker : une approche structurelle)*. PhD thesis, Grenoble Alpes University, France, 2023. URL: `https://tel.archives-ouvertes.fr/tel-04249805`.

53 Valentin Gledel, Nacim Oijid, Sébastien Tavenas, and Stéphan Thomassé. On the complexity of client-waiter and waiter-client games. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, July 8-11, 2025, Aarhus, Denmark*, volume 334 of *LIPIcs*, pages 89:1–89:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2025.89`, `doi:10.4230/LIPICS.ICALP.2025.89`.

54 Petr A. Golovach, Giannos Stamoulis, and Dimitrios M. Thilikos. Model-checking for first-order logic with disjoint paths predicates in proper minor-closed graph classes. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3684–3699. SIAM, 2023. `doi:10.1137/1.9781611977554.ch141`.

55 Timothy Gowers. Post on x. URL: `https://www.reddit.com/r/math/comments/1olctkm/gowers_on_using_ai_for_math_research/`.

56 Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *J. ACM*, 64(3):17:1–17:32, 2017. URL: `https://doi.org/10.1145/3051095`, `doi:10.1145/3051095`.

57 Sylvain Guillemot and Dániel Marx. Finding small patterns in permutations in linear time. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 82–101. SIAM, 2014. URL: `https://doi.org/10.1137/1.9781611973402.7`, `doi:10.1137/1.9781611973402.7`.

58 A. Gyárfás. Problems from the world surrounding perfect graphs. *Zastowania Matematyki Applicationes Mathematicae*, XIX:413–441, 1987.

59 Marc Heinrich. *Reconfiguration and combinatorial games. (Reconfiguration et jeux combinatoires)*. PhD thesis, University of Lyon, France, 2019. URL: `https://tel.archives-ouvertes.fr/tel-02294749`.

60 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

61 Klaus Jansen. Parameterized approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 39(4):1392–1412, 2010.

62 Dong Kang, Tom Kelly, Daniela Kühn, Abhishek Methuku, and Deryk Osthus. A proof of the Erdős–Faber–Lovász conjecture. *Annals of Mathematics*, 198(2):537 – 618, 2023. URL: https://doi.org/10.4007/annals.2023.198.2.2, doi:10.4007/annals.2023.198.2.2.

63 Frantisěk Kardoš. A computer-assisted proof of the barnette–goodey conjecture: Not only fullerene graphs are hamiltonian. *SIAM Journal on Discrete Mathematics*, 34(1):62–100, 2020.

64 CS Karthik, Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. In *50th Annual ACM Symposium on Theory of Computing*, pages 1283–1296, 2018.

65 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002. URL: https://doi.org/10.1006/jcss.2002.1829, doi:10.1006/JCSS.2002.1829.

66 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation II: undirected graphs. *ACM Trans. Algorithms*, 20(2):12, 2024. URL: https://doi.org/10.1145/3641105, doi:10.1145/3641105.

67 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation I: directed graphs. *J. ACM*, 72(1):5:1–5:38, 2025. URL: https://doi.org/10.1145/3706103, doi:10.1145/3706103.

68 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation III: complexity dichotomy for boolean csps parameterized by the number of unsatisfied constraints. *SIAM J. Comput.*, 54(4):1065–1137, 2025. URL: https://doi.org/10.1137/23m1553698, doi:10.1137/23M1553698.

69 Stephan Kreutzer. Algorithmic meta-theorems. In *International Workshop on Parameterized and Exact Computation*, pages 10–12. Springer, 2008.

70 Michael Lampis. The primal pathwidth SETH. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1494–1564. SIAM, 2025.

71 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Known algorithms on graphs of bounded treewidth are probably optimal. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 777–789. SIAM, 2011.

72 Nikolas Mählmann. *Monadically stable and monadically dependent graph classes: characterizations and algorithmic meta-theorems*. PhD thesis, Bremen University, Germany, 2024. URL: https://media.suub.uni-bremen.de/handle/elib/8304, doi:10.26092/ELIB/3338.

73 Heikki Mannila and Kari-Jouko Räihä. *The design of relational databases*. Addison-Wesley Longman Publishing Co., Inc., 1992.

74 Andrea Marino. *Analysis and enumeration: algorithms for biological graphs*, volume 6. Springer, 2015. doi:https://doi.org/10.2991/978-94-6239-097-3.

75 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

76 Sam Mattheus and Jacques Verstraete. The asymptotics of $r(4,t)$. *Annals of Mathematics*, 1999(2), 2024. URL: https://arxiv.org/abs/2306.04007, arXiv:2306.04007.

77 Robin A. Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *J. ACM*, 57(2), February 2010. URL: https://arxiv.org/abs/0903.0544, doi:10.1145/1667053.1667060.

78 Jaroslav Nesetril and Patrice Ossona de Mendez. Linear time low tree-width partitions and algorithmic consequences. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 391–400. ACM, 2006. URL: https://doi.org/10.1145/1132516.1132575, doi:10.1145/1132516.1132575.

79 Jaroslav Nesetril and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *Eur. J. Comb.*, 27(6):1022–1041, 2006. URL: https://doi.org/10.1016/j.ejc.2005.01.010, doi:10.1016/J.EJC.2005.01.010.

80 Jaroslav Nesetril and Patrice Ossona de Mendez. Grad and classes with bounded expansion i. decompositions. *Eur. J. Comb.*, 29(3):760–776, 2008. URL: https://doi.org/10.1016/j.ejc.2006.07.013, doi:10.1016/J.EJC.2006.07.013.

81 Jaroslav Nesetril and Patrice Ossona de Mendez. Grad and classes with bounded expansion III. restricted graph homomorphism dualities. *Eur. J. Comb.*, 29(4):1012–1024, 2008. URL: https://doi.org/10.1016/j.ejc.2007.11.019, doi:10.1016/J.EJC.2007.11.019.

82 Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012. URL: https://doi.org/10.1007/978-3-642-27875-4, doi:10.1007/978-3-642-27875-4.

83 Jinyoung Park and Huy Tuan Pham. A proof of the kahn-kalai conjecture. *J. Amer. Math. Soc.*, pages 235–243, 2024. URL: https://arxiv.org/abs/2203.17207, arXiv:2203.17207.

84 Christophe Paul, Evangelos Protopapas, Dimitrios M. Thilikos, and Sebastian Wiederrecht. Obstructions to erdös-pósa dualities for minors. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 31–52. IEEE, 2024. URL: https://doi.org/10.1109/FOCS61266.2024.00013, doi:10.1109/FOCS61266.2024.00013.

85 Md Lutfar Rahman and Thomas Watson. 6-uniform maker-breaker game is pspace-complete. *Comb.*, 43(3):595–612, 2023. URL: https://doi.org/10.1007/s00493-023-00026-7, doi:10.1007/S00493-023-00026-7.

86 Felix Reidl, Fernando Sánchez Villaamil, and Konstantinos S. Stavropoulos. Characterising bounded expansion by neighbourhood complexity. *Eur. J. Comb.*, 75:152–168, 2019. URL: https://doi.org/10.1016/j.ejc.2018.08.001, doi:10.1016/j.ejc.2018.08.001.

87 Matthieu Rosenfeld. Another Approach to Non-Repetitive Colorings of Graphs of Bounded Degree. *The Electronic Journal of Combinatorics*, 27(3), July 2020. URL: https://hal.science/hal-03583287, doi:10.37236/9667.

88 Matthieu Rosenfeld. The growth rate over trees of any family of sets defined by a monadic second order formula is semi-computable. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 776–795. SIAM, 2021.

89 Sigve Hortemo Sæther, Jan Arne Telle, and Martin Vatshelle. Solving #sat and MAXSAT by dynamic programming. *J. Artif. Intell. Res.*, 54:59–82, 2015. URL: https://doi.org/10.1613/jair.4831, doi:10.1613/JAIR.4831.

90 Thomas J. Schaefer. On the complexity of some two-person perfect-information games. *J. Comput. Syst. Sci.*, 16(2):185–225, 1978. URL: https://doi.org/10.1016/0022-0000(78)90045-4, doi:10.1016/0022-0000(78)90045-4.

91 Alex Scott and Paul Seymour. Induced subgraphs of graphs with large chromatic number. I. Odd holes. *Journal of Combinatorial Theory, Series B*, 121:68–84, 2016.

92 Yann Strozecki. Enumeration complexity. *Bulletin of EATCS*, 1(129), 2019.

93 David P Sumner. Subtrees of a graph and chromatic number. *The theory and applications of graphs*, pages 557–576, 1981.

94 Terence Tao. Post on mathstodon. URL: https://mathstodon.xyz/@tao/115493667607261044.

95 Szymon Torunczyk. Flip-width: Cops and robber on dense graphs. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 663–700. IEEE, 2023. URL: https://doi.org/10.1109/FOCS57990.2023.00045, doi:10.1109/FOCS57990.2023.00045.

96 Szymon Torunczyk. Evaluating first-order formulas in structured graphs (invited talk). In Sudeepa Roy and Ahmet Kara, editors, *28th International Conference on Database Theory, ICDT 2025, March 25-28, 2025, Barcelona, Spain*, volume 328 of *LIPIcs*, pages 3:1–3:2. Schloss

Dagstuhl - Leibniz-Zentrum für Informatik, 2025. URL: https://doi.org/10.4230/LIPIcs.ICDT.2025.3, doi:10.4230/LIPICS.ICDT.2025.3.

97 Xuding Zhu. Colouring graphs with bounded generalized colouring number. *Discret. Math.*, 309(18):5562–5568, 2009. URL: https://doi.org/10.1016/j.disc.2008.03.024, doi:10.1016/J.DISC.2008.03.024.

Twenty years of GdR IFM, seen from GT IQ

1 Context

Quantum computing is a quite young field of research, initiated in the early 80's when Feynman expressed the idea of using well-controlled quantum systems to simulate quantum physics, and which really took off in the mid 90's when Shor and Grover devised quantum algorithms that have a better complexity than their classical counterparts, respectively for prime factorisation and for the unstructured search problem. This aspect constitutes one of the best motivations for the field: the prospect of solving *some* problems more efficiently than what is possible classically, potentially exponentially so, depending on the problem at hand.

Quantum computing is also peculiar in that on the one hand, it covers a vast area of computer science – algorithmics, complexity theory, cryptography, error correcting codes, formal methods, artificial intelligence, ... –, and on the other hand, it is in close contact with physics. Indeed, the existing quantum computers are still in their infancy, and working with them efficiently for practical applications requires a deeper understanding of the underlying physics than what is required nowadays on a classical computer.

Long considered a relatively niche and quirky area of research, quantum computing has gained in traction over the past two decades, both in the academic and the industrial worlds. In the first, this is well illustrated by the surge in research positions in the domain, the integration of quantum computation to master programs, or even the creation of dedicated master programs to the topic, the development of conferences and workshops, etc... On the industrial side, big companies such as Google, Microsoft, IBM, Amazon, or Atos/Bull in France have started a serious quantum-related activity, while several startups have been brought to life, such as, just in France, Quandela, Pasqal, Alice&Bob, C12, Quobly, Welinq, ColibriTD, VeriQloud, etc... On top of these several big actors (EDF, Total, Crédit Agricole, ...) have begun monitoring developments in quantum computing technology in order to better assess the potential benefits of such technology for their businesses. All this now constitutes a rich ecosystem, fostered by recent, massive public and private funding (see e.g. the "Plan Quantique National").

2 Quantum algorithms and complexity

The development of quantum algorithms has been a driving force behind the rapid advancement of quantum computing since its early stages. While the field began with quantum algorithms for toy problems [53, 130], some of the main quantum algorithms, even after all progress in the field, are the foundational quantum algorithm to factor numbers in polynomial time [129] or unstructured database search [78].

In the 2000s, theoretical advancements focused on refining and generalizing quantum algorithms, often with ideas that are inspired both from Computer Science and Physics. Techniques that were fundamental in the first quantum algorithms were refined, generalized and culminated into more general technique that could be applied in different contexts, such as the quantum Fourier transform, which has fundamental to quantum learning theory [13], and the quantum amplitude amplification/estimation [21], that gives a quadratic speedup on "sampling" from a desired event. We also saw the development and consolidation of models of computation that had far-reaching applications such as Adiabatic Quantum

Computing [63] or Quantum Walks [4]. In adiabatic quantum computing, we encode the solution of a (potentially hard) problem into a Hamiltonian, a physical object that describes the evolution of quantum systems, and its groundstate, the state that minimizes the energy of the Hamiltonian. However, finding the groundstate of such a Hamiltonian directly is itself a hard problem. Adiabatic quantum computing offers a solution for it: we start from the groundstate of an easy Hamiltonian, and gradually turn it into the groundstate of the Hamiltonian of interest. Initially proposed for solving specific optimization problems, it was later shown that adiabatic quantum computing is an universal model for quantum computing [63], and it is a building block of modern frameworks such as VQE and QAOA that we describe below. Quantum walks emerged as a powerful framework within quantum algorithm design, offering a quantum analogue to classical random walks. Unlike their classical counterparts, quantum walks leverage superposition and interference to explore complex spaces more efficiently, enabling exponential speedups for certain problems [38] and they have found applications in a wide range of areas. We also have seen the proposal of new algorithmic techniques, such as HHL [82], an algorithm for solving linear systems of equations exponentially faster than classical methods, paving the way for more recent applications in machine learning and data analysis [92, 20]. Moreover, quantum algorithms also played a key role in the development of post-quantum cryptography, both from a cryptanalysis perspective, and as a tool to provide security proofs, such as celebrated the worst-case to average-case quantum reduction by Regev to support the hardness of LWE [121].

The 2010s and 2020s marked the beginning of the Noisy Intermediate-Scale Quantum (NISQ) era, characterized by the availability of quantum computers with limited qubits and high error rates. This era shifted the focus to hybrid quantum-classical algorithms, which combined quantum and classical computing to mitigate hardware limitations. Algorithms like the Variational Quantum Eigensolver (VQE) [119] and the Quantum Approximate Optimization Algorithm (QAOA) [62] were developed to tackle optimization and quantum chemistry problems, making quantum computing more accessible for near-term applications. The rise of quantum machine learning also gained momentum, with algorithms such as Quantum Support Vector Machines (QSVM) [120] and Quantum Neural Networks (QNNs) [125] exploring the intersection of quantum computing and artificial intelligence. Interestingly, some of these quantum machine learning algorithms have been later *dequantized* [134], leading to new efficient classical algorithms for a variety of problems. Moreover, new tools and frameworks, such as Quantum Singular-Value Transformation (QSVT) and block-encoding of matrices [72], provide a new conceptual way of developing quantum algorithms, and provide a significant improvement for important physical tasks such as Hamiltonian simulation.

From a complexity perspective, we have seen many fundamental results in the early 2000s. With Kitaev's seminal result on defining QMA, the quantum version of NP, and proving that the local Hamiltonian problem, a natural problem in condensed matter physics, is QMA-complete [96], initialized the field which is now called Hamiltonian complexity [70]. This field has gained a lot of attention since it not only allows us to study problems related to complexity theory and hardness of approximation, but also enables the study of the structure in low-energy states of interest from a Physics perspective. In parallel, we have also seen the development of techniques for proving lower-bounds in the quantum setting in different setups such as query complexity [9, 17] and communication complexity [32]. Such techniques have also influenced other fields in classical TCS, e.g., it has been shown, via lower bounds on quantum communication complexity, that TSP requires superpolynomial linear programs [65]. In the 2010s, we have seen the development of Hamiltonian complexity, and the study of the quantum PCP conjecture [5]. The classical PCP theorem connects

many key topics of classical complexity theory: hardness of approximation, efficient proof verification, hardness of finding the best game strategies. One of the most important open questions in quantum complexity theory today states whether a similar conjecture also holds in the quantum setting. We have also seen the development of quantum interactive proof systems, which extend NP and adds interaction between polynomially bounded verifiers and unbounded provers. Using techniques from classical optimization, it has been shown that in the single prover case, quantum and classical interactive proof systems have the same computational power [87]. However, in the multi-prover case, we have seen the breakthrough result that multiple untrusted entangled provers can be used to prove even undecidable problems such as the Halting problem [89]. This has not only profound implications for complexity theory, but one of its main consequences is to disprove the long-standing Connes embedding conjecture in Operator Algebra that had been open for decades [137].

3 Quantum cryptography and communication

Quantum cryptography emerged as an area of interest following the invention of quantum key distribution (QKD) [18] in the 1980s, marking a pivotal moment in the history of cryptography. The development of QKD represented a significant leap forward, introducing a method by which two parties could share a secure key over an insecure channel without any risk of interception by eavesdroppers. Since its inception, the field of quantum cryptography has expanded rapidly, addressing many privacy related problems that classical cryptography was previously unable to solve. A prominent example is the concept of quantum money [140], which presents a groundbreaking approach to using the "no-cloning" principle of quantum physics to create digital currency that cannot be counterfeited. As quantum cryptography has continued to evolve, it has been realized that information-theoretic security (i.e. security against even unbounded adversaries) also has its limitations in the quantum setting [106, 109].

In the 2000s, the field of quantum cryptography was mainly focused on improving the security and the efficiency of QKD protocols. First of all, there was a lot of effort in defining the desired security definitions [123], pushing protocols closer to practical setups [77] and proving their security – which sometimes come many years later [102]. Many of these improvements, which are still currently being pushed forward, come from redefining notions of information theory to the quantum setting [141], leading to a plethora of quantities such as quantum entropies and divergence, that enable tighter security proofs leading to improvements on QKD rate. Secondly, there has been a lot of effort in finding models under which we can achieve cryptographic protocols with provable security. Two examples of these directions are the bounded storage model [47], where the adversary has only limited quantum memory, or verifiable delegation of quantum computation: a classical client is able to delegate some quantum computation to a quantum server, who can solve the computation and prove to the client that the output is correct. The first protocols managed to prove this type of result but with clients with limited quantum resources [27, 66, 26, 7].

In the 2010s and 2020s, there was an impressive growth of the field, which branched in many directions. Firstly, new security notions such as device-independence, allowed us to find protocols where we have minimal trust in the quantum devices used in cryptographic protocols, e.g. in QKD [136] or to achieve protocols for verifiable delegation with quantum computation by classical clients [122, 44]. Secondly, inspired by classical cryptography, there was a big development on quantum cryptography under computational assumptions. Here, the goals are two-fold: to provide quantum protocols for classical functionalities while relying on a weaker computational assumption [76, 16], but also to provide protocols for quantum

functionalities [58, 55]. Within this research direction, it has been realized that considering assumptions that are inherently quantum (such as the existence of pseudo-random states), we can have computational cryptography even if P = NP [88, 100], which cannot be done classically. There is an intense line of research in quantum cryptography to understand the difference of variants of quantum cryptographic assumptions, and in particular to find the minimal one. Finally, there was also the "reborn of uncloneability". It has been shown that the fact that quantum states cannot be generically cloned can be used to achieve many more applications than just quantum money schemes, and we have today many protocols with one-time properties, i.e. we can prove that their functionalities can be used a single time [80], with uncloneable properties [29, 146], where the information is exclusively revealed to a single party, or even with certified deletion, guaranteeing that an untrusted party has destroyed information [28].

4 Quantum error-correction and fault-tolerant quantum computation

As quantum hardware is expected to be quite noisy, hence prone to errors, it is paramount to try and mitigate them. The first quantum error correcting code is due to Shor in the 90s, who showed in particular that through measurements, continuous errors could actually be brought to only two kinds of discrete errors: bit-flips and phase-flips. Over the years, many new error correction schemes, that heavily rely on the theory of classical codes, have been proposed [103, 14, 60, 105]. Notice that to have a good, practical error correction scheme, it is not enough to have a good code, as one also needs to efficiently decode it (which in all generality is an NP-hard problem for linear codes).

CSS codes are special cases of quantum error correcting codes, that can be built out of two classical linear codes, making their study easier. This constitutes one of the most studied classes of codes. To date, one of the codes that seems the most promising – and which turns out to be a CSS code –, due to its conceptual simplicity, its good theoretical performance, and its compliance with most reasonable qubit layouts, is the surface code [67]. It was introduced in the early 2000's, building on top of Kitaev's toric code [97], and it encodes a single logical qubit into an array of $n \times n$ physical qubits – the size of the array allowing us to control the error tolerance of the scheme: the code distance scales as $\sqrt{n}$.

This scaling capability is however not on par with the best classical codes, whose distance and number of logical qubits scale linearly with n. A lot of effort has hence been devoted to the study of *low-density parity check codes* [23], which benefit from efficient decoding algorithms and can approach channel capacities in the classical case. A breakthrough was [135] that shows that quantum LDPC codes may have both a non-vanishing rate[1] and a better-than-logarithmic distance. The question of the existence of an LDPC code with linear rate and linear distance was open for a long time and only positively answered in 2021 [113, 104].

This is particularly important if one wants to overcome the *threshold* problem [6, 99]: applying an error correcting scheme implies applying to the quantum memory some additional quantum gates, which themselves can bring new errors, hence, for an error correcting code to be useful on a given hardware, the error rates of the gates have to be below a certain threshold. The threshold of course depends on the chosen scheme and its properties. The threshold theorem has been extended in [64] for constant-overhead fault-tolerance using expander codes.

[1] the ratio of the number of logical qubits over the number of physical qubits

A recent avenue for obtaining better such threshold is dynamical codes, such as Floquet codes [83, 54] i.e. stabilizer codes (themselves a generalisation of CSS codes) that evolve over time. They may circumvent certain no-go theorems that apply to static stabilizer codes, exhibit reduced complexity in error detection, can often be implemented on lattices with low qubit connectivity and employ simple, two-qubit measurements.

Quantum error correction obviously benefits from results in its classical counterpart, but it so happens that the converse is also true. For instance, a result on locally decodable codes was provided in [91], following a quantum argument. Since then, several results in quantum error correction have been provided hand-in-hand with their classical counterparts (e.g. [25, 113]).

An important aspect of error correction is the ability to translate an algorithm or protocol defined on ideal logical qubits, into the encoded realm (e.g. a logical 1-qubit gate will translate into a series of logical gates on the array of qubits used for the surface code). Most error correction schemes natively handle a non-universal subset of gates (called Clifford gates), and need an extra push to reach universal computation. This additional computational resource is usually provided as a state (called *magic state*) [35, 71], which is then incorporated into the circuit. Providing this resource as a state allows us to better control the quality of that resource. Since magic states are usually hard to create, they may go through a round of distillation, whereby we take several low-quality magic states and create a better-quality one. Interestingly, this process itself uses error correcting codes.

With the advent of actual quantum computers, an obvious goal for manufacturers was to reach the threshold for a given error correction scheme using their hardware. This achievement has only recently been announced for the first time [60].

Physicists have also approached the problem of qubit robustness, and have proposed new physical implementations of qubits that allow to correct errors at a more fundamental level. The cat qubit [61], for instance, is a quantum harmonic oscillator that makes use of its coherent states as its logical 0 and 1 states, and which at a more abstract level can be considered as a qubit that is free from phase-flip errors. In such systems, one can go back to classical codes to correct for the remaining type of errors (bit-flips).

With such a setup that is easier to handle, a manufacturer with fault-tolerant quantum computation as its goal can more easily try to *co-design* its chip together with an error-correcting scheme. Since the qubit layout will most probably remain 2-dimensional (array-like), one may use the 2nd dimension in the case of the cat-qubit as a way to implement 2D classical codes [124].

5 Programming languages and compilation

Twenty years ago, several programming languages were already introduced [126], trying to identify and promote concepts deemed useful for quantum computation, that would allow the programmer to abstract away from the low-level inner workings of a putative quantum computer, and reason at a more conceptual level. A few imperative approaches existed at the time, that introduced run-time checks to ensure the validity of the execution of the program. This approach arguably raises several problems, first due to the probabilistic nature of quantum computation (is there an error because of a bug in the code, or because the quantum data unexpectedly changed?), and second because of the cost of quantum resources: one may prefer to classically and statically check the program before it is run on a quantum machine. This is allowed by functional programming languages.

All of these prototypical programming languages were either theoretical constructions

meant to study the properties of the concepts used as commands for the language, or simply lacked the ability to program large-scale algorithms. One of the first practical programming languages that overcomes this scalability issue is Quipper [75]. It is a higher-order functional programming language embedded in Haskell, and it comes with a categorical semantics.

Several scalable programming languages have since then been developed, often linked to a specific platform (Qiskit, developed and used by IBM, Q# by Microsoft, MyQLM by Atos/Eviden, Cirq by Google, Perceval by Quandela...). These usually have a preferred gate set (driven by the quantum hardware they interface with), and have to compile, in their own way (e.g. [132, 84]), the high-level code provided by the user to this set of gates. On top of complying with the imposed gate set, one has to account for the topological constraints of the quantum chip: indeed, physically, qubits have a given geographical location, and multi-qubit gates usually cannot be applied on qubits that are far apart. Solutions to this problem often use heuristics to classical optimisation problems, e.g. from graph theory in [131].

Since the lifetime of quantum memory is short and quantum gates may incur new errors, it is paramount to make the quantum circuit we want to execute as short as possible (without changing the results of the computation). Different metrics exist, depending on the setting and the kind of errors we want to avoid the most: circuit depth, circuit size (or gate count), number of ancillas (the additional qubits that are used for storage during the computation), ... Quantum volume is a more recent notion that takes into account both the full memory size, and either the depth or the gate count, leading to a more practical metric that one wants to minimise when optimising a circuit, but which is also used by manufacturers to gauge the quality of their device [46] (they look at the maximum volume they can get before the error rate becomes prohibitive). On most fault-tolerant architectures, the Clifford subset of gates is much simpler to implement than the others, it is hence customary to try and optimise the non-Clifford gate count. In other settings, two-qubit gates incur more errors than single-qubit ones, hence we are interested in those cases in the two-qubit gate count. In certain fragments of quantum computation, optimisation is fairly simple and can be done in polynomial time [143, 108], in other cases, heuristics are being sought to reach a better but non-optimal result [10, 110, 31].

Many quantum algorithms require very fine rotation gates (e.g. the QFT in Shor's algorithm that requires the $\frac{\pi}{2^n}$ on $n-1$ qubits). Such gates are believed to be very hard to implement, especially fault-tolerantly. Thankfully, the Solovay-Kitaev theorem [95] provides a way to turn any such ideal circuit into a circuit that uses a fixed (universal) gate-set with a very good gate-count w.r.t. the target accuracy of the transformation. This complexity has even been improved for specific gate sets, such as Clifford+T [127].

Compilation is also concerned with integrating classical data (part of the problem's parameters) into the quantum circuit. A main historical such component is the so-called quantum oracle, which, given a boolean function f, implements either $|x\rangle \mapsto (-1)^{f(x)} |x\rangle$ or $|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$, for which there now exist several synthesis methods (e.g. [68, 144]).

Other such primitives include state preparation (encoding a given vector into a quantum state) [74, 148], or block-encoding (encoding an arbitrary given matrix as part of a quantum circuit) [133]. These last routines have recently gained interest because of their use in the powerful QSP and QSVT algorithms [73].

6 Formal methods and models of computation

Formal methods for quantum computation arose when the first quantum programming languages were defined, and needed to be given a semantic. As discussed above, some of

their most important motivations is the analysis and providing of guaranties to quantum programs, in a classical manner.

Theoretical quantum programming languages, such as the quantum λ-calculus [128], can be defined specifically with a given set of paradigmatic features in mind (linearity, higher-order, ...). It is then possible to study the properties of such sets of features, and ensure some properties on the language and its semantic, such as type safety, soundness (the fact the semantic cannot identify observationally different programs), full abstraction (the fact that the semantic identifies two programs if and only if they are observationally equivalent, see e.g. [40] for the result for the quantum λ-calculus). Although not meant to be practical, these may serve as foundations for programming languages such as those mentioned in the previous section.

One may for instance make use of deductive verification (using Hoare triple) together with an intermediate language to give and check specifications, as in the QBricks system [37], where algorithms such as Shor's were verified using the Why3 proof assistant. Other verification frameworks make for instance use of Rocq [115]. Other methods to improve the trust in the fact that an implementation is bug-free include type systems [48] and abstract interpretations [118, 145]. One may also make sure that the cost of the quantum circuit that is being built is under control, for instance by using implicit complexity (being able to write the program in a dedicated language ensures the polynomial time of the execution [81]), or by assigning a semantic to the program that takes this cost into account (e.g. [15]).

Formal methods are also being used to verify or certify the classical processing of the program. For instance, certain optimisers for quantum circuits has been checked using the Rocq proof assistant [85].

Low-level formal methods are interested in the gate-based behaviour and verification of quantum circuits and similar graphical representations. Quantum decision diagrams [142] for instance make use of a weighted version of decision diagrams – usually used for boolean function analysis – to perform equivalence checks of quantum circuits. They have been used to check the compilation flow of the Qiskit software [33]. More complex automata-based structures (e.g. [3]) are also being investigated and provide very good results.

The equations that govern the behaviour of quantum logic gates can be used to perform verification or to simplify the circuits we consider, by simple local manipulations of the circuit. It is natural to wonder whether a given set of such equations entirely captures the semantical equivalence of circuits. This central property is called the completeness of the equational theory. The first complete presentation for quantum circuits was only given recently, in 2023 [42]. Interestingly, it derives from an analogous result in linear optics. The result has since been improved [41], to reach minimality: the fact that none of the equations in the presentation is a consequence of the others.

Category theory has been used to provide alternative representations of quantum processes, in a way that in particular eliminates the rather rigid unitarity constraint, leading to the family of ZX-like calculi [43], whose main feature is the ability to deform the diagrams (representations of quantum operators) at will, without changing their semantics. In practice, such diagrams get rid of a lot of bureaucracy and become easier to manipulate than circuits. Completeness has also been one of the focuses there [138], and on top of verification [116], these can be used for instance in circuit optimisation [56, 93] and emulation of quantum circuits [94], or as it was first intended, to bridge between different models of quantum computation [57, 50, 36].

Indeed, the prominent way to devise and represent computations, the circuit model, is not the only one at our disposal. In this model, the focus is put on the unitary part (i.e. where

there is no interaction with the environment), as measurements can be deferred to the very end [111]. Another model of computation, the measurement-based model, instead pushes the core of the computation to the measurements, after a round of entangling gates [24]. Due to the probabilistic nature of measurements, the question of performing deterministic computations by applying corrections later in the execution has become central. Solutions to this question interestingly serve as a way to turn from a measurement-based scheme to a full-fledged quantum circuit [49, 30]. We may also mention the topological quantum computer which performs computations by manipulating anyons and swapping them [98] (although the existence of the kind of anyons required here is still unsure, but now benefits from good evidence [86]). Since this operation is non-involutive, computations can be modeled by braids from knot theory.

We have already mentioned adiabatic quantum computing in the context of algorithms and complexity, with results like [8] showing its polynomial equivalence with quantum circuits, hence providing a bridge between the continuous and the digital worlds. Another such bridge can be made by discretising the continuous space-time and simulating the evolution of a quantum system using circuits. This results in quantum cellular automata, whose local functioning is very simple, and from which emerge the physical phenomena one wants to simulate. The main concern of quantum cellular automata is quite naturally to ensure that its evolution converges to the simulated one when the time steps and space steps tend to 0 [12].

We close this section by mentioning a way to generalise quantum circuits that came to light in recent years: the capacity, allowed by quantum theory, to have superpositions of orders of execution of operators [39, 101], not only of data. The study of indefinite causal order is an ongoing research topic that may redefine the way we think about quantum computation, and with potential far reaching consequences, for instance due to its link with quantum gravity [52] (to quickly explain that link, relativity tells us that masses bend space-time around them, but quantum theory tells us that a particle may be in a superposition of two different places, resulting in a superposition of two space-time curvatures, where the causal order of two distinct events may be different). The most prototypical example of this new resource is the quantum switch, which, given two circuits U and V and an input state $|\psi\rangle$, applies a superposition of the two possible orders of U and V to $|\psi\rangle$, i.e. $|\psi\rangle \mapsto (\alpha UV + \beta VU)\,|\psi\rangle$. Physical realisations of this have been performed, although it is still debated whether it really is the quantum switch that was implemented or merely a simulation of it. A generalisation of the quantum switch may be used to show that indefinite causal orders yield a computational advantage over fixed-ordered quantum computations [11].

7 Near-term algorithms and quantum advantage

With the development of quantum devices, we have achieved what is called the NISQ area, where we have **N**oisy **I**ntermediate-**S**cale **Q**uantum devices. While these devices are not sufficient for implementing quantum algorithms with provable advantage such as Shor, HHL, and others, they are a landmark on the progress on quantum technologies and they open an era where we can start to investigate the behaviour of quantum devices in practice.

More concretely, with access to quantum devices, we can now start to study how quantum heuristics behave to solve problems of interest in practice. For example, the variational algorithms (see above) have been developed to accommodate these limitations. They use short, parameterised quantum circuits whose parameters are to be optimised classically. These can in theory be applied to many problems, but they face many challenges (in particular the so-called barren plateau, where the parameters' optimisation seems to be stuck in a local

but not global optimum). Lately, they have drawn attention for their potential for machine learning [45], they seem to be well designed for quantum chemistry (i.e. simulating chemical systems) [79].

The promise of solving some problems more efficiently quantumly than classically, has fueled the development of quantum computers, together with the definition of relevant problems for such demonstration, and provides a good way to assess the validity of a quantum device. The goal is to perform a computation on a quantum computer, that would take much longer classically – the precise scales vary from an author to the other, but think a few minutes quantumly vs at least thousands of years classically.

Theoretically, finding a suitable problem relies heavily on complexity theory. One such problem is Boson Sampling [1], which can be physically setup by, say, scattering photons through linear interferometers. This problem is related to computing permanents of matrices, a known hard problem, and it has been shown that an efficient simulation of the Boson Sampling problem would imply the collapse of the polynomial hierarchy to its third level.

Since 2019, several platforms have claimed to have reached quantum supremacy with the Random Circuit Sampling setup, starting with Google [59]. There is still some discussion as to whether this milestone really has been crossed, for several of these claims have been hindered by the existence of good classical simulation algorithms that brought the computation time to reasonable scales. These usually use tensor networks, and exploit the fact that current quantum computers are still very noisy [147].

8 Quantum HPC

The advent of the NISQ era has underlined the need for powerful classical resources to run hand in hand with the quantum algorithms. Some of them (like the variational ones) require a heavy classical treatment to optimise quantum circuit parameters [139]; others use the quantum primitives to get a rough estimate of the solution of a problem, that can then be refined classically, or with several backs-and-forth with the quantum processing unit [112]. Many numerical stability issues arising in quantum algorithms such as QSVT, HHL and variational solvers could be mitigated by the use of HPC-inspired methods, or by cleverly interfacing the classical and quantum parts. HPC may also be used in the preprocessing step, where for instance circuit synthesis may prove rather resource intensive [114, 51].

Since the NISQ-era computers are very limited in memory size, cutting the circuit we want to execute into chunks and stitching together the results obtained on each smaller circuit has been proposed [117, 107]. Finding the best spots to cut the circuit, and dealing with the data, requires large and fast classical capabilities.

In the LSQ era (the large-scale, fault tolerant era), it is expected that performing error correction will require fast classical methods to not hinder the quantum part. Some schemes have already been proposed with HPC methods in mind [105].

More pragmatically, HPC has already been used to simulate quantum circuits (as pointed out above). Different methods have been explored for that purpose, the most naive (although amenable to many optimisation possibilities) is the state-vector, where an n-qubit quantum state is represented by a 2^n-sized complex vector, and the application of quantum gates results in modifications in the vector [90]. Another method is that of tensor networks, where the state is represented by a set of multidimensional matrices connected together following a predefined topology. This structure may be more amenable to simulate the effect of quantum operators, and benefits from efficient data compression [19]. Other methods include stabiliser-based simulation, whereby we use the fact that the Clifford fragment is

efficiently simulable [2], so we try to decompose the non-Clifford parts as linear combinations of Clifford parts. This method has the benefit of scaling exponentially with the number of non-Clifford gates rather than the number of qubits [22].

Finally, developments have already started in the integration of quantum computing in the workflow of HPC developers. For instance, the Q-pragma [69] is a C++ framework developed to offload specific tasks to the quantum processing unit. Workflow abstractions and workflow management are being investigated [34] to propose the best ways to interact with the QPU.

9 Conclusion

The field of quantum computing has boomed over the past 20 years, following the advent of actual quantum computers. Once an almost completely theoretical field, it now evolves hand-in-hand with experimenters, who can now test some of the theoretical results, and pose new problems. The putative large-scale quantum computing era, where virtually any quantum algorithm or protocol can be implemented using fault-tolerant schemes, promises powerful results, as well as interesting new challenges, in the way we program, verify, or implement such algorithms.

Contributors

Alex Bredariol Grilo (CNRS, LIP6) and Renaud Vilmart (Inria, LMF).

References

1 Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. URL: https://theoryofcomputing.org/articles/v009a004, doi:10.4086/toc.2013.v009a004.

2 Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70(5), November 2004. URL: http://dx.doi.org/10.1103/PhysRevA.70.052328, doi:10.1103/physreva.70.052328.

3 Parosh Aziz Abdulla, Yo-Ga Chen, Yu-Fang Chen, Lukáš Holík, Ondřej Lengál, Jyun-Ao Lin, Fang-Yi Lo, and Wei-Lun Tsai. Verifying quantum circuits with level-synchronized tree automata. *Proc. ACM Program. Lang.*, 9(POPL), January 2025. URL: https://doi.org/10.1145/3704868, doi:10.1145/3704868.

4 Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh V. Vazirani. Quantum walks on graphs. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 50–59. ACM, 2001. URL: https://doi.org/10.1145/380752.380758, doi:10.1145/380752.380758.

5 Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum pcp conjecture. *SIGACT News*, 44(2):47–79, June 2013. URL: https://doi.org/10.1145/2491533.2491549, doi:10.1145/2491533.2491549.

6 Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008. URL: https://doi.org/10.1137/S0097539799359385, arXiv:https://doi.org/10.1137/S0097539799359385, doi:10.1137/S0097539799359385.

7 Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 453–469. Tsinghua University

Press, 2010. URL: http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/35.html.

8 Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Review*, 50(4):755–787, 2008. URL: https://doi.org/10.1137/080734479, arXiv:https://doi.org/10.1137/080734479, doi:10.1137/080734479.

9 Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 636–643. ACM, 2000.

10 Matthew Amy, Dmitri Maslov, and Michele Mosca. Polynomial-time t-depth optimization of clifford+t circuits via matroid partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1476–1489, October 2014. URL: http://dx.doi.org/10.1109/TCAD.2014.2341953, doi:10.1109/tcad.2014.2341953.

