Quelques études de l'aléatoire en informatique

Jean-Marie Le Bars

Préparée au sein de l'université de Caen Normandie

Quelques études de l'aléatoire en informatique

Présentée par Jean-Marie Le Bars

Habilitation à diriger les recherches soutenue publiquement le 29 juin 2016 à 10h30
devant le jury composé de

Mr Duchon Philippe	Professeur à l'université de Bordeaux LABRI	Rapporteur
Mr Charrier Christophe	Maître de conférences HDR université de Caen Normandie, GREYC	Examinateur
Mr Grandjean Etienne	Professeur à l'université de Caen Normandie GREYC	Examinateur
Mr Marion Jean-Yves	Professeur à l'université de Lorraine, LORIA	Rapporteur
Mr Rosenberger Christophe	Professeur à l'ENSICAEN, GREYC	Directeur HDR
Mr Tillich Jean-Pierre	CR HDR INRIA Paris	Rapporteur

Première partie

Préambule

Je vais m'attacher dans cette partie à montrer la cohérence de mes travaux de recherche. Bien que reposant sur plusieurs domaines de l'informatique clairement distincts, toutes mes études font intervenir l'aléatoire. C'est cette notion d'aléatoire – vue sous différents aspects– qui unifie mon travail. La figure 1 reprend les périodes de travail concernant ces différents domaines. Nous avons d'un côté des domaines de recherche de l'informatique mathématique que sont la logique, l'algorithmique, la complexité et la combinatoire. D'un autre côté, nous avons des domaines en sécurité informatique, avec des études en cryptographie, tatouage et biométrie. Ces domaines d'application sont transversaux aux domaines fondamentaux, ils offrent des motivations scientifiques complémentaires. J'ai aussi indiqué les objets sur lesquels j'ai effectué mes études : les graphes orientés, les fonctions booléennes et les triangulations de Delaunay.

Je ne vais donc pas détailler mes résultats de recherche, mais plutôt insister sur mon cheminement qui m'a conduit à explorer différentes thématiques.