11 Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-controlled ordering of gates. *Physical Review Letters*, 113(25), December 2014. URL: http://dx.doi.org/10.1103/PhysRevLett.113.250402, doi:10.1103/physrevlett.113.250402.

12 P. Arrighi. An overview of quantum cellular automata. *Natural Computing*, 18(4):885–899, September 2019. URL: http://dx.doi.org/10.1007/s11047-019-09762-6, doi:10.1007/s11047-019-09762-6.

13 Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, June 2017. URL: https://doi.org/10.1145/3106700.3106710, doi:10.1145/3106700.3106710.

14 Benjamin Audoux and Alain Couvreur. On tensor products of css codes. *Annales de l'Institut Henri Poincaré D, Combinatorics, Physics and their Interactions*, 6(2):239–287, March 2019. URL: http://dx.doi.org/10.4171/aihpd/71, doi:10.4171/aihpd/71.

15 Martin Avanzini, Georg Moser, Romain Péchoux, Simon Perdrix, and Vladimir Zamdzhiev. Quantum Expectation Transformers for Cost Analysis. In *Symposium on Logic In Computer Science LICS '22*, Haifa, Israel, August 2022. URL: https://inria.hal.science/hal-03540366.

16 James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 467–496. Springer, 2021.

17 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.

18 Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *EEE International Conference on Computers, Systems and Signal Processing*, 1984.

19 Aleksandr Berezutskii, Minzhao Liu, Atithi Acharya, Roman Ellerbrock, Johnnie Gray, Reza Haghshenas, Zichang He, Abid Khan, Viacheslav Kuzmin, Dmitry Lyakh, Danylo Lykov, Salvatore Mandrà, Christopher Mansell, Alexey Melnikov, Artem Melnikov, Vladimir Mironov, Dmitry Morozov, Florian Neukart, Alberto Nocera, Michael A. Perlin, Michael Perelshtein, Matthew Steinberg, Ruslan Shaydulin, Benjamin Villalonga, Markus Pflitsch, Marco Pistoia, Valerii Vinokur, and Yuri Alexeev. Tensor networks for quantum computing. *Nature Reviews Physics*, 7(10):581–593, July 2025. URL: http://dx.doi.org/10.1038/s42254-025-00853-1, doi:10.1038/s42254-025-00853-1.

20 Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549, 2017.

21 Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation, 2002. URL: http://dx.doi.org/10.1090/conm/305/05215, doi:10.1090/conm/305/05215.

22 Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. Simulation of quantum circuits by low-rank stabilizer decompositions. *Quantum*, 3:181,

September 2019. URL: http://dx.doi.org/10.22331/q-2019-09-02-181, doi:10.22331/q-2019-09-02-181.

23 Nikolas P. Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2(4), October 2021. URL: http://dx.doi.org/10.1103/PRXQuantum.2.040101, doi:10.1103/prxquantum.2.040101.

24 H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, January 2009. URL: http://dx.doi.org/10.1038/nphys1157, doi:10.1038/nphys1157.

25 Jop Briet and Ronald de Wolf. Locally Decodable Quantum Codes. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 219–230, Dagstuhl, Germany, 2009. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2009.1813, doi:10.4230/LIPIcs.STACS.2009.1813.

26 Anne Broadbent. How to verify a quantum computation. *Theory Comput.*, 14(1):1–37, 2018. URL: https://doi.org/10.4086/toc.2018.v014a011, doi:10.4086/TOC.2018.V014A011.

27 Anne Broadbent, Joseph F. Fitzsimons, and Elham Kashefi. Universal blind quantum computation. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, Atlanta, Georgia, USA, October 25-27, 2009*, pages 517–526. IEEE Computer Society, 2009. URL: https://doi.org/10.1109/FOCS.2009.36, doi:10.1109/FOCS.2009.36.

28 Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 92–122, 2020.

29 Anne Broadbent and Sébastien Lord. Uncloneable Quantum Encryption via Oracles. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 4:1–4:22, 2020.

30 Daniel E Browne, Elham Kashefi, Mehdi Mhalla, and Simon Perdrix. Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics*, 9(8):250–250, August 2007. URL: http://dx.doi.org/10.1088/1367-2630/9/8/250, doi:10.1088/1367-2630/9/8/250.

31 Timothée Goubault de Brugière, Marc Baboulin, Benoît Valiron, Simon Martiel, and Cyril Allouche. Reducing the depth of linear reversible quantum circuits. *IEEE Transactions on Quantum Engineering*, 2:1–22, 2021. doi:10.1109/TQE.2021.3091648.

32 Harry Buhrman and Ronald de Wolf. Communication complexity lower bounds by polynomials. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, pages 120–130. IEEE Computer Society, 2001.

33 Lukas Burgholzer, Rudy Raymond, and Robert Wille. Verifying results of the ibm qiskit quantum circuit compilation flow. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 356–365, 2020. doi:10.1109/QCE49297.2020.00051.

34 Silvina Caino-Lores, Daniel Claudino, Eugene Dumitrescu, Travis S. Humble, Sonia Lopez Alarcon, and Elaine Wong. *Rethinking Programming Paradigms in the QC-HPC Context*, page 84–91. Springer Nature Switzerland, 2024. URL: http://dx.doi.org/10.1007/978-3-031-61763-8_8, doi:10.1007/978-3-031-61763-8_8.

35 Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, September 2017. URL: http://dx.doi.org/10.1038/nature23460, doi:10.1038/nature23460.

36 Kostia Chardonnet, Marc de Visme, Benoît Valiron, and Renaud Vilmart. The many-worlds calculus. *Logical Methods in Computer Science*, Volume 21, Issue 2, May 2025. URL: http://dx.doi.org/10.46298/lmcs-21(2:13)2025, doi:10.46298/lmcs-21(2:13)2025.

37 Christophe Chareton, Sébastien Bardin, François Bobot, Valentin Perrelle, and Benoît Valiron. *An Automated Deductive Verification Framework for Circuit-building Quantum Programs*,

page 148–177. Springer International Publishing, 2021. URL: http://dx.doi.org/10.1007/978-3-030-72019-3_6, doi:10.1007/978-3-030-72019-3_6.

38 Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 59–68. Association for Computing Machinery, 2003.

39 Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, and Benoit Valiron. Quantum computations without definite causal structure. *Physical Review A*, 88(2), August 2013. URL: http://dx.doi.org/10.1103/PhysRevA.88.022318, doi:10.1103/physreva.88.022318.

40 Pierre Clairambault and Marc de Visme. Full abstraction for the quantum lambda-calculus. *Proc. ACM Program. Lang.*, 4(POPL), December 2019. URL: https://doi.org/10.1145/3371131, doi:10.1145/3371131.

41 Alexandre Clément, Noé Delorme, and Simon Perdrix. Minimal equational theories for quantum circuits. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '24, New York, NY, USA, 2024. Association for Computing Machinery. URL: https://doi.org/10.1145/3661814.3662088, doi:10.1145/3661814.3662088.

42 Alexandre Clément, Nicolas Heurtel, Shane Mansfield, Simon Perdrix, and Benoît Valiron. A complete equational theory for quantum circuits. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2023. doi:10.1109/LICS56636.2023.10175801.

43 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, April 2011. URL: http://dx.doi.org/10.1088/1367-2630/13/4/043016, doi:10.1088/1367-2630/13/4/043016.

44 Andrea Coladangelo, Alex B. Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: New schemes for verifiable delegated quantum computation, with quasilinear resources. *Theory Comput.*, 20:1–87, 2024.

45 Brian Coyle, Snehal Raj, Natansh Mathur, El Amine Cherrat, Nishant Jain, Skander Kazdaghli, and Iordanis Kerenidis. Training-efficient density quantum machine learning. *npj Quantum Information*, 11(1), November 2025. URL: http://dx.doi.org/10.1038/s41534-025-01099-6, doi:10.1038/s41534-025-01099-6.

46 Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3), September 2019. URL: http://dx.doi.org/10.1103/PhysRevA.100.032328, doi:10.1103/physreva.100.032328.

47 Ivan B. Damgard, Serge Fehr, Louis Salvail, and Christian Schaffner. Cryptography in the bounded quantum-storage model. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, page 449–458, USA, 2005. IEEE Computer Society. URL: https://doi.org/10.1109/SFCS.2005.30, doi:10.1109/SFCS.2005.30.

48 Liliane-Joy Dandy, Emmanuel Jeandel, and Vladimir Zamdzhiev. Type-safe quantum programming in idris. In Thomas Wies, editor, *Programming Languages and Systems*, pages 507–534, Cham, 2023. Springer Nature Switzerland.

49 Vincent Danos and Elham Kashefi. Determinism in the one-way model. *Phys. Rev. A*, 74:052310, Nov 2006. URL: https://link.aps.org/doi/10.1103/PhysRevA.74.052310, doi:10.1103/PhysRevA.74.052310.

50 Niel de Beaudrap and Dominic Horsman. The zx calculus is a language for surface code lattice surgery. *Quantum*, 4:218, 2020.

51 T. Goubault de Brugière, M. Baboulin, B. Valiron, and C. Allouche. Quantum circuits synthesis using householder transformations. *Comput. Phys. Commun.*, 248:107001, 2020.

52 Anne-Catherine de la Hamette, Viktoria Kabel, Marios Christodoulou, and Časlav Brukner. Indefinite causal order and quantum coordinates. *Phys. Rev. Lett.*, 135:141402, Oct 2025. URL: https://link.aps.org/doi/10.1103/bnkn-4p3f, doi:10.1103/bnkn-4p3f.

53 David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 12 1992. URL: https://doi.org/10.1098/rspa.1992.0167, arXiv:https://royalsocietypublishing.org/rspa/article-pdf/439/1907/553/68698/rspa.1992.0167.pdf, doi:10.1098/rspa.1992.0167.

54 Arpit Dua, Nathanan Tantivasadakarn, Joseph Sullivan, and Tyler D Ellison. Engineering 3d floquet codes by rewinding. *PRX Quantum*, 5(2):020305, 2024.

55 Yfke Dulek, Alex B. Grilo, Stacey Jeffery, Christian Majenz, and Christian Schaffner. Secure multi-party quantum computation with a dishonest majority. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 12107 of *Lecture Notes in Computer Science*, pages 729–758. Springer, 2020.

56 Ross Duncan, Aleks Kissinger, Simon Perdrix, and John Van De Wetering. Graph-theoretic simplification of quantum circuits with the zx-calculus. *Quantum*, 4:279, 2020.

57 Ross Duncan and Simon Perdrix. Rewriting measurement-based quantum computations with generalised flow. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming*, pages 285–296, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

58 Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Secure two-party quantum evaluation of unitaries against specious adversaries. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference*, volume 6223 of *Lecture Notes in Computer Science*, pages 685–706, 2010.

59 Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, October 2019. URL: http://dx.doi.org/10.1038/s41586-019-1666-5, doi:10.1038/s41586-019-1666-5.

60 Rajeev Acharya et al. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, December 2024. URL: http://dx.doi.org/10.1038/s41586-024-08449-y, doi:10.1038/s41586-024-08449-y.

61 Ulysse Réglade et al. Quantum control of a cat qubit with bit-flip times exceeding ten seconds. *Nature*, 629(8013):778–783, May 2024. URL: http://dx.doi.org/10.1038/s41586-024-07294-3, doi:10.1038/s41586-024-07294-3.

62 Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014. URL: https://arxiv.org/abs/1411.4028, arXiv:1411.4028.

63 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution, 2000. URL: https://arxiv.org/abs/quant-ph/0001106, arXiv:quant-ph/0001106.

64 Omar Fawzi, Antoine Grospellier, and Anthony Leverrier. Constant overhead quantum fault tolerance with quantum expander codes. *Commun. ACM*, 64(1):106–114, December 2020. URL: https://doi.org/10.1145/3434163, doi:10.1145/3434163.

65 Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2), May 2015. URL: https://doi.org/10.1145/2716307, doi:10.1145/2716307.

66 Joseph F. Fitzsimons and Elham Kashefi. Unconditionally verifiable blind quantum computation. *Phys. Rev. A*, 96:012303, Jul 2017. URL: https://link.aps.org/doi/10.1103/PhysRevA.96.012303, doi:10.1103/PhysRevA.96.012303.

67 Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3), September 2012. URL: http://dx.doi.org/10.1103/PhysRevA.86.032324, doi:10.1103/physreva.86.032324.

68 Peng Gao, Yiwei Li, Marek Perkowski, and Xiaoyu Song. Realization of quantum oracles using symmetries of boolean functions. *Quantum Information and Computation*, 20(5 & 6):418–448, May 2020. URL: http://dx.doi.org/10.26421/QIC20.5-6-4, doi:10.26421/qic20.5-6-4.

69 Arnaud Gazda and Océane Koska. A pragma based c++ framework for hybrid quantum/classical computation. *Science of Computer Programming*, 236:103119, September 2024. URL: http://dx.doi.org/10.1016/j.scico.2024.103119, doi:10.1016/j.scico.2024.103119.

70 Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum hamiltonian complexity. *Found. Trends Theor. Comput. Sci.*, 10(3):159–282, 2015. URL: https://doi.org/10.1561/0400000066, doi:10.1561/0400000066.

71 Craig Gidney and Austin G. Fowler. Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation. *Quantum*, 3:135, April 2019. URL: https://doi.org/10.22331/q-2019-04-30-135, doi:10.22331/q-2019-04-30-135.

72 András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 193–204, 2019.

73 András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19, page 193–204. ACM, June 2019. URL: http://dx.doi.org/10.1145/3313276.3316366, doi:10.1145/3313276.3316366.

74 Niels Gleinig and Torsten Hoefler. An efficient algorithm for sparse quantum state preparation. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 433–438, 2021. doi:10.1109/DAC18074.2021.9586240.

75 Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. Quipper: a scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '13, page 333–342, New York, NY, USA, 2013. Association for Computing Machinery. URL: https://doi.org/10.1145/2491956.2462177, doi:10.1145/2491956.2462177.

76 Alex B Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in miniqcrypt. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 531–561. Springer, 2021.

77 Frédéric Grosshans and Philippe Grangier. Continuous variable quantum cryptography using coherent states. *Phys. Rev. Lett.*, 88:057902, Jan 2002. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.057902, doi:10.1103/PhysRevLett.88.057902.

78 Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997. URL: https://link.aps.org/doi/10.1103/PhysRevLett.79.325, doi:10.1103/PhysRevLett.79.325.

79 Shaojun Guo, Jinzhao Sun, Haoran Qian, Ming Gong, Yukun Zhang, Fusheng Chen, Yangsen Ye, Yulin Wu, Sirui Cao, Kun Liu, Chen Zha, Chong Ying, Qingling Zhu, He-Liang Huang, Youwei Zhao, Shaowei Li, Shiyu Wang, Jiale Yu, Daojin Fan, Dachao Wu, Hong Su, Hui Deng, Hao Rong, Yuan Li, Kaili Zhang, Tung-Hsun Chung, Futian Liang, Jin Lin, Yu Xu, Lihua Sun, Cheng Guo, Na Li, Yong-Heng Huo, Cheng-Zhi Peng, Chao-Yang Lu, Xiao Yuan, Xiaobo Zhu, and Jian-Wei Pan. Experimental quantum computational chemistry with optimized unitary coupled cluster ansatz. *Nature Physics*, 20(8):1240–1246, June 2024. URL: http://dx.doi.org/10.1038/s41567-024-02530-z, doi:10.1038/s41567-024-02530-z.

80 Aparna Gupte, Jiahui Liu, Justin Raizes, Bhaskar Roberts, and Vinod Vaikuntanathan. Quantum one-time programs, revisited. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, 2025.

81 Emmanuel Hainry, Romain Péchoux, and Mário Silva. A polytime quantum programming language. *ACM Transactions on Quantum Computing*, 7(1), November 2025. URL: https://doi.org/10.1145/3769851, doi:10.1145/3769851.

82 Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15), October 2009. URL: http://dx.doi.org/10.1103/PhysRevLett.103.150502, doi:10.1103/physrevlett.103.150502.

83 Matthew B. Hastings and Jeongwan Haah. Dynamically Generated Logical Qubits. *Quantum*, 5:564, October 2021. URL: https://doi.org/10.22331/q-2021-10-19-564, doi:10.22331/q-2021-10-19-564.

84 Nicolas Heurtel, Andreas Fyrillas, Grégoire De Gliniasty, Raphaël Le Bihan, Sébastien Malherbe, Marceau Pailhas, Eric Bertasi, Boris Bourdoncle, Pierre-Emmanuel Emeriau, Rawad Mezher, Luka Music, Nadia Belabas, Benoît Valiron, Pascale Senellart, Shane Mansfield, and Jean Senellart. Perceval: A Software Platform for Discrete Variable Photonic Quantum Computing. *Quantum*, 7:931, February 2023. URL: https://hal.science/hal-03874624, doi:10.22331/q-2023-02-21-931.

85 Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, and Michael Hicks. A verified optimizer for quantum circuits. *Proc. ACM Program. Lang.*, 5(POPL), January 2021. URL: https://doi.org/10.1145/3434318, doi:10.1145/3434318.

86 Mohsin Iqbal, Nathanan Tantivasadakarn, Ruben Verresen, Sara L. Campbell, Joan M. Dreiling, Caroline Figgatt, John P. Gaebler, Jacob Johansen, Michael Mills, Steven A. Moses, Juan M. Pino, Anthony Ransford, Mary Rowe, Peter Siegfried, Russell P. Stutz, Michael Foss-Feig, Ashvin Vishwanath, and Henrik Dreyer. Non-abelian topological order and anyons on a trapped-ion processor. *Nature*, 626(7999):505–511, February 2024. URL: http://dx.doi.org/10.1038/s41586-023-06934-4, doi:10.1038/s41586-023-06934-4.

87 Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. QIP = PSPACE. *J. ACM*, 58(6):30:1–30:27, 2011. URL: https://doi.org/10.1145/2049697.2049704, doi:10.1145/2049697.2049704.

88 Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*, volume 10993 of *Lecture Notes in Computer Science*, pages 126–152. Springer, 2018. URL: https://doi.org/10.1007/978-3-319-96878-0_5, doi:10.1007/978-3-319-96878-0_5.

89 Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. Mip*=re, 2022. URL: https://arxiv.org/abs/2001.04383, arXiv:2001.04383.

90 Tyson Jones, Anna Brown, Ian Bush, and Simon C. Benjamin. Quest and high performance simulation of quantum computers. *Scientific Reports*, 9(1), July 2019. URL: http://dx.doi.org/10.1038/s41598-019-47174-9, doi:10.1038/s41598-019-47174-9.

91 Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 106–115, New York, NY, USA, 2003. Association for Computing Machinery. URL: https://doi.org/10.1145/780542.780560, doi:10.1145/780542.780560.

92 Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, Berkeley, CA, USA, January 9-11, 2017*, volume 67 of *LIPIcs*, pages 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

93 Aleks Kissinger and John van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Phys. Rev. A*, 102:022406, Aug 2020. URL: https://link.aps.org/doi/10.1103/PhysRevA.102.022406, doi:10.1103/PhysRevA.102.022406.

94 Aleks Kissinger, John van de Wetering, and Renaud Vilmart. Classical Simulation of Quantum Circuits with Partial and Graphical Stabiliser Decompositions. In François Le Gall and Tomoyuki Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2022.5, doi:10.4230/LIPIcs.TQC.2022.5.

95 A Yu Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, December 1997. URL: http://dx.doi.org/10.1070/RM1997v052n06ABEH002155, doi:10.1070/rm1997v052n06abeh002155.

96 Alexei Y. Kitaev, A. H. Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate studies in mathematics*. American Mathematical Society, 2002. URL: https://bookstore.ams.org/gsm-47/.

97 A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003. URL: http://dx.doi.org/10.1016/S0003-4916(02)00018-0, doi:10.1016/s0003-4916(02)00018-0.

98 A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, January 2003. URL: http://dx.doi.org/10.1016/S0003-4916(02)00018-0, doi:10.1016/s0003-4916(02)00018-0.

99 Emanuel Knill, Raymond Laflamme, and Wojciech H. Zurek. Resilient quantum computation. *Science*, 279(5349):342–345, 1998. URL: https://www.science.org/doi/abs/10.1126/science.279.5349.342, arXiv:https://www.science.org/doi/pdf/10.1126/science.279.5349.342, doi:10.1126/science.279.5349.342.

100 William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1589–1602. ACM, 2023. URL: https://doi.org/10.1145/3564246.3585225, doi:10.1145/3564246.3585225.

101 Hlér Kristjánsson, Giulio Chiribella, Sina Salek, Daniel Ebler, and Matthew Wilson. Resource theories of communication. *New Journal of Physics*, 22(7):073014, July 2020. URL: http://dx.doi.org/10.1088/1367-2630/ab8ef7, doi:10.1088/1367-2630/ab8ef7.

102 Anthony Leverrier. Security of continuous-variable quantum key distribution via a gaussian de finetti reduction. *Phys. Rev. Lett.*, 118:200501, May 2017. URL: https://link.aps.org/doi/10.1103/PhysRevLett.118.200501, doi:10.1103/PhysRevLett.118.200501.

103 Anthony Leverrier, Simon Apers, and Christophe Vuillot. Quantum xyz product codes. *Quantum*, 6:766, July 2022. URL: http://dx.doi.org/10.22331/q-2022-07-14-766, doi:10.22331/q-2022-07-14-766.

104 Anthony Leverrier and Gilles Zémor. Quantum tanner codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 872–883, 2022. doi:10.1109/FOCS54457.2022.00117.

105 Anthony Leverrier and Gilles Zémor. Decoding quantum tanner codes. *IEEE Transactions on Information Theory*, 69(8):5100–5115, 2023. doi:10.1109/TIT.2023.3267945.

106 Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 1997.

107 Angus Lowe, Matija Medvidović, Anthony Hayes, Lee J. O'Riordan, Thomas R. Bromley, Juan Miguel Arrazola, and Nathan Killoran. Fast quantum circuit cutting with randomized measurements. *Quantum*, 7:934, March 2023. URL: https://doi.org/10.22331/q-2023-03-02-934, doi:10.22331/q-2023-03-02-934.

108 Dmitri Maslov and Ben Zindorf. Depth optimization of cz, cnot, and clifford circuits. *IEEE Transactions on Quantum Engineering*, 3:1–8, 2022.

109 Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical review letters*, 1997.

110 Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, Daniel J Egger, Stefan Filipp, Andreas Fuhrer, Jay M Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3(3):030503, June 2018. URL: http://dx.doi.org/10.1088/2058-9565/aab822, doi:10.1088/2058-9565/aab822.

111 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

112 Arnaud Gazda Océane Koska, Marc Baboulin. A mixed-precision quantum-classical algorithm for solving linear systems. *IPDPS 2025 - 39th IEEE International Parallel and Distributed Processing Symposium Workshops*, pages 501–508, 2025.

113 Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical ldpc codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 375–388, New York, NY, USA, 2022. Association for Computing Machinery. URL: https://doi.org/10.1145/3519935.3520017, doi:10.1145/3519935.3520017.

114 Anouk Paradis, Jasper Dekoninck, Benjamin Bichsel, and Martin Vechev. Synthetiq: Fast and versatile quantum circuit synthesis. *Proc. ACM Program. Lang.*, 8(OOPSLA1), April 2024. URL: https://doi.org/10.1145/3649813, doi:10.1145/3649813.

115 Jennifer Paykin, Robert Rand, and Steve Zdancewic. Qwire: a core language for quantum circuits. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, POPL '17, page 846–858, New York, NY, USA, 2017. Association for Computing Machinery. URL: https://doi.org/10.1145/3009837.3009894, doi:10.1145/3009837.3009894.

116 Tom Peham, Lukas Burgholzer, and Robert Wille. Equivalence checking of quantum circuits with the zx-calculus. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 12(3):662–675, 2022. doi:10.1109/JETCAS.2022.3202204.

117 Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters*, 125(15), October 2020. URL: http://dx.doi.org/10.1103/PhysRevLett.125.150504, doi:10.1103/physrevlett.125.150504.

118 Simon Perdrix. *Quantum Entanglement Analysis Based on Abstract Interpretation*, page 270–282. Springer Berlin Heidelberg. URL: http://dx.doi.org/10.1007/978-3-540-69166-2_18, doi:10.1007/978-3-540-69166-2_18.

119 Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, 2014.

120 Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, Sep 2014. URL: https://link.aps.org/doi/10.1103/PhysRevLett.113.130503, doi:10.1103/PhysRevLett.113.130503.

121 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), September 2009. URL: https://doi.org/10.1145/1568318.1568324, doi:10.1145/1568318.1568324.

122 Ben W. Reichardt, Falk Unger, and Umesh V. Vazirani. Classical command of quantum systems. *Nat.*, 496(7446):456–460, 2013. URL: https://doi.org/10.1038/nature12035, doi:10.1038/NATURE12035.

123 Renato Renner. Security of quantum key distribution, 2006. URL: https://arxiv.org/abs/quant-ph/0512258, arXiv:quant-ph/0512258.

124 Diego Ruiz, Jérémie Guillaud, Anthony Leverrier, Mazyar Mirrahimi, and Christophe Vuillot. Ldpc-cat codes for low-overhead quantum computing in 2d. *Nature Communications*, 16(1), January 2025. URL: http://dx.doi.org/10.1038/s41467-025-56298-8, doi:10.1038/s41467-025-56298-8.

125 Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11), 2014.

126 Peter Selinger. *A Brief Survey of Quantum Programming Languages*, page 1–6. Springer Berlin Heidelberg, 2004. URL: http://dx.doi.org/10.1007/978-3-540-24754-8_1, doi:10.1007/978-3-540-24754-8_1.

127 Peter Selinger. Efficient clifford+t approximation of single-qubit operators. *Quantum Info. Comput.*, 15(1–2):159–180, January 2015.

128 Peter Selinger and Benoît Valiron. Quantum Lambda Calculus. In Simon J. Gay and Ian Mackie, editors, *Semantic Techniques in Quantum Computation*, pages 135–172. Cambridge University Press, Cambridge, November 2009. URL: https://www.mscs.dal.ca/~selinger/papers/qlambdabook.pdf, doi:10.1017/CBO9781139193313.005.

129 P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, page 124–134, 1994. doi:10.1109/SFCS.1994.365700.

130 D.R. Simon. On the power of quantum computation. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 116–123, 1994. doi:10.1109/SFCS.1994.365701.

131 Marcos Yukio Siraichi, Vinicius Fernandes Dos Santos, Caroline Collange, and Fernando Magno Quintão Pereira. Qubit allocation as a combination of subgraph isomorphism and token swapping. In *OOPSLA*, volume 3, pages 1–29, Athens, Greece, October 2019. URL: https://inria.hal.science/hal-02316820, doi:10.1145/3360546.

132 Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. t|ket): a retargetable compiler for NISQ devices. *Quantum Science and Technology*, 6(1):014003, November 2020. URL: http://dx.doi.org/10.1088/2058-9565/ab8e92, doi:10.1088/2058-9565/ab8e92.

133 Christoph Sünderhauf, Earl Campbell, and Joan Camps. Block-encoding structured matrices for data input in quantum computing. *Quantum*, 8:1226, January 2024. URL: https://doi.org/10.22331/q-2024-01-11-1226, doi:10.22331/q-2024-01-11-1226.

134 Ewin Tang. Dequantizing algorithms to understand quantum advantage in machine learning. *Nature Reviews Physics*, 4, 2022.

135 Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–1202, 2014. doi:10.1109/TIT.2013.2292061.

136 Umesh Vazirani and Thomas Vidick. Fully device independent quantum key distribution. *Commun. ACM*, 62(4):133, March 2019.

137 Thomas Vidick. MIP*=RE: A negative resolution to Connes' embedding problem and Tsirelson's problem. *ICM International Congress of Mathematicians 2022 July 6-14. Sections 12-14*, 2022.

138 Renaud Vilmart. A near-minimal axiomatisation of zx-calculus for pure qubit quantum mechanics. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2019. doi:10.1109/LICS.2019.8785765.

139 Aleksander Wennersteen, Kemal Bidzhiev, Mauro D'Arcangelo, Matthieu Moreau, Anton Quelle, Alexandre Dauphin, and Mourad Beji. *Hybrid Quantum Classical Algorithms: A Cloud On-Demand Viewpoint*, page 117–125. Springer Nature Switzerland, 2026. URL: http://dx.doi.org/10.1007/978-3-032-13855-2_11, doi:10.1007/978-3-032-13855-2_11.

140 Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1), 1983.

141 Mark M. Wilde. *Quantum Information Theory*. Cambridge University Press, 2 edition, 2017.

142 Robert Wille, Stefan Hillmich, and Lukas Burgholzer. *Decision Diagrams for Quantum Computing*, page 1–23. Springer International Publishing, August 2022. URL: http://dx.doi.org/10.1007/978-3-031-15699-1_1, doi:10.1007/978-3-031-15699-1_1.

143 Bujiao Wu, Xiaoyu He, Shuai Yang, Lifu Shou, Guojing Tian, Jialin Zhang, and Xiaoming Sun. Optimization of cnot circuits on limited-connectivity architecture. *Physical Review Research*, 5(1):013065, 2023.

144 Shuai Yang, Wei Zi, Bujiao Wu, Cheng Guo, Jialin Zhang, and Xiaoming Sun. Efficient quantum circuit synthesis for sat-oracle with limited ancillary qubit. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(3):868–877, 2024. doi:10.1109/TCAD.2023.3325974.

145 Nengkun Yu and Jens Palsberg. Quantum abstract interpretation. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, PLDI 2021, page 542–558, New York, NY, USA, 2021. Association for Comput-

ing Machinery. URL: https://doi.org/10.1145/3453483.3454061, doi:10.1145/3453483.3454061.

146 Mark Zhandry. Quantum lightning never strikes the same state twice. In *Advances in Cryptology – EUROCRYPT 2019*, pages 408–438, Cham, 2019. Springer International Publishing.

147 Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.041038, doi:10.1103/PhysRevX.10.041038.

148 Julien Zylberman and Fabrice Debbasch. Efficient quantum state preparation with walsh series. *Phys. Rev. A*, 109:042401, Apr 2024. URL: https://link.aps.org/doi/10.1103/PhysRevA.109.042401, doi:10.1103/PhysRevA.109.042401.

Les vingt ans du GdR IFM, vus du GT « Calculabilités »

Les thématiques fédérées par le GT Calculabilités du GdR IFM s'inscrivent dans une tradition centrale de la recherche en informatique théorique et en logique mathématique : comprendre les limites fondamentales du calcul, leurs variations selon les modèles, et leurs interactions avec d'autres domaines des mathématiques.

Si le GT en tant que structure n'a qu'une dizaine d'années [1], les axes scientifiques qu'il rassemble constituent un fil directeur majeur des activités du GdR IFM sur cette période, avec un approfondissement constant des résultats et une diversification des modèles étudiés.

1 Calculabilité classique, logique et degrés

Un premier socle scientifique concerne la théorie classique de la calculabilité et ses liens profonds avec la logique. Les travaux menés sur les hiérarchies arithmétiques et analytiques, les degrés de calculabilité et les relations de réduction ont permis d'affiner considérablement la compréhension des frontières entre décidabilité, indécidabilité et définissabilité. Au cours des vingt dernières années, plusieurs contributions ont mis en évidence des séparations fines entre notions de calculabilité proches, ainsi que des phénomènes de non-robustesse sous relativisation ou sous contraintes de ressources, montrant que des variations apparemment mineures dans les modèles peuvent entraîner des changements profonds de pouvoir de calcul. Par exemple, les méthodes utilisées pour comprendre le problème de Post, à savoir l'existence de problèmes qui ne sont ni calculables ni calculatoirement énumérables, ont été approfondies et bien mieux comprises. On est passé d'un système à priorité à un système calquant les mécanismes de forcing utilisés en théorie des ensembles, permettant de ne plus avoir à décrire les mécanismes de calculs pour se concentrer sur ce qui est calculé.

Dans cette dynamique, les interactions avec les mathématiques à rebours et la combinatoire effective ont joué un rôle structurant. Partant d'un énoncé de combinatoire (e.g. si on colorie en deux couleurs les arêtes d'un graphe infini, il existe un sous-graphe induit infini mono-chromatique), on cherche d'une part à trouver les axiomes minimaux de l'arithmétique du second ordre permettant de prouver ce résultat, et d'autre part à comprendre sa nature calculatoire (e.g. l'ensemble mono-chromatique est-il calculable si le graphe est calculable ?). Ceci permet de relier des principes combinatoires précis à des niveaux bien identifiés de calculabilité, offrant une lecture computationnelle de résultats logiques fondamentaux, et renforçant le dialogue entre logique mathématique et théorie du calcul. Cette perspective est particulièrement visible dans l'étude systématique de théorèmes de type Ramsey – l'exemple précédent correspondant au théorème de Ramsey pour les paires [19] – et montre comment la calculabilité se lit à travers des hiérarchies fines et des séparations délicates.

Un second fil directeur a porté sur l'articulation entre *hasard effectif* et *degrés / généricité*. Mathématiquement, il existe plusieurs façons de dire qu'une propriété est vraie presque partout : d'un point de vue mesurable (l'ensemble des points où elle est vraie est de mesure pleine) ou topologiquement (l'ensemble des points où elle est vraie est un ouvert dense), qui ont toutes les deux leur pendant calculatoire. Sur ce sujet, des résultats identifient précisément ce que des oracles aléatoires permettent (ou ne permettent pas) de calculer en matière de généricité, et réciproquement [3].

1. La première édition des journées du GT ont eu lieu à Fontainebleau en 2015.

2 Calculabilité sur les réels et analyse effective

Un axe majeur des vingt dernières années concerne la calculabilité sur les structures continues, c'est-à-dire des structures mettant en jeu des éléments ayant la cardinalité des réels. Les travaux sur l'analyse effective et la calculabilité sur les espaces métriques et topologiques ont permis de mieux comprendre la nature algorithmique des objets analytiques classiques, au-delà du cadre discret, le point de départ étant le fait que toute fonction de $\mathbb{R}$ dans $\mathbb{R}$ (par exemple) est continue si et seulement si elle est calculable relativement à un oracle. Ce fait se généralise aux ensembles fermés et compacts, et à n'importe quel espace topologique, une fois qu'on a une *représentation* de cet espace.

L'étude des espaces représentables a montré que des propriétés analytiques classiques telles que la continuité ou la convergence ne garantissent en rien une bonne effectivité, et que des phénomènes systématiques de non-calculabilité peuvent apparaître même dans des cadres très réguliers [10]. Les liens établis entre topologie descriptive et calculabilité ont ainsi permis de préciser la structure effective d'espaces mathématiques usuels.

Un résultat représentatif de cette approche est l'effectivisation d'objets probabilistes continus et pas seulement des suites infinies de bits. Par exemple, dans [17], sont étudiés des espaces métriques généraux sur des objets mathématiques plus complexes (mesures, fonctions, treillis), donnant une compréhension robuste des notions de calculabilité et de hasard dans des cadres non purement discrets. Dans un autre registre, les journées du GT ont régulièrement mis en avant des contributions portant sur la calculabilité de structures topologiques et analytiques, où les questions de représentation effective sont centrales et où l'on cherche à comprendre ce que signifie concrètement « calculer » dans un espace topologique [9].

3 Calcul analogique, systèmes continus et complexité

Les modèles de calcul continus et analogiques constituent un autre axe structurant du GT. Des travaux consacrés aux équations différentielles comme modèles de calcul ont montré que des classes de calculabilité sur les réels peuvent être capturées de manière intrinsèque par des systèmes dynamiques continus, et en particulier par des équations différentielles polynomiales effectives [6].

Ces résultats contribuent à mettre sur un pied d'égalité conceptuel calcul discret et calcul continu : le calcul analogique devient un cadre où l'on peut formuler des questions de pouvoir de calcul et de complexité avec une précision comparable au cadre classique. Il a été montré que l'on peut également définir des notions naturelles de complexité dans ces modèles : le temps correspond à la longueur des trajectoires [8], et l'espace à la précision requise pour représenter les configurations [4].

Un autre apport marquant est l'étude de la *robustesse* : il ne s'agit pas seulement de savoir ce qui est calculable en théorie, mais ce qui demeure calculable sous perturbations réalistes (approximation, bruit, erreurs). Des résultats mettent en évidence des phénomènes de stabilité et de non-stabilité qui structurent l'espace des modèles analogiques [5]. Dans le même esprit, la clarification du lien entre le paradigme GPAC (Shannon) et l'analyse calculable, ainsi que la description des fonctions générées, ont joué un rôle important dans l'unification du paysage des modèles analogiques.

4 Systèmes dynamiques discrets, pavages et universalité

Les systèmes dynamiques discrets — automates cellulaires, dynamique symbolique, pavages — ont constitué un terrain privilégié pour étudier l'émergence du calcul universel. Une

ligne forte des vingt dernières années est que des systèmes localement simples peuvent simuler des calculs universels, et donc engendrer de l'indécidabilité pour des propriétés dynamiques naturelles.

Un résultat emblématique est le développement de constructions *pavages apériodiques par point fixe*. Il existe des liens forts entre pavages et calculabilité, qui s'illustrent par exemple par le fait qu'un diagramme espace-temps d'une machine de Turing peut être vu comme un pavage. Cependant toutes les constructions de problèmes indécidables, à différents niveaux de la hiérarchie arithmétique, s'appuyaient sur l'existence de pavages apériodiques, dont les constructions historiques sont en général combinatoires ou géométriques (et donc pas de nature calculatoire). Dans [13], les auteurs montrent qu'on peut obtenir un pavage apériodique par des arguments de calculabilité et plus exactement par le point fixe de Kleene : une machine qui écrit son propre code se transforme en un pavage qui se désubstitue en lui-même.

Cette approche présente à la fois un intérêt conceptuel — en fournissant une explication computationnelle unifiée de l'apériodicité — et un intérêt technique, la flexibilité de la construction permettant d'imposer des propriétés supplémentaires, afin de rendre par exemple l'ensemble de pavages robuste aux erreurs locales [13], ou minimal.

Par ailleurs, l'étude de la dynamique de réseaux d'automates a mis en évidence des phénomènes de type « théorème de Rice » en complexité : tester des propriétés non triviales du comportement dynamique devient rapidement intractable, fournissant une explication structurelle aux difficultés de vérification de dynamiques finies [15]. Dans le même esprit, plusieurs travaux sur les automates cellulaires et les systèmes discrets ont analysé finement la complexité et la décidabilité de propriétés dynamiques telles que la prédiction, l'atteignabilité ou la stabilité, contribuant à clarifier les liens entre dynamique symbolique, calculabilité et complexité [14].

5 Information algorithmique et hasard

Un autre axe important concerne l'aléa algorithmique et la théorie de l'information algorithmique. Les notions classiques (complexité de Kolmogorov, hasard, normalité) ont été revisitées sous l'angle de contraintes de ressources, en particulier dans des modèles à mémoire finie.

Une contribution marquante, explicitement identifiée dans le cadre du GT, est la structuration d'une *théorie de l'information algorithmique à états finis*. Il était connu de longue date que la normalité de Borel est étroitement liée aux machines à états finis et aux mécanismes de compression à mémoire bornée, mais ces résultats restaient dispersés.

Des travaux récents ont proposé un cadre unifié fournissant des versions à états finis de notions fondamentales de l'information algorithmique, telles que la complexité de Kolmogorov, la probabilité a priori et la complexité conditionnelle. Une contribution technique centrale est l'introduction de la notion de *mesure de complexité superadditive*, qui permet de donner des preuves plus conceptuelles et uniformes de nombreux résultats auparavant techniques [18].

6 Calculabilité transfinie et modèles de calcul proches

Le GT s'est aussi intéressé aux modèles de calcul à temps transfini, qui, contrairement aux automates cellulaires et machines de Turing classiques, ne s'arrêtent pas en un temps fini, mais en un temps ordinal. L'idée est d'ajouter, en plus du mécanisme classique permettant de passer d'un temps t au temps $t+1$ (ordinal successeur), un mécanisme expliquant comment

passer au temps t si on dispose de l'état du système à tous les temps $t' < t$ (ordinal limite). Ces modèles servent d'interface naturelle entre calculabilité et logique transfinie.

Dans ce cadre, il a été montré que les *ordinaux admissibles*, qui jouent en logique un rôle central en tant qu'ordinaux bien clos pour les fonctions élémentaires Σ_1, sont profondément liés à des propriétés algorithmiques précises des machines. Plusieurs travaux ont permis de préciser ces correspondances, en reliant la structure des calculs des machines de Turing à temps infini à la hiérarchie des ensembles constructibles de Gödel. Ceci résume la direction générale de plusieurs articles [2, 1, 11, 12] qui couvrent un spectre assez large de résultats, de la caractérisation de tout admissible comme premier ordinal non récursif avec un oracle réel (qui est explicitement proposé), jusqu'au temps minimal nécessaire pour construire les ensembles de Gödel.

7 Ouvertures contemporaines

Enfin, les activités récentes du GT montrent une ouverture vers des modèles contemporains issus de l'informatique moderne, en particulier l'apprentissage automatique et les réseaux de neurones. Abordés avec les outils de la calculabilité, de la dynamique et de la complexité, ces modèles sont analysés du point de vue de leur pouvoir de calcul, de la décidabilité de problèmes de vérification et des limites intrinsèques de leur expressivité [7]. Ces travaux prolongent naturellement les questions classiques du GT, en montrant que des modèles largement utilisés en pratique soulèvent des questions fondationnelles profondes.

Dans un autre registre, des travaux récents ont montré comment des approches issues de l'analyse de programmes et de la sécurité de l'information peuvent fournir des outils conceptuels pour raisonner sur la complexité, illustrant une autre facette des interactions entre calculabilité, analyse de programmes et contrôle de l'information [16].

8 Conclusion

La rétrospective des activités du GT Calculabilités sur les vingt dernières années met en évidence une communauté scientifique cohérente, structurée autour de questions de fond sur les limites du calcul. En combinant logique, analyse, dynamique et complexité, les travaux menés ont contribué à une compréhension plus unifiée et plus profonde de la calculabilité dans une grande diversité de modèles. Ce positionnement transversal constitue un socle solide pour les développements futurs.

Contributeurs.

Olivier Bournez, Julien Cervelle, Bruno Durand, Mathieu Hoyrup, Ludovic Patey, Romain Péchoux, Emmanuel Rauzy, Alexander Shen.

Références

1 Kenza Benjelloun and Bruno Durand. Infinite Time Turing Machines for elementary proofs on recursive reals. In *JAF 2024 - Journées sur les Arithmétiques Faibles*, Passau, Germany, September 2024. URL : `https://hal-lirmm.ccsd.cnrs.fr/lirmm-04509148`.

2 Kenza Benjelloun, Bruno Durand, and Grégory Lafitte. Writability power of ITTMs : ordinals and constructible sets. working paper or preprint, February 2023. URL : `https://hal-lirmm.ccsd.cnrs.fr/lirmm-04505369`.

3 Laurent Bienvenu and Christopher P Porter. On the interplay between effective notions of randomness and genericity. *The Journal of Symbolic Logic*, 84(1) :393–407, 2019.

4 Manon Blanc and Olivier Bournez. The complexity of computing in continuous time : space complexity is precision. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *ICALP : Annual International Colloquium on Automata, Languages and Programming 2024 (ICALP'2024)*. LIPICS, July 2024. doi:10.4230/LIPIcs.ICALP.2024.129.

5 Manon Blanc and Olivier Bournez. Quantifiying the robustness of dynamical systems. Relating time and space to length and precision. 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 17 :1–17 :20, Dagstuhl, Germany, 2024. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2024.17.

6 Olivier Bournez, Manuel L. Campagnolo, Daniel S. Graça, and Emmanuel Hainry. Polynomial differential equations compute all real computable functions on computable compact intervals. *Journal of Complexity*, 23(3) :317–335, June 2007. URL : http://dx.doi.org/10.1016/j.jco.2006.12.005, doi:10.1016/j.jco.2006.12.005.

7 Olivier Bournez, Johanne Cohen, and Adrian Wurm. A Universal Uniform Approximation Theorem for Neural Networks. In Paweł Gawrychowski, Filip Mazowiecki, and Michał Skrzypczak, editors, *50th International Symposium on Mathematical Foundations of Computer Science (MFCS 2025)*, volume 345 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29 :1–29 :20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL : https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.MFCS.2025.29, doi:10.4230/LIPIcs.MFCS.2025.Kol29.

8 Olivier Bournez, Daniel Silva Graça, and Amaury Pouly. Polynomial time corresponds to solutions of polynomial ordinary differential equations of polynomial length. *Journal of the ACM*, 64(6) :38 :1–38 :76, 2017. doi:10.1145/3127496.

9 Vasco Brattka and Emmanuel Rauzy. Effective second countability in computable analysis. In Arnold Beckmann, Isabel Oitavem, and Florin Manea, editors, *Crossroads of Computability and Logic : Insights, Inspirations, and Innovations*, volume 15764 of *LNCS*, pages 19–33, Cham, 2025. Springer. 21st Conference on Computability in Europe. doi:10.1007/978-3-031-95908-0_2.

10 Antonin Callard and Mathieu Hoyrup. Descriptive complexity on non-polish spaces. In *International Symposium on Teoretical Aspects of Computer Science (STACS)*, 2020.

11 Merlin Carl, Bruno Durand, Grégory Lafitte, and Sabrina Ouazzani. Admissibles in gaps. In Jarkko Kari, Florin Manea, and Ion Petre, editors, *Unveiling Dynamics and Complexity - 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12-16, 2017, Proceedings*, volume 10307 of *Lecture Notes in Computer Science*, pages 175–186. Springer, 2017. URL : https://doi.org/10.1007/978-3-319-58741-7_18, doi:10.1007/978-3-319-58741-7_18.

12 Bruno Durand and Grégory Lafitte. An algorithmic approach to characterizations of admissibles. In Florin Manea, Barnaby Martin, Daniël Paulusma, and Giuseppe Primiero, editors, *Computing with Foresight and Industry - 15th Conference on Computability in Europe, CiE 2019, Durham, UK, July 15-19, 2019, Proceedings*, volume 11558 of *Lecture Notes in Computer Science*, pages 181–192. Springer, 2019. URL : https://doi.org/10.1007/978-3-030-22996-2_16, doi:10.1007/978-3-030-22996-2_16.

13 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Fixed-point tile sets and their applications. *Journal of Computer and System Sciences*, 78(3) :731–764, 2012.

14 Enrico Formenti. Complexity of local, global and universality properties in finite dynamical systems. In Jérôme Durand-Lose and György Vaszil, editors, *Machines, Computations, and Universality - 9th International Conference, MCU 2022, Debrecen, Hungary, August 31 - September 2, 2022, Proceedings*, volume 13419 of *Lecture Notes in Computer Science*, pages 1–11. Springer, 2022. URL : https://doi.org/10.1007/978-3-031-13502-6_1, doi:10.1007/978-3-031-13502-6_1.

15 Guilhem Gamard, Pierre Guillon, Kevin Perrot, and Guillaume Theyssier. Rice-like theorems for automata networks. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, 2021.

16 Emmanuel Hainry, Bruce M. Kapron, Jean-Yves Marion, and Romain Péchoux. Declassification policy for program complexity analysis. 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 41 :1–41 :14. ACM, 2024. URL : `https://doi.org/10.1145/3661814.3662100`, `doi:10.1145/3661814.3662100`.

17 Mathieu Hoyrup and Cristóbal Rojas. Computability of probability measures and martin-löf randomness over metric spaces. *Information and Computation*, 207(7) :830–847, 2009.

18 Alexander Kozachinskiy and Alexander Shen. Automatic kolmogorov complexity, normality, and finite-state dimension revisited. *Journal of Computer and System Sciences*, 118 :75–107, 2021.

19 Ludovic Patey and Keita Yokoyama. The proof-theoretic strength of ramsey's theorem for pairs and two colors. *Advances in Mathematics*, 330 :1034–1070, 2018.

Programmes, vérification, preuve, automates et logique

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, memoryless 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, 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 Internation 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, 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 Vandenhove. 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. *TheoretiCS*, 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 Koussainov, 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 BC(Σ_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, 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, 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 rpqs. *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 mu-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 Schwarzentruber. 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-Schwarzentruber.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, 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, 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. *TheoretiCS*, 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, 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, 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.

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, 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 MichałSzynwelski. 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, 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, 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, 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 internationnaly 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 decidablity, 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 lead 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 Vandenhove. 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 Miroslaw Kutylowski and Jaideep Vaidya, editors, *Computer Security - ESORICS 2014 - 19th European Symposium on Research in Computer Security, Wroclaw, 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, Slawomir 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 Simácek. 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, doi:10.2168/LMCS-3(1:8)2007.

73 Florian Renkin, Philipp Schlehuber-Caissier, Alexandre Duret-Lutz, and Adrien Pommellet. Dissecting ltlsynt. *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 attachées 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 21ème 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 librairies 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 finiment 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 Aloïs Brunel, Marco Gaboardi, Damiano Mazza, and Steve Zdancewic. A core quantitative coeffect 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 Zamdzhiev. 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 Nguyên 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 ∞-categories (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/1lKaq7watGGl3xvjqw9qHjm6SDPFJ2-0o/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`.

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`.

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. Firore 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`.

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 Melliès. 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`.

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 Théret. 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/btl172`, `doi:10.1093/bioinformatics/btl172`.

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 Koeppl, 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 Koeppl, 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 Soljacic, 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=Ozo7qJ5vZi.

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.

Calcul formel, arithmétique et cryptographie

Les vingt ans du GdR IFM, vus du GT Calcul Formel

L'objet premier de ce corpus synthétique est d'opérer un bilan rétrospectif sur l'évolution du calcul formel sur approximativement les vingt dernières années, c'est-à-dire du début des années 2000 et jusqu'à fin 2025, en essayant de distiller ses caractéristiques ainsi que l'évolution scientifique de ses thématiques de recherche et en indiquant quelques éléments saillants et faits marquants qui nous semblent pertinents pour cette période. Pour ce faire, nous avons opté pour recueillir le témoignage des responsables d'équipes, et plus largement les chercheuses et chercheurs, qui composent notre communauté scientifique. Ce qui suit est une brève synthèse structurée de ces retours.

1 Introduction

Le **calcul formel** est une discipline à l'intersection et à l'interface de l'informatique et des mathématiques, plus spécifiquement des mathématiques constructives.

L'aspect informatique consiste en la **conception d'algorithmes manipulant de façon exacte des objets et expressions mathématiques formels comme des polynômes ou des séries** ainsi qu'en l'étude et l'analyse de leur complexité. Le calcul se veut soit exact, soit précis et certifié. Parmi ces algorithmes, on peut citer ceux dédiés aux *briques fondamentales* comme la multiplication d'entiers, de polynômes ou de matrices, ceux réduisant des problèmes classiques à ces briques, comme la division ou l'inversion, ou encore ceux s'appuyant sur des structures plus générales comme les polynômes à plusieurs variables ou les matrices à coefficients polynomiaux.

L'aspect mathématique consiste à **construire, ou à expliciter formellement un objet, par exemple pour prouver son existence** (par opposition à l'énoncé de son existence en exploitant le principe du tiers exclu). On peut notamment citer ici le théorème fondamental de l'algèbre qui a désormais une preuve élémentaire purement constructive [7]. De telles preuves sont aussi en particulier des algorithmes et se prêtent donc aux mêmes traitements : implémentations, analyses de complexité, etc. Cette concordance entre preuves et programmes a notamment apporté un éclairage fort utile aux preuves (factorisation, simplification, etc.). La richesse et l'utilité de ce mariage sont apparues de manière de plus en plus évidente au cours des dernières décennies, dans un cadre bien plus large que le calcul formel, comme en témoigne par exemple l'engouement actuel de la communauté mathématique pour les assistants de preuves tels que Rocq ou Lean. Ainsi, des preuves mathématiques très longues, comme la preuve du théorème de l'ordre impair établissant que tout groupe fini d'ordre impair est résoluble, ont pu être ainsi entièrement vérifiées [9] [1].

À l'intersection de la preuve et de l'algorithmique, le développement des *mathématiques expérimentales* s'est accéléré au cours de la dernière décennie. Le calcul concret, possible grâce aux implémentations disponibles et de plus en plus performantes et précises, permet d'**intuiter des nouvelles propriétés ou des nouvelles structures mathématiques** qui seront éventuellement prouvées ultérieurement.

1. La preuve de Feit et Thompson publiée en 1963 remplit un numéro entier de 255 pages du Pacific Journal of Mathematics.

2 Spécificités de la recherche en calcul formel

Il est très important de mentionner que les avancées considérées aujourd'hui majeures en calcul formel (que ce soit en mathématiques constructives ou en complexité algorithmique) se sont opérées sur le **temps long**. Les résultats marquants qui sont reportés aujourd'hui (fin 2025) sur ce document sont le fruit d'un **effort continu, persistant et de longue haleine** qui a commencé 20 (voire 30) années plus tôt et qui est difficilement perceptible pour les non-spécialistes durant toute la période de ses développements.

On notera par ailleurs une tendance relativement marquante des deux dernières décennies de la recherche en calcul formel : on cherche moins à concevoir et à implémenter des algorithmes résolvant les problèmes dans leur généralité, et on s'intéresse davantage à des sous-classes structurées. Toujours dans un souci de gain de performance, des algorithmes probabilistes ont été par ailleurs conçus et implémentés. Un exemple frappant est la résolution de systèmes polynomiaux par calcul accéléré de bases de Gröbner en combinant l'arithmétique multi-modulaire avec des méthodes probabilistes (certifiables sous conditions de généricité souvent vérifiées en pratique) [8].

3 Algorithmes et complexité

Le calcul exact mène naturellement à des études de complexité en pire cas qui ne sont pas toujours en accord avec ce que l'utilisateur observe expérimentalement. Ces écarts n'ont cessé de motiver des **améliorations notables des bornes de complexités théoriques** soit en exploitant des structures particulières (comme les symétries ou le caractère creux pour ne citer que les plus communes) avant de s'attaquer éventuellement au cas le plus général, soit en améliorant la complexité de certaines briques de base comme la multiplication. Dans ce dernier cas, améliorer ne serait-ce qu'un peu la complexité d'une de ces briques a bien souvent un impact majeur et transversal sur plusieurs bornes de complexité théorique. Nous mentionnerons deux exemples saillants.

1. **Complexité de la multiplication**. Dans le cas des entiers (resp. des polynômes univariés), nous connaissons déjà une réponse partielle : elle est, au pire, quasi-linéaire en la taille (resp. le degré) n. Plus précisément, il existe $a > 0$ tel que le produit de deux entiers de taille n (resp. polynômes de degré n) se calcule en $O(n(\log n)^a)$ opérations binaires (resp. arithmétiques). En 1971, Schönhage et Strassen ont montré que, dans le cas des entiers, a est au plus $1+o(1)$ en conjecturant qu'il serait possible de se débarrasser du $o(1)$. Ces vingt dernières années, cette borne pire-cas théorique n'a cessé d'être améliorée par paliers successifs jusqu'à atteindre la borne présumée $O(n \log n)$ en 2021, **prouvant ainsi une conjecture ouverte depuis un demi-siècle** [10]. Notons également que la complexité théorique du produit de matrices a été améliorée à plusieurs reprises, bien que la quasi-optimalité $O(n^2(\log n)^a)$ reste encore incertaine et en tout cas hors d'atteinte pour le moment.

2. **Réduction de calculs en algèbre linéaire**. L'inversion de matrice, la décomposition LU, le calcul du noyau ou encore le calcul du déterminant ne sont pas plus difficiles que le calcul du produit de matrices carrées. Cependant, le calcul du polynôme caractéristique échappait jusqu'ici à cette règle hormis sous des hypothèses de *généricité* ou via des méthodes probabilistes. En 2021, le premier algorithme pour le **calcul du polynôme caractéristique** dont la complexité en temps déterministe est la même que celle du produit a été proposé [14].

Une des thématiques historiques du calcul formel, la résolution exacte des systèmes polynomiaux d'équations et/ou d'inéquations et inégalités dans le cas réel, a aussi connu des avancées majeures. Alors que les situations pire cas peuvent avoir des sorties de taille gigantesque, les situations génériques et/ou structurées ont permis d'exploiter des leviers menant à des algorithmes dont la complexité est polynomiale voire quasi-optimale en la taille de l'entrée et de la sortie [15].

Le calcul formel ne se cantonne pas seulement au calcul exact. La communauté s'est fortement mobilisée pour développer des **algorithmes symboliques-numériques** afin de tirer partie de la puissance du calcul numérique. À cet égard, l'usage des librairies multi-précisions comme MPFR a été instrumental [2]. On peut citer en particulier les nombreux algorithmes à base de méthode d'homotopie et/ou de déformation, qui sont très efficaces en grande dimension. Mentionnons également la **résolution du 17e problème de Smale**, posant la question du calcul d'une solution approchée d'un système polynomial en temps moyen polynomial, tout d'abord via un algorithme probabiliste puis via un algorithme déterministe [3, 11]. Notons par ailleurs la résolution récente d'un problème majeur et fondamental qui a résisté pendant un demi-siècle : l'**évaluation numérique multi-points d'un polynôme univarié** en un nombre d'opérations binaires quasi-linéaire en son degré [13].

4 Calcul formel et mathématiques

Dans certaines disciplines mathématiques, comme la combinatoire ou la géométrie algébrique réelle, **le traitement algorithmique**, ou constructif, des objets manipulés a grandement contribué à des **avancées novatrices**. Des améliorations quantitatives notables ont été obtenues par ce biais sur les 20 dernières années [2]. En guise d'illustration, citons deux exemples frappants qui montrent la puissance et l'intérêt de l'approche constructive pour **enrichir le corpus de résultats mathématiques**.

1. Estimations précises des **constantes impliquées dans l'inégalité de Łojasiewicz dans le cadre semi-algébrique**. Dans sa version analytique originale, cette inégalité borne la distance d'un point quelconque au zéro le plus proche d'une fonction analytique. Les estimations récentes obtenues dans le cas semi-algébrique par des méthodes algorithmiques sont **plus précises que celles obtenues jusque ici par les géomètres** [1].
2. Le **17e problème de Hilbert** qui demande si tout polynôme multivarié qui ne prend aucune valeur strictement négative est une somme de carrés de fonctions rationnelles. Ce résultat a été démontré par Emil Artin au début du 20e siècle mais sa preuve était non constructive. En particulier aucune borne sur le degré des numérateurs et dénominateurs n'était connue. Des preuves constructives avec des bornes primitives récursives (tours d'exponentielles dont les hauteurs dépendaient du nombre de variables) ont été données plus d'un demi-siècle plus tard. Seulement très récemment, des **bornes élémentaires ont été établies (tours d'exponentielles à hauteur bornée), résolvant ainsi un problème ouvert de longue date**. La preuve utilise notamment les *sous-résultants* qui font partie des objets de base étudiés en calcul formel [12].

Pour souligner l'utilité des **mathématiques expérimentales**, à l'intersection du calcul formel et des mathématiques, on citera les marches de Gessel en combinatoire qui constituent un modèle classique de chemins dans le quart de plan, où un marcheur part de l'origine et effectue des pas parmi l'ensemble $\{\rightarrow, \nearrow, \leftarrow, \swarrow\}$. Pendant longtemps, la nature exacte de la

2. `https://www.mpfr.org`, prix du logiciel libre de recherche 2025, catégorie scientifique et technique.

série génératrice comptant ces chemins est restée ouverte, en particulier sa D-finitude. Le problème a été résolu en 2010 : la **série génératrice associée aux marches de Gessel est algébrique** et donc D-finie, c'est-à-dire solution d'une équation différentielle à coefficients polynomiaux [5]. La preuve repose sur une combinaison d'outils de calcul formel, dont la capacité à effectuer des calculs concrets et corrects. Il s'est agit de **calculer des équations fonctionnelles** satisfaites par les premiers termes de la série, puis de **prouver leur validité** pour la série complète.

Nous finissons cette section avec une mention spéciale pour l'usage des méthodes constructives et du calcul formel dans le cadre spécifique de l'algèbre différentielle avec des implémentations matures et raisonnablement efficaces qui ont naturellement trouvé des applications en biologie et en physique. On citera **les améliorations notables qui ont été apportées au calcul de plusieurs types d'intégrales premières** (algébriques, élémentaires, rationnelles etc.) ainsi qu'à la théorie de **l'élimination différentielle** qui demande une mise en forme particulière du système d'équations fort semblable à la forme triangulaire dans le cadre purement algébrique. L'usage des *chaînes régulières* par exemple a permis un **calcul effectif des bases de Ritt-Raudenbush** (analogue au théorème de base de Hilbert) [16].

5 Logiciel et reproductibilité des résultats

La communauté de calcul formel s'est toujours montrée volontaire pour implémenter, que ce soit à bas niveau comme en `C` ou à plus haut niveau dans un logiciel de calcul formel, les algorithmes qu'elle développe. Les fins sont multiples : pouvoir reproduire des résultats, résoudre des problèmes venant d'applications, déterminer les limites des méthodes et des algorithmes mais aussi développer de nouvelles approches en s'aidant de l'ordinateur. Ainsi, une **évolution logicielle notable s'est produite en calcul formel sur les 20 dernières années**. Elle s'est opérée sur deux niveaux.

1. D'abord, l'apparition de librairies spécialisées de plus en plus performantes qui permettent un calcul exact ou certifié. Citons par exemple FLINT (théorie des nombres, calcul modulaire) et LinBox (algèbre linéaire) [3]. Ce gain en performance n'est pas uniquement dû à de bonnes implémentations mais aussi à des avancées notables et décisives en algorithmique, par exemple [6].

   S'ajoute à cela une **hybridation numérique/symbolique** qui a été extrêmement profitable pour **gagner en performance sans sacrifier l'exactitude des calculs**. On mentionnera la librairie PariGP qui se spécialise entre autres en calcul efficace en théorie des nombres et est lauréate de deux prix prestigieux en 2021 et 2024 [4], ou encore msolve (qui implémente notamment l'algorithme F4 de Faugère) bibliothèque devenue incontournable pour la résolution des systèmes polynomiaux multivariés [5], sans oublier le projet Differential Algebra implémentant l'état de l'art des méthodes pour la résolution de l'équivalent différentiel des systèmes polynomiaux [6].

2. Ensuite, et jusqu'au début des années 2000, les logiciels ou systèmes de calcul formel (*computer algebra systems*) les plus utilisés, bien qu'issus des laboratoires de calcul scientifique, évoluaient pour devenir propriétaires tout en gardant une étroite collaboration

3. `https://flintlib.org`, `https://linalg.org`.
4. `https://pari.math.u-bordeaux.fr`, ACM SIGSAM Richard Dimick Jenks Memorial Prize 2021 et prix science ouverte du logiciel libre de la recherche 2024, catégorie communauté.
5. `https://msolve.proj.lip6.fr`
6. `https://codeberg.org/francois.boulier/DifferentialAlgebra/`

avec la communauté, citons par exemple Maple ou Magma [7]. L'avènement de **plateformes libres** (open source), à l'instar de SageMath ou Macaulay2 [8], qui agrègent de nombreux composants autour du calcul numérique et symbolique, a non seulement permis une **diffusion plus large**, mais aussi et surtout de faciliter et d'**uniformiser l'accès aux librairies spécialisées**.

6 Dissémination du calcul formel

Aujourd'hui SageMath est typiquement utilisé dans l'enseignement de l'option calcul formel en agrégation de mathématiques. Par ailleurs l'outillage logiciel développé au sein de la communauté calcul formel a eu un impact majeur sur des communautés adjacentes. Le champ d'application de ces outils n'a cessé de s'étendre, modifiant ainsi les pratiques de certaines communautés. Par exemple, les **calculs de bases de Gröbner** sont devenus presque routiniers en **cryptographie** pour la cryptanalyse de **crypto-systèmes post-quantiques**, de primitives cryptographiques symétriques.

Le **télescopage créatif**, et ses différentes évolutions [4], pour le calcul d'équations différentielles linéaires (et parfois de formes closes) pour des séries formelles issues de calculs d'intégration ou de sommation multiples à paramètres sert désormais d'outil incontournable pour montrer des équivalences ou des égalités connues, voire pour calculer de nouvelles sommes. Ceci permet notamment le calcul des intégrales de Feynman et est par exemple utilisé au LHC (CERN).

On mentionnera aussi les récentes collaborations industrielles en robotique et en théorie du contrôle [9]. La combinaison de techniques classiques de résolution de systèmes polynomiaux, notamment le théorème de Kantorovich, avec des implémentations numériques robustes et efficaces (dans Julia) ont permis le passage à l'échelle, nécessaire pour de telles applications.

Contributeurs :

Jérémy Berthomieu et Khalil Ghorbal

Références

1 Lorenzo Baldi, Bernard Mourrain, and Adam Parusiński. On Łojasiewicz inequalities and the effective Putinar's Positivstellensatz. *Journal of Algebra*, 662 :741–767, 2025. doi:10.1016/j.jalgebra.2024.08.022.

2 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2006. doi:10.1007/3-540-33099-2.

3 Carlos Beltrán and Luis Miguel Pardo. Smale's 17th problem : average polynomial time to compute affine and projective solutions. *J. Amer. Math. Soc.*, 22(2) :363–385, 2009. doi:10.1090/S0894-0347-08-00630-9.

4 Alin Bostan, Frédéric Chyzak, Pierre Lairez, and Bruno Salvy. Generalized Hermite reduction, creative telescoping and definite integration of D-finite functions. In Éric Schost, editor, *ISSAC'18*, pages 95–102. ACM Press, 2018. doi:10.1145/3208976.3208992.

7. https://maplesoft.com, https://magma.maths.usyd.edu.au/magma
8. https://www.sagemath.org, https://macaulay2.com
9. Safran Electronics & Defense, https://pace.gitlabpages.inria.fr

5 Alin Bostan and Manuel Kauers. The complete Generating Function for Gessel Walks is Algebraic. *Proceedings of the American Mathematical Society*, 138, 2010. doi:10.1090/s0002-9939-2010-10398-2.

6 Javad Doliskani, Pascal Giorgi, Romain Lebreton, and Éric Schost. Simultaneous conversions with the residue number system using linear algebra. *ACM Trans. Math. Softw.*, 44(3), January 2018. doi:10.1145/3145573.

7 Michael Eisermann. The Fundamental Theorem of Algebra Made Effective : An Elementary Real-algebraic Proof via Sturm Chains. *The American Mathematical Monthly*, 119(9) :pp. 715–752, 2012. doi:10.4169/amer.math.monthly.119.09.715.

8 Jean-Charles Faugère and Chenqi Mou. Sparse FGLM algorithms. *Journal of Symbolic Computation*, 80 :538–569, 2017. doi:https://doi.org/10.1016/j.jsc.2016.07.025.

9 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 *Interactive Theorem Proving*, pages 163–179. Springer, 2013. doi:10.1007/978-3-642-39634-2_14.

10 David Harvey and Joris van der Hoeven. Integer multiplication in time $O(n \log n)$. *Annals of Mathematics*, 193(2) :563 – 617, 2021. doi:10.4007/annals.2021.193.2.4.

11 Pierre Lairez. A Deterministic Algorithm to Compute Approximate Roots of Polynomial Systems in Polynomial Average Time. *Found. Comput. Math.*, 17(5) :1265–1292, October 2017. doi:10.1007/s10208-016-9319-7.

12 Henri Lombardi, Daniel Perrucci, and Marie-Françoise Roy. An elementary recursive bound for effective Positivstellensatz and Hilbert 17th problem. *Memoirs of the American Mathematical Society*, 263, 2020.

13 Guillaume Moroz. New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1090–1099, 2022. doi:10.1109/FOCS52979.2021.00108.

14 Vincent Neiger and Clément Pernet. Deterministic computation of the characteristic polynomial in the time of matrix multiplication. *Journal of Complexity*, 67 :101572, 2021. doi:10.1016/j.jco.2021.101572.

15 Mohab Safey El Din and Éric Schost. A Nearly Optimal Algorithm for Deciding Connectivity Queries in Smooth and Bounded Real Algebraic Sets. *J. ACM*, 63(6), January 2017. doi:10.1145/2996450.

16 William Y. Sit. *The Ritt-Kolchin Theory for Differential Polynomials*, pages 1–70. World Scientific Publishing, 2002. doi:10.1142/9789812778437_0001.

Les vingt ans du GdR IFM, vus du GT « Arithmétique des ordinateurs »

L'arithmétique des ordinateurs a pour but de travailler sur les représentations des nombres, les algorithmes de calcul et les architectures pour calculer plus vite, plus précisément, à moindre coût ou de façon plus sûre. Ce domaine s'appuie sur les mathématiques, l'informatique et l'architecture des circuits. Les domaines d'application sont très variés : calcul scientifique, cryptographie, preuves assistées par ordinateur, traitement du signal, spatial... La communauté du GT-ARITH s'est structurée ces 20 dernières années autour de plusieurs thématiques.

1 Maîtriser la précision numérique

Une thématique importante du GT ARITH est la **maîtrise de la précision numérique**. Bien que ce sujet soit ancien, il est primordial pour les académiques et les industriels de savoir si leurs programmes calculent bien. Cela inclut à la fois une borne sur les erreurs d'arrondis et une vérification qu'il n'y a pas de comportement exceptionnel (par exemple *overflow* ou division par zéro). Un large choix d'outils a été développé sur cette thématique, sur des bases statistiques (CADNA), d'interprétation abstraite (Fluctuat, EVA) ou d'arithmétique d'intervalles (Gappa). Hors des cas d'applications de ces outils, la littérature s'est enrichie de nombreuses preuves d'algorithmes ou de théorèmes flottants. Un tel exemple récent est un algorithme d'estimation rapide et fiable du risque de collision spatiale, actuellement embarqué dans le mini-satellite expérimental OPS-SAT de l'Agence spatiale européenne [1]

Une évolution dans cette thématique est l'apparition de démonstrations formelles de ces résultats (bibliothèque Flocq [2] notamment) qui permet plus de garanties sur la correction du programme ou de l'algorithme.

Avec le développement de ces outils, la maîtrise de la précision numérique a pu passer à l'échelle avec des applications HPC (high-performance computing). D'ailleurs, le GT ARITH est devenu un GT commun entre le GdR IFM et le GdR C4P (Calcul : Paradigmes, Parallélisme, Performance, Précision) créé très récemment.

2 Arithmétique virgule-flottante

L'arithmétique **virgule-flottante** est un outil fondamental commun aux membres du GT ARITH. Afin de présenter le plus clairement possible le standard IEEE-754 [1], mais aussi de motiver des améliorations potentielles à y apporter, un groupe dirigé par Jean-Michel Muller a rédigé une somme [12] qui est actuellement considérée comme la référence internationale sur le sujet.

Une des améliorations significatives portée par une partie des membres du GT est la promotion de l'arrondi correct lors de l'évaluation de larges classes de fonctions mathématiques (essentiellement, les fonctions élémentaires). Au cours des 20 dernières années, des progrès substantiels ont été faits par plusieurs membres du GT. Ils permettent à présent l'évaluation des fonctions élémentaires avec arrondi correct en précision double (la précision de prédilection du calcul scientifique).

1. https ://lejournal.cnrs.fr/articles/un-algorithme-pour-eviter-les-debris-spatiaux

Le GT a également été fortement influencé par les **évolutions matérielles** des fabricants de puces. Il y a 20 ans, les formats disponibles et répandus étaient le *binary32* et le *binary64*, anciennement dénommés simple et double précisions. Même si la multiprécision en logiciel (MPFR) était disponible, le matériel se concentrait sur ces deux formats. Avec l'essor de l'IA, un grand nombre de formats plus petits sont maintenant disponibles, notamment sur les GPUs (*fp16, bfloat16, fp8, fp6, fp4*). En lien avec la maîtrise de la précision numérique, il est utile de savoir quels algorithmes fonctionnent encore à si petite précision et comment émuler une précision plus grande.

On peut aussi utiliser ces formats à bon escient en essayant d'optimiser la vitesse d'exécution, la mémoire ou la quantité de données échangées (en HPC), c'est le but de la *mixed-precision*.

Tout cela reflète le **lien fort entre matériel et logiciel** dans ce GT puisque les évolutions matérielles influent sur nos applications mais nos résultats influent sur le matériel (usage du *round-to-odd* notamment).

En plus de ces formats fondés sur la virgule flottante, se sont développés d'**autres formats de nombres** plus adaptés à certaines applications. Par exemple, LNS (*Logarithmic Number System*) stocke une approximation du logarithme du nombre, rendant les multiplications simples et les additions complexes [13]. D'autres formats plus exotiques ont été développés sur la période. On peut citer la famille de format posits qui ont une taille variable de la mantisse et de l'exposant, ce qui les rend plus dynamiques mais bien plus complexes à implémenter en matériel [4].

Une autre application où la précision numérique est importante est celle des **filtres numériques**. Ces programmes sont composés de calculs (en virgule flottante ou fixe) à l'intérieur d'une boucle. Le nombre important d'itérations rend les bornes d'erreurs difficiles à trouver et à prouver. Sur les filtres LTI (*Linear Time-Independant*), ce problème a été résolu avec une borne d'erreur prouvée optimale [8]. Autour de cette application, plusieurs outils ont été réalisés pour estimer les intervalles dynamiques et la qualité numérique nécessaire pour une implémentation en C ou sur FPGA [16].

3 Arithmétique et cryptographie

Les activités du GT-ARITH trouvent aussi des liens importants avec le GT-C2. Les approches **arithmétiques pour la cryptographie** ont pour but de permettre des calculs rapides et sûrs sur des grands nombres (au plus quelques milliers de bits) dans des corps finis. Dans le cadre de la cryptographie classique, les opérations modulaires sont très présentes. Ce travail reste pertinent avec l'évolution des standards cryptographiques et de l'architecture des processeurs. À titre d'exemple les processeurs Intel proposent des instructions pour calculer sur des corps finis de caractéristique deux depuis quelques années déjà.

Certaines représentations sont utiles dans ce domaine. Les représentations **RNS** (Residue Number System) et **PMNS** (Polynomial Modular Number Systems), qui ont l'avantage de paralléliser certaines opérations, offrent de bonnes performances dans ce contexte. Par exemple un des algorithmes-phares en cryptographie homomorphe, l'algorithme CKKS [9], nécessite des calculs sur des polynômes de très haut degré avec des coefficients de très grande taille et tire avantage de ces représentations. De plus, sous certaines conditions, ces représentations sont redondantes et permettent plusieurs écritures pour chaque valeur. Il devient alors possible d'introduire de l'aléa dans les écritures et d'effectuer des calculs dont l'exécution sur processeur **réduit les fuites** pouvant être exploitées par un attaquant [7, 3].

Plus généralement, l'algorithmique des **réseaux euclidiens** joue un rôle-clé en cryptographie à clé publique depuis une trentaine d'années. Ainsi, l'algorithme LLL, fondamental au sein de ce domaine, est un outil de cryptanalyse essentiel qui a été rendu pratique grâce aux implémentations flottantes [15, 11] actuellement disponibles dans le logiciel `fplll`. Depuis une quinzaine d'années, cette algorithmique est également au cœur de la conception de nouvelles primitives cryptographiques. En effet, avec la menace de plus en plus prégnante que représente l'apparition d'un ordinateur quantique, les techniques cryptographiques à clé publique classiques ne sont plus sûres. C'est pourquoi, le NIST (National Institute of Standards and Technology) a ouvert en 2017 un concours pour développer des algorithmes résistant à cette menace. Ces nouveaux algorithmes, dits **post-quantiques**, s'appuient sur de nouveaux problèmes (notamment dans des réseaux euclidiens, mais aussi sur des codes correcteurs, sur des isogénies, etc.). De nouvelles approches arithmétiques ont démontré la possibilité d'implémentations rapides et sûres de plusieurs propositions à ce concours.

4 Arithmétique et opérateurs matériels

Les activités du GT-ARITH sont très liées au support matériel et suivent deux approches. La première consiste à exploiter au mieux les ressources offertes par les processeurs modernes. Ce sont des approches logicielles qui cherchent à utiliser le parallélisme et l'hétérogénéité de ces processeurs. L'autre approche consiste à développer des **architectures d'opérateurs** ou d'applications complètes sur FPGA ou ASIC.

Ces 20 dernières années des outils pour réaliser des implémentations matérielles ont été développés dans la communauté du GT-ARITH. Dans les implémentations matérielles, la précision n'est pas fixe et peut être librement choisie par le développeur. La maîtrise de la qualité numérique en **virgule fixe** et le choix de la précision optimale est difficile mais permet de construire des architectures de petite taille, basse consommation ou encore pour du calcul approché [10].

Par exemple, **FloPoCo** [6, 5] est un générateur de cœurs arithmétiques pour différentes technologies cible (FPGA, VLSI, ...). Le premier but de cet outil est de concevoir des opérateurs ad-hoc permettant d'obtenir des résultats plus précis avec moins de matériel et en moins de temps. Le deuxième but est de permettre un calcul juste. Les opérateurs FloPoCo sont soigneusement conçus pour garantir qu'aucun bit inutile pour le résultat final ne soit calculé.

Une troisième approche intermédiaire est explorée en adaptant le jeu d'instructions des processeurs. Le jeu d'instructions ouvert `Risc-V` prévoit la customisation de son jeu d'instruction. Cette approche permet d'utiliser des arithmétiques adaptées à l'application visée [14].