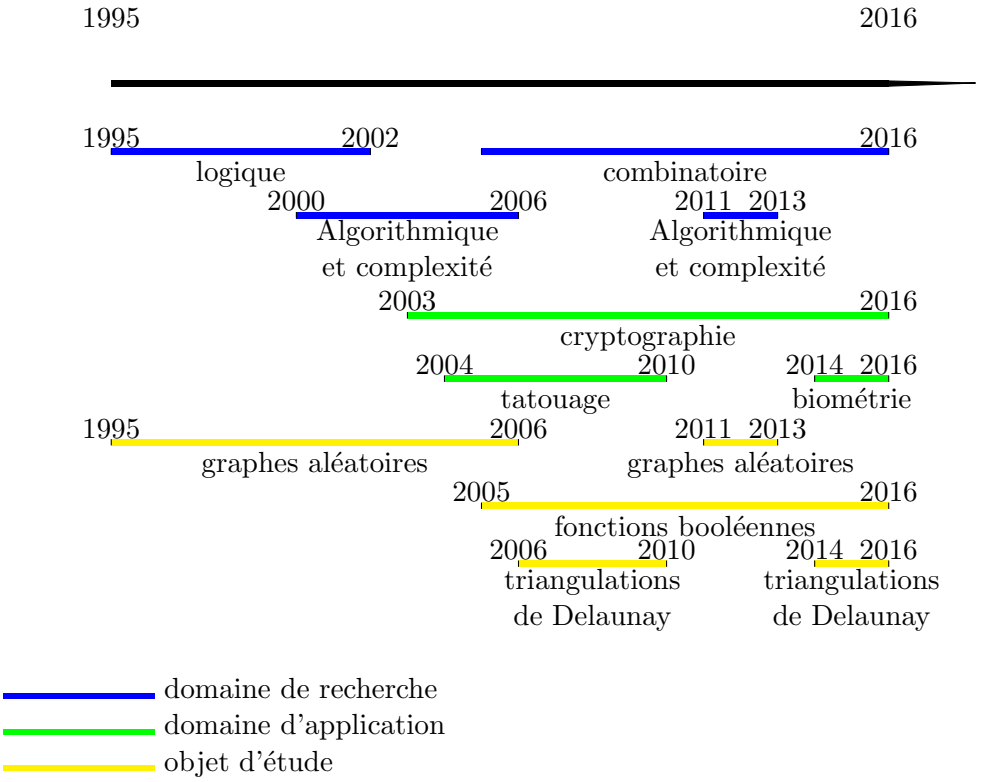


FIGURE 1 – Chronologie des recherches

Tout d'abord, j'ai suivi des études mathématiques à l'université de Caen, j'ai obtenu ma maîtrise de mathématiques pures en 1993 et mon DEA d'algorithmique et d'arithmétique en 1994. J'ai découvert et apprécié l'informatique théorique par les cours de théorie des langages et de décidabilité et complexité délivrés par Patrick Dehornoy. Je me suis aperçu que cela correspondait plus à mes envies et mes compétences

alors qu'auparavant j'étais plus intéressé par la théorie des nombres. C'est dans cet état d'esprit que j'ai souhaité faire de la recherche et je me suis naturellement tourné vers Etienne Grandjean pour commencer une thèse en 1995, en logique informatique, plus précisément en théorie des modèles finis. Ce domaine peut être vu comme une branche de la logique mathématique –la théorie des modèles– restreinte aux structures finies. Du fait qu'elle porte sur des structures finies elle amène des applications dans divers domaines de l'informatique théorique comme la théorie des bases de données, la complexité descriptive et la théorie des langages formels. Mon sujet de thèse portait sur les lois 0-1, il s'agit d'étudier les probabilités asymptotiques des propriétés définissables dans une logique. J'ai pu intégrer la communauté nationale sur ce sujet – le GT (groupe de travail) CMF, Complexité et modèles finis– et la communauté internationale présente notamment dans la conférence européenne CSL (Computer Science Logic) et la conférence internationale LICS (Logic In Computer Science). Mes résultats de contre-exemples de lois 0-1 –basés sur des variantes de la propriété de noyau– ont été obtenus en fin de thèse. Ils ont été remarqués comme une avancée significative sur ce sujet par des chercheurs majeurs du domaine comme Kolaitis, Vardi, Pacholski et Gurevich. Ces résultats m'ont permis d'obtenir le Kleene award, le meilleur papier étudiant, à la conférence de LICS 98. J'ai également reçu un des deux accessits du prix de thèse SPECIF 98. Dans les deux cas, j'ai reçu cela comme une reconnaissance de la valeur de mon travail et cela m'a encouragé à poursuivre dans le milieu de la recherche. J'ai soutenu ma thèse en janvier 1998, et après une année d'ATER j'ai obtenu un poste de maître de conférences toujours à l'université de Caen en septembre 1999 dans l'équipe Algorithmique (qui deviendra l'équipe AMACC à partir de 2010).

Dès 1998, j'ai découvert d'autres GT du GDR Informatique Mathématique, les GT C2 (Codage et Cryptographie) et Aléa. C'est surtout le GT Aléa –dédié à l'analyse d'algorithmes et à l'analyse des propriétés des structures aléatoires discrètes– qui m'a intéressé à ce moment-là. Il avait pour leader charismatique Philippe Flajolet qui a constitué au fils des ans une communauté conséquente mêlant mathématiciens, physiciens et informaticiens. Cette communauté est particulièrement soudée, elle se retrouve tous les ans pendant une semaine pour les journées Aléa où tous les participants –des étudiants aux chercheurs confirmés– ont la possibilité de faire un exposé devant un public attentif et passionné. Sur l'invitation de Brigitte Vallée, professeur d'informatique à l'université de Caen et membre de cette communauté, j'ai exposé mes travaux de recherche en février 1998. Je suis retourné de nombreuses fois à ces journées et j'ai souvent eu l'occasion de présenter mes différentes recherches.