Contributeurs et contributrices :

Sylvie Boldo, Nicolas Brisebarre et Laurent-Stéphane Didier.

Références

1 IEEE standard for floating-point arithmetic. *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pages 1–84, July 2019. doi:10.1109/IEEESTD.2019.8766229.

2 Sylvie Boldo and Guillaume Melquiond. *Computer Arithmetic and Formal Proofs : Verifying Floating-point Algorithms with the Coq System.* ISTE Press - Elsevier, December 2017.

3 Jérôme Courtois, Lokman Abbas-Turki, and Jean-Claude Bajard. Resilience of randomized RNS arithmetic with respect to side-channel leaks of cryptographic computation. *IEEE Transactions on Computers*, 68(12) :1720–1730, 2019.

4 Florent De Dinechin, Luc Forget, Jean-Michel Muller, and Yohann Uguen. Posits : the good, the bad and the ugly. In *Proceedings of the Conference for Next Generation Arithmetic 2019*, pages 1–10, 2019.

5 Florent De Dinechin and Martin Kumm. Application-specific arithmetic. *Cham : Springer International Publishing*, 2024.

6 Florent De Dinechin and Bogdan Pasca. Designing custom arithmetic data paths with FloPoCo. *IEEE Design & Test of Computers*, 28(4) :18–27, 2011.

7 Laurent-Stéphane Didier, Fangan-Yssouf Dosso, Nadia El Mrabet, Jérémy Marrez, and Pascal Véron. Randomization of arithmetic over polynomial modular number system. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 199–206. IEEE, 2019.

8 T. Hilaire. From filters/controllers to code – contributions to fixed-point arithmetic implementations under accuracy constraint. Habilitation à Diriger des Recherches (HDR) – Sorbonne Université, 2024.

9 Joon-Woo Lee, Eunsang Lee, Yongwoo Lee, Young-Sik Kim, and Jong-Seon No. High-precision bootstrapping of rns-ckks homomorphic encryption using optimal minimax polynomial approximation and inverse sine function. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 618–647. Springer, 2021.

10 D. Menard, R. Rocher, O. Sentieys, N. Simon, L-S. Didier, T. Hilaire, B. Lopez, E. Goubault, S. Putot, F Vedrine, A. Najahi, G. Revy, L Fangain, C. Samoyeau, F Lemonnier, and C. Clienti. Design of fixed-point embedded systems (defis). In ECSI European Electronic Chips and Systems Initiative, editors, *2012 Conference on Design and Architectures for Signal and Image Processing (DASIP), Karlsruhe, Germany, October 23 - 25, 2012*, pages 365–366. ECSI - European Electronic Chips and Systems design Initiative, 2012.

11 Ivan Morel, Damien Stehlé, and Gilles Villard. H-LLL : using Householder inside LLL. In Jeremy R. Johnson, Hyungju Park, and Erich L. Kaltofen, editors, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2009, Seoul, Republic of Korea, July 29-31, 2009, Proceedings*, pages 271–278. ACM, 2009. URL : `https://doi.org/10.1145/1576702.1576740`, `doi:10.1145/1576702.1576740`.

12 Jean-Michel Muller, Nicolas Brunie, Florent de Dinechin, Claude-Pierre Jeannerod, Mioara Joldes, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, and Serge Torres. *Handbook of floating-point arithmetic*. Birkhäuser/Springer, Cham, second edition, 2018. URL : `https://doi.org/10.1007/978-3-319-76526-6`, `doi:10.1007/978-3-319-76526-6`.

13 Jean-Michel Muller, Alexandre Scherbyna, and Arnaud Tisserand. Semi-logarithmic number systems. *IEEE Trans. Computers*, 47(2) :145–151, 1998. URL : `https://doi.org/10.1109/12.663760`, `doi:10.1109/12.663760`.

14 Geneviève Ndour, Tiago Trevisan Jost, Anca Molnos, Yves Durand, and Arnaud Tisserand. Evaluation of variable bit-width units in a RISC-V processor for approximate computing. In Francesca Palumbo, Michela Becchi, Martin Schulz, and Kento Sato, editors, *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF 2019, Alghero, Italy, April 30 - May 2, 2019*, pages 344–349. ACM, 2019. URL : `https://doi.org/10.1145/3310273.3323159`, `doi:10.1145/3310273.3323159`.

15 Phong Q. Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3) :874–903, 2009. URL : `https://doi.org/10.1137/070705702`, `doi:10.1137/070705702`.

16 R. Rocher, D. Menard, O. Sentieys, and P. Scalart. Accuracy evaluation of fixed-point based LMS algorithm. *Digital Signal Processing*, 20(3) :640–652, 2010. URL : `http://hal.inria.fr/inria-00450935/en/`, `doi:doi:10.1016/j.dsp.2009.10.007`.

Les vingts ans du GdR IFM, vus du GT « Codage et cryptographie »

Ces vingt dernières années ont été marquées par une accélération spectaculaire de la recherche dans les thématiques de la cryptographie et des codes correcteurs d'erreurs avec l'émergence de nombreux sujets nouveaux. Les applications de la cryptographie ont explosé et beaucoup d'outils théoriques datant de plus de 40 ans sont maintenant utilisés en pratique. La communauté internationale s'est très largement développée et le GT C2, commun aux GdR IFM et SI (Sécurité Informatique), a fait de même. Son nombre de membres a considérablement augmenté et le GT s'est structuré avec les journées C2 (devenues annuelles depuis 2025, réunissant plus de 200 participants par édition) et son séminaire (trois journées pleines par an, dont une hors de Paris).

De nouveaux algorithmes et de nouvelles manières de prouver des calculs sont apparus, certains transformant totalement la recherche. Plusieurs avancées viennent de résultats et techniques en codes correcteurs d'erreurs, rapprochant ainsi ces deux parties de la communauté, notamment sur les problèmes de cryptographie à base de codes correcteurs d'erreurs.

Les recherches du GT ont été menées dans tous les sous-thèmes fondateurs de C2, tels que la cryptographie symétrique, asymétrique et les codes correcteurs d'erreurs. Les membres du GT se sont fortement mobilisés sur toutes ces thématiques, en partie aiguillonnés par les compétitions de standardisation du NIST et le développement du quantique. Dans la suite, les avancées sont présentées selon ce découpage thématique, mais plusieurs résultats sont transversaux ou mobilisent des outils propres à plusieurs sous-thèmes.

1 Cryptographie symétrique.

Ce domaine a été structuré principalement par les compétitions du NIST (*National Institute of Standards and Technology*) et la conception de nouveaux schémas plus efficaces. Depuis 2001, l'Advanced Encryption Standard (AES) est largement déployé pour remplacer le DES, notamment grâce aux instructions matérielles présentes dans certains processeurs. Après 25 ans de cryptanalyse, seulement 7 tours parmi les 10 de la version avec une clef de 128 bits sont atteignables par la cryptanalyse, les versions de 192 et 256 bits présentant d'autres faiblesses.

Au début des années 2000, une série d'attaques contre les fonctions de hachage a été proposée (attaques différentielles contre MD4, MD5, SHA-1 et contre le mode opératoire de Merkle-Damgård). À noter qu'un fort impact du GT fut la première collision sur SHA-1 [13], amenant au retrait de cette fonction de hachage de plusieurs standards. Pour sélectionner une nouvelle fonction, la compétition SHA-3 a été lancée, et son vainqueur, KECCAK [3], a introduit la structure éponge utilisant une permutation publique et pouvant avoir une sortie de taille arbitraire. Cette construction permet d'obtenir différentes primitives cryptographiques comme des fonctions pseudo-aléatoire à sortie extensible (XOF), tel SHAKE, des codes d'authentification de messages, et des schémas de chiffrement authentifié.

Depuis 2010, les compétitions de chiffrement léger pour les environnements contraints, de chiffrement authentifié (devenu une norme de facto après toutes les attaques contre TLS dans les années 2010), et de fonction de hachage pour les mots de passe, ont aussi amené des progrès remarquables avec les vainqueurs ASCON, SCRYPT et ARGON2. De plus, la problématique de la cryptographie à bas coût à utiliser dans des systèmes ultra-légers a renouvelé certains critères de performance et en a défini de nouveaux, par exemple la faible latence. Enfin, aujourd'hui la recherche s'oriente vers de nouvelles primitives symétriques

efficaces sur $\mathbb{F}_p$, plus adaptées à un usage combiné avec les systèmes de preuves, les signatures, ou le chiffrement homomorphe. Très naturellement, ces travaux, accompagnés de leur volet cryptanalytique, ont vu émerger de nouveaux critères d'évaluation de performances et de nouvelles classes d'analyse (« cryptanalyse » ou « attaques ») : attaques par invariants linéaire et non-linéaire, attaques différentielles à clef fixée...

Du côté de la cryptanalyse, des progrès remarquables ont ainsi été obtenus sur plusieurs fronts. De nouvelles techniques ont émergé, comme les algorithmes reposant sur la *division property* qui permettent des attaques intégrales plus précises contre MISTY1 [14], l'utilisation de MILP (Mixed Integer Linear Programming) pour rechercher automatiquement les attaques. Enfin, les attaques génériques exploitant les graphes des fonctions ont permis des avancées importantes à la fois en termes de cryptanalyse et de preuves de sécurité. Du côté des preuves, des progrès ont été effectués pour obtenir des schémas et modes opératoires dont la sécurité est supérieure à la borne dérivée du paradoxe des anniversaires grâce à des outils avancés de probabilités.

Enfin l'éventuelle arrivée de l'ordinateur quantique a conduit la communauté à définir un cadre essentiellement quantique pour la cryptanalyse (« superposition ») dans lequel on peut considérer de nouvelles classes d'attaques.

2 Codes correcteurs d'erreurs : évolution des problèmes.

Au cours des vingt dernières années, les codes correcteurs ont connu une évolution marquante : initialement conçus pour résoudre des problèmes classiques de communication et assurer la fiabilité des transmissions, ils sont progressivement devenus des outils de première importance dans des domaines beaucoup plus larges. Leur structure mathématique s'est révélée particulièrement adaptée à la cryptographie moderne, notamment dans la construction de systèmes basés sur les preuves à divulgation nulle de connaissance, où l'on exploite leurs propriétés de robustesse et de cohérence [2]. Ces outils sont très demandés par exemple pour améliorer les blockchains. Parallèlement, l'essor du stockage distribué et de la nécessité de garantir l'intégrité des données a conduit au développement de codes localement testables et localement décodables, permettant de vérifier ou de récupérer une information en n'accédant qu'à une petite partie des données [6]. Ces travaux illustrent à quel point la théorie des codes a évolué et s'est diversifiée. Enfin, dans le monde quantique lui-même, des constructions de bons codes quantiques (codes LDPC quantiques, codes de Tanner quantiques) ont été proposées, avec plusieurs percées fondamentales. Ces codes quantiques sont essentiels pour lutter contre la décohérence et le bruit inhérents au contexte quantique, et permettre la réalisation de l'ordinateur quantique.

3 Cryptographie post-quantique.

La menace de l'ordinateur quantique a déclenché une révolution dans le domaine, explicitement concrétisée en 2016 par l'appel du NIST à la conception de systèmes dits post-quantiques. En effet, l'algorithme de Shor [12] permettrait, avec un ordinateur quantique suffisamment puissant et efficace, d'attaquer une grande partie des constructions asymétriques utilisées aujourd'hui dont la sécurité repose sur les problèmes de la factorisation et du logarithme discret. De nouveaux problèmes conjecturés difficiles aussi pour les ordinateurs quantiques, alternatifs à la factorisation et au logarithme discret, ont été très étudiés ces dernières années. Ils reposent sur les réseaux euclidiens, les codes, les isogénies entre courbes, et les systèmes polynomiaux multivariés. Les fonctions de hachage permettent aussi de construire des schémas

de signature dits post-quantiques. Enfin, de nouveaux schémas issus de techniques de calcul sécurisé multipartite (MPC) ont permis d'obtenir des signatures très efficaces.

Dans ce contexte, le NIST a organisé plusieurs compétitions pour mettre en place de nouveaux standards de chiffrement et de signature à clé publique. Proposer un candidat à de tels appels demande un travail important de conception, détermination des paramètres, preuve de sécurité, programmation d'implémentation de référence et mise en place de vecteurs de tests et de spécification. Ces compétitions se déroulent en plusieurs étapes, avec une élimination de candidats à chaque étape jusqu'au choix des finalistes. La première compétition, qui a commencée en 2016, a abouti à la sélection de cinq premiers standards : le mécanisme d'encapsulation de clé (et chiffrement) CRYSTALS-Kyber [4] (maintenant standardisé sous le nom de ML-KEM), la signature CRYSTALS-Dilithium [7] (standardisée sous le nom de ML-DSA), la signature Falcon (encore en cours de standardisation), la signature Sphincs+ (standardisée sous le nom de SLH-DSA) et le mécanisme d'encapsulation de clé (et chiffrement) HQC (encore en cours de standardisation). Parmi ces constructions, les trois premières font reposer leur sécurité sur des problèmes difficiles sur les réseaux euclidiens, la dernière sur un problème issu de la théorie des codes.

Historiquement, le cryptosystème de McEliece, quasi-contemporain de RSA, est fondateur de la cryptographie à clé publique à base de codes correcteurs. Mais, depuis une vingtaine d'années, la cryptographie reposant sur les réseaux euclidiens s'est imposée comme la solution la plus aboutie et mieux comprise pour remplacer la cryptographie asymétrique utilisée aujourd'hui. En plein essor depuis les travaux de Regev en 2005 [10], elle permet de construire des schémas de chiffrement et de signature à la fois sûrs et efficaces, et également des constructions avec des fonctionnalités avancées allant jusqu'au chiffrement complètement homomorphe.

Suite à la première compétition, une seconde a été lancée en 2023 pour proposer d'autres signatures avec comme contrainte que leur sécurité ne devrait pas reposer sur les problèmes difficiles sur les réseaux euclidiens. Certaines des soumissions proposées à cette compétition, qui est encore en cours, reposent sur des problèmes issus du domaine des codes correcteurs, d'autres sur le problème de la résolution d'équations algébriques.

Enfin, pour la cryptographie fondée sur les isogénies, une attaque dévastatrice [5, 9, 11] contre le schéma de chiffrement SIKE en 2022 a finalement été transformée en outil de construction puissant, permettant la construction de nouveaux schémas beaucoup plus efficaces, comme par exemple la signature SQIsign, et remettant sur le devant de la scène les courbes elliptiques.

La recherche a encore des progrès à faire pour mieux comprendre la sécurité de ces schémas, obtenir des schémas plus efficaces, des implémentations sûres ou des constructions avancées. En particulier, la cryptanalyse quantique progresse pour améliorer l'algorithme de Shor et comprendre la sécurité quantique des nouvelles hypothèses.

4 Cryptographie classique.

Ces nouvelles considérations ne doivent pas nous faire négliger la cryptographie classique à base de logarithme discret (sur les corps finis ou sur les courbes elliptiques) ou de factorisation qui est quasiment la seule utilisée en pratique. Dans ces domaines, le problème du logarithme discret sur les corps finis s'est finalement révélé facile à résoudre dans les corps de petite caractéristique, grâce un algorithme « quasi polynomial » [1]. Ces attaques ont été conduites en pratique dans des réalisations logicielles. En particulier, un effort continu sur deux décennies a assuré la permanence et la qualité du logiciel CADO-NFS de factorisation, qui a

notamment permis d'obtenir des records toujours inégalés de factorisation de clé publique RSA (250 chiffres, 828 bits).

Depuis les années 2000, la cryptographie à base de logarithme discret sur les courbes elliptiques remplace RSA. Le coup de grâce est donné en 2018 avec le passage à TLS 1.3 qui promeut la notion de *forward secrecy* pour le chiffrement, même si RSA reste utilisé pour les signatures dans les certificats. Les courbes sécurisées et les implémentations sur Internet utilisent les modèles d'Edwards EdDSA et X25519. Les courbes elliptiques présentant un couplage permettent de concevoir de nouveaux schémas avec des fonctionnalités avancées comme le chiffrement basé sur les attributs ou fonctionnel. La recherche pour la sécurité des courbes, l'efficacité des couplages, le hachage vers les courbes, a ouvert de nouveaux sujets de recherche. Les courbes permettent aujourd'hui de concevoir des systèmes de preuves succinctes et non-interactives, comme celles utilisées dans certaines blockchains.

5 Sécurité des calculs.

Une petite révolution est apparue en 2009 quand Gentry a montré qu'il était possible de construire des schémas complètement homomorphes [8], offfrant ainsi la possibilité de calculer sur des données chiffrées sans les déchiffrer. Les constructions existantes d'un tel chiffrement reposent toutes sur les hypothèses de réseaux euclidiens. Plusieurs générations de systèmes ont été proposées et le domaine est très actif. La cryptographie symétrique a aussi proposé des constructions efficaces avec le chiffrement cherchable, grâce auquel il est possible d'effectuer des recherches par mots-clés dans une base de données chiffrée. Enfin, le calcul sécurisé entre plusieurs parties a effectué des progrès impressionnants, permettant son utilisation pratique et la protection de la vie privée dans différentes applications. Il permet à un groupe de participants, chacun détenant son propre secret, de calculer une fonction de tous leurs secrets, par exemple le max, sans qu'aucun participant ne révèle aux autres son propre secret. Il a été montré au niveau international, que contrairement aux idées reçues, le calcul multipartite est performant en pratique, et il en a suivi un développement accéléré de la recherche dans ce domaine. Comme mentionné précédemment, le calcul multipartite sécurisé a aussi servi de concept clé pour la construction d'algorithmes de signature, à travers le paradigme *MPC in the head*.

6 Implémentations sécurisées.

Depuis la fin des années 90, les attaques par canaux auxiliaires forment un domaine de recherche très important au sein de la communauté CHES. Aujourd'hui, l'implémentation des systèmes classiques est bien comprise et les preuves de sécurité par masquage, à la suite des travaux fondateurs de Ishai, Sahai et Wagner, et différents modèles de sécurité, ont permis des avancées majeures. Dans ce domaine, les outils statistiques d'apprentissage ont eu une influence très importante. L'utilisation des outils formels a aussi amené de réels progrès pour la vérification des preuves, jusqu'à l'implémentation de la cryptographie. Enfin, les attaques logicielles ont aussi fait leur apparition depuis 2005 et les mécanismes matériels d'accélération sont également finement étudiés.

Contributeurs et contributrices.

Daniel Augot, Pierre-Alain Fouque, Eleonora Guerrini et Adeline Roux-Langlois.

Références

1 Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 1–16. Springer, Berlin, Heidelberg, May 2014. doi:10.1007/978-3-642-55220-5_1.

2 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 701–732. Springer, Cham, August 2019. doi:10.1007/978-3-030-26954-8_23.

3 Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 313–314. Springer, Berlin, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_19.

4 Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber : A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018. doi:10.1109/EuroSP.2018.00032.

5 Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 423–447. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_15.

6 Alexandros G. Dimakis, Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. *IEEE Trans. Inf. Theory*, 56(9) :4539–4551, 2010.

7 Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium : A lattice-based digital signature scheme. *IACR TCHES*, 2018(1) :238–268, 2018. URL : https://tches.iacr.org/index.php/TCHES/article/view/839, doi:10.13154/tches.v2018.i1.238-268.

8 Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. doi:10.1145/1536414.1536440.

9 Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 448–471. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_16.

10 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. doi:10.1145/1060590.1060603.

11 Damien Robert. Breaking SIDH in polynomial time. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 472–503. Springer, Cham, April 2023. doi:10.1007/978-3-031-30589-4_17.

12 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5) :1484–1509, 1997.

13 Marc Stevens, Pierre Karpman, and Thomas Peyrin. Freestart collision for full SHA-1. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part I*, volume 9665 of *LNCS*, pages 459–483. Springer, Berlin, Heidelberg, May 2016. doi:10.1007/978-3-662-49890-3_18.

14 Yosuke Todo. Integral cryptanalysis on full MISTY1. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 413–432. Springer, Berlin, Heidelberg, August 2015. doi:10.1007/978-3-662-47989-6_20.

Combinatoire
et systèmes dynamiques

Twenty Years of GdR IFM, seen from GT ALEA

1 Introduction

Discrete structures play a central role in modern scientific research. They allow us to model particles, data, and their relationships, and more broadly to understand how complex systems emerge from simple components. Whether it is data structures in computer science, the organization of matter in statistical physics, or large biological molecules such as DNA and RNA, discrete models provide powerful tools for describing and analyzing real-world phenomena.

The GT ALEA aims to explore these models, to develop new theoretical approaches, and to promote their applications across disciplines. In fact, the *raison d'être* of the ALEA group is the study of discrete random structures that appear in all sciences, mainly fundamental computer science and algorithms, discrete mathematics and probabilities, statistical physics, and more marginally, bioinformatics. A distinctive feature of the group is the systematic use of combinatorial insights – particularly techniques from analytic combinatorics – to analyze and understand these structures.

More precisely, one fundamental line of inquiry concerns simulation methods – the design, programming, and analysis of algorithms that generate these structures efficiently. Alongside this, researchers investigate algorithms operating on these structures (or on more elementary structures such as lists of numbers to be sorted), characterizing their typical behavior. A second major area involves the study of parameters of random discrete objects: investigating quantities such as the average height of a large random tree or determining the limiting distribution of its height after appropriate normalization. Closely related is the study of scaling limits, which seeks to demonstrate that the normalized global object converges in distribution to a limiting random object, typically of a continuous nature. A third major direction is enumerative in nature, focusing on the computation and analysis of generating functions for combinatorial families. This is traditionally done via solving functional equations satisfied by these generating functions. Their study amplified over the past twenty years, with the emergence of a variety of new tools and techniques to solve them, at the interface of many subfields of mathematics. Beyond these core themes, the ALEA group studies numerous other random objects, each with their own characteristic questions (probabilistic cellular automata, random geometry, particle systems, ...), driven by a keen interest in models from theoretical physics and the rich, often unexpected, phenomena they reveal.

This survey presents a selective panorama of recent developments addressing each of these questions, with a particular emphasis on contributions from the ALEA working group. Its aim is not only to describe major advances, but also to highlight the conceptual unification emerging from these diverse domains, and to illustrate how the combination of combinatorial insight, analytic techniques, and probabilistic methods continues to reshape the field.

2 Analysis of algorithms

The precise and systematic analysis of algorithm performance dates back to the work of Donald Knuth, who developed techniques for studying both worst-case and average-case scenarios. The methodology for average-case analysis relies on discrete probability and combinatorics, particularly through their central object: generating functions. For example, if the number of occurrences of a fixed pattern X in a random text of length n is required to

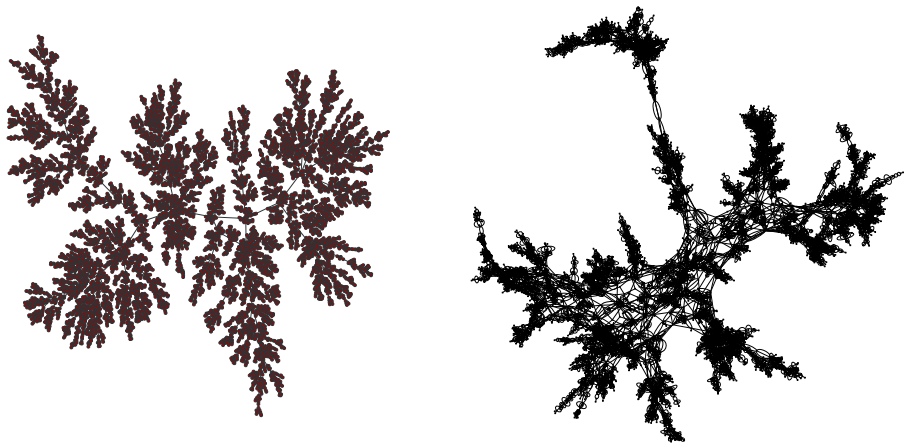


Figure 1 Simulation of a uniform tree with 10,000 nodes and of a uniform quadrangulation with 10,000 faces.

analyze an algorithm, this approach consists of studying its *generating function*, the formal power series in two variables $A_X(z,u) = \sum_{n,k\geq 0} p_{n,k} z^n u^k$, where $p_{n,k}$ is the probability that a word of length n contains k occurrences of X. This generating function captures all the necessary information about the distribution of the number of occurrences of the pattern and presents two fundamental advantages: first, it allows us to obtain an explicit expression directly without computing the $p_{n,k}$'s (or implicitly as a solution of a system of functional equations); second, the desired information (expectation, standard deviation, limit distribution, large deviations, and more) can be extracted using ready-made theorems from complex analysis, by viewing the generating function as a holomorphic function of the parameters z and u. This approach has been systematized by the ALEA community, notably under the impulsion of Philippe Flajolet, and is fully described in his seminal book *Analytic Combinatorics* [31], written with Robert Sedgewick.

Nowadays, a thorough examination of the algorithms implemented in the standard libraries of modern programming languages reveals significant deviations from traditional textbook algorithms. Even with classical data structures and algorithms, such as hash tables or sorting methods, a closer look at the source code shows that engineers often diverge from the expected paths, creating new algorithms or combining existing techniques in innovative ways.

An emblematic example is the sorting algorithm TimSort, designed by the `Python`'s engineer Tim Peters in the early 2000's, which progressively gained popularity and which is now used in `Java`, `Rust`, etc. Its story has been eventful, and the final version of TimSort, its formal proof of correctness, and its complexity [5, 33] were established with the decisive help of academic researchers from the GT ALEA. Importantly, TimSort remained unnoticed by the academic community for more than 10 years despite being widely used in real life. Eventually, the techniques and metrics used for the theoretical study of TimSort led the academic researchers to propose new sorting algorithms, including the PowerSort algorithm [38], which was recently adopted by `Python` in 2021 as their main sorting procedure.

Members of GT ALEA also began, about fifteen years ago, to incorporate architectural elements into computer models and to develop techniques for refining algorithm analyses in order to better describe their actual performance. For example, an internal processor mechanism, branch prediction, tries to anticipate what the next instruction will be during a

conditional jump (such as an `if` or `while`) in order to better take advantage of the speed-up provided by pipelining. Adding this feature to the computer model allows for a complexity analysis that is closer to what is observed in practice. This has been done by the community for the dual-pivot variant of QUICKSORT used in `Java` [37], as well as for standard pattern-matching algorithms [39]. In an orthogonal direction, fundamental algorithms were revisited and modified while taking this mechanism into account; this led to the discovery of surprising variants (such as a binary search that does not split in the middle to make prediction easier), which were proved theoretically and validated experimentally [6].

3 Random generation

A (uniform) random generation algorithm aims at efficiently sampling an object of a given size n from a combinatorial class (e.g. of trees, permutations, partitions, graphs, maps,...) uniformly at random. A prime motivation is to obtain simulations of large random objects and conjecture their asymptotic behavior, in particular for parameters that are difficult to analyze by generating function techniques (e.g. distance parameters). Providing a random sample obeying various sorts of syntactic and semantic constraints can also be useful to test correctness of software or of conjectures.

Random generation is closely related to counting (and to exhaustive generation): if the counting coefficients of a class have a simple closed-form formula (for instance Catalan numbers for binary trees) then combinatorial proofs of the formula can usually be turned into linear time random samplers. More generally, a class can be efficiently sampled if it can be set in bijection to another class that can be efficiently sampled; in the ALEA community this has proved particularly fruitful for the random generation of maps (embedded graphs) and of deterministic automata. This can also be combined with rejection techniques (e.g. if the correspondence is only injective), giving interesting complexity analysis problems [7]. In a broader context, symbolic combinatorics provides a general framework to formulate decompositions of combinatorial classes, which can be applied to exact and asymptotic enumeration on one hand, and to efficient random sampling on the other hand. Two random generation frameworks have been developed: the recursive method [32] that is based on the counting coefficients and recurrences, and Boltzmann samplers [26] that are based on generating functions. Boltzmann sampling achieves simple and faster (linear time) random samplers, at the cost of a slight loss of control on the output size. It has become a reference over the last 20 years and has been actively investigated by the ALEA community. Extensions [29, 12], improvements [23, 45] (e.g. to achieve linear time complexity for exact-size sampling, in particular for trees), and automatization [42, 41] (for the computation of the radius of convergence and evaluation of the generating functions) have been developed, as well as implementations in computer algebra systems such as Maple and SageMath.

Another well-known approach for random generation in a combinatorial class is via Markov chain Monte Carlo techniques, a wide-ranging method that can be applied to classes that are difficult to count (such as self-avoiding walks). Under easy-to-check conditions (ergodicity and symmetry), the distribution after t steps converges as $t \to \infty$ to the uniform distribution on objects of size n. The mixing time is the number of steps required to be close to uniform, its dependency on n being often polynomially bounded (or even logarithmic, for instance in the famous card-shuffling dynamic). A so-called cutoff phenomenon is also frequently observed, namely a sharp transition (at the mixing time) between being far from uniform and being close to uniform. Showing the order of magnitude of the mixing time, and whether a cutoff occurs, can be very challenging. Recent progress has been made [44]

on exhibiting general properties of the Markov chain that determine whether cutoff occurs. Finally, coupling from the past, due to Propp and Wilson, is a generic technique to sample exactly from the limit distribution of an ergodic Markov chain satisfying some easy-to-check conditions. Its efficient implementation has been the subject of active research, in particular for structures encoded by a height function [25, 28].

4 Understanding the behavior of large combinatorial structures

Gaining a better understanding of the statistical behavior of large combinatorial structures has long been an important goal of the ALEA community. One of the motivations is to analyze data structures – and the algorithms associated with them – on realistic data models. Three decades ago, most probabilistic results concerning random combinatorial structures focused on parameters of these structures. For example, it was proved that the height H_n of a tree of size n is of order $\sqrt{n}$, with a standard deviation of the same order. The most precise results concerned convergence in distribution: for instance, the law of $H_n/\sqrt{n}$ converges as $n \to +\infty$ ([30]).

A new point of view emerged in the 1990s, following the work of Aldous [4] and Pitman [40] on scaling limits of random trees: the entire tree, suitably rescaled (distances in the tree are divided by $\sqrt{n}$), converges in distribution. The limit is a continuous random tree, which can already be perceived in the simulation, see Figure 1. These convergence results are global theorems, which imply – as by-products – an infinite number of more classical parameter convergence results. Moreover, the limit is the same for many different tree models, a phenomenon referred to as *universality*.

These results and techniques spread throughout the ALEA community in the early 21st century. Indeed, obtaining such results requires a very detailed understanding of the combinatorial structures under study, as well as expertise in probabilistic methods for proving convergence of complex structures.

The ALEA community now contains dozens of contributions in this specialty, and includes a significant proportion of the world's leading experts in the field.

Among the most notable achievements are the scaling limit theorems – and, in particular, the identification of the Brownian map as the limit – for random maps such as uniform quadrangulations with n faces, together with numerous variants motivated by problems in theoretical physics ([21, 36]). Many other studies focus on graph limits (for example, the shape of giant components in random graphs [1]), random trees, families of paths, tableaux, and permutations (including the definition of permutons as limits of pattern-avoiding permutation tableaux, the analogue of graphons in graph theory [9]).

5 Discrete differential equations

In enumerative combinatorics, we seek to determine the number $a(n)$ of objects of size n in a combinatorial class (trees, permutations, paths, etc.). The most common approach consists of finding an unambiguous decomposition of these objects into smaller objects, which translates into a recurrence relation on the numbers $a(n)$. Often, we find a decomposition, but to deduce a recurrence, we are forced to consider, in addition to the size of the objects, an extra parameter (or even several ones). This parameter is sometimes called "catalytic": without it, there is no recurrence. We must therefore work with an array of numbers, $a(n; k)$, rather than with a sequence $a(n)$.

In the most favorable cases, the recurrence will be expressed as a functional equation

on the associated generating function, $A(t;x) = \sum_{n,k} a(n;k)t^n x^k$. We would then like to answer, from such an equation, some natural questions about the power series $A(t;x)$, or about its specialization $A(t;1) = \sum_n a(n)t^n$: is this generating function a rational function? Does it satisfy a polynomial equation, or at least a differential equation?

A particularly interesting class of functional equations, common in combinatorics, is that of *discrete differential equations*, that is, those using the operator Δ defined by

$$\Delta A(t;x) := \frac{A(t;x) - A(t;0)}{x}.$$

The first equations of this type appeared as early as the 1960s, for counting maps; then in the 1970s, for counting paths. Methods for solving them were then outlined, before it was established in algebra that their solutions were always *algebraic* (i.e., solutions of a polynomial equation). Members of the ALEA group subsequently contributed significantly to a better understanding of this result, to making it effective through adapted algorithms, and to exploiting it in extreme situations where the equations are particularly complex.

Beyond this, a broader class of equations, involving discrete derivatives in several variables, has been the subject of extensive work in our working group. As soon as derivatives with respect to two variables are involved, systematic algebraicity is lost — even if some cases do remain algebraic. Significant efforts to classify solutions have focused on a particular family of equations, considered typical, which involve two catalytic variables and only first-order derivatives. These equations arise from counting paths, with prescribed steps, confined to a quarter of a plane [16]. An astonishing variety of tools has been used to study them: probability theory, basic algebra on formal power series, computer algebra, complex analysis, and difference Galois theory [43, 24, 15]. It is striking how many people with diverse backgrounds this question now brings together. And the story is far from over: only the simplest range of equations is currently well understood. It should be noted that the tools developed in this study have already been adapted to similar equations that were not part of the initial set [8].

6 Interactions with physics

The interactions between the ALEA community and physics have grown significantly over the last 20 years. Among these fruitful interactions are combinatorial maps. Initially studied by Tutte in the 1960s in connection with the 4-color theorem, the study of maps resumed entirely independently in physics in the 1980s, within the context of two-dimensional quantum gravity. Indeed, as collections of discrete surfaces, maps provide a simple yet rich model of random geometries.

The advent of the Cori-Vauquelin-Schaeffer (CVS) bijection, between planar maps and decorated trees, triggered an unprecedented collaboration, which took place within the ALEA group, between combinatorists, physicists and probabilists. On the one hand, the CVS bijection allows access to information inaccessible through physicists' methods [18]; on the other hand, some of their methods, such as integrable hierarchies, are gradually making their way into combinatorics [19]. These interactions have taken the form of new meetings called the "Maps Days" (Journées Cartes) since 2012. They have given rise to an ever-expanding body of new results on map enumeration [17, 13], new bijections [10, 3, 20, 11], and the discovery of scaling limits [34].

These interactions continue today, for example, through the study of maps decorated with statistical physics models (i.e., matter coupled with 2-dimensional quantum gravity!) [2], and the study of higher-dimensional analogues of maps [14, 35].

Other aspects at the intersection of physics, probability, and combinatorics have also entered the GT, for example, probabilistic cellular automata, which are cellular automata whose states evolve according to rules determined by probabilities [27]. We can also mention particle models like the asymmetric simple exclusion process [22], also at the intersection with algebraic combinatorics and the GT CombAlg.

We conclude this text with a few more words about the connection with the GT CombAlg. Some objects commonly studied by the ALEA community also possess an algebraic facet: this is the case, for example, with maps through permutation factorizations and with Young tableaux via representations of classical groups. However, this facet is not central to ALEA's activities, but rather to those of CombAlg. Indeed, ALEA focuses on the statistical properties of combinatorial objects, while CombAlg uses these objects to represent algebraic structures and operations.

Contributors.

Marie Albenque, Valentin Bonzom, Alin Bostan, Mireille Bousquet-Mélou, Philippe Duchon, Éric Fusy, Jean-François Marckert and Cyril Nicaud.

References

1 L. Addario-Berry, N. Broutin, and C. Goldschmidt. The continuum limit of critical random graphs. *Probab. Theory Related Fields*, 152(3-4):367–406, 2012. doi:10.1007/s00440-010-0325-4.

2 M. Albenque, L. Ménard, and G. Schaeffer. Local convergence of large random triangulations coupled with an Ising model. *Transactions of the American Mathematical Society*, 374(1):175–217, October 2020. doi:10.1090/tran/8150.

3 Marie Albenque and Dominique Poulalhon. A generic method for bijections between blossoming trees and planar maps. *Electron. J. Combin.*, 22(2):Paper 2.38, 44, 2015. doi:10.37236/3386.

4 David Aldous. *The continuum random tree. II. An overview*, volume 167 of *London Math. Soc. Lecture Note Ser.*, pages 23–70. Cambridge Univ. Press, Cambridge, 1991. doi:10.1017/CBO9780511662980.003.

5 Nicolas Auger, Vincent Jugé, Cyril Nicaud, and Carine Pivoteau. On the worst-case complexity of TimSort. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*, pages 4:1–4:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.4.

6 Nicolas Auger, Cyril Nicaud, and Carine Pivoteau. Good predictions are worth a few comparisons. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, Orléans, France, February 17-20, 2016*, volume 47 of *LIPIcs*, pages 12:1–12:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICS.STACS.2016.12.

7 Axel Bacher and Andrea Sportiello. Complexity of anticipated rejection algorithms and the Darling-Mandelbrot distribution. *Algorithmica*, 75(4):812–831, 2016. doi:10.1007/s00453-015-0040-8.

8 Jean-Luc Baril, Mireille Bousquet-Mélou, Sergey Kirgizov, and Mehdi Naima. The ascent lattice on Dyck paths. *Electron. J. Combin.*, 32(2):Paper No. 2.36, 42, 2025. URL: https://doi.org/10.37236/13436, doi:10.37236/13436.

9 Frédérique Bassino, Mathilde Bouvel, Valentin Féray, Lucas Gerin, and Adeline Pierrot. The Brownian limit of separable permutations. *Ann. Probab.*, 46(4):2134–2189, 2018. doi:10.1214/17-AOP1223.

10 O. Bernardi and É. Fusy. Unified bijections for maps with prescribed degrees and girth. *J. Combin. Theory Ser. A*, 119(6):1351–1387, 2012. doi:10.1016/j.jcta.2012.03.007.

11 J. Bettinelli. Slit-slide-sew bijections for bipartite and quasibipartite plane maps. *Electron. J. Combin.*, 27(3):Paper 3.4, 23, 2020. doi:10.37236/9069.

12 Olivier Bodini and Alice Jacquot. Boltzmann samplers for v-balanced cycles. *Theoret. Comput. Sci.*, 502:55–63, 2013. doi:10.1016/j.tcs.2012.03.048.

13 V. Bonzom, G. Chapuy, and M. Dołęga. Enumeration of non-oriented maps via integrability. *Algebraic Combinatorics*, 5(6):1363–1390, December 2022. doi:10.5802/alco.268.

14 V. Bonzom and L. Lionni. Counting gluings of octahedra. *The Electronic Journal of Combinatorics*, 24(3), aug 2017. doi:10.37236/6503.