Dans les années qui ont suivi ma thèse, j'ai surtout orienté mes recherches sur les thématiques du GT Aléa. Sur la figure 1, les études du noyau de la période de 2000 à 2006 sont regroupés sous le domaine *Algorithmique et complexité*. Cela recouvre cependant des thématiques variées : complexité, théorie des jeux, transitions de phase, combinatoire analytique. Je précise que je n'étais pas du tout un spécialiste des graphes aléatoires pendant ma thèse qui était plus orientée en logique. J'ai donc aussi pendant ces années d'après thèse renforcé mes connaissances sur ce sujet et affiné mes méthodes de preuves. J'ai également intégré le groupe de recherche internatio-

nal AofA (Analysis of Algorithms) qui est sur les mêmes thématiques qu'Aléa et qui organise une conférence internationale annuelle.

Cette période marque un tournant dans ma carrière de chercheur. En effet, de part ma formation en mathématiques pures, j'étais plus attiré par des preuves formelles et là, j'ai eu l'occasion d'implémenter des algorithmes dont j'ai testé l'efficacité sur différentes distributions pour évaluer la difficulté de trouver un noyau dans un graphe aléatoire. Je me suis aussi intéressé dès 2003 à un sujet d'étude de cryptographie en *information hiding*, la difficulté de retrouver un noyau caché dans un graphe, problème pour lequel je n'ai obtenu une solution satisfaisante qu'en 2013 avec Marco Illengo, alors post doctorant en mathématiques à l'université de Caen.

Parallèlement à ces nouvelles recherches sur le noyau, j'ai poursuivi mon travail sur les lois 0-1 mais cette fois ci en logique modale. Nous avons vu avec Valentin Goranko comment exprimer cette propriété et définir des variantes en logique modale. J'ai aussi obtenu en 2002 un contre-exemple de lois 0-1 contredisant un résultat célèbre de Joseph Halpern et Bruce Kapron de 1994.

J'ai obtenu une délégation CNRS de deux ans de septembre 2003 à août 2005 à l'université Paris-Nord, plus précisément au LIPN, dans l'équipe OCAD (optimisation Combinatoire et Algorithmique Distribuée). J'ai mis en application mes compétences en combinatoire analytique acquises lors des journées Aléa en utilisant les séries génératrices avec Cyril Banderier et Vlady Ravelomanana sur les noyaux dans les graphes ayant un seul cycle ou un seul circuit. Ma contribution visait plus la détermination des séries génératrices et l'obtention des formes closes de celles-ci que l'extraction des coefficients et le calcul de leur asymptotique qui requièrent une bonne maîtrise de l'analyse complexe, heureusement mes collègues étaient plus à l'aise que moi sur cette partie.

Un nouveau tournant dans ma carrière a aussi eu lieu pendant cette délégation, nous avons souhaité avec Alfredo Viola –alors professeur invité au LIPN–d'utiliser la combinatoire analytique– pour dénombrer le nombre de fonctions booléennes 1-résilientes. La 1-résilience et plus généralement la k-résilience est un critère des fonctions booléennes important pour leur utilisation en cryptographie symétrique (notamment pour combiner ou filtrer des LFSR, les registres à décalage à rétroaction linéaire). Nous avons réussi à compter et énumérer (qui sont deux problèmes bien distincts) les fonctions 1-résilientes jusqu'à 8 variables. Nous avons déterminé les séries génératrices en jeu, mais elles se sont avérées insuffisantes pour résoudre notre problème d'énumération. Notre travail a été autant combinatoire qu'algorithmique. Nous avons poursuivi cette recherche en proposant le codage énumératif et la génération aléatoire de ces fonctions. La méthode d'énumération –que nous avons appelée méthode des classes– a alors été alliée à la méthode récursive, elle même issue de la méthode du dictionnaire introduite par Flajolet. Cette fois-ci le travail a été essentiellement de l'algorithmique et de la programmation, l'implémentation a nécessité un « tuning » complexe afin que les compromis en terme de mémoire entre les pré-calculs et les traitements dynamiques soient optimisés.