15 A. Bostan and M. Kauers. The complete generating function for Gessel walks is algebraic. *Proc. Amer. Math. Soc.*, 138(9):3063–3078, 2010. With an appendix by M. van Hoeij. doi:10.1090/S0002-9939-2010-10398-2.

16 M. Bousquet-Mélou and M. Mishna. Walks with small steps in the quarter plane. In *Algorithmic probability and combinatorics*, volume 520 of *Contemp. Math.*, pages 1–39. Amer. Math. Soc., Providence, RI, 2010. doi:10.1090/conm/520/10252.

17 M. Bousquet-Mélou and A. Elvey Price. The generating function of planar Eulerian orientations. *Journal of Combinatorial Theory, Series A*, 172:105183, May 2020. doi:10.1016/j.jcta.2019.105183.

18 J. Bouttier, É. Fusy, and E. Guitter. On the two-point function of general planar maps and hypermaps. *Annales de l'Institut Henri Poincaré D, Combinatorics, Physics and their Interactions*, 1(3):265–306, July 2014. doi:10.4171/aihpd/8.

19 S. R. Carrell and G. Chapuy. Simple recurrence formulas to count maps on orientable surfaces. *Journal of Combinatorial Theory, Series A*, 133:58–75, jul 2015. doi:10.1016/j.jcta.2015.01.005.

20 G. Chapuy, V. Féray, and É. Fusy. A simple model of trees for unicellular maps. *J. Combin. Theory Ser. A*, 120(8):2064–2092, 2013. URL: https://doi.org/10.1016/j.jcta.2013.08.003, doi:10.1016/j.jcta.2013.08.003.

21 Philippe Chassaing and Gilles Schaeffer. Random planar lattices and integrated super-Brownian excursion. *Probab. Theory Related Fields*, 128(2):161–212, 2004. doi:10.1007/s00440-003-0297-8.

22 S. Corteel and L. K. Williams. Staircase tableaux, the asymmetric exclusion process, and Askey-Wilson polynomials. *Proceedings of the National Academy of Sciences*, 107(15):6726–6730, March 2010. doi:10.1073/pnas.0909915107.

23 Luc Devroye. Simulating size-constrained Galton-Watson trees. *SIAM J. Comput.*, 41(1):1–11, 2012. doi:10.1137/090766632.

24 T. Dreyfus, C. Hardouin, J. Roques, and M. F. Singer. On the nature of the generating series of walks in the quarter plane. *Invent. Math.*, 213(1):139–203, 2018. doi:10.1007/s00222-018-0787-z.

25 Philippe Duchon. Combinatoire des configurations de boucles compactes, 2008. Habilitation à diriger des recherches. URL: https://hal.science/tel-00414805/.

26 Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combin. Probab. Comput.*, 13(4-5):577–625, 2004. doi:10.1017/S0963548304006315.

27 Nazim Fatès and Lucas Gerin. Examples of fast and slow convergence of 2D asynchronous cellular systems. In *Cellular Automata*, pages 184–191. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-79992-4_24.

28 Stefan Felsner, Daniel Heldt, Sandro Roch, and Peter Winkler. Block coupling and rapidly mixing k-heights. *Preprint*, 2024. URL: https://arxiv.org/abs/2410.08992.

29 Philippe Flajolet, Éric Fusy, and Carine Pivoteau. Boltzmann sampling of unlabeled structures. In *Proceedings of the Fourth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2007, New Orleans, Louisiana, USA, January 06, 2007*, pages 201–211. SIAM, 2007. doi:10.1137/1.9781611972979.5.

30 Philippe Flajolet, Zhicheng Gao, Andrew Odlyzko, and Bruce Richmond. The distribution of heights of binary trees and other simple trees. *Combin. Probab. Comput.*, 2(2):145–156, 1993. doi:10.1017/S0963548300000560.

31 Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, Cambridge, 2009. URL: https://doi.org/10.1017/CBO9780511801655, doi:10.1017/CBO9780511801655.

32 Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoret. Comput. Sci.*, 132(1-2):1–35, 1994. doi:10.1016/0304-3975(94)90226-7.

33 Elahe Ghasemi, Vincent Jugé, Ghazal Khalighinejad, and Helia Yazdanyar. Galloping in fast-growth natural merge sorts. *Algorithmica*, 87(2):242–291, 2025. doi:10.1007/s00453-024-01285-6.

34 J.-F. Le Gall. Uniqueness and universality of the Brownian map. *Ann. Probab.*, 41(4):2880–2960, 2013. doi:10.1214/12-AOP792.

35 L. Lionni and J.-F. Marckert. Iterated foldings of discrete spaces and their limits: Candidates for the role of Brownian map in higher dimensions. *Mathematical Physics, Analysis and Geometry*, 24(4), November 2021. doi:10.1007/s11040-021-09410-5.

36 Jean-François Marckert and Abdelkader Mokkadem. Limit of normalized quadrangulations: the Brownian map. *Ann. Probab.*, 34(6):2144–2202, 2006. doi:10.1214/009117906000000557.

37 Conrado Martínez, Markus E. Nebel, and Sebastian Wild. Analysis of branch misses in Quicksort. In Robert Sedgewick and Mark Daniel Ward, editors, *Proceedings of the Twelfth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2015, San Diego, CA, USA, January 4, 2015*, pages 114–128. SIAM, 2015. URL: https://doi.org/10.1137/1.9781611973761.11, doi:10.1137/1.9781611973761.11.

38 J. Ian Munro and Sebastian Wild. Nearly-optimal mergesorts: Fast, practical sorting methods that optimally adapt to existing runs. In Yossi Azar, Hannah Bast, and Grzegorz Herman, editors, *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*, pages 63:1–63:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ESA.2018.63.

39 Cyril Nicaud, Carine Pivoteau, and Stéphane Vialette. Branch prediction analysis of Morris-Pratt and Knuth-Morris-Pratt algorithms. In Paola Bonizzoni and Veli Mäkinen, editors, *36th Annual Symposium on Combinatorial Pattern Matching, CPM 2025, Milan, Italy, June 17-19, 2025*, volume 331 of *LIPIcs*, pages 8:1–8:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.CPM.2025.8.

40 J. Pitman. *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2006. Lectures from the 32nd Summer School on Probability Theory held in Saint-Flour, July 7–24, 2002, With a foreword by Jean Picard. URL: http://bibserver.berkeley.edu/csp/april05/bookcsp.pdf, doi:10.1007/b11601500.

41 Carine Pivoteau and Bruno Salvy. Effective asymptotics of combinatorial systems. *Preprint*, 2025. URL: https://arxiv.org/abs/2508.20008.

42 Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial structures: well-founded systems and Newton iterations. *J. Combin. Theory Ser. A*, 119(8):1711–1773, 2012. doi:10.1016/j.jcta.2012.05.007.

43 K. Raschel. Counting walks in a quadrant: a unified approach via boundary value problems. *J. Eur. Math. Soc. (JEMS)*, 14(3):749–777, 2012. doi:10.4171/JEMS/317.

44 Justin Salez. Cutoff for non-negatively curved Markov chains. *J. Eur. Math. Soc. (JEMS)*, 26(11):4375–4392, 2024. doi:10.4171/jems/1348.

45 Andrea Sportiello. Boltzmann sampling of irreducible context-free structures in linear time. *Preprint*, 2021. URL: https://arxiv.org/abs/2410.08992.

Twenty Years of GdR IFM, seen from GT Algebraic Combinatorics

1 Introduction

Combinatorics is the study of discrete structures. These structures are ubiquitous in mathematics, computer science and their fields of application. They are used to model how data are structured within computer memory, to analyse algorithms, and also to understand larger-scale mechanisms, such as the organisation of telecommunications networks, fuel in a nuclear power plant, and varietal mixtures within cultivated fields.

Algebraic combinatorics, which is the GT CombAlg's area of expertise, has two facets: the study of discrete structures using algebraic tools and the study of algebraic structures on discrete objects. To illustrate the first perspective, consider a nuclear power plant. The uranium in this plant lasts for three cycles. In each cycle, the fuel must be present in specific locations. The type of questions studied by the members of the GT is therefore as follows: How can the fuel be repositioned in an optimal way while minimising the distance travelled by each rod?[1] This study involves a detailed examination of the symmetric group and its representations (encoded by symmetric functions). To illustrate the second point of view, let us consider a task scheduling problem, such as the production of different car models on an assembly line. In this type of problem, the order of precedence between tasks is fundamental (it makes no sense to start by painting the car body), and this problem is intrinsically linked to the partial order between tasks. Another example of a topic addressed by the GT is the study of (co-)products or products on combinatorial objects. The existence of these structures allows their study to be reduced to that of elementary building blocks. For example, words (or tensor algebra on letters) are obtained by concatenating letters (the elementary building blocks of words); if a property is stable by concatenation and true for letters, it will be true for words. Furthermore, each word can be written uniquely as a concatenation of its letters: this provides a way to generate all words once and only once. This problem is simple in terms of words, but leads to challenging questions when transposed to certain tree structures, or words that satisfy specific conditions, such as parking functions.

Over the past 20 years, there has been a remarkable expansion from symmetric functions, which were the focus of activity at the beginning of the millennium, towards more geometric (matroids, polytopes), algebraic (cluster algebras, Hopf algebras, categories) and bijective aspects. At the heart of these new research topics lies the study of partial orders, or posets, and their links with algebraic structures, such as algebras and operads and their geometric realisations (polytopes). We detail some of these aspects in the following sections, without aiming to be exhaustive: symmetric functions (Section 2), algebras and bialgebras (Section 3), posets (Section 4), operads and species (Section 5), geometric combinatorics (Section 6), bijective methods (Section 7), additive combinatorics and combinatorial number theory (Section 8) and finally, last but not least, formal proof and experimental algebraic combinatorics (Section 9).

[1] see Martin Desombre's PhD thesis [35] for instance.

2 Symmetric functions and generalisations

Symmetric functions and representations of the symmetric group

Symmetric functions are obtained as series of symmetric polynomials. They have been studied extensively for their connection to representations of the symmetric group. There are several bases for symmetric functions: elementary symmetric polynomials, power-sum polynomials, and Schur polynomials. Schur polynomials, indexed by partitions of integers, encode the irreducible representations of the symmetric group, while power-sum polynomials encode their characters. The change from one basis to another is a very difficult question that motivated the introduction of Macdonald polynomials, polynomials in several variables that instantiate into power sums and Schur polynomials in particular.

MacDonald polynomials, coinvariant spaces and Schubert polynomials

Research on MacDonald polynomials led to the introduction of covariant spaces whose characters are given by these polynomials. The first example of a covariant space is the covariant algebra $R_1(n)$, which is the algebra of polynomials in n variables quotiented by the elementary symmetric polynomials $e_1, \ldots, e_n$. It is a central element of algebraic combinatorics through its links with the theory of representations of the symmetric group, algebraic geometry and combinatorics. This space has been generalised with k sets of variables (spaces $R_k(n)$), and more recently by adding l sets of antisymmetric variables, giving multigraded spaces $R_{k,l}(n)$. The case $R_{2,0}(n)$ of diagonal coinvariants (see [11, 60, 59, 64, 61, 7, 62]) has been a driving force with Haiman's calculation of the Frobenius characteristic and its combinatorial interpretation (the shuffle conjecture proved by Carlsson and Mellit [21]). For the rest, $R_{1,1}(n)$ is now partially understood [78] and some results are known (see [32, 31, 30, 29, 67]), but many conjectures remain unproven. This has led to the development of q,t-combinatorics [58, 63].

Schubert polynomials indexed by permutations of S_n form a basis for the space $R_1(n)$, which extends to a basis for polynomials in an infinite number of variables. The combinatorics of these polynomials has been the subject of constant activity [Knutson-Miller, Lascoux, Weigandt, ...] with extensions motivated by geometry. Algebraically, recent work [73] allows us to reformulate their definition and extend it. In another direction, quasi-symmetric polynomials were introduced by Stanley and then Gessel more than forty years ago in the context of P-partition theory. They form an algebra that is the terminal object in the theory of combinatorial Hopf algebras [3]. They form a natural and useful intermediate space between polynomials and symmetric polynomials, and have been studied intensively since their introduction [73].

3 Combinatorial algebras and bialgebras

Connes-Kreimer Hopf algebra, renormalisation and applications to physics

The theory of Hopf algebras and their applications, particularly in combinatorics, experienced a significant revival in the late 1990s, thanks to the work of Connes, Kreimer and their collaborators on renormalisation. This procedure, used in quantum field theory and described recursively in the 1960s, allows the extraction of poles from iterative integrals defined on certain graphs, known as Feynman diagrams. The ultimate goal is to predict the value of certain physical quantities such as the mass of an electron very precisely. In addition to providing theoretical clarifications, this work has renewed the approach, particularly in

computer science, to renormalisation, as well as to Dyson-Schwinger equations [Kreimer, Yeats]. Furthermore, the ideas behind this discovery have been successfully applied in other fields. Combined with Terry Lyons' rough path theory (a theory widely interpreted in a Hopf-algebraic framework, see also [Zambotti]), an approach using tree-based Hopf algebras led Martin Hairer to construct a solution to the KPZ stochastic differential equation and then to develop regularity structures [15], work for which he was awarded the **Fields Medal in 2014**. Regularity structures are based on a type of Hopf algebras known as 'double' Hopf algebras, i.e. equipped with a second coproduct with good compatibilities: the first example of such an object [20] is based on rooted trees. The double bigebras introduced have enabled the study of Butcher series in numerical analysis (solving PDEs). Other objects have been equipped with a double structure: graphs, posets, matrices, hypergraphs, words, etc. [Foissy, Manchon,...], leading to an intrinsic and purely algebraic definition of certain invariants, such as chromatic or Ehrhart polynomials.

Combinatorial Hopf algebras

In parallel with this work, many other combinatorial Hopf algebras have been discovered, based on different types of graphs, words, matrices, etc., originating from different areas of mathematics: theoretical computer science [66, 55, 36, 65], control theory (Hopf algebras associated with Fliess groups [Grey]), probability theory (free probability theory, which can be largely formulated in terms of Hopf algebras of noncrossing partitions [Arizmendi, Nica, Speicher]), etc.

Cluster algebras

The theory of cluster algebras, introduced around 2000 by Fomin and Zelevinsky [51, 49, 50, 10, 52], has since undergone rapid development, including in algebraic combinatorics. Among the notable results is the spectacular proof of their conjectures [56]. It involves regular graphs whose vertices are the clusters at the heart of this theory. In a number of important cases, these are finite graphs, which are identified with the graphs formed by the edges and vertices of certain polytopes. Finally, cluster theory allows us to equip these finite graphs with natural orientations, which define partial orders. This provides a very rich context in which polyhedral geometry, order theory and cluster mutation interact in subtle ways. This context has served as a model and inspiration for many developments. One example is a translation of all this cluster combinatorics in terms of finite Coxeter groups, initiated by Reading, under the name of Cambrian lattices. This leads in particular to a major extension to the case of finite non-crystallographic Coxeter groups, but also to new perspectives on Coxeter groups. The field has been given the name of Coxeter-Catalan combinatorics. A remarkable point is the new interpretation given to very classical combinatorial objects. Thus, the lattice of non-crossing partitions of Kreweras and the lattice of binary trees of Tamari are now understood to be associated, by a general construction, with Coxeter groups of type A, which are the symmetric groups.

4 Posets

Tamari lattices, weak order and generalisations

Around 2005, a particular interest in the algebra of planar binary trees (PBT) [71, 65] and its connection to other combinatorial Hopf algebras gave rise to several articles highlighting its central role in numerous applications in combinatorics and mathematics. In particular, the

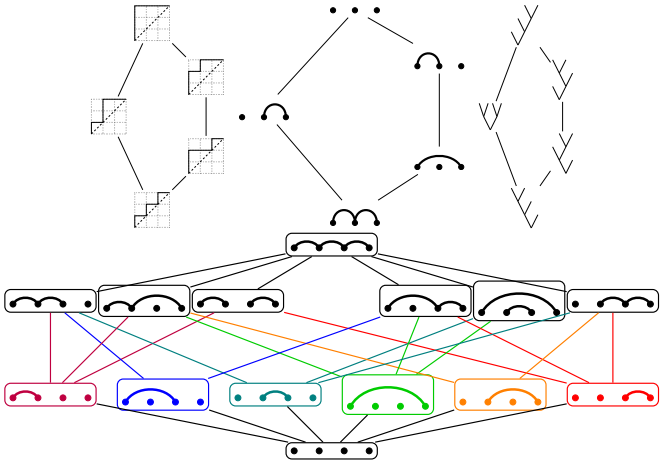


Figure 1 From left to right: Tamari lattice on Dyck paths, noncrossing partitions and planar binary trees, lattice of noncrossing partitions on four elements.

study of different bases of PBT revealed a very natural and profound partial order structure (poset) on binary trees: the Tamari order. This order coincides with the order introduced by Dov Tamari in 1962 on the correct parenthesisations of an expression. Loday and Ronco notably showed the link between Tamari's order and the Hopf algebra of tree shuffles. Since then, a great deal of research has been devoted to a better understanding of this order and its many generalisations (other quotients of the weak order, generalisation to other Coxeter types, generalisation of Dyck paths of multiple types, etc.) (see, for example [24, 26, 5, 23, 34, 74].)

Lattice properties

The research conducted also covers classical concepts in the field of partial orders and lattices, such as semi-distributivity and congruence-uniformity, but has also led to new concepts and constructions, including H. Thomas's definition of thin lattices and the Reading-Mühle fragment order, which generalises lattices of non-crossing partitions, or 'middle' orders, which are partial orders that are both finer than the weak (Bruhat) order and coarser than the strong Bruhat order, and which are distributive lattices [14].

Incidence algebra of a poset

A relatively new direction in representation theory focusses on incidence algebras of finite partial orders. These algebras have the advantage of having a subset of representations of a highly combinatorial nature, which offers useful leverage. One of the questions that arises is the study of partial orders whose derived category has a certain periodicity of the Coxeter functor. This property, which defines fractional Calabi-Yau categories, is expected for all Cambrian lattices and has only been proven for Tamari lattices. A surprising relationship has recently been obtained [72] between this periodicity and combinatorial dynamics.

5 Operads and species

Species of structure are a categorical construction introduced in the 1980s by Joyal. They are analogous to Flajolet's combinatorial classes and have recently made a comeback in the

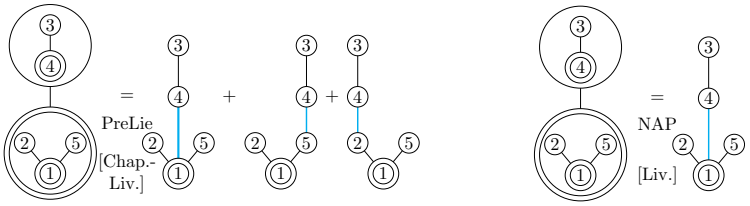


Figure 2 Two ways of sending a tree of trees to a linear combination of trees, corresponding to two operads on the species of trees: the PreLie operad [25] and the NAP operad [69]

combinatorial arsenal. They are moreover linked with linear logic (GT LHC) [47] and operad theory.

Operads

Operads have emerged as a fundamental tool in algebraic topology, and their development in this field continues. They have recently begun to spread more widely into different areas of mathematics. The theory of operads could be described as the algebra of trees, by analogy with the algebra of words. The interest of operads in algebraic combinatorics is thus similar to that long recognised in simpler and more classical algebraic structures, such as associative algebras or Hopf algebras. Defining an operad on combinatorial objects allows or requires these objects to be organised and better understood. Conversely, the study of a given operad can lead to the discovery of new combinatorial objects. The theory of operads and its many variants (cyclic operads, modular operads, properads, reconnectads, etc.) also has important connexions with Kontsevich's graph complexes, which are very important and still mysterious structures.

Generalised bialgebras

Numerous links between Hopf algebras, bialgebras and operads have also been highlighted. One of them is the notion of triple of operads, introduced by Loday [70], based on the fact that free algebras can be endowed with a bialgebraic structure in many cases. The primitive elements then inherit an additional structure, based on an operad that can be difficult to grasp [16, 18, 17, 19, 48, 69], but knowledge of which provides very rich information about the bialgebras thus obtained.

Topology of posets and operads

One final aspect worth mentioning is the link between the topology of posets and operads. This link, established in the case of partition posets by Fresse [53] and extended to decorated partition posets by Vallette [80], was initially used to demonstrate algebraic properties on operads (Koszulity). Today, it is one of the tools used to prove topological properties of posets (Cohen-Macaulay character, shellability).

6 Geometric combinatorics

Polytope combinatorics

Polytope combinatorics has been very fruitful in recent years. Among the results obtained, we can mention Santos'counterexample to Hirsch's conjecture in connection with the complexity

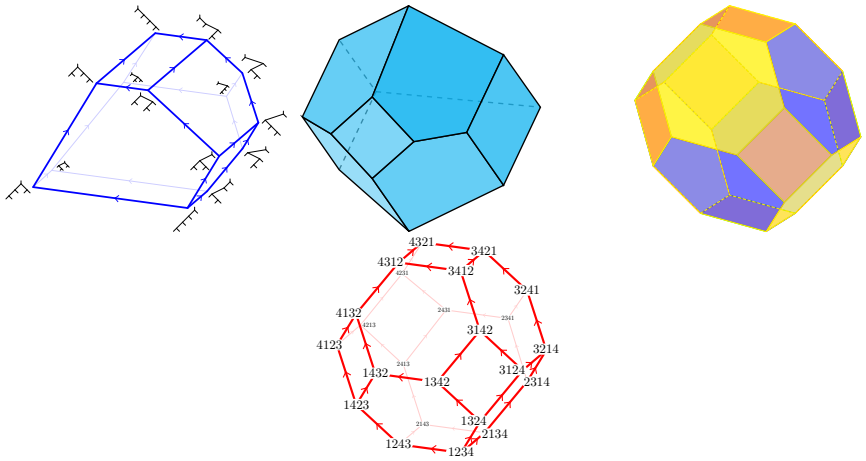


Figure 3 Some polytopes :associahedron (on the left) and permutohedron (on the right)

of the simplex algorithm [79], **awarded the Fulkerson Prize**, the study of Postnikov's generalised permutohedra [77] and all their combinatorial and algebraic ramifications, and the discovery of the amplituhedron [6] in relation to particle physics, and the multiple works on positive geometries.

Link with posets

The field of research linking posets and polytopes has also become very active in recent years. Indeed, the weak order and Tamari lattice mentioned in the previous section form the 1-skeleton of polytopes (the permutohedron and the associahedron, respectively). Among the directions explored is the construction of new examples by a simultaneous quotient procedure of the lattice and the polytope, such as quotientopes [76] or permutrees [75], which encompass and generalise the Cambrian lattices associated with symmetric groups.

Matroids

Matroid theory was introduced in 1935, as an abstraction of the combinatorial properties of linear dependence in vector spaces. The finer structure of oriented matroids, taking into account the signs in these dependencies, has been developed since the 1970s. These structures satisfy various axioms and can be seen as combinatorial abstractions of graph theory, linear algebra, and convex geometry. Over the past thirty years, applications have been developed in a wide variety of fields.

Internationally, matroids have recently been in the spotlight thanks to the work of June Huh, **recipient of the Fields Medal in 2022**. In particular, Adiprasito, Huh and Katz proved the famous unimodality conjecture ([1]), according to which the sequence of coefficients of the characteristic polynomial of a matroid is log-concave, establishing a link with Hodge theory in algebraic geometry. Some of the recent research in the field has followed this lead by combining these two approaches. From a combinatorial point of view, this research falls within the traditional field of matroid polynomials, which has been studied continuously since the 1950s (notably the Tutte polynomial, which generalises the above-mentioned polynomials, see [46]), but also beyond. On aspects related to Tutte's polynomial, in an enumerative and bijective approach, a theory of 'active bijection' describing the properties of (oriented) matroids on an ordered set has been developed in a series of articles since 2002 by Gioan

and Las Vergnas. Other specific results on these aspects include, for example: a proof of the Merino-Welsh conjecture in a special case [68]; a generalisation in Hopf algebras [39]; a theory of 'matroid set functions' by Lass in 1997 and 2024.

On geometric aspects, in hyperplane arrangement theory, oriented matroids are classic underlying structures, both as rich and practical combinatorial tools and as potential topological generalisations with, for example, various works on polytopes and geometric representations of combinatorial structures by Padrol or Pilaud, and double pseudolines by Pocchiola. Furthermore, it is worth mentioning that matroids are also related in algebraic geometry to the stratification of Grassmannians and thus appeared in the work of Laurent Lafforgue (which earned him the **Fields Medal in 2002**), leading to the still open question of when a matroid polytope can be divided into matroid polytopes, for which results in specific cases have been obtained [27, 28]. Finally, we should mention a nascent theory developed since 2015 by Chepoi, Knauer, Chalopin, Philibert et al. of 'Oriented Matroid Complexes' which generalises both oriented matroids and lopsided sets, fundamental objectsin learning theory.

7 Bijective methods

Planar maps

Planar map enumeration has its roots in the seminal work of W. Tutte in the 1960s. It is at the heart of modern developments in probability theory concerning random maps (explored by the GT Alea) and is also linked to questions in mathematical physics. An unexpected connection has emerged and is still developing between the enumeration of different types of maps and the enumeration of intervals in posets, for most variants of Tamari lattices, in particular greedy Tamari orders [34]. This has led in particular to new elegant bijections and a better understanding of the important classical bijections that connect blossoming trees and maps.

Polyominos

A polyomino is a finite union of square unit cells in the plane, whose interior is connected. The problem of enumerating polyominos according to the number of cells is a famous and difficult problem. This is why different subclasses of polyominos have been introduced in order to push the limits of enumeration techniques. Convexity, i.e. the fact that the intersection of the polyomino with any column or row is connected, is one of the best-known restrictions that have been studied in the literature. Many different methods have been successfully applied to enumerate convex polyominoes and some subfamilies: classic approaches include linear constructions involving generating trees, grammar-type decompositions, and bijective constructions.

In recent years, in the context of discrete tomography, a new classification of convex polyominoes has been proposed: a polyomino is said to be k-convex if each pair of cells can be connected by a momotonic path with at most k changes of direction. The problem of enumerating these k-convex polyominoes according to perimeter raised in [22] has led, for example, for $k = 2$, to the development of a new method that consists of generating these polyominoes by 'inflating' smaller polyominoes [38], or, in the directed case, to the introduction of an original bijective approach [13]. Guttmann and Massazza recently obtained a set of recurrences to calculate the number of k-convex polyominoes in polynomial time and in $O(n^4)$ space [57]. Another notable result is that of Bacher [8], who introduces new

bijective methods for explicitly calculating the average value of three parameters of directed polyominoes with given area and perimeter. His constructions are based on a Viennot bijection between directed animals and coin stacks and also apply to polyominoes on the triangular lattice. Finally, the class of fighting fish was introduced [37], which can be seen as surfaces that generalise directed polyominoes without holes by allowing branches: fighting fish are counted as non-separable planar maps and thus provide a link to the very active field of maps. Most recently, new subclasses of vertically convex polyominoes have been studied in relation to Motzkin or Fibonacci words [9].

Permutations

Over the last 20 years, the study of excluded patterns has seen a sharp acceleration, in particular with the exact enumeration of numerous classes of permutations that avoid one or more patterns, and with the emergence of new systematic approaches, such as Vatter's enumeration schemes [81] or the method of combinatorial exploration [4]. Vatter's enumeration schemes are an algorithmic method based on the natural recursive structure of permutation classes, which allows a recurrence equation to be constructed automatically from local construction rules. Combinatorial exploration consists of systematically exploring how pattern restrictions interact with the usual decomposition operations of the objects studied. This method has also been applied to certain classes of polyominos, words, and matrices. Although the famous open problem of enumerating permutations excluding the pattern 1324 has not been solved, significant progress has been made thanks to the exact enumeration of well-chosen subclasses [12]. Finally, even if these are asymptotic results rather than exact enumerations, it is worth mentioning the question of constructing natural limit objects for large random objects: indeed, this line of research has motivated, beyond questions of excluded patterns, numerous enumeration works for the statistics of pattern occurrence numbers in permutation classes.

8 Additive combinatorics and combinatorial number theory

Combinatorial identities and q-series

In recent years, combinatorial identities and q-series have come under the spotlight thanks to new connections. In particular, it is possible to prove quadratic duality relations between solutions of q-hypergeometric equations without using q-difference Galois theory, in a manner completely analogous to the classical case of hypergeometric differential equations [Beukers, Jouhet, Roques]. Furthermore, new considerations of graded quotient rings related to arc spaces in algebraic geometry have led to the discovery of Andrews-Gordon-type partition identities generalising those of Rogers-Ramanujan: this discovery [2] came as a surprise given the abundance of literature in the field. Finally, it was discovered that the modular structure of the generating series of partition cores that appear naturally in representation theory made it possible to enumerate them, thus opening up new fields of exploration in both algebra and combinatorics (see, for example, the work of Alpoge, Brunat, Chapelier-Laget, Gerber, Granville, Jacon, Jouhet, Lecouvey, Ono, and Wahiche).

Combinatorial number theory

Work in combinatorial number theory has mainly focused on *numerical semigroups*, i.e. submonoids of natural numbers with finite complement. These semigroups can be described as the results of the coin problem: what sums can be formed with 2 and 5 euro coins?

Introduced in the 19th century by Sylvester, these remain the subject of many open problems, including Wilf's conjecture[2] (1987) and that of Bras-Amoros (2008) [43, 40, 54, 41, 42, 33]. Another aspect concerns the growth of cardinals of iterated sum sets $A_n = \{\sum_{i=1}^{n} x_i, x_i \in A\}$ for A a subset of an abelian group or semigroup. The first general bounds date back to 1970, with Plünnecke's inequalities. These classical inequalities have been improved recently [44, 45].

9 Formal proof and experimental algebraic combinatorics

Combinatorics and formal proof

In recent years, the algebraic combinatorics community has grown closer to the mathematical formalisation community in several ways: first, algebraic combinatorics benefits from its unique position, studying both advanced algebraic structures (Hopf algebras, Operads, etc.) and non-trivial algorithmic objects. This intertwining of mathematics and algorithmics poses formalisation challenges that push proof systems and their libraries to their limits. Secondly, the question of how much trust can be placed in computing systems is becoming increasingly pressing as their use in proofs increases. It is becoming important to explore how to perform combinatorics more reliably. As published proofs are also becoming increasingly complex, we must also look at how they can be verified. One example is the proof of Littlewood-Richardson's rule, where many false proofs have been accepted, some for more than two decades.

Software environment

From the perspective of algebraic combinatorics, the last 20 years have been characterised by significant developments in the software environment and an increasing use of machine experimentation for the study and understanding of discrete combinatorial objects. Experimental algebraic combinatorics offers an innovative approach to exploring discrete structures through an ingenious combination of theoretical results, algorithms and experiments. For example, it allows us to construct mathematical objects with certain desired properties or to conjecture certain phenomena. To pool the software development efforts required, the community has gradually structured itself to share libraries (SF, guess, ACE, muEC for instance). However, the underlying general-purpose calculation systems remained closed, imposing ethical, scientific, technical and practical limitations. A major breakthrough was the launch in 2005, spearheaded by William Stein, of the open-source general-purpose computing system SageMath, based on Python and a large number of specialised libraries. The contribution to our community is considerable, both in practical terms (calculation, experimentation, prototyping) and in structural terms (creation of an open, sustainable, reproducible ecosystem). At the heart of these advances is the Sage-Combinat community (initially MuPAD-Combinat), which since 2000 has brought together combinatorial researchers from around the world who wish to turn to open development. Aong all the significant contributions, let us cite Nicolas Thiéry's, who led the design and development of the category infrastructure that allows the management of large hierarchies of algebraic structures (groups, rings, fields, Hopf algebras, ...).

[2] See `https://paisajes.math.cnrs.fr/Semigroupes-numeriques-et-conjecture-de-Wilf-3865.html` and `https://images.math.cnrs.fr/les-extraordinaires-predictions-du-reverend-walker/` for instance.

High-performance computing

Typical search spaces in algebraic combinatorics grow exponentially with the size of the input, which is known as the combinatorial explosion. In this regard, algorithmic efficiency is of the utmost importance. Recent technological developments in processors (multi-core, vector units) and computing environments (distributed computing, use of GPUs) offer several ways to improve machine experimentation. To take full advantage of these technologies, it is necessary to design new algorithms for algebraic combinatorics that are adapted to the various technical constraints. This algorithmic research will require a complete rethinking of existing algorithms and the introduction of new ways of representing the underlying mathematical objects.

Contributors.

Frédéric Chapoton, Bérénice Delcroix-Oger, Enrica Duchi, Loïc Foissy, Jean Fromentin, Emeric Gioan, Florent Hivert, Frédéric Jouhet, Philippe Nadeau, Jean-Christophe Novelli, Vincent Pilaud, Viviane Pons, Adrian Tanasa, Nicolas Thiéry. We are also grateful to Bérénice Delcroix-Oger and Vincent Pilaud for providing the images that illustrate this document.

References

1 Karim Adiprasito, June Huh, and Eric Katz. Hodge theory for combinatorial geometries. *Ann. of Math. (2)*, 188(2):381–452, 2018. URL: https://doi.org/10.4007/annals.2018.188.2.1, doi:10.4007/annals.2018.188.2.1.

2 Pooneh Afsharijoo, Jehanne Dousse, Frédéric Jouhet, and Hussein Mourtada. New companions to the Andrews-Gordon identities motivated by commutative algebra. *Adv. Math.*, 417:40, 2023. Id/No 108946. doi:10.1016/j.aim.2023.108946.

3 Marcelo Aguiar, Nantel Bergeron, and Frank Sottile. Combinatorial Hopf algebras and generalized Dehn-Sommerville relations. *Compos. Math.*, 142(1):1–30, 2006. doi:10.1112/S0010437X0500165X.

4 Michael H. Albert, Christian Bean, Anders Claesson, Émile Nadeau, Jay Pantone, and Henning Ulfarsson. Combinatorial exploration: An algorithmic framework for enumeration, 2024. URL: https://arxiv.org/abs/2202.07715, arXiv:2202.07715.

5 D. Albertin, V. Pilaud, and J. Ritter. Removahedral congruences versus permutree congruences. *Eletronic Journal of Combinatorics*, 28(4), 2021. doi:10.37236/10214.

6 Nima Arkani-Hamed and Jaroslav Trnka. The Amplituhedron. *Journal of High Energy Physics*, 2014(10):30, October 2014. doi:10.1007/JHEP10(2014)030.

7 D. Armstrong, A. Garsia, J. Haglund, B. Rhoades, and B. Sagan. Combinatorics of Tesler matrices in the theory of parking functions and diagonal harmonics. *J. Comb.*, 3(3):451–494, 2012. doi:10.4310/JOC.2012.v3.n3.a7.

8 Axel Bacher. Average site perimeter of directed animals on the two-dimensional lattices. *Discrete Mathematics*, 312(5):1038–1058, 2012. URL: https://www.sciencedirect.com/science/article/pii/S0012365X1100505X, doi:https://doi.org/10.1016/j.disc.2011.11.008.

9 Jean-Luc Baril, Sergey Kirgizov, José Ramírez, and Diego Villamizar. The Combinatorics of Motzkin Polyominoes. *Discrete Applied Mathematics*, 364:1–15, 2025. URL: https://hal.science/hal-04396241, doi:10.1016/j.dam.2024.12.002.

10 Arkady Berenstein, Sergey Fomin, and Andrei Zelevinsky. Cluster algebras. III: Upper bounds and double Bruhat cells. *Duke Math. J.*, 126(1):1–52, 2005. doi:10.1215/S0012-7094-04-12611-9.

11 F. Bergeron, A. M. Garsia, M. Haiman, and G. Tesler. Identities and positivity conjectures for some remarkable operators in the theory of symmetric functions. volume 6, pages 363–420.

1999. Dedicated to Richard A. Askey on the occasion of his 65th birthday, Part III. URL: https://doi.org/10.4310/MAA.1999.v6.n3.a7, doi:10.4310/MAA.1999.v6.n3.a7.

12 David Bevan, Robert Brignall, Andrew Elvey Price, and Jay Pantone. A structural characterisation of av(1324) and new bounds on its growth rate. *European Journal of Combinatorics*, 88:103115, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0195669820300366, doi:https://doi.org/10.1016/j.ejc.2020.103115.

13 A. Boussicault, S. Rinaldi, and S. Socci. The number of directed k-convex polyominoes. *Discrete Mathematics*, 343(3):111731, 2020. URL: https://www.sciencedirect.com/science/article/pii/S0012365X19304091, doi:https://doi.org/10.1016/j.disc.2019.111731.

14 Mathilde Bouvel, Luca Ferrari, and Bridget Eileen Tenner. Between Weak and Bruhat: The Middle Order on Permutations. *Graphs and Combinatorics*, 41(2):34, February 2025. doi:10.1007/s00373-024-02885-3.

15 Y. Bruned, M. Hairer, and L. Zambotti. Algebraic renormalisation of regularity structures. *Invent. Math.*, 215(3):1039–1156, 2019. doi:10.1007/s00222-018-0841-x.

16 Emily Burgunder. Infinite magmatic bialgebras. *Adv. Appl. Math.*, 40(3):309–329, 2008. doi:10.1016/j.aam.2006.12.005.

17 Emily Burgunder and Bérénice Delcroix-Oger. Structure theorems for dendriform and tridendriform algebras. In *Algebraic combinatorics, resurgence, moulds and applications (CARMA). Volume 1*, pages 19–66. Berlin: European Mathematical Society (EMS), 2020. doi:10.4171/204-1/2.

18 Emily Burgunder and Ralf Holtkamp. Partial magmatic bialgebras. *Homology Homotopy Appl.*, 10(2):59–81, 2008. URL: intlpress.com/hha/v10/n2/, doi:10.4310/HHA.2008.v10.n2.a3.

19 Emily Burgunder and María Ronco. Tridendriform structure on combinatorial Hopf algebras. *J. Algebra*, 324(10):2860–2883, 2010. doi:10.1016/j.jalgebra.2010.07.010.

20 Damien Calaque, Kurusch Ebrahimi-Fard, and Dominique Manchon. Two interacting Hopf algebras of trees: a Hopf-algebraic approach to composition and substitution of B-series. *Adv. Appl. Math.*, 47(2):282–308, 2011. doi:10.1016/j.aam.2009.08.003.

21 Erik Carlsson and Anton Mellit. A proof of the shuffle conjecture. *J. Am. Math. Soc.*, 31(3):661–697, 2018. doi:10.1090/jams/893.

22 G. Castiglione, A. Frosini, A. Restivo, and S. Rinaldi. Enumeration of l-convex polyominoes by rows and columns. *Theoretical Computer Science*, 347(1):336–352, 2005. URL: https://www.sciencedirect.com/science/article/pii/S030439750500469X, doi:https://doi.org/10.1016/j.tcs.2005.06.031.

23 Cesar Ceballos and Clément Chenevière. On linear intervals in the alt \(\nu\)-Tamari lattices. *Combinatorial Theory*, 4(2), 2024. doi:10.5070/C64264254.

24 Frédéric Chapoton. Sur le nombre d'intervalles dans les treillis de Tamari. *Sémin. Lothar. Comb.*, 55:b55f, 18, 2005.

25 Frédéric Chapoton and Muriel Livernet. Pre-lie algebras and the rooted trees operad. *Int. Math. Res. Not.*, 2001(8):395–408, 2001. doi:10.1155/S1073792801000198.

26 Grégory Châtel and Viviane Pons. Counting smaller elements in the Tamari and m-Tamari lattices. *J. Comb. Theory, Ser. A*, 134:58–97, 2015. doi:10.1016/j.jcta.2015.03.004.

27 Vanessa Chatelain and Jorge Luis Ramírez Alfonsín. Matroid base polytope decomposition. *Adv. Appl. Math.*, 47(1):158–172, 2011. doi:10.1016/j.aam.2010.04.005.

28 Vanessa Chatelain and Jorge Luis Ramírez Alfonsín. Matroid base polytope decomposition II: Sequences of hyperplane splits. *Adv. Appl. Math.*, 54:121–136, 2014. doi:10.1016/j.aam.2013.11.003.

29 Sylvie Corteel, Matthieu Josuat-Vergès, and Anna Vanden Wyngaerd. Combinatorics of the Delta conjecture at $q = -1$. *Algebr. Comb.*, 7(1):17–35, 2024. doi:10.5802/alco.329.

30 Michele D'Adderio and Alessandro Iraci. Some consequences of the valley delta conjectures. *Ann. Comb.*, 29(1):25–46, 2025. doi:10.1007/s00026-023-00663-1.

31 Michele D'Adderio, Alessandro Iraci, and Anna Vanden Wyngaerd. *Decorated Dyck paths, polyominoes, and the delta conjecture*, volume 1370 of *Mem. Am. Math. Soc.* Providence, RI: American Mathematical Society (AMS), 2022. doi:10.1090/memo/1370.

32 Michele D'Adderio and Anton Mellit. A proof of the compositional Delta conjecture. *Adv. Math.*, 402:17, 2022. Id/No 108342. doi:10.1016/j.aim.2022.108342.

33 Manuel Delgado, Shalom Eliahou, and Jean Fromentin. A verification of Wilf's conjecture up to genus 100. *J. Algebra*, 664:150–163, 2025. doi:10.1016/j.jalgebra.2024.10.028.

34 Aram Dermenjian. Maximal degree subposets of ν-Tamari lattices. *Electron. J. Comb.*, 30(2):research paper p2.43, 40, 2023. doi:10.37236/11571.

35 Martin Desombre, Baptiste Magnaudet, Jean-Michel Do, Jean Fromentin, and Sébastien Verel. Harmonic analysis of a repositioning plan optimization problem. In *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. 2025.

36 G. Duchamp, F. Hivert, and J.-Y. Thibon. Noncommutative symmetric functions. VI. Free quasi-symmetric functions and related algebras. *International Journal of Algebra and Computation*, 12(5):671–717, 2002. doi:10.1142/S0218196702001139.

37 E Duchi, V Guerrini, S Rinaldi, and G Schaeffer. Fighting fish*. *Journal of Physics A: Mathematical and Theoretical*, 50(2):024002, dec 2016. URL: https://doi.org/10.1088/1751-8121/50/2/024002, doi:10.1088/1751-8121/50/2/024002.

38 Enrica Duchi, Simone Rinaldi, and Gilles Schaeffer. The number of z-convex polyominoes. *Advances in Applied Mathematics*, 40(1):54–72, 2008. URL: https://www.sciencedirect.com/science/article/pii/S0196885806002004, doi:https://doi.org/10.1016/j.aam.2006.07.004.

39 Clément Dupont, Alex Fink, and Luca Moci. Universal Tutte characters via combinatorial coalgebras. *Algebr. Comb.*, 1(5):603–651, 2018. doi:10.5802/alco.35.

40 Shalom Eliahou. Wilf's conjecture and Macaulay's theorem. *J. Eur. Math. Soc. (JEMS)*, 20(9):2105–2129, 2018. doi:10.4171/JEMS/807.

41 Shalom Eliahou. A graph-theoretic approach to Wilf's conjecture. *Electron. J. Comb.*, 27(2):research paper p2.15, 31, 2020. doi:10.37236/9106.

42 Shalom Eliahou and Jean Fromentin. Gapsets and numerical semigroups. *J. Comb. Theory, Ser. A*, 169:19, 2020. Id/No 105129. doi:10.1016/j.jcta.2019.105129.

43 Shalom Eliahou and Michel Kervaire. Minimal sumsets in finite solvable groups. *Discrete Math.*, 310(3):471–479, 2010. doi:10.1016/j.disc.2009.03.024.

44 Shalom Eliahou and Eshita Mazumdar. Iterated sumsets and Hilbert functions. *J. Algebra*, 593:274–294, 2022. doi:10.1016/j.jalgebra.2021.11.019.

45 Shalom Eliahou and Eshita Mazumdar. Optimal bounds on the growth of iterated sumsets in abelian semigroups. *Ann. Inst. Fourier*, 75(6):2321–2339, 2025. doi:10.5802/aif.3674.

46 Joanna A. Ellis-Monaghan and Iain Moffatt, editors. *Handbook of the Tutte polynomial and related topics*. Boca Raton, FL: CRC Press, 2022. doi:10.1201/9780429161612.

47 M. Fiore, N. Gambino, M. Hyland, and G. Winskel. The Cartesian closed bicategory of generalised species of structures. *J. Lond. Math. Soc., II. Ser.*, 77(1):203–220, 2008. doi:10.1112/jlms/jdm096.

48 Loïc Foissy. Bidendriform bialgebras, trees, and free quasi-symmetric functions. *J. Pure Appl. Algebra*, 209(2):439–459, 2007. doi:10.1016/j.jpaa.2006.06.005.

49 Sergey Fomin and Andrei Zelevinsky. Cluster algebras. I: Foundations. *J. Am. Math. Soc.*, 15(2):497–529, 2002. doi:10.1090/S0894-0347-01-00385-X.

50 Sergey Fomin and Andrei Zelevinsky. Cluster algebras. II: Finite type classification. *Invent. Math.*, 154(1):63–121, 2003. URL: hdl.handle.net/2027.42/46591, doi:10.1007/s00222-003-0302-y.

51 Sergey Fomin and Andrei Zelevinsky. Y-systems and generalized associahedra. *Ann. Math. (2)*, 158(3):977–1018, 2003. doi:10.4007/annals.2003.158.977.

52 Sergey Fomin and Andrei Zelevinsky. Cluster algebras. IV: Coefficients. *Compos. Math.*, 143(1):112–164, 2007. doi:10.1112/S0010437X06002521.

53 Benoit Fresse. Koszul duality of operads and homology of partition posets. In *Homotopy theory: relations with algebraic geometry, group cohomology, and algebraic K-theory. Papers from the international conference on algebraic topology, Northwestern University, Evanston, IL, USA, March 24–28, 2002*, pages 115–215. Providence, RI: American Mathematical Society (AMS), 2004.

54 Jean Fromentin and Florent Hivert. Exploring the tree of numerical semigroups. *Math. Comput.*, 85(301):2553–2568, 2016. doi:10.1090/mcom/3075.

55 Israel M. Gelfand, Daniel Krob, Alain Lascoux, Bernard Leclerc, Vladimir S. Retakh, and Jean-Yves Thibon. Noncommutative symmetric functions. *Advances in Mathematics*, 112:218, 1995.

56 Mark Gross, Paul Hacking, Sean Keel, and Maxim Kontsevich. Canonical bases for cluster algebras. *J. Am. Math. Soc.*, 31(2):497–608, 2018. doi:10.1090/jams/890.

57 Guttmann, Anthony J. and Massazza, Paolo. Efficient counting of k-convex polyominoes. *RAIRO-Theor. Inf. Appl.*, 59:11, 2025. URL: https://doi.org/10.1051/ita/2025011, doi:10.1051/ita/2025011.

58 J. Haglund. Conjectured statistics for the q,t-Catalan numbers. *Adv. Math.*, 175(2):319–334, 2003. doi:10.1016/S0001-8708(02)00061-0.

59 J. Haglund, M. Haiman, N. Loehr, J. B. Remmel, and A. Ulyanov. A combinatorial formula for the character of the diagonal coinvariants. *Duke Math. J.*, 126(2):195–232, 2005. URL: https://doi.org/10.1215/S0012-7094-04-12621-1, doi:10.1215/S0012-7094-04-12621-1.

60 J. Haglund and N. Loehr. A conjectured combinatorial formula for the Hilbert series for diagonal harmonics. *Discrete Math.*, 298(1-3):189–204, 2005. doi:10.1016/j.disc.2004.01.022.

61 J. Haglund, J. Morse, and M. Zabrocki. A compositional shuffle conjecture specifying touch points of the Dyck path. *Can. J. Math.*, 64(4):822–844, 2012. doi:10.4153/CJM-2011-078-4.

62 J. Haglund, J. B. Remmel, and A. T. Wilson. The delta conjecture. *Trans. Amer. Math. Soc.*, 370(6):4029–4057, 2018. URL: https://doi.org/10.1090/tran/7096, doi:10.1090/tran/7096.

63 James Haglund. *The q,t-Catalan numbers and the space of diagonal harmonics. With an appendix on the combinatorics of Macdonald polynomials*, volume 41. Providence, RI: American Mathematical Society (AMS), 2008.

64 James Haglund. A polynomial expression for the Hilbert series of the quotient ring of diagonal coinvariants. *Adv. Math.*, 227(5):2092–2106, 2011. doi:10.1016/j.aim.2011.04.013.

65 F. Hivert, J.-C. Novelli, and J.-Y. Thibon. The algebra of binary search trees. *Theoretical Computer Science*, 339(1):129–165, 2005. doi:10.1016/j.tcs.2005.01.012.

66 Michael Hoffman. Quasi-shuffle algebras and applications. *EMS Press eBooks*, February 2020. doi:10.4171/205-1/5.

67 Alessandro Iraci, Philippe Nadeau, and Anna Vanden Wyngaerd. Smirnov words and the delta conjectures. *Adv. Math.*, 452:41, 2024. Id/No 109793. doi:10.1016/j.aim.2024.109793.

68 Kolja Knauer, Leonardo Martínez-Sandoval, and Jorge Luis Ramírez Alfonsín. A Tutte polynomial inequality for lattice path matroids. *Adv. Appl. Math.*, 94:23–38, 2018. doi:10.1016/j.aam.2016.11.008.

69 Muriel Livernet. A rigidity theorem for pre-Lie algebras. *J. Pure Appl. Algebra*, 207(1):1–18, 2006. doi:10.1016/j.jpaa.2005.10.014.

70 Jean-Louis Loday. *Generalized bialgebras and triples of operads*, volume 320 of *Astérisque*. Paris: Société Mathématique de France, 2008. URL: smf4.emath.fr/Publications/Asterisque/2008/320/html/smf_ast_320.html.

71 Jean-Louis Loday and María O. Ronco. Hopf algebra of the planar binary trees. *Adv. Math.*, 139(2):293–309, 1998. doi:10.1006/aima.1998.1759.

72 René Marczinzik, Hugh Thomas, and Emine Yıldırım. On the interaction of the Coxeter transformation and the rowmotion bijection. *J. Comb. Algebra*, 8(3-4):359–374, 2024. doi:10.4171/JCA/101.

73 Philippe Nadeau, Hunter Spink, and Vasu Tewari. Schubert polynomial expansions revisited. *Forum Math. Sigma*, 13:25, 2025. Id/No e106. doi:10.1017/fms.2025.10068.

74 Eva Philippe and Vincent Pilaud. Geometric realizations of the s-weak order and its lattice quotients. *Journal of the London Mathematical Society*, 112(3):e70268, 2025. doi:10.1112/jlms.70268.

75 V. Pilaud and V. Pons. Permutrees. *Algebraic Combinatorics*, 1(2):173–224, 2018. doi:10.5802/alco.1.

76 Vincent Pilaud and Francisco Santos. Quotientopes. *Bulletin of the London Mathematical Society*, 51(3):406–420, 2019. doi:10.1112/blms.12231.

77 Alexander Postnikov. Permutohedra, Associahedra, and Beyond. *International Mathematics Research Notices*, 2009(6):1026–1106, January 2009. doi:10.1093/imrn/rnn153.

78 Brendon Rhoades and Andrew Timothy Wilson. The Hilbert series of the superspace coinvariant ring. *Forum Math. Pi*, 12:35, 2024. Id/No e16. doi:10.1017/fmp.2024.14.

79 Francisco Santos. A counterexample to the Hirsch Conjecture | Annals of Mathematics. *Annals of Mathematics*, 176(1):383–412, 2012.

80 Bruno Vallette. Homology of generalized partition posets. *J. Pure Appl. Algebra*, 208(2):699–725, 2007. doi:10.1016/j.jpaa.2006.03.012.

81 Vincent Vatter. Enumeration schemes for restricted permutations. *Combinatorics, Probability and Computing*, 17(1):137–159, 2008. doi:10.1017/S0963548307008516.

Twenty Years of GdR IFM, seen from GT Dynamical Systems, Automata, and Algorithms

1 Introduction

The main research topics studied by the SDA2 workgroup are at the frontier between theoretical computer science, combinatorics, number theory, discrete geometry, ergodic theory and topological dynamics. Some of them are common with others groups of the GDR, mainly GT Alea and GT Calculabilités. In addition to their inherent appeal, scientific motivations also arise from physical questions — such as those related to quasicrystals — and biological ones, like protein folding or DNA/RNA assembly.

We highlight here some major results and research directions of the last 20 years.

The notion of *symbolic system* is at the core of the studies of SDA2. Symbolic systems are discrete time dynamical systems, whose elements are colorings of $\mathbb{Z}^d$ by a finite alphabet which respect some local constraints (such a set of colorings is called a *subshift*). The dynamics may be the *shift* action or more generally a *cellular automaton* (also called sliding block-code): an update rule which is the same for each cell and only depends on a finite neighborhood. These systems are also computational models.

2 Cellular Automata

Problems related to cellular automata are at the heart of the research activities of the members of the SDA2 group. Recent advances have changed the way these objects are viewed, in particular their connection to the structure of the subshift on which they act.

A fruitful approach to studying cellular automata (CA) is to interpret them within the framework of dynamical systems, i.e., as continuous shift-commuting functions, thereby establishing a canonical link with symbolic dynamics. This perspective, initiated by Hedlund, makes it possible to use tools from topology and ergodic theory and reveals that cellular automata exhibit a wide diversity of dynamical behaviors, ranging from the simplest to the most complex.

A classical problem consists in studying all reversible cellular automata (called automorphisms) that preserve a given subshift. Recently, a series of papers by S. Petite, F. Durand, A. Maass, and S. Donoso, V. Cyr, B. Kra [17, 16] have led to a good understanding of these automorphisms. In particular, they show that the growth rate of the complexity of the subshift restricts the algebraic properties of its automorphisms, such as their growth rate and the presence of distorsion elements. It follows that the automorphism group of subshifts with linear complexity is genuinely constrained, and that neither the Baumslag–Solitar group $BS(1,n)$, $n \geq 2$, nor $SL(3,\mathbb{Z})$ can occur as subgroups of the automorphism group of a zero-entropy subshift. This constitutes the first known algebraic restriction for a zero-entropy system.

By contrast, A. Callard and V. Salo [11] have shown that the automorphism group of the full shift contains a distorted automorphism. For multidimensional subshifts of finite type, the situation is rather mysterious: P. Guillon, E. Jeandel, J. Kari, and P. Vanier [29] present an example of such a system whose automorphism group has an undecidable word problem .

Classification, *i.e.* whether two systems are isomorphic, is a central problem in dynamical systems, in particular for families that arise in a wide range of contexts, such as substitution subshifts. Usually invariants are used for this purpose, but currently known invariants are not

sufficient. In 2022, F. Durand and J. Leroy [22] showed that it is possible to decide whether a given cellular automaton defines a map between two prescribed minimal substitution subshifts. Building on these results, they provide an algorithm which, given two minimal substitution subshifts, determines whether one is a factor of the other and, as a direct consequence whether they are isomorphic.

3 Links between arithmetic and coding

The coding in a (finite memory) computer of an (irrationnal) real number or the coordinates of a vector is a classical but fondamental problem in computer science. Proceeding by approximations even with a very high number of digits is not enough when we have to iterate computations using this real, e.g. when computing one single orbit of a translation on the d-dimensional torus.

Substitutions are combinatorial objects (one replaces a letter by a word) which produce sequences by iteration and generate symbolic dynamical systems. These systems, produced by this elementary algorithmic process, have a highly ordered self-similar structure. Substitutions occur in many mathematical fields (combinatorics on words, digital geometry, ergodic theory and spectral theory, Diophantine approximation and transcendence, as well as in theoretical computer science or physics). The connections with numeration systems are numerous and natural. For example, the analogy between substitutions and beta-numeration is underscored by the work of Rauzy and Thurston, which led to the *Pisot conjecture*. This conjecture posits that a substitution satisfying certain arithmetic conditions generates a symbolic system that is measure-theoretically equivalent (i.e. up to a statistically negligible set) to a rotation on a torus.

Inspired by Rauzy's approach and generalizing substitutions to S-adic systems (given by sequence of substitutions), V. Berthé, W. Steiner and J. Thuswaldner [7], as well as the N. Pytheas Fogg and C. Noûs group [27], give codings in terms of multidimensional continued fractions algorithms.

In particular, given a continued fraction algorithm with exponential convergence, these sequences lead to renormalization schemes that naturally produce symbolic codings of toral translations and bounded remainder sets at all scales. Moreover, these examples have the lowest possible factor complexity (more precisely, the number of factor of length n is$2n + 1$). These examples include classical algorithms such as the Jacobi–Perron, Brun, Cassaigne–Selmer, and Arnoux–Rauzy algorithms.

4 Conjectures in symbolic dynamics

We present here several well-known long-standing conjectures that have been resolved over the past twenty years by members of SDA2.

4.1 Cobham's Theorem

Cobham's theorem (1969) is a deep result in theoretical computer science, linking finite automata, formal languages, and number theory. It concerns automatic sets: subsets of integers recognizable by a finite automaton reading their digits in base p. Cobham's theorem states that the only sequences recognizable in two multiplicatively independent bases (e.g., 2 and 3) are ultimately periodic. This theorem reveals a profound connection between automata, numeration bases, and periodicity, highlighting the difficulty of transcribing an automatic sequence from one base to another.

S. Eilenberg criticized the original proof of Cobham's theorem (1969) as "correct, long, and difficult" [23], prompting the search for a more accessible proof. Cobham later clarified that p-recognizable sets are the images, under letter-to-letter morphisms, of fixed points of constant-length substitutions [14]. The theorem was extended by Semenov to recognizable subsets of $\mathbb{N}^d$, giving rise to the Cobham–Semenov theorem, and was then generalized to non-standard numeration systems (such as the Fibonacci numeration system). More accessible proofs subsequently emerged, notably via logical approaches (Michaux, Villemaire [42]) and dynamical approaches (Durand [21]), linking automaticity, logic, and dynamical systems. Finally, more recent works (Fabre, Bruyère-Point, Bès [44]) generalized this framework to non-standard numeration systems associated with Pisot numbers, further extending the scope of the theorem.

The Cobham theorem has been extended far beyond its original setting, touching geometry, dynamics, algebra, and even analysis, with applications to tilings, formal power series, and real numbers. In particular, the works of Boigelot, Brusten, and Bruyère [8] generalized Cobham's theorem to recognizable subsets of $\mathbb{R}$ or $\mathbb{R}^d$ in standard numeration systems, paving the way for a geometric and dynamical interpretation. From the viewpoint of algebraic geometry, the work of Christol (1979) characterized algebraic formal power series in $\mathbb{F}_q((t))$ by p-automatic sequences [38]. Subsequently, Kedlaya generalized this result to Hahn series $\mathbb{F}_q((t^{\mathbb{Q}}))$ via quasi-automatic functions. Adamczewski and Bell extended Cobham's theorem (2008) to quasi-automatic functions [1], showing that a sequence of coefficients represents two algebraic power series in distinct characteristics if and only if these series are rational. Beyond finite fields, and more recently (2024), B. Adamczewski and C. Faverjon proved that an irrational real number can be automatic in two multiplicatively independent integer bases using a quasi-automatic function and Mahler's method [2].

4.2 The $\mathcal{S}$-adic Conjecture

Recent approaches show that zero-entropy systems have a very restrictive combinatorial structure. This viewpoint, initiated by Morse and Hedlund for Sturmian sequences, has been enriched by other examples (substitutive systems, Toeplitz systems, interval exchange transformations, dendric sequences, etc.). This intuition, developed between the 1970s and 1990s, led to the $\mathcal{S}$-adic conjecture for subshifts with the lowest possible complexity.

There exists an $\mathcal{S}$-adic characterization (i.e., in terms of concatenations of monoid morphisms) of subshifts with linear complexity.

Formulated by S. Ferenczi and attributed to B. Host and the Marseille school, this conjecture was resolved in 2023 thanks to the work of B. Espinoza (a PhD student of F. Durand), who solved it in the minimal case. His characterization is explicit. He also proposes a characterization of systems with non-superlinear complexity [26]. These structural theorems will allow for a better understanding and interpretation of these systems.

4.3 Nivat's Conjecture

It has long been observed that low complexity systems exhibit a rigid structure. One of the earliest illustrations of this principle was highlighted by Morse and Hedlund in 1938, who showed that a bi-infinite sequence with few distinct words (more precisely, when for some length n the number of factors of length n is less than n) must necessarily be periodic. Motivated by the discovery of quasicrystals, which display strong long-range order

while remaining aperiodic, the question arises as to the weakest condition that still implies aperiodicity.

In 1997, Maurice Nivat formulated a conjecture concerning two-dimensional configurations (colorings of the integer lattice $\mathbb{Z}^2$) and their local patterns. More precisely, if for some integers $n, m \geq 1$ the number $P_c(m, n)$ of distinct patterns (restrictions) of a configuration c on arbitrary $n \times m$ rectangles is at most nm, then the conjecture asserts that c must be periodic in some direction.

Since then, Nivat's conjecture has been the subject of extensive research. It is, however, known to be false in dimension $d \geq 3$ (Sander–Tijdeman) and for non-convex domains (Cassaigne [13]). In particular, Cassaigne exhibited periodic examples that do not satisfy the complexity assumption. Numerous partial results toward the conjecture have been established. For instance, there are proofs showing the existence of a period whenever $P_c(n, m) \leq \alpha nm$ for various constants $\alpha < 1$: Epifanio–Koskas–Mignosi [24] obtained $\alpha = 1/144$, later improved to $\alpha = 1/16$ by Quas–Zamboni [45]. At present, the best bound for α was obtained by Cyr and Kra, with $\alpha = 1/2$. The optimization of this constant with respect to the size of the alphabet was achieved by Colle and Garibaldi. Sander and Tijdeman established periodicity when $P_c(2, m) \leq 2m$ for some $m \geq 1$, and similarly Cyr and Kra obtained the same result when $P_c(3, m) \leq 3m$.

Jarkko Kari and Balázs Szabados achieved a major breakthrough by introducing an algebraic approach to Nivat's conjecture. Their method relies on tools from algebraic geometry, notably Hilbert's Nullstellensatz. In 2019, Kari and Étienne Moutot further developed this algebraic approach [37]. They proved that any two-dimensional configuration c of low complexity (with respect to a convex shape) contains, in its orbit closure (the smallest subshift containing c), a periodic configuration. This result means that c contains arbitrarily large periodic regions, even if it is not globally periodic. To date, these works represent the strongest progress toward Nivat's conjecture.

All these results suggest that the appropriate framework for understanding low-complexity sequences is to interpret them in terms of first-order formulas in Presburger arithmetic.

4.4 In search of the smallest set of Wang tiles

The domino problem, which asks whether a set of tiles can tile the plane $\mathbb{Z}^2$, was first formulated by Wang in the 1960s. A key ingredient in the proof of undecidability is the existence of an aperiodic set of tiles. While the initial proof paid little attention to the number of tiles (the first construction used more than 20,000 tiles), the search for the smallest tileset soon became a goal. The number of tiles was gradually reduced: first by Berger himself to 104 tiles, then by others over time (92 tiles [39], 35 tiles [46], 16 tiles [3], 14 tiles [36], 13 tiles [15]). Most of these constructions are based on self-similarity, but perhaps the most remarkable is that of [36], which introduces a completely new method for creating aperiodic tilesets based on representing real numbers along the rows of the tilings.

For a long time, the 13-tile bound could not be improved, but a definitive answer was given by Jeandel and Rao [34] recently: there exists a set of 11 tiles that tiles the plane aperiodically, and no such set exists with 10 tiles. This result was obtained both through an exhaustive computer search and by a manual proof on the candidate aperiodic tilesets identified. The resulting 11 tile tileset turned out to be self-similar, like most of the previously constructed aperiodic tilesets.

5 Computability and dynamical systems

A major shift in perspective occurred around 2010 with the use of computability theory to study discrete dynamical systems. Whereas previously the tool of choice had been language theory and computability was seen as preventing their study[1], it is now considered one of the standard tools.

This shift in perspective was initiated by the article [32], which introduced tilings with computably enumerable constraints: effective subshifts. One of the major results of that paper, later refined in [20, 5], is that 1-dimensional effective subshifts are exactly the projections of rows of shifts of finite type, i.e., subshifts defined by a finite set of forbidden patterns. This result later became a tool to easily construct subshifts satisfying certain properties.

Shortly after [32], computability was used for the first time to characterize a dynamical aspect of subshifts: their entropy [33]. This line of research has been fruitful, and many results from the SDA2 community follow this approach and characterize isomorphism invariants of subshifts or cellular automata through computability: invariants based on pattern growth [12, 30], projective subdynamics [28], limit sets of cellular automata [10, 25, 9, 31], characterization of periods using complexity classes [35], and more.

6 Tilings and groups

The structures on which tilings are traditionally considered are the discrete plane $\mathbb{Z}^2$ or the continuous plane $\mathbb{R}^2$. A major difference with one-dimensional tilings (tilings on $\mathbb{Z}$) is the undecidability of the domino problem in the plane. A recent line of research, whose goal is in part to understand the boundary between decidability and undecidability of the domino problem, is to study it on groups.

The development of this topic followed the article [6], in which it is conjectured that the domino problem is decidable on a group if and only if the group is virtually free. An overview of these topics was written by members of the working group [4], and many articles in this direction have been produced by the SDA2 community. This line of research has also led to the introduction of new notions on groups originating from symbolic dynamics, such as the notion of a self-simulable group [47], that is, groups for which the classes of effective and sofic subshifts coincide.

7 Self-assembly and pumping

Another topic present in the working group concerns bio-inspired models of computation, and in particular self-assembly. This is a model close to Wang tiles, where colors are called "glues" and are assigned a certain strength. These tiles are placed one after another on the grid $\mathbb{Z}^2$, starting from a seed which is a pre-assembly of several tiles. The assembly takes place at a given temperature τ, which determines the minimal glue strength required for attachment: each tile must attach to one or more existing tiles in such a way that the sum of the strengths exceeds the temperature. Several variants of these models exist (see the surveys [43, 18]).

The study of the computational power of this model has been a major question over the past two decades. This model is intrinsically universal starting at temperature 2 [19],

[1] One of the authors of the reference book *An Introduction to Symbolic Dynamics and Coding* [40], Doug Lind, even wrote: "The swamp of undecidability: It's a place you don't want to go."

meaning that there exists a tileset capable of simulating all other tilesets. One of the main results obtained by members of the SDA2 community is that, by contrast, at temperature 1 it is not universal, and there exists a pumping lemma showing that such assemblies admit a finite description [41].

Contributors

Samuel Petite & Pascal Vanier.

References

1 Boris Adamczewski and Jason Bell. Function fields in positive characteristic: Expansions and cobham's theorem. *Journal of Algebra*, 319(6):2337–2350, 2008.

2 Boris Adamczewski and Colin Faverjon. Mahler's method in several variables and finite automata. *Annals of Math. to appear, preprint arXiv:2012.08283*, 2020.

3 Robert Ammann, Branko Grunbaum, and G. C. Shephard. Aperiodic tiles. *Discrete and Computational Geometry*, 8(1):1–25, December 1992.

4 Nathalie Aubrun, Sebastián Barbieri, and Emmanuel Jeandel. *About the Domino Problem for Subshifts on Groups*, page 331–389. Springer International Publishing, 2018. doi:10.1007/978-3-319-69152-7_9.

5 Nathalie Aubrun and Mathieu Sablik. Simulation of effective subshifts by two-dimensional subshifts of finite type. *Acta Applicandae Mathematicae*, 126(1):35–63, 2013. doi:10.1007/s10440-013-9808-5.

6 Alexis Ballier and Maya Stein. The domino problem on groups of polynomial growth. *Groups, Geometry, and Dynamics*, 12(1):93–105, March 2018. doi:10.4171/ggd/439.

7 Valérie Berthé, Wolfgang Steiner, and Jörg Thuswaldner. Multidimensional continued fractions and symbolic codings of toral translations. *Journal of the European Mathematical Society*, 25(12):4997–5057, 2022.

8 Bernard Boigelot and Julien Brusten. A generalization of Cobham's theorem to automata over real numbers. *Theoretical Computer Science*, 410(18):1694–1703, 2009.

9 Laurent Boyer, Martin Delacourt, Victor Poupet, Mathieu Sablik, and Guillaume Theyssier. μ-limit sets of cellular automata from a computational complexity perspective. *J. Comput. Syst. Sci.*, 81(8):1623–1647, 2015. doi:10.1016/j.jcss.2015.05.004.

10 Laurent Boyer, Victor Poupet, and Guillaume Theyssier. On the complexity of limit sets of cellular automata associated with probability measures. In Rastislav Královič and Paweł Urzyczyn, editors, *Mathematical Foundations of Computer Science 2006*, pages 190–201, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

11 Antonin Callard and Ville Salo. Distortion element in the automorphism group of a full shift. *Ergodic Theory and Dynamical Systems*, 44(7):1757–1817, 2024.

12 Antonin Callard, Léo Paviet Salomon, and Pascal Vanier. Computability of extender sets in multidimensional subshifts. In Olaf Beyersdorff, Michal Pilipczuk, Elaine Pimentel, and Kim Thang Nguyen, editors, *42nd International Symposium on Theoretical Aspects of Computer Science - STACS 2025*, volume 327 of *LIPIcs*, pages 21:1–21:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICS.STACS.2025.21.

13 Julien Cassaigne. Subword complexity and periodicity in two or more dimensions. In *Developments In Language Theory: Foundations, Applications, and Perspectives*, pages 14–21. World Scientific, 2000.

14 Alan Cobham. Uniform tag sequences. *Mathematical systems theory*, 6(1):164–192, 1972.

15 Karel Culik II. An aperiodic set of 13 Wang tiles. *Discrete Mathematics*, 160(1–3):245 – 251, 1996. doi:10.1016/S0012-365X(96)00118-5.

16 Van Cyr, John Franks, Bryna Kra, and Samuel Petite. Distortion and the automorphism group of a shift. *Journal of modern dynamics*, 13(0):147–161, 2018.

17 Sebastián Donoso, Fabien Durand, Alejandro Maass, and Samuel Petite. On automorphism groups of low complexity subshifts. *Ergodic Theory and Dynamical Systems*, 36(1):64–95, 2016.

18 David Doty. Theory of algorithmic self-assembly. *Communications of the ACM*, 55(12):78088, December 2012. doi:10.1145/2380656.2380675.

19 David Doty, Jack H. Lutz, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Damien Woods. The tile assembly model is intrinsically universal. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 302–310. IEEE, October 2012. doi:10.1109/focs.2012.76.

20 Bruno Durand, Andrei Romashchenko, and Alexander Shen. Effective Closed Subshifts in 1D Can Be Implemented in 2D. In *Fields of Logic and Computation*, number 6300 in Lecture Notes in Computer Science, pages 208–226. Springer, 2010. doi:10.1007/978-3-642-15025-8_12.

21 Fabien Durand. Cobham-Semenov theorem and $\mathbb{N}^d$-subshifts. *Theoretical Computer Science*, 42(1):1–22, 2008.

22 Fabien Durand and Julien Leroy. Decidability of the isomorphism and the factorization between minimal substitution subshifts. *Discrete Analysis*, 2022, August 2022. doi:10.19086/da.36901.

23 Samuel Eilenberg. *Automata, languages, and machines*. Academic press, 1974.

24 Chiara Epifanio, Michel Koskas, and Filippo Mignosi. On a conjecture on bidimensional words. *Theoretical computer science*, 299(1-3):123–150, 2003.

25 Solène J. Esnay, Alonso Núñez, and Ilkka Törmä. Arithmetical complexity of the language of generic limit sets of cellular automata. *Journal of Computer and System Sciences*, 134:20–41, 2023. doi:https://doi.org/10.1016/j.jcss.2023.01.002.

26 Bastián Espinoza. The structure of low complexity subshifts. *arXiv preprint*, (arXiv:2305.03096), 2023.

27 N Pytheas Fogg, Camille Noûs, Mélodie Andrieu, Nicolas Bédaride, Jean-François Bertazzon, Julien Cassaigne, Paul Mercat, and Thierry Monteil. Symbolic coding of linear complexity for generic translations on the torus, using continued fractions. *Journal of modern dynamics*, 20:527–596, 2024.

28 Pierre Guillon. Projective subdynamics and universal shifts. *Discrete Mathematics and Theoretical Computer Science*, DMTCS Proceedings vol. AP,...(Proceedings), January 2011. doi:10.46298/dmtcs.2969.

29 Pierre Guillon, Emmanuel Jeandel, Jarkko Kari, and Pascal Vanier. Undecidable word problem in subshift automorphism groups. In *International Computer Science Symposium in Russia*, pages 180–190. Springer, 2019.

30 Pierre Guillon and Charalampos Zinoviadis. *Densities and Entropies in Cellular Automata*, pages 253–263. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-30870-3_26.

31 Benjamon Hellouin de Menibus and Mathieu Sablik. Characterization of sets of limit measures of a cellular automaton iterated on a random configuration. *Ergodic Theory and Dynamical Systems*, 38(2):601–650, September 2016. doi:10.1017/etds.2016.46.

32 Michael Hochman. On the dynamics and recursive properties of multidimensional symbolic systems. *Inventiones Mathematicae*, 176(1):2009, April 2009.

33 Michael Hochman and Tom Meyerovitch. A characterization of the entropies of multi-dimensional shifts of finite type. *Annals of Mathematics*, 171(3):2011–2038, May 2010. doi:10.4007/annals.2010.171.2011.

34 Emmanuel Jeandel and Michaël Rao. An aperiodic set of 11 Wang tiles. *Advances in Combinatorics*, January 2021. doi:10.19086/aic.18614.

35 Emmanuel Jeandel and Pascal Vanier. Characterizations of periods of multi-dimensional shifts. *Ergodic Theory and Dynamical Systems*, 35:431–460, 4 2015. doi:10.1017/etds.2013.60.

36 Jarkko Kari. A small aperiodic set of Wang tiles. *Discrete Mathematics*, 160:259–264, 1996.

37 Jarkko Kari and Etienne Moutot. Nivat's conjecture and pattern complexity in algebraic subshifts. *Theoretical Computer Science*, 777:379–386, 2019.

38 Kiran S Kedlaya. Finite automata and algebraic extensions of function fields. *Journal de théorie des nombres de Bordeaux*, 18(2):379–420, 2006.

39 Donald E Knuth. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley Professional, 1997.

40 Douglas A. Lind and Brian Marcus. *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, New York, NY, USA, 1995.

41 Pierre-Étienne Meunier and Damien Regnault. Directed non-cooperative tile assembly is decidable. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.DNA.27.6.

42 Christian Michaux and Roger Villemaire. Presburger arithmetic and recognizability of sets of natural numbers by automata: new proofs of Cobham's and Semenov's theorems. *Annals of Pure and Applied Logic*, 77(3):251–277, 1996.

43 Matthew J. Patitz. An introduction to tile-based self-assembly and a survey of recent results. *Natural Computing*, 13(2):195–224, June 2013. doi:10.1007/s11047-013-9379-4.

44 Françoise Point and Véronique Bruyere. On the Cobham-Semenov theorem. *Theory of Computing Systems*, 30(2):197–220, 1997.

45 Anthony Quas and Luca Zamboni. Periodicity and local complexity. *Theoretical Computer Science*, 319(1-3):229–240, 2004.

46 Raphael M. Robinson. Undecidability and Nonperiodicity for Tilings of the Plane. *Inventiones Mathematicae*, 12(3):177–209, 1971. doi:10.1007/BF01418780.

47 Mathieu Sablik Sebastián Barbieri and Ville Salo. Self-simulable groups. *Transactions of the American Mathematical Society*, 2025. doi:10.1090/tran/9434.

Géométrie(s) et image

Twenty Years of GdR IFM, seen from GT Computational Geometry

General Presentation

The field of research of the GT GéoAlgo is *Discrete and Computational Geometry*.

Computational Geometry studies computational questions related to geometric objects. These questions often come from applied fields such as robotics (motion planning), computer graphics (mesh processing), data mining (multidimensional search), or optimization (linear programming). The focus is on provably correct algorithms with worst-case theoretical guarantees. Some algorithms are numerical and imply precision issues, but in most cases one assumes exact computations on real numbers, and the algorithms have a discrete flavor. *Discrete Geometry*, also called *Combinatorial Geometry*, studies combinatorial aspects of geometry. It is naturally very interrelated with Computational Geometry, and should not be confused with *Digital Geometry*, which studies discretization of geometric objects on grid-like spaces (with pixels and voxels). In the 2000s, *Computational Topology* emerged as a subfield of Computational Geometry and has grown significantly, in particular in the French community.

The flagship venue of the community is the *International Symposium on Computational Geometry* (SoCG). Additionally, publications occur in broader conferences in theoretical computer science and algorithms (STOC, FOCS, SODA), in machine learning (AISTATS, ICML, NeurIPS), and in more applied fields (Siggraph, SGP). Members participate in workshops in computer science (Dagstuhl) and mathematics (Oberwolfach). The community has dedicated journals (*Discrete & Computational Geometry* and *Journal of Computational Geometry*), but also publishes in journals with a broader scope (J. ACM, SIAM J. Comput., FOCM, and journals in algorithms, combinatorics, or geometry). The *Handbook of Discrete and Computational Geometry* [7] provides an excellent presentation of the field.