J'ai cherché avec Viola et d'autres chercheurs à reprendre la méthode des classes pour d'autres propriétés cryptographiques des fonctions booléennes. Je me suis aperçu

que le principal obstacle provenait de la nécessité de considérer pour une fonction booléenne à la fois la représentation sous forme de table de vérité et sous forme algébrique normale. J'étudie actuellement les fonctions coïncidentes avec Hayat Cheballat et Morgan Barbier, ces fonctions ont la particularité de faire correspondre ces deux représentations.

Entre 2004 et 2008, j'ai participé à une ACI sécurité nommée Tadorne sur le tatouage de données contraintes et structurées. Cet ACI a été dirigé par David Gross-Amblard que j'avais rencontré lors des journées CMF. Je me suis formé à la problématique du tatouage, mais aussi aux spécificités des données géographiques que nous avons privilégiées pour nos études. Un juriste est intervenu pendant cette ACI pour nous expliquer comment une preuve de propriété se constitue. Il a en particulier expliqué le rôle de l'aléatoire, car les preuves exploitent plus l'improbabilité que la certitude. Il faut donc être en mesure d'effectuer une modélisation aléatoire afin de montrer qu'un événement demeure très improbable.

Cela m'a permis de mieux comprendre comment associer à une recherche académique un réel cadre applicatif, depuis je conserve cela en tête lors de mes études en sécurité informatique. J'ai co-dirigé avec Jacques Madelaine la thèse de Cyril Bazin de 2006 à 2010. Le décalage entre le début de l'ACI et le commencement de la thèse est dû au temps nécessaire pour être bien formé sur le tatouage et les documents géographiques, mais surtout pour avoir un sujet clairement défini, la thématique du tatouage de documents étant très peu explorée. La méthode que nous avons retenue consiste à extraire de l'aléatoire de petites parties du document appelées sites –construites sur une triangulation de Delaunay des points du document– et de modifier ces sites afin d'obtenir des biais statistiques. La validation nécessite des statistiques et un protocole d'évaluation rigoureux. Ce travail fut très différent de mes autres recherches, il m'a permis de découvrir comment fonctionne une étude basée sur une démarche expérimentale. De plus, j'ai apprécié pour ce premier encadrement de thèse de pouvoir intervenir avec un rôle différent, ma contribution étant surtout d'aider à formaliser le travail de mon étudiant en favorisant ses idées et ses orientations.

Pendant la période 2011-2013, j'ai repris mes recherches sur la propriété de noyau. Avec Marco Illengo, nous avons réussi à calculer la probabilité asymptotique de cette propriété sur les graphes creux (sparse graphs en anglais), problème auquel je m'étais attaqué sans succès auparavant avec Wenceslas Fernandez de la Vega et qui semblait particulièrement coriace. Nous avons également réussi à mesurer la difficulté de retrouver un noyau caché dans un graphe aléatoire et à proposer une utilisation en cryptographie (preuve de propriété sans divulgation de connaissance).

J'ai décidé de changer d'équipe de recherche en janvier 2014. J'ai intégré une autre équipe du GREYC, l'équipe Monétique et Biométrie. Je m'intéresse depuis aux aspects aléatoires de structures discrètes intervenant dans les domaines de la Biométrie et de la Confiance. Ce changement d'équipe me permet, d'une part, d'être confronté à l'expérimentation (évaluation des méthodes proposés) alors que cet aspect de la recherche était très peu considéré dans ma précédente équipe et, d'autre part, d'avoir de meilleures opportunités d'encadrements de thèses. Je souhaitais pouvoir continuer de travailler sur des types d'activités de recherche variées comme je l'avais fait au-

paravant, mais dans une équipe où toutes ses activités sont valorisées. La recherche de liens entre la combinatoire et l'algorithmique sur des structures aléatoires, la recherche d'algorithmes performants, la recherche de l'aléatoire dans des données réelles, l'évaluation et la validation avec une approche plus expérimentale sont autant d'investigations qui me tiennent à cœur.