The French community plays a vibrant role on the international scene. It gathers researchers from CNRS (sections 2 and 3, and to a lesser extent section 1), Inria, and many universities, and holds a biyearly meeting, the *Journées de Géométrie Algorithmique*, supported by the GDR. Below, we review the main trends of the last 20 years, with a strong focus on the topics that are studied by the French researchers, and also emphasize connections with other working groups (GTs) of the GDR-IFM.

1 Geometric algorithms in $\mathbb{R}^d$ with theoretical guarantees

Delaunay triangulations, Voronoï diagrams, and polytopes.

These fundamental concepts are studied since the beginnings of Computational Geometry and they remain a vibrant area of research. Over the last twenty years, while new algorithms have been developed in the traditional Euclidean setting, the focus on Delaunay triangulations and Voronoï diagrams has also widely expanded: refined algorithms have been developed for weighted and constrained variants (see e.g. [11] for an application in computational optimal transport), for robustness in practical meshing pipelines, and for extensions to more complex metric spaces—including curved geometries such as hyperbolic surfaces (see below). At the same time, the study of convex polytopes has continued to grow, driven by the need to handle high-dimensional data and models. Because polytopes admit different representations whose

sizes can significantly vary, designing efficient algorithms to approximate a polytope—or even estimate its volume—has become a key challenge. These lines of work illustrate how classical geometric structures continue to evolve in response to modern computational demands.

Geometric algorithms.

Standard algorithmic problems are revisited in a geometric context. Prominently, any family of geometric objects can be studied through its *intersection graph*, and classical packing and covering problems (and variations) are attacked, sometimes in an approximate way, by exploiting properties of the graph itself and by using other properties and concepts (VC-dimension, ε-nets, ...). For this purpose, algorithms on intersection graphs, e.g. for coloring, TSP, and separators, have been developed. As a particularly fruitful example, efficient approximation schemes have been discovered recently using local search, leading to algorithms that are very simple (but highly non-trivial to analyze). For the specific case of the plane, other types of graphs are relevant, such as *visibility graphs*, whose construction and properties have been widely studied until around 2015.

Numerical and symbolic computations.

Most geometric algorithms are designed under the assumption that computations are performed over the real numbers, while practical implementation manipulate fixed-precision floating-point numbers, integers, or algebraic numbers. This discrepancy may yield undesirable behavior (crashing, entering infinite loop, producing incorrect output) and addressing this properly led to the development of the *exact geometric computation* paradigm at the core of the CGAL library. Evaluation of geometric predicates (e.g., deciding whether a point does belong to a line defined by two other points) is not spared, yielding important interactions between the GTs "Calcul Formel" and GéoAlgo. A notable example was identified in 2018: a vulnerability in Skia graphics library (used by Chrome etc.) caused by a non-state-of-the-art implementation of a 2D convex-hull algorithm. Key results in the field include the first algorithm to remove intersections in 3D triangle meshes with fixed-precision output vertices, addressing a long-standing practical need for industrial and academic implementations. Another result is the development of efficient algorithms for drawing algebraic plane curves, approximating both the curves and self-intersections while preserving topology and controlling approximation error. A further achievement is the first practical and robust computation of 3D quadric intersections using algorithms that reduce the degree of algebraic numbers involved. A recent trend focuses on hyberbolic geometry where cascaded constructions of points make it critical to design algorithms with predicates and algebraic numbers of low degree.

Probabilistic analysis of geometric algorithms.

Over the past two decades, the probabilistic analysis of geometric algorithms has gained prominence as a way to temper the often pessimistic nature of worst-case bounds—such as the classical exponential upper bounds for the simplex method. This shift has motivated the introduction of more realistic input models and average-case analyses. A major conceptual breakthrough in the early 2000s was the development of smoothed analysis, which blends worst-case and average-case perspectives; its application to the simplex algorithm by Spielman and Teng had a profound impact on the field. In France, this line of research fostered closer interactions between communities (GTs) in analysis of algorithms and stochastic geometry, notably through several national collaborative projects in the 2010s. Over the last fifteen

years, French researchers have made significant contributions in low-dimensional settings, including advances in smoothed analysis, probabilistic approximation of polytopes, and the study of random walks in geometric contexts. These works illustrate the increasingly rich interplay between geometry, randomness, and algorithmic efficiency.

2 Meshes, triangulations, geometric and topological inference

Meshes/triangulations and in particular *Delaunay complexes* have been a cornerstone of computational geometry from its inception. The interest in meshing is motivated by e.g. applications in numerical PDEs (finite element methods) and graphics. Up to the year 2000 the study of meshing was limited to low dimensions and mostly Euclidean space. Since then, the research has gone into new directions with an emphasis on high dimensions and curved spaces: surfaces, hyperbolic spaces, and more generally Riemannian manifolds. The manifolds we want to mesh occur in various forms: submanifolds that are known from samples (this combines meshing with shape reconstruction discussed below), submanifolds that are given by (not necessarily algebraic) equations, or abstract Riemannian manifolds (given by local charts ignoring the ambient space, which avoids the curse of dimensionality). For instance, one important line of work has been to adapt the classical algorithmic toolbox for Delaunay triangulation so that it works beyond Euclidean settings. This has been initiated in toroidal and hyperbolic geometries, both of which are relevant for applications, e.g., in materials sciences, and many questions still abound in this active area of research. The works in the community have many facets: algorithm development, theoretical guarantees of correctness, complexity theory, implementation and practical running time.

Shape reconstruction is a classical problem in computational geometry, closely related to meshing. Given a finite set of data points that sample an unknown shape, the problem consists in building an approximation (ideally a triangulation) of the shape based solely on the data points. In the early 2000s, the first methods with theoretical guarantees were proposed for surfaces in $\mathbb{R}^3$. Around 2005, the focus started to shift toward *manifolds* in high dimensional ambient spaces, motivated by applications to *machine learning*. A milestone was obtained with the work of Niyogi, Smale and Weinberger in 2008, which provided sampling conditions under which an offset of the data points recovers the homotopy type of the underlying manifold [16]. Since then, significant effort of the French community has gone into generalizing and improving the bounds of that seminal paper. In a high-dimensional ambient space, classical algorithms based on the computation of the Delaunay complex of the data points suffer from the curse of dimensionality. Alternative methods have been proposed that scale well with the ambient dimension and come with guarantees [1]. This led to developing new data structures for encoding simplicial complexes: the simplex tree and the blocker-skeleton data structure, which are both at the core of the Gudhi library (cf. §3 and 6). In a recent breakthrough, manifold reconstruction has been expressed as finding minimal cycles (either for an ℓ_1-norm or for a lexicographic order, yielding a topologically correct and efficient algorithm for surfaces in $\mathbb{R}^3$. The geometric assumptions that underpins the correctness of many algorithms mentioned above are most often formulated in terms of the reach [6]: the largest distance under which a point is guaranteed to admit a unique projection on the manifold. A new metric characterization of the reach was provided in [2] which proved useful for (statistical) learning. The reach is intricately linked with the medial axis, a skeleton which lies in the middle of a shape and captures the homotopy type and plays an important role in graphics, though being difficult to compute. Significant effort has gone into computing and simplifying the medial axis.

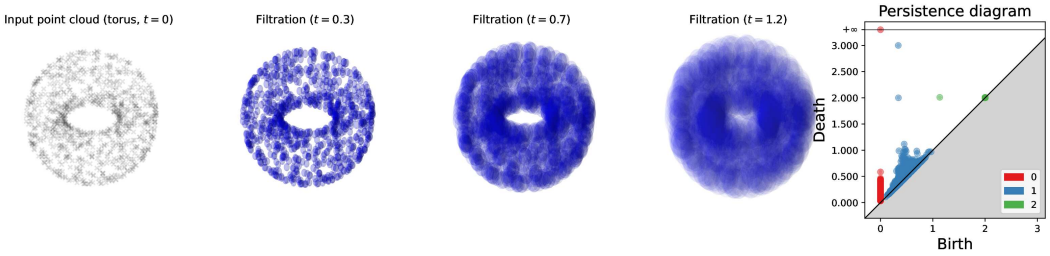


Figure 1 Čech filtration and resulting persistence diagram built on top of a point cloud (1000 points sampled uniformly on the surface of a torus). The right plot depicts the resulting PD: red points correspond to homology of dimension 0 (connected components), blue points to dimension 1 (loops) and green points to dimension 2 (cavities). The three points far away from the diagonal (the two blue points and the green one) testify for the two loops generating the torus and the corresponding cavity, points closer to the diagonal (birth $\simeq$ death) correspond to features that are merged quickly after appearing in the filtration (i.e. persisted less longer), often considered as topological noise.

3 Topological Data Analysis

Topological Data Analysis is a relatively recent machinery in computational topology that aims at computing *quantitative* topological descriptors of structured objects (graphs, points sampled on a manifold, time series, etc.). The field emerged in the early 2000s, and has since found application domains such as computer graphics, material science and computational biology to name a few. Rooted in algebraic topology, and more precisely in the theory of *persistent homology*, the TDA pipeline can be summarized in the following way: given a topological space $\mathcal{X}$ and a map $f : \mathcal{X} \to \mathbb{R}$, consider the *filtration* $K_t = \{x \in \mathcal{X},\ f(x) \leq t\}$. For instance, one may take $\mathcal{X} = \mathbb{R}^d$ and $f(x) = \min_{1 \leq i \leq n} \|x - x_i\|$ where $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ is a point cloud, yielding the so-called Čech filtration. One then records the pairs of values (t_b, t_d) at which a given topological feature (such as connected components, loops, cavities, etc.) appears in K_{t_b} and disappears in K_{t_d}.[1] The resulting (multi-)set of pairs is called the *persistence diagram* (PD) $\mathrm{PD}(f)$ of f. See Figure 1. A fundamental property of PDs is their *stability*: similar filtrations yield similar PDs [3], where similarity of PDs is measured using optimal transport-type distances.

By the mid-2010s, PDs began to be used in machine learning as static features derived from data. Because the space of PDs has a non-linear structure, their integration into downstream models is non-trivial, and typically relies on vectorizations techniques, such as kernel methods, or data-driven representations based on specific deep neural network architectures. More recently, to further strengthen the connections between TDA and machine learning, the concept of persistence-based *topological optimization* has emerged. A rigorous framework for differentiating composite maps of the form $f \mapsto \mathrm{PD}(f) \mapsto \ell(\mathrm{PD}(f)) \in \mathbb{R}$ for some loss functions ℓ as been introduced in [15]. A series of works has since been devoted to studying, implementing, and refining gradient-based optimization schemes—an area that continues to be particularly active within the French TDA community.

In addition, the field has seen significant advances in the context of multiparameter

[1] In practice, K_t is represented by a combinatorial object called a simplicial complex (which should have, or closely approximate, the same homotopy type as the corresponding sublevel set), and PDs are computed using matrix reduction algorithms.

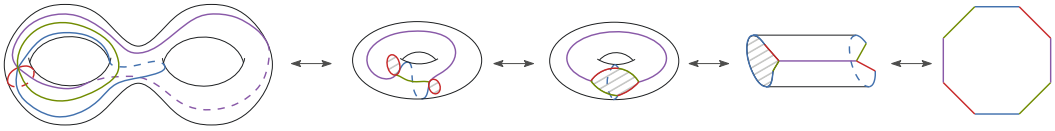


Figure 2 Cutting a double-torus (an orientable surface of genus two) into a disk using a *polygonal schema*.

persistence, that is, when the map f is valued in $\mathbb{R}^d$ with $d \geq 2$. In contrast to the case $d = 1$, the lack of total order on $\mathbb{R}^d$ prevents one from defining canonical analogues of PDs. The design of topological features based on multiparameter persistence has been studied deeply in the field, with the French community being a driving force, with the design, theoretical analysis, and implementation of two topological features for multiparameter persistence: the *signed barcode* and *candidate decompositions*. Multiparameter topological optimization has then quickly followed as a new application field in TDA, fueled by these new features.

These efforts materialize by the development of the open-source library Gudhi (`Python` / `C++`), which contains most of TDA descriptors with fast implementations in `C++`, as well as easy-to-use pipelines in `Python` for data science. As such, it has become the leading library for TDA, with thousands of downloads per week. With the same ambition, the development of multiparameter persistence has been made easier thanks to the multipers library, whose core implementations rely on `Gudhi`.

4 Low-dimensional computational topology

While algorithmic aspects are implicit in topology since its inception, a systematic development of algorithms for topological problems emerged only in the late 1990s [5]. The French community is particularly active in low-dimensional computational topology, where the ambient space is a surface or a 3-manifold (a space locally homeomorphic to $\mathbb{R}^3$), with some non-trivial topology (e.g., "surfaces with handles" such as the torus). Like topological data analysis, this field relies on algebraic topology, but it has a very distinct flavor: the applications are very different, and low dimension allows us to work with homotopy, which is finer than homology. The relevant tools from computer science include graph algorithms, complexity theory, and fixed-parameter tractability. In this area, the GT-GéoAlgo has connections with several other GTs, notably CoA and Graphs, and is also related to other fields, such as graph drawing. Relevant applications are found in computer graphics, geometry processing, and molecular biology.

The **2-dimensional case**, namely computational topology for graphs on surfaces [4], started to flourish 20 years ago [14]. The most basic questions are algorithmic formulations of well-studied mathematical problems: Are two input curves or graphs on a surface equivalent, for various topological notions of equivalence (homotopy, homology, isotopy)? Given a self-intersecting graph or multicurve, what is the minimum number of crossings that must remain after homotopically deforming it on a surface? Around the same time, algorithms for computing shortest curves and graphs with given topological properties blossomed, e.g., shortest non-contractible or non-separating closed curves; this requires an appropriate discretization of a metric surface.

While the above line of research is still active, several directions have emerged more recently. Applications to graph problems, e.g., generalized cut and flow problems, have been found: combining topological subroutines with tools from algorithmic graph theory (treewidth, minors, etc.), faster algorithms are obtained assuming the input graph is planar or,

more generally, is embedded on a fixed surface—often these algorithms are fixed-parameter tractable in the genus of the surface. Generalizations of planar structures (e.g., Schnyder woods) to surfaces are investigated. Compact data structures for surface triangulations are discovered. Efficient algorithms are described to compute well-studied mathematical quantities related to surfaces, such as their spectra. Algorithms and data structures are being developed to handle hyperbolic surfaces and compute Delaunay triangulations and Voronoi diagrams on such spaces.

In the **3-dimensional case**, new topological features and knotting phenomena start to appear. Their investigation from a computational point of view was initiated about twenty years ago in a seminal work [10] and our understanding has been steadily improving since then, see for example [13]. In terms of exact recognition algorithms, this area is populated with peculiar questions at all ends of the complexity spectrum. For instance, the problem of deciding if a closed curve is knotted is now known to be in NP ∩ co-NP but no polynomial-time algorithm is known, while the best-known algorithm to check whether two knots are equivalent is elementary recursive but the problem is not even known to be NP-hard. Topological questions in 2D on homotopy, homology, etc. also adapt naturally to 3-manifolds, leading to new problems which are often harder to solve algorithmically. Over the past decades, the computational geometry community in France and abroad has largely contributed to mapping the 3-dimensional complexity landscape, in particular through the design of hardness proofs.Another computational aspect of this field is that topological insights are often garnered via the development of highly-optimized specific software tailored for the analysis of knots and 3-manifolds, which then fuels experimental works.

Given the dearth of knowledge on exact recognition algorithms, it is natural to investigate *invariants* which provide a coarser way to differentiate topological objects. Topological invariants of algebraic nature, such as quantum invariants of knots and 3-manifolds whose introduction dates back to the 80s, have been recently studied from the computational point of view. Their computational hardness has been established thanks to connections with quantum computing [12], and efficient algorithms have been designed through the extension of techniques from parameterized complexity. This phenomenon has, in turn, fueled a new interest for the study of the structural properties of combinatorial representations of knots, 3-manifolds, and other low dimensional objects, notably with the extension of techniques from structural graph theory to the context of low dimensional topology, or the study of the complexity of hard problems on inputs of restricted topology.

5 Discrete Geometry, geometric and topological combinatorics

Discrete geometry studies the structure of discrete objects in a geometric space, from finite point sets in the plane to arrangements of n-dimensional convex bodies. Foundational problems in this area deal with packing (Kepler's conjecture), polytopes and triangulations (Hilbert's third problem on decomposing polyhedra, Hirsch's conjecture on the diameter of polytopes, the g-conjecture on the face numbers of triangulated spheres), point configurations (the Erdös-Szekeres problem, distinct and repeated distances, integer distance sets), etc. There are strong ties between discrete geometry and computational geometry around questions of complexity, simulation, discretization, etc. and the two communities merged to some extent in the 1980's.

Within the French community, several lines of research in discrete geometry have been particularly active over the last two decades. Disk and ball packings in dimensions two and three continue to attract attention: although the monumental proof of the Kepler

conjecture [9] settles the case of a single radius, the landscape becomes far richer when balls with multiple radii are allowed, raising challenging questions of optimal structure and computational optimization. Progress has also been made on the study of oriented matroids, a combinatorial and algebraic structure arising from point configurations and hyperplane arrangements, with connections to the complexity class ETR (Existential Theory of the Reals) [17], to probabilistic properties of geometric structures, and to refined combinatorial classifications. Hitting sets problems (finding a small number of points/lines/hyperplanes that intersect all elements of a collection of sets) have likewise seen strong advances, especially regarding the existence and computation of hitting sets and the influence that the geometry (convexity, half-spaces, etc.) has on the size of hitting sets. Equipartition problems (seeking for a single hyperplane cutting a collection of objects in halves, e.g. ham-sandwich theorem)—often involving partitions by systems of hyperplanes—have developed new links to classical conjectures such as Mahler's. Beyond these themes, new combinatorial objects inspired by geometric constructions have been introduced, including generalized associahedra and related polyhedral families, and have deepened the study of reconfiguration graphs such as flip graphs of triangulations and decomposition complexes, often in connection with low-dimensional topology (Section 4).

In the last 20 years, the development of algebraic and topological methods have been two driving forces in discrete geometry. A spectacular example was the *polynomial method* [8] which builds on ruled surface theory in real algebraic geometry and led to solutions to the *joint problem* and Erdös' *distinct distances problem* and the algorithmic technique of *polynomial partitioning*. Another landmark is Adiprasito's proof of the *g-conjecture*, which draws on the connections between face rings and toric varieties and yields extensions to simplicial complexes of some classical results on graphs (*e.g.* a linear bound on the number of facets in terms of the number of ridges, a generalization of Heawood's inequality on which complexes embed in manifolds, etc.). Other examples include the sharpening of several fundamental results in extremal combinatorics for *semi-algebraic (hyper)graphs*, the developments around the *configuration space/test map* scheme (*e.g.* the topological Tverberg theorem), etc.

6 Software development in Computational Geometry

The French community is involved in the development of several open-source software related to computational geometry: gudhi for Topological Data Analysis, contributions to `Maple` (root finding algorithms), Isotop (drawing of planar curves with correct topology), to name a few.

A major contribution of the French community to discrete and computational Geometry over the past decades lies in the development and long-term maintenance of the Computational Geometry Algorithms Library (CGAL). Initiated in 1996, CGAL has grown into the most influential and widely used software library in computational geometry. It provides robust, efficient, and well-documented implementations of geometric algorithms for both academic and industrial use. The project now comprises more than 700,000 lines of code, is downloaded about 10,000 times per year, and serves over 200 commercial users worldwide. With around 20 active developers and supervised by an editorial board in charge of project management and code review, CGAL continues to be a vibrant and sustainable initiative, supported by academic partners who have made long-term commitments to its maintenance and evolution. The library is distributed under both open-source and commercial licenses, ensuring accessibility for research as well as reliability for industrial applications. The private company GeometryFactory, an integral partner of the project (including 3 members of the

editorial board), provides professional support and assists industrial users in integrating CGAL components into their software. In recognition of its enduring impact, CGAL received the "Test of Time Award" at the 2023 Symposium on Computational Geometry (SoCG), underscoring its role as a cornerstone in the field's computational infrastructure.

Contributors.

Dominique Attali, Mathieu Carrière, Xavier Goaoc, Théo Lacombe, Clément Maria, Arnaud de Mesmay, Guillaume Moroz, Mathijs Wintraecken.

References

1 Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*. Cambridge University Press, 2018.

2 Jean-Daniel Boissonnat, André Lieutier, and Mathijs Wintraecken. The reach, metric distortion, geodesic convexity and the variation of tangent spaces. *Journal of Applied and Computational Topology*, 3(1):29–58, Jun 2019.

3 Frédéric Chazal, Vin De Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*, volume 10. Springer, 2016.

4 Éric Colin de Verdière. Computational topology of graphs on surfaces. In Jacob E. Goodman, Joseph O'Rourke, and Csaba Toth, editors, *Handbook of Discrete and Computational Geometry*, chapter 23, pages 605–636. CRC Press LLC, third edition, 2018.

5 Tamal K. Dey, Herbert Edelsbrunner, and Sumanta Guha. Computational topology. In Bernard Chazelle, Jacob E. Goodman, and Richard Pollack, editors, *Advances in Discrete and Computational Geometry – Proc. 1996 AMS-IMS-SIAM Joint Summer Research Conf. Discrete and Computational Geometry: Ten Years Later*, number 223 in Contemporary Mathematics, pages 109–143. AMS, 1999.

6 Herbert Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.

7 Jacob E. Goodman, Joseph O'Rourke, and Csaba Tóth, editors. *Handbook of discrete and computational geometry*. CRC Press LLC, Boca Raton, FL, third edition, 2018.

8 Larry Guth. *Polynomial methods in combinatorics*, volume 64. American Mathematical Soc., 2016.

9 Thomas C. Hales. A proof of the kepler conjecture. *Annals of mathematics*, pages 1065–1185, 2005.

10 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *Journal of the ACM (JACM)*, 46(2):185–211, 1999.

11 Jun Kitagawa, Quentin Mérigot, and Boris Thibert. Convergence of a newton algorithm for semi-discrete optimal transport. *Journal of the European Mathematical Society*, 21(9):2603–2651, 2019.

12 Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory Comput.*, 11(1):183–219, 2015.

13 Marc Lackenby. Algorithms in 3-manifold theory. *Surveys in Differential Geometry*, 25, 2022.

14 Francis Lazarus, Michel Pocchiola, Gert Vegter, and Anne Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proceedings of the 17th Annual Symposium on Computational Geometry (SOCG)*, pages 80–89. ACM, 2001.

15 Jacob Leygonie, Steve Oudot, and Ulrike Tillmann. A framework for differential calculus on persistence barcodes. *Foundations of Computational Mathematics*, 22(4):1069–1131, 2022.

16 Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

17 Marcus Schaefer, Jean Cardinal, and Tillmann Miltzow. The existential theory of the reals as a complexity class: A compendium. *arXiv preprint arXiv:2407.18006*, 2024.

Twenty Years of GdR IFM, seen from GT Discrete Geometry and Mathematical Morphology

The overview of the working group on *Géométrie Discrète et Morphologogie Mathématique* (GDMM) is structured around five major scientific axes developed continuously from 2005 to 2025: discrete geometric objects, interactions in combinatorial optimization, topology for imaging, watershed transforms for segmentation, and morphological filtering. These axes are complemented by a transversal axis devoted to software development and open, reproducible science.

1 Discrete geometry: multigrid convergence and discrete calculus

Discrete geometry, also known as digital geometry, is primarily concerned with defining a consistent geometry on subsets of the discrete grid $\mathbb{Z}^d$, or more generally on lattices or tilings. It was originally studied within the framework of the geometry of numbers in mathematics, with strong links to arithmetic and convex geometry. Since the 1970s, the scope of discrete geometry has been profoundly influenced and transformed by the emergence of digital images—first in 2D, then rapidly in 3D with computed tomography and nuclear magnetic resonance imaging. The subsets under study have become discretizations of real physical shapes, or even a means of efficiently encoding the geometry of virtual shapes. Questions that were initially purely mathematical have thus been reformulated in more applied terms, with on the one hand the constraint of developing a geometry capable of addressing concrete problems (such as measuring length or area, estimating tangents, normals, or curvatures, identifying smooth or convex regions, corners, or edge lines), and on the other hand algorithmic constraints, such as decidability or efficiency in time and memory.

Whereas research conducted in the 1980s–2000s was essentially focused on the study of linear/affine geometry—particularly the definition and recognition of discrete objects (lines, planes, circles) using arithmetic or combinatorial methods—the last twenty years have been marked by the study of the asymptotic consistency of discrete geometry results with those of Euclidean geometry. The property of **multigrid convergence** of a geometric estimator precisely guarantees that a finer discretization of a Euclidean object leads to a more accurate geometric estimation based solely on the discretized object. Gauss had already shown that counting the number of integer points inside a convex volume provides a good approximation of its volume. While multigrid convergence of estimators for perimeter, length, or moments was already known in the early 2000s (see, e.g., [28]), it was only in the following decade that multigrid convergence of local geometric estimators, such as tangents or normals, was established, using a wide variety of approaches in 2D: maximal segments [19, 35], binomial convolutions [21], and polynomial approximations [50]. Approaches for 3D surfaces have also been proposed, such as binomial convolutions [22] or planarity probing [31], though with limited guarantees. This was followed by the development of "integral" or "geometric measure theory" approaches, which estimate differential geometric quantities via carefully chosen integrals, leading to more robust methods [13, 17, 42]. The current theoretical state of the art is achieved by normals computed using integral invariants [30], which also enable the estimation of all curvatures [13]. These convergent normals induce an area estimator on discretized surfaces in any dimension, and more generally the convergence of any surface integral [34]. Since 2021, the theory of corrected normal currents has been the best approach for curvature estimation [32]. Initially developed for digital surfaces (see Figure 1), its theoretical framework and stability also make it the state of the art for polygonal surfaces

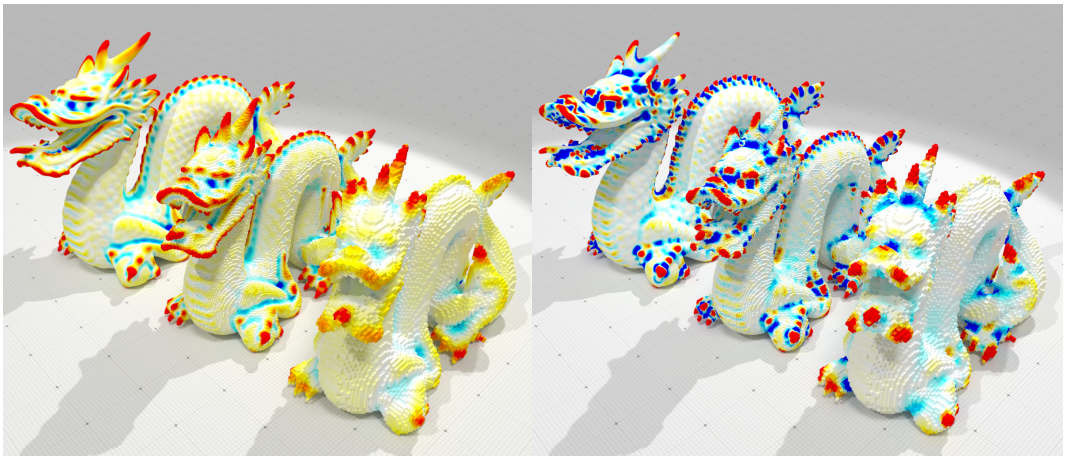


Figure 1 Convergent estimation of mean (left) and Gaussian (right) curvatures on digital surfaces [32], implemented in the DGtal library.

[33] and point clouds [29].

The development of a convergent discrete differential geometry has fostered the emergence of **discrete exterior calculus** on digital surfaces over the past decade. Discrete exterior calculus was initially approached through methods originating from theoretical physics [41] or from graph theory and electrical networks [26]. To obtain convergence properties, methods developed for triangulated or polygonal surfaces [18, 20] were adapted to the case of digital surfaces. The incorporation of convergent estimators of normal vectors into this calculus induces the convergence of classical differential operators, such as the gradient, the Laplacian, or the divergence [9, 53]. This framework then enables the geometric processing of digital data via these operators (e.g., spectral decomposition of the Laplacian, piecewise-smooth regularization of Mumford–Shah type, geodesics, UV parameterization).

2 Combinatorial optimization and discrete geometry

The founders of digital geometry envisioned a new computational paradigm suited to numerical data modeled as sets of pixels or voxels. The initial idea of providing a fully discrete computational model for data that are themselves perfectly digital has given way to a scientific environment in which discrete and continuous tools have come together and combined to address increasingly complex problems.

Hermann Minkowski's geometry of numbers, the study of polytopes with integer vertices, combinatorial optimization, geometric optimization, algorithmic geometry, combinatorial geometry, and discrete geometry have not diverged but, on the contrary, have converged toward a set of questions that do not constitute isolated compartments but rather a continuum of problems that are both combinatorial and geometric.

Let us take the example of Multi-Agent Path Finding (MAPF), which involves navigating a fleet of robots moving on a grid, as seen in competitions such as the League of Robot Runners or CG:SHOP 2021. Digital geometry is fundamental here: the robots operate in a discretized space, their trajectories are combinatorial, yet the search for a global optimum (minimizing total time, avoiding collisions, respecting spatial constraints) fully falls under geometric and algorithmic optimization. Such problems perfectly illustrate the convergence of the two fields: a discrete numerical geometric space tackled using all possible continuous or

discrete optimization methods—SAT solvers, MIP, graph- and geometry-based metaheuristics, or even Deep Learning.

In the age of combinatorial games on smartphones, many games—from the old tic-tac-toe to Tetris, Numberlink, Block Blast, and Flow Free—take place on a 2D grid. The combinatorial richness of the grid has fascinated mathematicians since ancient times: from Euler's Latin squares to John Conway's Game of Life, it has become a favorite testing ground for algorithmic complexity. Geometric problems such as tiling and discrete tomography are examples of this. The last twenty years have seen an explosion of results on the decidability, complexity, and approximation of geometric problems on grids, where the boundaries between combinatorics, optimization, and geometry are blurred.

Digital geometry has also greatly benefited from these interactions. For example, determining whether a subset S of $\mathbb{Z}^2$ is digitally convex can be decided in linear time. The problem of covering S, assumed to be hole-free, with a minimal number of balls (not containing any integer point outside S) can be solved in polynomial time, whereas these problems are typically hard in classical computational geometry. The convex skull problem can be solved digitally in $O(k^3)$ time for a set of k integer points, whereas for a polygon with n vertices, the complexity is $O(n^7)$. This demonstrates that classical problems in computational geometry change character when considered in the framework of digital geometry, leading to original theoretical developments. Nevertheless, there are strong similarities, one of which is the central role played by convexity, which is itself the subject of new developments. The computational geometry community has also taken up combinatorial problems from digital geometry, such as polyomino folding or polycube unfolding [51]. The french community is very active on these topics, as illustrated by the 1st place for the team "*Les Shadocks*" in the CG:SHOP (Computational Geometry: Solving Hard Optimization Problems) challenge in 2021, 2022, 2024 and 2026.

Today, the boundary between digital geometry, algorithmics, and optimization has never been more porous. Deep learning introduces yet another dimension: geometric machine learning models learn to reconstruct, simplify, or parameterize surfaces directly from numerical data. This hybridization of the discrete, continuous and statistical spaces opens a new era in which numerical geometry becomes simultaneously a tool for analysis, optimization, representation, and modeling.

3 Topology for imaging

Following the example of discrete geometry, whose main goal is to develop concepts in Cartesian spaces ($\mathbb{Z}^d$) that are consistent with Euclidean geometry ($\mathbb{R}^d$), in the 1960s work was also initiated to develop topological frameworks adapted to digital imaging and dedicated to coherently modeling the topology of the continuous scenes underlying images. The historical work carried out during the first decades, in conjunction with methodological and technological advances in 2D and then 3D digital imaging, initially focused on the development of (1) discrete topological frameworks based on $\mathbb{Z}^d$ (digital topology, cubic complexes); (2) discrete versions of classical topological invariants; (3) operational tools for the development of topology-based transformations (e.g., skeletonization), under the impetus of pioneers such as Rosenfeld, Pfaltz, Khalimsky, Kovalevsky… Until the turn of the 2000s, this work was mainly dedicated to binary images.

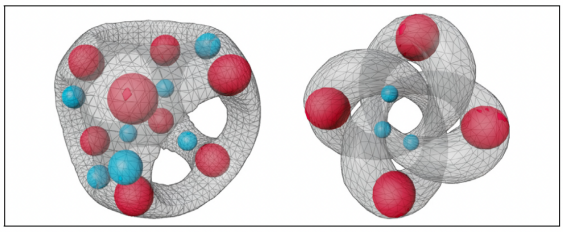
During the period 2005-2025, the French community, within the GT GDMM, contributed to major advances in this field, with the extension of concepts in terms of image dimensions (3D, 4D) and valuation (grayscale images, labeled images), but also through the generalization

of notions and the creation of new links with continuous topology and connective frameworks developed in mathematical morphology.

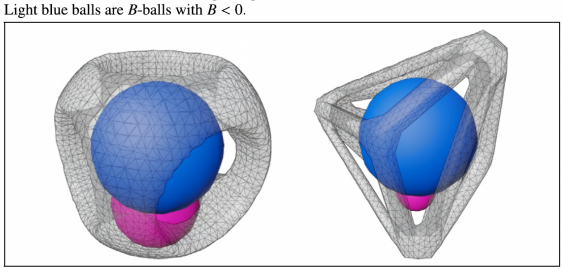
The main contributions of the first decade (2005-2015) concerned work on homotopic transformations, led by researchers such as Bertrand, Couprie... In particular, the historical notion of "simple point", widely studied in the context of binary 3D digital topology during the 1990s, was extended in multiple ways: in dimension (4D) [15], in valuation (grayscale, labels [40]), in cardinality (simple sets). The extension of the notion of simple points from digital topology to complex models has also led to the proposal of more general concepts of homotopic transformations in complexes, notably with the proposal of critical kernels [5], which provide guarantees of topological and geometric optimality (directly related to median axes) for objects obtained by decreasing homotopic transformations, particularly in parallel paradigms. Another notable contribution during the same period concerns the formal proof of the compatibility of discrete and continuous topology frameworks [39].

During the second decade (2015-2025), the spectrum of contributions became much more diverse. Without claiming to be exhaustive, the following advances are particularly noteworthy. (a) The conditions for preserving topological properties during the continuous-discrete transition, initially explored in the mid-1980s, were extended [36] to also consider alternative paradigms (e.g., discrete-discrete in the context of "re-digitization"), leading in particular to homotopic preservation guarantees for geometric transformations [44]. (b) The notion of a "well-composed set" proposed in the 1990s has been extensively developed and consolidated in different frameworks and dimensions, making it possible, on the one hand, to provide dimensional guarantees on the objects manipulated [7], but also to develop sufficient conditions for symmetrically managing the open/closed paradigm, leading in particular to the proposal of truly self-dual hierarchical structures for image modeling [6]. (c) An axiomatic framework based on completions, which allows several key concepts (simple homotopy, homology, etc.) to be formalized inductively and related to each other, has been proposed in the context of simplicial complexes (obviously related to cubic complexes) [2].

Although continuous efforts were made during this period to develop effective concepts and algorithms for calculating homology groups and related descriptors [25, 46] (e.g., Betti numbers), the recent democratization of topological data analysis (TDA), as well as the advent of deep learning (DL) and the relevance of integrating topological priors into their model, have (re)motivated the development of sustained research on topological invariants. The most recent work, initiated at the end of the period and offering prospects for the coming years, focuses in particular on the notion of persistence accessible in hierarchical structures (morphological trees) modeling digital images, which echoes the persistence implemented in TDA. In this context, the construction of topological loss functions based on trees [48], the integration of binary invariants into these trees resulting in grayscale invariants [45], or the links between the notion of dynamics and homological persistence [8], are all avenues of research that could consolidate the bridges between discrete topology, mathematical morphology, and TDA in the coming years. Finally, we can cite very recent work that links discrete Morse theory and the watershed framework [4] and, more generally, the notions of "Morse sequences" and "Morse frames" [3], which revisit discrete Morse theory and pave the way for efficient computation of persistence in homology and cohomology in the framework of simplicial complexes.



(a) hole-balls of 0-holes (including non-present holes) for *hollow-cubic-ball* and *knot*. Light blue balls are B-balls with $B < 0$.



(b) hole-balls of non-present 2-holes for *hollow-cubic-ball* and *triakis-tetrahedron*. Magenta balls are T-balls with $T < 0$.

Figure 2 Calculation of cavity measurements in volumetric objects defined by surface meshes. Illustration based on [23].

4 Watershed and image segmentation

The concept of the watershed, studied since the 19th century by scientists such as Camille Jordan and James Clerk Maxwell, describes how a landscape can be divided into catchment basins, each associated with a minimum toward which water flows. The watershed line separates these basins and corresponds to locations where water can flow toward different minima. In the late 1970s, this idea was adapted to image processing by interpreting an image as a topographic surface where pixel intensity represents altitude, allowing images to be segmented into coherent regions. Efficient algorithms were developed in the 1990s and incorporated into many software tools, including medical imaging and photo editing. However, in 2005 it was shown that previous definitions either rely on unrealistic continuity assumptions for digital images or fail to satisfy certain desirable properties.

To overcome earlier limitations, the concept of the watershed was redefined using edge-weighted graphs [16]. This new definition, based on the intuitive drop of water principle, satisfies important properties that previous discrete definitions lacked. In particular, it establishes a duality between catchment basins defined by steepest-descent paths and their separation defined by flow divergence. The key result shows that watersheds correspond to cuts of minimum spanning forests rooted at the graph's minima, linking the concept to a classical problem in combinatorial optimization and creating connections with algorithms, topology, morphological filtering, and hierarchical analysis.

For example, a new generic combinatorial optimization problem on graphs parameterized by two exponent values p and q, which allows several known methods to be unified, has been defined and studied. Depending on the values of p and q, the solution is a minimal cut, a random walker cut, a Voronoi diagram, or finally, when p tends to infinity, a new combinatorial object called a power watershed [14]. The power watershed is a minimum spanning forest cut (thus, a watershed cut) for which certain forests of the same weight are discriminated by a secondary optimality criterion. This result links several methods that are useful in image analysis and combines their advantages.