Bien sûr, tous les aspects de l'aléatoire n'apparaissent pas simultanément dans toutes les études, mais j'espère continuer à m'y confronter, en particulier par l'encadrement de thèses.

J'ai eu la chance de pouvoir peu de temps après mon arrivée dans l'équipe Monétique et Biométrie co-encadrer deux étudiants en thèse en biométrie sur les empreintes digitales. Je me suis initié à ce domaine dont j'ai trouvé quelques similitudes avec le tatouage de données géographiques, spécialement sur la façon dont intervient l'aléatoire. D'autre part, la triangulation de Delaunay apparaît comme un outil très utile dans ces deux domaines.

Zhigang Yao, le premier étudiant, a déjà soutenu en juillet 2015 et Benoît Vibert, le second, devrait soutenir en décembre 2016. Ma motivation principale pour l'obtention de l'HDR est de pouvoir très rapidement diriger de nouvelles thèses, cette fois-ci en tant que directeur officiel, l'équipe offrant de fortes opportunités pour cela. Normalement, dès septembre 2016, je devrais diriger une thèse avec une allocation ministérielle sur les mots de passe et une autre avec un financement CIFRE sur les *blockchain*.

Concernant la structure du document de l'HDR, celui-ci comporte une partie reprenant mon CV (activités d'enseignement, d'administration et de recherche), une partie sur la propriété de noyau sur les graphes aléatoires, une partie sur l'énumération et la génération aléatoire des fonctions booléennes et une partie sur le tatouage et la biométrie. Pour chacune de ces trois dernières parties, le premier chapitre amène une introduction non technique, avec le contexte et un résumé des contributions.

Finalement, une dernière partie expose mon projet de recherche.

Deuxième partie

Curriculum vitæ

Chapitre 1

Synthèse de la carrière

État civil

Jean-Marie Le Bars
Maître de conférences
Section CNU : 27
Date de naissance : 2 décembre 1966
Situation familiale : marié, 4 enfants
Établissement : Université de Caen Normandie
Laboratoire GREYC (UMR 6072)
Équipe de recherche : Monétique & Biométrie
Adresse électronique : jean-marie.lebars@unicaen.fr
page Web : https ://www.greyc.fr/fr/users/lebarsj
Adresse personnelle : 7 rue des loisirs 14610 Epron

Expérience professionnelle

depuis 1999 - maître de conférences à l'UCBN
2003-2004 - délégation de deux ans au LIPN
1998 - poste d'ATER à temps complet à l'UCBN, enseignant au CNAM
1994 - 1997 - Vacataire en informatique et en mathématiques à l'UCBN

Parcours universitaire

janvier 1998 - Thèse de doctorat de l'UCBN intitulée « Probabilités asymptotiques et pouvoir d'expression des fragments de la logique du second ordre ».
Rapporteurs : P. G. Kolaitis, J. F. Lynch, M. de Rougemont
Jury : S. Abitboule, P. Dehornoy, W. Fernandez de la Vega, E. Grandjean (directeur), J. Stern, P. Toffin, B. Vallée.
1994 - DEA d'algorithmique et d'arithmétique à l'UCBN
1993 - Maîtrise de Mathématiques à l'UCBN

Enseignements

Algorithmique et programmation
Mathématiques discrètes
Programmation orientée objet

Probabilités et statistiques
Génération et tests aléatoires
Algorithmes probabilistes
Outils algorithmiques pour la cryptographie