In practice, the images and data contain different levels of detail whose importance varies depending on the application. It is therefore appropriate to consider a hierarchical representation in which watershed basins are progressively merged as one moves up the scale levels. To this end, a filtering process, known as connected closing, is considered. It consists of "filling in" certain minima according to a relevance criterion and the desired degree of severity. By varying the severity, a series of filtered images and their corresponding watershed lines are generated. Other types of hierarchies are studied within the community: component trees, trees of shapes, binary partition trees, quasi-flat zones and alpha trees, as well as links with ultrametric distances and hierarchical clustering theory, such as single-linkage clustering or the HDBSCAN method. The international state of the art in image segmentation generally includes a hierarchical segmentation step, most often obtained using the watershed transform. The most recent work in our community consists of integrating segmentation algorithms into deep learning models in order to predict segmentations [12].

This methodological development was accompanied by algorithmic development based on schemes such as Kruskal, Prim, or Boruvka algorithms, leading to improvements in the complexity and efficiency in the computation of these various structures. Several scenarios were studied, leading to interactive, distributed, parallel, out-of-core algorithms capable of handling large data (gigapixels) and integrating the developed operators on GPUs, making them easier to integrate into deep learning chains.

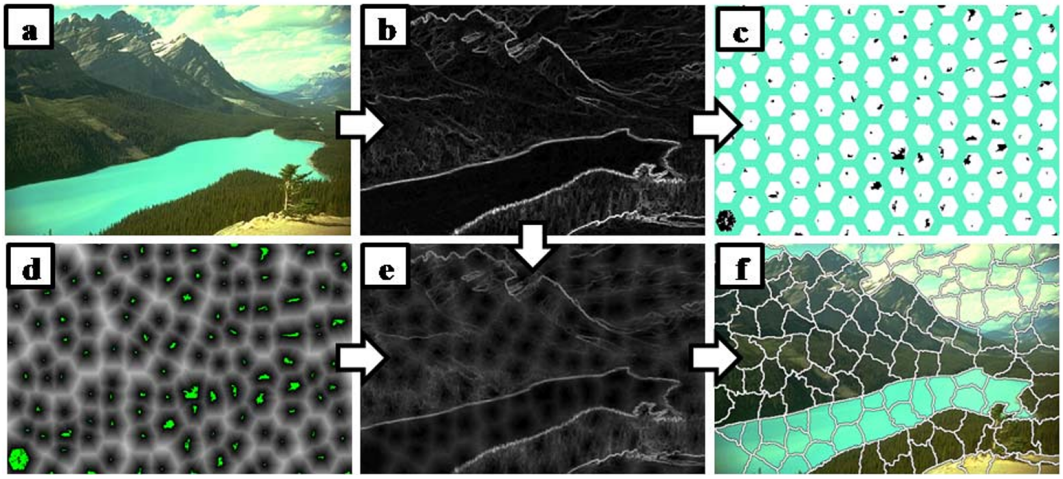


Figure 3 Illustration of the waterpixel method, which segments the image (see f) into regular regions distributed across the image, with contours that adhere to those of the image. To do this, a watershed is calculated from the image (e) obtained by combining the contours (b) of the original image (a) with a Voronoi diagram (d) of the minima positioned regularly throughout the image (c). Illustration based on [38].

5 Morphological filtering

Mathematical morphology is known worldwide as a mathematically well-founded technique and theory, inspired by image processing problems, which constitute its main field of application. The theory of morphological filtering is based on the algebraic structure of lattices. The elementary operators, dilation and erosion, are defined as applications of one lattice to another (or to itself) that commute with the supremum and infimum, respectively, forming pairs called adjunctions. Many more complex filters are constructed by composing

these basic operators. Unlike classical image processing operators, which are generally linear and invertible, mathematical morphology operators are nonlinear, non-invertible, and have specific properties such as idempotence and growth. This characteristic results in a controlled loss of information that leads to powerful and targeted filtering. In particular, these operators can be used to remove structures that do not meet certain geometric criteria, such as width, volume, or contrast. Over the past twenty years, the French community has contributed significantly to the evolution of this field by extending conventional operators to more complex frameworks that are better suited to modern applications.

In practice, mathematical morphology operators are often parameterized by a structuring element, i.e., a shape used to guide the effect of the operator in its definition space, which is generally a discrete grid with integer coordinates. On the one hand, spatially variant and adaptive morphological operators have been developed. These operators allow the shape of the structuring element to be varied according to its position in space. This makes it possible to adapt the filtering to the content of an image or image area [37]. This may involve varying the direction of a linear structuring element to filter fine structures such as a vascular network, or reducing the regularization effect in high-contrast areas of the image [43]. On the other hand, the study of morphological operators in spaces of graphs, hypergraphs, simplicial complexes, point clouds, fuzzy and bipolar sets, or even logical relations is a very active field. The goal is, for example, to perform fine-grained operations on the topology of objects or to formalize spatial relations such as “is to the right of”, “is located between”, “is intertwined with”, or even to detect certain patterns in pieces of music.

Another category of operators, called connected operators, acts solely by removing or preserving connected components of the object or its complement. An important property of these operators is that they can only remove contours: creating new contours or moving existing contours is not possible. This leads to filtering that does not blur, unlike standard operators. The selection of components to be removed is based on geometric or photometric attributes. To effectively represent these components and evaluate their attributes, images are represented by hierarchical structures that encode the inclusion of related components. Component trees (directed [49] or undirected [10]) or level set trees [24] are among the most studied in recent decades. Our community has proposed increasingly efficient algorithms for calculating and processing them.

Regardless of the structure of the definition space, another interesting area of research concerns the processing of non-scalar images. In this case, there is no natural order to the values. A choice must be made to define a lattice suited to the nature of the data and the application objectives of the processing. This theoretical challenge has given rise to numerous proposals, particularly for color imaging (as opposed to binary or grayscale imaging) [1], multi- or hyperspectral imaging (as in astronomy or remote sensing) [52], and even tensor imaging, available in certain MRI modalities.

Finally, the most recent work aims to integrate mathematical morphology operators into deep learning models. This involves, for example, inferring the shape of the structural elements to be used to perform a given task [27], or determining whether a neural network model can be more effective by replacing certain convolutional layers with mathematical morphology operators [47]. Another challenge is to train a neural network to produce results with desired topological characteristics: number of desired minima or maxima, removal of connected components corresponding to false positives, or reconnection of true positives [48].

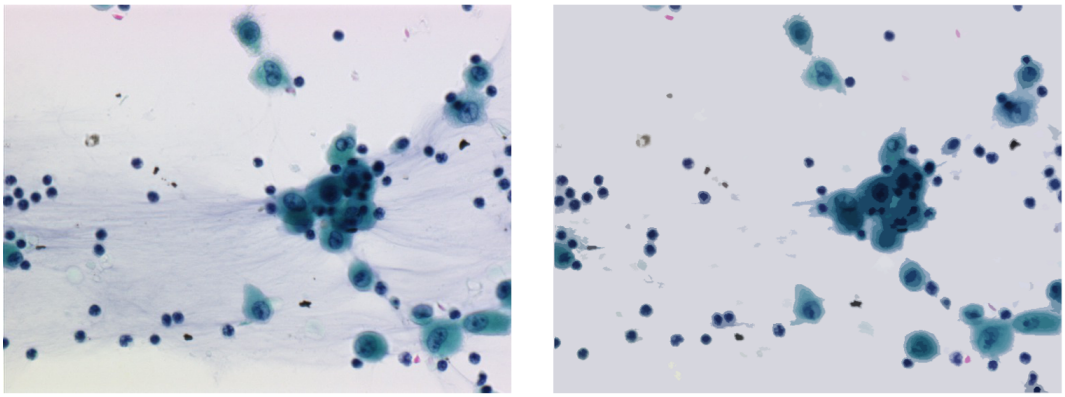


Figure 4 Simplification (right) of a microscopic color image of the bronchi (left) in cytology: connected filtering removes non-circular shapes in order to highlight the nuclei and cytoplasm. Illustration based on [11].

6 Reproducible research and software development

The GDMM community is committed to reproducible research through the provision of open codes and data. Among the first initiatives was the development of the Pink library. It contains more than 400 image processing algorithms and operators, many of which originate from the work of the GDMM community as a whole. We can also mention the development of the OLENA-MILENA-PYLENE library, which focuses on a software genericity approach: the same code must work for "all" possible data types. For image processing, we can also mention the Higra library, which is a reference for hierarchical analysis of images and data in mathematical morphology and is capable of interfacing with modern deep learning tools, as is MorphoLayers dedicated to this interaction between mathematical morphology and deep learning, which uses Keras/Tensorflow for this purpose.

The DGtal library is a collaborative effort which brings together the main algorithms from French research in discrete geometry. It is currently the benchmark and the state-of-the-art library in digital geometry, with more than 400,000 lines of code, 160 pages of documentation and hundreds of forks. It is developed in C++ and includes data-structures, algorithms, tools and interfaces for processing and visualizing images and geometric objects in arbitrary dimension, with also a front-end in Python. DGtal received the *software award* at the *Symposium on Geometry Processing* in 2016.

To complete this overview, we would like to highlight the efforts of the GDMM community to structure and promote reproducible research. In particular, the GDMM community is active in the editorial board of the journal IPOL (Image Processing OnLine), in which each article contains a text on an algorithm and its source code with an online demonstrator and an archive of the experiments carried out. It also helped establishing the RRPR (Reproducible Research in Pattern Recognition) workshop in the international image analysis landscape. This workshop has been held every two years since 2016 during the ICPR international conference (a major event organized by the International Association for Pattern Recognition, IAPR). Since 2024, RRPR has been a full-fledged technical committee of the IAPR (designated TC-22), with a vice-chair from the GDMM community, while discrete geometry and mathematical morphology form TC-18.

Contributors.

Jean Cousty, Yan Gerard, Jacques-Olivier Lachaud, Phuc Ngo, Nicolas Passat, and Isabelle Sivignon.

References

1 Jesús Angulo. Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis. *Computer Vision and Image Understanding*, 107(1):56–73, 2007.

2 Gilles Bertrand. Completions, perforations and fillings. In *DGMM, Proceedings*, pages 137–151, 2021.

3 Gilles Bertrand. Morse sequences: A simple approach to discrete Morse theory. *Journal of Mathematical Imaging and Vision*, 67:16, 2025.

4 Gilles Bertrand, Nicolas Boutry, and Laurent Najman. Discrete Morse functions and watersheds. *Journal of Mathematical Imaging and Vision*, 65:787–801, 2023.

5 Gilles Bertrand and Michel Couprie. On parallel thinning algorithms: Minimal non-simple sets, P-simple points and critical kernels. *Journal of Mathematical Imaging and Vision*, 35:23–35, 2009.

6 Nicolas Boutry, Thierry Géraud, and Laurent Najman. How to make n-D plain maps defined on discrete surfaces alexandrov-well-composed in a self-dual way. *Journal of Mathematical Imaging and Vision*, 61:849–873, 2019.

7 Nicolas Boutry, Laurent Najman, and Thierry Géraud. Equivalence between digital well-composedness and well-composedness in the sense of Alexandrov on n-d cubical grids. *Journal of Mathematical Imaging and Vision*, 62:1285–1333, 2020.

8 Nicolas Boutry, Laurent Najman, and Thierry Géraud. Some equivalence relation between persistent homology and morphological dynamics. *Journal of Mathematical Imaging and Vision*, 64:807–824, 2022.

9 Thomas Caissard, David Coeurjolly, Jacques-Olivier Lachaud, and Tristan Roussillon. Laplace–Beltrami operator on digital surfaces. *Journal of Mathematical Imaging and Vision*, 61(3):359–379, 2019.

10 Edwin Carlinet and Thierry Géraud. A comparative review of component tree computation algorithms. *IEEE TIP*, 23(9):3885–3895, 2014.

11 Edwin Carlinet and Thierry Géraud. Mtos: A tree of shapes for multivariate images. *IEEE Transactions on Image Processing*, 24(12):5330–5342, 2015.

12 Giovanni Chierchia and Benjamin Perret. Ultrametric fitting by gradient descent. *Advances in Neural Information Processing Systems*, 32, 2019.

13 David Coeurjolly, Jacques-Olivier Lachaud, and Jérémy Levallois. Multigrid convergent principal curvature estimators in digital geometry. *Computer Vision and Image Understanding*, 129:27–41, 2014.

14 Camille Couprie, Leo Grady, Laurent Najman, and Hugues Talbot. Power watershed: A unifying graph-based optimization framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(7):1384–1399, 2011.

15 Michel Couprie and Gilles Bertrand. New characterizations of simple points in 2D, 3D, and 4D discrete spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:637–648, 2009.

16 Jean Cousty, Gilles Bertrand, Laurent Najman, and Michel Couprie. Watershed cuts: Minimum spanning forests and the drop of water principle. *IEEE TPAMI*, 31(8):1362–1374, 2009.

17 Louis Cuel, Jacques-Olivier Lachaud, Quentin Mérigot, and Boris Thibert. Robust geometry estimation using the generalized voronoi covariance measure. *SIAM Journal on Imaging Sciences*, 8(2):1293–1314, 2015.

18 Fernando De Goes, Andrew Butts, and Mathieu Desbrun. Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics (TOG)*, 39(4):110–1, 2020.

19 François De Vieilleville, Jacques-Olivier Lachaud, and Fabien Feschet. Convex digital polygons, maximal digital straight segments and convergence of discrete geometric estimators. *Journal of Mathematical Imaging and Vision*, 27(2):139–156, 2007.

20 Mathieu Desbrun, Anil N Hirani, Melvin Leok, and Jerrold E Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.

21 Henri-Alex Esbelin, Rémy Malgouyres, and Colin Cartade. Convergence of binomial-based derivative estimation for c2 noisy discretized curves. *Theoretical Computer Science*, 412(36):4805–4813, 2011.

22 Sébastien Fourey and Rémy Malgouyres. Normals estimation for digital surfaces based on convolutions. *Computers & Graphics*, 33(1):2–10, 2009.

23 Yann-Situ Gazull, Alexandra Bac, and Aldo Gonzalez-Lorenzo. Computing geometrical measures of topological holes. *Computer-Aided Design*, 163:103563, 2023.

24 Thierry Géraud, Edwin Carlinet, Sébastien Crozet, and Laurent Najman. A quasi-linear algorithm to compute the tree of shapes of n d images. In *International symposium on mathematical morphology and its applications to signal and image processing*, pages 98–110, 2013.

25 Aldo Gonzalez-Lorenzo. *Computational Homology Applied to Discrete Objects*. PhD thesis, Aix-Marseille University, France, 2016.

26 Leo J Grady and Jonathan R Polimeni. *Discrete calculus: Applied analysis on graphs for computational science*, volume 3. Springer, 2010.

27 Romain Hermary, Guillaume Tochon, Élodie Puybareau, Alexandre Kirszenberg, and Jesús Angulo. Learning grayscale mathematical morphology with smooth morphological layers. *Journal of Mathematical Imaging and Vision*, 64(7):736–753, 2022.

28 Reinhard Klette and Azriel Rosenfeld. *Digital geometry - geometric methods for digital picture analysis*. Morgan Kaufmann, 2004.

29 Jacques-Olivier Lachaud, David Coeurjolly, Céline Labart, Pascal Romon, and Boris Thibert. Lightweight curvature estimation on point clouds with randomized corrected curvature measures. *Computer Graphics Forum*, 42(5):e14910, 2023.

30 Jacques-Olivier Lachaud, David Coeurjolly, and Jérémy Levallois. Robust and convergent curvature and normal estimators with digital integral invariants. In *Modern Approaches to Discrete Curvature*, pages 293–348. Springer, 2017.

31 Jacques-Olivier Lachaud, Xavier Provençal, and Tristan Roussillon. Two plane-probing algorithms for the computation of the normal vector to a digital plane. *Journal of Mathematical Imaging and Vision*, 59(1):23–39, 2017.

32 Jacques-Olivier Lachaud, Pascal Romon, and Boris Thibert. Corrected curvature measures. *Discrete & Computational Geometry*, 68(2):477–524, 2022.

33 Jacques-Olivier Lachaud, Pascal Romon, Boris Thibert, and David Coeurjolly. Interpolated corrected curvature measures for polygonal surfaces. *Computer Graphics Forum*, 39(5):41–54, 2020.

34 Jacques-Olivier Lachaud and Boris Thibert. Properties of gauss digitized shapes and digital surface integration. *Journal of Mathematical Imaging and Vision*, 54(2):162–180, 2016.

35 Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing*, 25(10):1572–1587, 2007.

36 Étienne Le Quentrec, Loïc Mazo, Étienne Baudrier, and Mohamed Tajine. Local turn-boundedness: A curvature control for continuous curves with application to digitization. *Journal of Mathematical Imaging and Vision*, 62:673–692, 2020.

37 Romain Lerallut, Étienne Decencière, and Fernand Meyer. Image filtering using morphological amoebas. *Image and Vision Computing*, 25(4):395–404, 2007.

38 Vaïa Machairas, Matthieu Faessel, David Cárdenas-Peña, Théodore Chabardes, Thomas Walter, and Etienne Decenciere. Waterpixels. *IEEE Transactions on Image Processing*, 24(11):3707–3716, 2015.

39 Loïc Mazo, Nicolas Passat, Michel Couprie, and Christian Ronse. Digital imaging: A unified topological framework. *Journal of Mathematical Imaging and Vision*, 44:19–37, 2012.

40 Loïc Mazo, Nicolas Passat, Michel Couprie, and Christian Ronse. Topology on digital label images. *Journal of Mathematical Imaging and Vision*, 44:254–281, 2012.

41 Christian Mercat. Discrete Riemann surfaces and the Ising model. *Communications in Mathematical Physics*, 218(1):177–216, 2001.

42 Quentin Mérigot, Maks Ovsjanikov, and Leonidas J Guibas. Voronoi-based curvature and feature estimation from point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 17(6):743–756, 2010.

43 Odyssée Merveille, Hugues Talbot, Laurent Najman, and Nicolas Passat. Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(2):304–317, 2018.

44 Phuc Ngo, Nicolas Passat, Yukiko Kenmochi, and Hugues Talbot. Topology-preserving rigid transformation of 2d digital images. *IEEE Transactions on Image Processing*, 23:885–897, 2014.

45 Nicolas Passat, Julien Mendes Forte, and Yukiko Kenmochi. Morphological hierarchies: A unifying framework with new trees. *Journal of Mathematical Imaging and Vision*, 65:718–753, 2023.

46 Samuel Peltier, Sylvie Alayrangues, Laurent Fuchs, and Jacques-Olivier Lachaud. Computation of homology groups and generators. *Computers & Graphics*, 30:62–69, 2006.

47 Valentin Penaud–Polge, Santiago Velasco-Forero, and Jesus G. Angulo. Group equivariant morphological networks. *SIAM Journal on Imaging Sciences*, 18(4):2236–2276, 2025.

48 Benjamin Perret and Jean Cousty. Component tree loss function: Definition and optimization. In *International Conference on Discrete Geometry and Mathematical Morphology*, pages 248–260, 2022.

49 Benjamin Perret, Jean Cousty, Olena Tankyevych, Hugues Talbot, and Nicolas Passat. Directed connected operators: Asymmetric hierarchies for image filtering and segmentation. *IEEE TPAMI*, 37(6):1162–1176, 2014.

50 Laurent Provot and Yan Gérard. Estimation of the derivatives of a digital function with a convergent bounded error. In *International Conference on Discrete Geometry for Computer Imagery*, pages 284–295, 2011.

51 Lydie Richaume, Eric Andres, Gaëlle Largeteau-Skapin, and Rita Zrour. Unfolding h-convex manhattan towers. *J. Comb. Optim.*, 44(4):3023–3037, 2022.

52 Santiago Velasco-Forero and Jesus Angulo. Supervised ordering in $\mathbb{R}^p$: Application to morphological processing of hyperspectral images. *IEEE TIP*, 20(11):3301–3308, 2011.

53 Colin Weill-Duflos, David Coeurjolly, and Jacques-Olivier Lachaud. Corrected Laplace–Beltrami operators for digital surfaces. *Journal of Mathematical Imaging and Vision*, 67(2):11, 2025.

Les vingt ans du GdR IFM, vus du GT « Modélisation géométrique »

Le début de notre communauté remonte à 1995 en tant que groupe de travail du GDR-PRC AMI (cadre du pôle Informatique Graphique) et de l'AFIG. Le GDR AMI a cessé d'exister fin 1997 et a cédé la place au GDR ALP (Algorithmique, Langages, Programmation), avec un pôle Informatique graphique et le groupe de travail Modélisation Géométrique s'est poursuivi. Depuis 2006, le GTMG fait partie du GDR IM (Informatique Mathématique, puis IFM en 2024) et du GDR IG (puis IGRV en 2014). Le GT travaille sur la définition de modèles géométriques, la description d'objets en tant que courbes et surfaces paramétriques, maillages, nuages de points, voxels et volumes. Il s'intéresse donc à la description, aux traitements, à la caractérisation de modèles discrets, continus ou semi-continus, ponctuels, linéiques, surfaciques, volumiques en dimension 2, 3 ou plus. La transformation entre les modèles est également un enjeu important, ainsi que la cohérence topologique de ceux-ci.

Le GT reste proche de l'application de ses travaux, en spécialisant modèles et traitements. Parmi les applications, la production industrielle (modélisation, façonnage, analyse) à toujours pris une place importante, ce qui explique le lien avec la communauté SMAI-AFA et les écoles d'ingénieurs comme l'ENSAM. L'aspect créatif et artistique des objets produits, à une échelle moindre, est toujours actif dans le groupe, on peut citer notamment les travaux des laboratoires LJK, ACROE, CRESTIC.

1 Topologie

La topologie est un point important du groupe, car la cohérence des modèles se base également sur ces concepts. On peut citer les travaux sur les cartes généralisées (XLIM). Les propriétés topologiques, les groupes d'homologie, le contrôle d'entités topologiques pendant la construction d'un objet prennent de l'importance depuis une dizaine d'années (ICUBE, XLIM, LIS, Figure 1). Les cartes combinatoires multirésolution décrivent des maillages de dimension quelconque. Des opérateurs topologiques et géométriques permettent de travailler d'abord en dimension 3 sur des maillages tétraédriques et hexaédriques, puis plus généralement sur des topologies arbitraires avec une approche multi-échelle. Les membres du groupe développent aussi une grande expertise en analyse topologique de données à travers des travaux sur la réduction de dimension de nuages de points par exemple. Finalement l'extraction de caractéristiques topologiques, comme l'axe médian reste un sujet très actif (Figure 1).

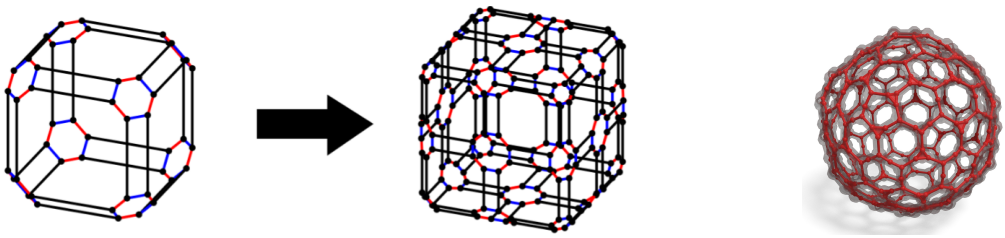


Figure 1 À gauche : Règles de subdivision pour un cube représenté sous forme de G-cartes (Repris de [24]. À droite : Extraction de l'axe médian d'une forme (Repris de [7]).

2 Reconstruction

La reconstruction ou la création d'un lien entre des espaces de dimensions différentes est généralement une transformation de modèles géométriques. Cela couvre les opérations classiques de triangulation 2D et 3D (sujet proche des thématiques du GT GDMM), de remaillage (INRIA Sophia-Antipolis) et a pris une force importante lors de la généralisation de la production des nuages de points (scanner laser, photogrammétrie) dans les années 2000. Les applications ont permis de définir de nouveaux traitements notamment en production industrielle (Scan2CAD, Scan2BIM, équipe du LISPEN) afin de filtrer, caractériser et annoter sémantiquement les données produites. Rapidement, les traitements se sont portés sur les nuages de points directement (*i.e.* sans voisinage, sans notion de topologie) afin d'éviter l'étape de maillage (consommatrice en temps, erreurs et espace, et qui constitue en soi une première interprétation de la donnée brute acquise). On peut citer les travaux du LIRIS ou de l'IRIT. Il est apparu également l'idée de joindre la géométrie à d'autres modalités comme l'intensité de laser, ou la couleur, permettant de spécialiser ou de lever des ambiguïtés dans les traitements (LIRIS). Cette thématique a fortement bénéficié du bond de l'IA à partir de 2010.

3 Caractérisation

La caractérisation est une thématique importante pour le retour d'information vers l'expertise de plus haut niveau. Elle a été motrice des algorithmes de détection, de segmentation, d'extraction de formes (*features*).

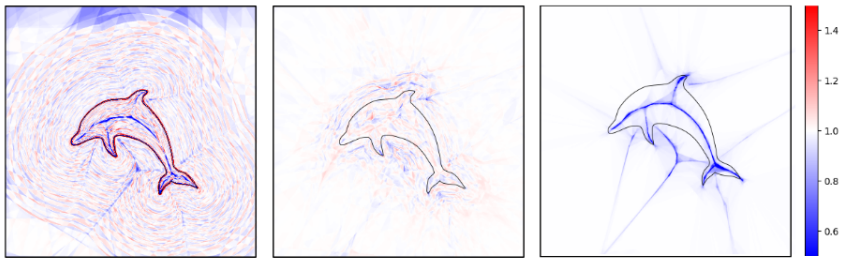
L'IA a un impact important sur la caractérisation d'objets : les processus de segmentation construits précédemment par analyses locale et globale, par référence à un dictionnaire de formes, par des systèmes experts ont pu être remplacés par des modèles supervisés ou non. Cela a rapidement posé à la communauté le problème de la création de modèles pour l'entraînement, ce qui reprend les règles de construction (CAD, architecture, mécanique). Cela permet également de comparer des objets 3D sous forme de graphes de propriétés géométriques et topologiques (GNN - *Graph neural network*, travaux du XLIM et LISPEN). L'IA a également un fort impact sur la caractérisation de nuages de points, où la topologie est complexe à extraire, et que l'apprentissage permet d'inférer.

4 Transformation de formes, *levels set*, transport optimal

La communauté du GT MG a également beaucoup contribué au développement de méthodes de transport optimal pour la préservation de la matière, notamment pour le recalage de nuages de points, ou pour l'interpolation de formes, avec la contrainte de conservation de masse proposée par le transport (transport semi-discret symétrisé par exemple), avec le défaut de déchirement des formes induit par le transport qui vise à minimiser des distances de déplacement (INRIA Nancy, LIRIS). Plus récemment la communauté s'est emparé des problèmes de *morphing* entre formes à volume constant mais en se basant sur la *level set equation*, en formulation implicite neuronale permettant de garantir une advection à volume constant (LIRIS).

5 Représentations neuronales implicites

Les travaux sur les surfaces implicites avaient été nombreux avant les années 2000 (ICUBE, IRIT, LIRIS) et ont trouvé une seconde dynamique récemment. Dans la lignée des NeRF



max $||\nabla f|| = 13.2$ max $||\nabla f|| = 1.43$ max $||\nabla f|| = 0.998$

(a) Classical MLP **(b)** MLP with eikonal regularization **(c)** Lipschitz neural network

Figure 2 Les réseaux 1-Lipschitz garantissent que la fonction distance signée estimée est bien 1-lipschitz. Repris de [9].

est apparue (ou revenue) l'idée de paramétriser les fonctions implicites par des réseaux de neurones, c'est à dire d'optimiser des paramètres d'un réseau pour représenter, par exemple la fonction implicite d'une forme. Il est ensuite possible à la façon d'un *Physically Informed Neural Network* (PINN) de rajouter des contraintes d'EDP, comme par exemple l'équation eikonale qui doit être satisfaite par une fonction distance signée. Cette représentation est utile par exemple pour extraire des axes médians de manière robuste (LIRIS). On peut aussi contraindre par construction le réseau à produire des fonctions 1-lipschitz (INRIA Grenoble, voir Figure 2). Il est également possible de paramétriser temporellement cette fonction implicite pour modéliser des morphings de formes (LIRIS). Cela permet à nouveau d'intégrer des contraintes sous formes d'équations aux dérivées partielles et d'apprendre des champs d'advection avec certaines bonnes propriétés (voir au dessus).

6 Spécifications formelles et preuves en modélisation géométrique, contraintes

L'analyse qualitative des systèmes de contraintes utilise classiquement des méthodes combinatoires (le plus souvent des graphes, parfois des matroïdes). Cette analyse est indispensable, vu la taille des systèmes de contraintes, mais elle ne fonctionne bien que pour les systèmes bien contraints, que ce soit en 2D ou en 3D ; elle ne peut détecter que les erreurs les plus simples, dites structurelles, comme dans : $f(x, y, z) = g(z) = h(z) = 0$ qui sur-contraint l'inconnue z. En 3D, cette analyse se heurte à de sérieuses difficultés, comme la caractérisation des graphes rigides. Plus généralement, de nombreux théorèmes de géométrie ou de topologie provoquent des dépendances non structurelles entre les contraintes (ICUBE, LIB). Comme dans les autres domaines, il faut utiliser de manière systématique en modélisation géométrique des techniques avancées de spécifications formelles et de preuves de propriétés et de programmes, de préférence assistées par ordinateur. La résolution numérique (y compris par des méthodes d'analyse par intervalles) est guidée par l'analyse qualitative des systèmes de contraintes permettant de détecter les erreurs (sous- ou sur-contraintes) et de décomposer les systèmes bien contraints.

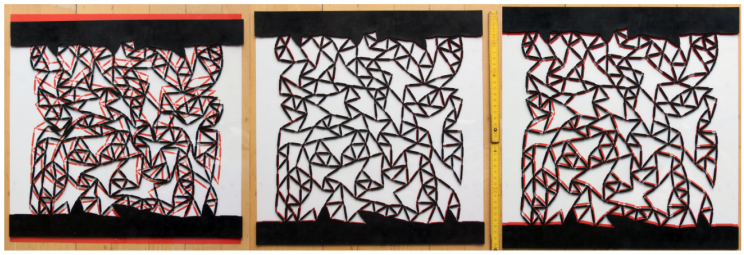


Figure 3 Optimisation géométrique pour la conception de méta-matériau auxetique. Repris de [2].

7 Analyse multirésolution, streaming et compression

Le lien entre topologie et géométrie est très fort, essentiel pour décrire une surface, des bords, des frontières. Les surfaces de subdivision ont longtemps tiré ce thème (IRIT, LIB). Leur facilité à raffiner une géométrie a fait de la compression une des applications de ces méthodes. La difficulté est alors de construire des maillages initiaux suffisamment légers mais dont la subdivision conduirait à un maillage dense identique à l'initial. Cela nécessite la maîtrise de subdivision inverse et le travail sur les schémas de subdivision (assurer la localité, la contraction, les contraintes de continuité...). Pour cela l'analyse multi-résolution et notamment les transformées en ondelettes sont toujours d'actualité.

Dorénavant, les méthodes NeRF et Gaussian Splat sont plus efficaces en terme de visualisation et s'attachent moins à la mesure de la géométrie (INRIA Sophia-Antipolis, IRIT).

Cette thématique porte également l'analyse de la rugosité des surfaces et la caractérisation des surfaces fractales (LIB). On représente le plus souvent des surfaces lisses par des surfaces de subdivision et des surfaces rugueuses par des surfaces fractales (IFS). Un formalisme commun (basé sur les IFS) a été mis en place (de manière théorique).

8 Structures géométriques

Un dernier axe de recherche actif dans le GT MG est l'optimisation de la géométrie pour l'impression 3D (Nancy, Grenoble) avec la prise en compte de caractéristiques souhaitées (compressibilité dans une direction par exemple). La figure 3 montre un exemple d'une telle optimisation pour un méta-matériau auxétique.

9 Logiciels et valorisation de la communauté MG

— membre du consortium CGAL : importante bibliothèque de géométrie algorithmique (LIRIS, INRIA Sophia-Antipolis)
— CGoGN, plateforme logicielle de modélisation géométrique et topologique de l'équipe IGG, basée sur CGAL (ICUBE)
— Jerboa, un modeleur géométrique à base de règles de transformation de graphes, modélisation basée sur la topologie, XLIM, Poitiers

Contributeurs et contributrices.

Julie Digne, Samuel Peltier, Romain Raffin.

Références

1 Dominique Attali, Mattéo Clémot, Bianca Dornelas, and André Lieutier. When alpha-complexes collapse onto codimension-1 submanifolds. In *Proceedings of the Symposium on Computational Geometry (SoCG)*, 2025.

2 Georges-Pierre Bonneau, Stefanie Hahmann, and Johana Marku. Geometric construction of auxetic metamaterials. *Computer Graphics Forum*, 40(2) :291–303, 2021.

3 Nicolas Bonneel and Julie Digne. A survey of optimal transport for computer graphics and computer vision. *Computer Graphics Forum*, 42(2) :439–460, 2023.

4 Alexander Braune, Mark Gillespie, Yiying Tong, and Mathieu Desbrun. Discrete torsion of connection forms on simplicial meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2025.

5 Anh Quoc Bui, Gilles Rougeron, Géraldine Morin, and Simone Gasparini. ROI-GS : Interest-based local quality 3D gaussian splatting. In *IEEE Visual Communications and Image Processing (VCIP)*, 2025.

6 Camille Buonomo, Julie Digne, and Raphaëlle Chaine. Volume preserving neural shape morphing. *Computer Graphics Forum (Proceedings of SGP)*, 44(5), 2025.

7 Mattéo Clémot and Julie Digne. Neural skeleton : Implicit neural representation away from the surface. *Computers & Graphics*, 114 :368–378, 2023.

8 Mattéo Clémot, Julie Digne, and Julien Tierny. Topological autoencoders++ : Fast and accurate cycle-aware dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, 32(2) :1622–1639, 2026.

9 Guillaume Coiffier and Louis Béthune. 1-Lipschitz neural distance fields. *Computer Graphics Forum (Proceedings of SGP)*, 43(5) :i–x, 2024.

10 Vincent Commin, Samuel Peltier, Arthur Cavalier, and Sébastien Horna. Clock mechanism generation using interaction graphs. *Computers & Graphics*, 2025.

11 Khoa Do, David Coeurjolly, Pooran Memari, and Nicolas Bonneel. Linear-time transport with rectified flows. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2025.

12 Amine Farhat, Alexandre Bléron, Romain Vergne, and Joëlle Thollot. Motion ribbons : Parametrized surfaces for depicting motion effects. *Computers & Graphics*, 129 :104227, 2025.

13 Alex Fernandes, Steve Oudot, and François Petit. Computation of gamma-linear projected barcodes for multiparameter persistence. *Journal of Applied and Computational Topology*, 9(2) :12, 2025.

14 Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. Foldover-free maps in 50 lines of code. *ACM Transactions on Graphics*, 40(4), 2021.

15 Guillaume Gisbert, Raphaëlle Chaine, and David Coeurjolly. Inpainting holes in folded fabric meshes. *Computers & Graphics*, 114 :201–209, 2023.

16 Diego Gomez, Bingchen Gong, and Maks Ovsjanikov. FourieRF : Few-shot NeRFs via progressive fourier frequency control. In *International Conference on 3D Vision (3DV)*, 2025.

17 Chems Eddine Himeur, Thibault Lejemble, Thomas Pellegrini, Mathias Paulin, Loïc Barthe, and Nicolas Mellado. PCEDNet : A lightweight neural network for fast and interactive edge detection in 3D point clouds. *ACM Transactions on Graphics*, 2021.

18 V. N. Huynh, H. H. Nguyen, and R. Raffin. A versatile multi-space DBSCAN framework for rough surface object segmentation. *Multimedia Tools and Applications*, 84 :39473–39497, 2025.

19 Maylis Jouvencel, Razmig Kéchichian, Julie Digne, and Sébastien Valette. SCONet : Convolutional occupancy networks for multi-organ segmentation. In *IEEE International Symposium on Biomedical Imaging - ISBI*, pages 1–5. IEEE, 2025.

20 Mathieu Ladeuil, Marc Trabucato, Alexis Vaisse, and Noura Faraj. Weighted feature graph via hierarchical clustering. In *Eurographics Posters*, 2025.

21 Vadim Lebovici, Jan-Paul Lerch, and Steve Oudot. Local characterization of block-decomposability for multiparameter persistence modules. *Homology, Homotopy and Applications*, 2025.

22 Léopold Maillard, Nicolas Sereyjol-Garros, Tom Durand, and Maks Ovsjanikov. DeBaRA : denoising-based 3D room arrangement generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

23 Niv Maruani, Yifan Wang, Matthew Fisher, Pierre Alliez, and Mathieu Desbrun. ShapeShifter. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

24 Romain Pascual, Hakim Belhaouari, Agnès Arnould, and Pascale Le Gall. Inferring topological operations on generalized maps : Application to subdivision schemes. *Graphics and Visual Computing*, 6 :200049, 2022.

25 Samuel Peltier, Géraldine Morin, and Damien Aholou. Tubular parametric volume objects : Thickening a piecewise smooth 3D stick figure. *Computer Aided Geometric Design*, 85, 2021.

26 Mathieu Pietri, Eric Remy, Vincent Penné, and Jean-Luc Mari. Real-time live compression of dynamic 3D meshes for client-side GPU cloud gaming using skinning decomposition. *The Visual Computer*, 2025.

27 Clément Poull, Christian Gentil, Céline Roudet, Lucie Druoton, and Michael Roy. Second order differential properties of tensor product fractal surfaces. In *GRAPP*, 2025.

28 Yuang Shi, Simone Gasparini, Géraldine Morin, Chenggang Yang, and Wei Tsang Ooi. Sketch and patch : Efficient 3D Gaussian representation for man-made scenes. In *ACM Multimedia Systems Conference (MMVE)*, 2025.

29 Ramana Sundararaman, Nicolas Donati, Simone Melzi, Etienne Corman, and Maks Ovsjanikov. Deformation recovery : Localized learning for detail-preserving deformations. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2024.

30 Lucas Vergez, Arnaud Polette, and Jean-Philippe Pernot. Multi-part kinematic constraint prediction for automatic generation of CAD model assemblies using graph convolutional networks. *Computer-Aided Design*, 2025.

31 Giulio Viganò, Maks Ovsjanikov, and Simone Melzi. NAM : neural adjoint maps for refinement of shape correspondences. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2025.

32 Chao Zhang, Arnaud Polette, Romain Pinquié, Gregorio Carasi, Henri De Charnace, and Jean-Philippe Pernot. eCAD-Net : Editable parametric cad models reconstruction from Dumb B-Rep models using deep neural networks. *Computer-Aided Design*, 178, 2025.