Responsabilités actuelles

Directeur-adjoint du département d'informatique (2011-2015)
Responsable des formations d'informatique (depuis 2011)
Directeur des études de la licence d'informatique (depuis 2011)
Porteur des maquettes de la licence d'informatique (maquette 2012-2017 et 2017-2022)
Membre élu du conseil de l'UFR (depuis 2011)

Recherche

Études algorithmiques et combinatoires des structures aléatoires
Outils algorithmiques pour la sécurité (cryptographie, biométrie, tatouage)
Mise en place et étude de la sécurité de systèmes d'authentification

Publications

4 revues internationales
13 conférences internationales avec actes et comité de lecture,
1 conférence nationale avec comité de lecture.

Rayonnement

Co-organisation de quatre conférences, deux conférences internationales (AofA2001, Number, sequences, Lattices 2010) et deux conférences nationales (Aléa2005, école Jeunes chercheurs en algorithmique et calcul formel 2000)
Comité de lecture de multiples conférences internationales et revues
Expertise de projets (1 ANR, 1 ACI)

Encadrement de thèses

Une thèse soutenue en 2010 (encadrement à 50%), une thèse soutenue en 2015 (encadrement à 33%)
Une thèse en cours (encadrement à 33%)

Chapitre 2

Activités pédagogiques

2.1 Responsabilités pédagogiques

Mise en place de la licence d'informatique
Depuis que je suis en poste à l'UCBN, je me suis toujours activement impliqué dans le suivi de nos formations en informatique, en particulier pour la licence d'informatique. La réforme LMD nous a permis d'élaborer pour la première fois un parcours informatique sur les trois premières années de la licence, auparavant la licence d'informatique se faisait en une année après un DEUG de mathématiques ou de physique-chimie (SM). J'ai participé en 2003-2004 avec Patrice Enjalbert, alors président de la commission pédagogique, à ce vaste chantier où toute notre formation a dû être remise à plat. J'ai piloté en 2010 la maquette de la mention informatique de la licence sciences et technologies (regroupant toutes les mentions de l'UFR sciences) pour son renouvellement pour la période 2012-2017. Il faut noter que la mention informatique a été la seule sur toutes les licences de l'UCBN à obtenir une évaluation A+ par l'AERES en partie grâce à un bon compromis entre une finalité scientifique généraliste et professionnelle. J'ai mis en place pour la prochaine maquette 2017-2022 un comité de pilotage. L'effectif total des trois années de licence a considérablement augmenté, il est passé de 200 en 2012-2013 à 350 en 2014-2015, ce qui complexifie considérablement son organisation en partie pour le suivi des enseignements et le suivi des étudiants.

Promotion des formations du département d'informatique
J'ai animé à de nombreuses reprises des conférences pour présenter nos formations aux lycéens (journées du lycéen, salon de l'étudiant). J'interviens également auprès des étudiants de L1 et L2 pour leur présenter les différentes formations en informatique, licences professionnelles et master. J'ai participé à la rédaction de flyers présentant les formations du département d'informatique. Je participe depuis 5 ans à une conférence sur les métiers de l'informatique au salon de l'étudiant de Caen.

Méthodes pédagogiques
Je m'intéresse fortement à la mise en place de nouvelles méthodes pédagogiques. J'ai introduit pour la maquette 2012-2017 de la licence d'informatique un encadrement de

projet sur les deux semestres du L2 appelé TPA (Travaux Personnels Approfondis) qui permet un travail par groupe de quatre à six étudiants suivis par un enseignant tout au long de l'année. J'expérimente la pédagogie inversée dans la valeur de tests aléatoires en M1, les étudiants n'ont que des TP et les notions de cours sont incluses dans les énoncés des TP et expliquées à la demande en fonction des besoins des étudiants pendant la réalisation des TP.

2.2 Présentation synthétique des enseignements

Mes enseignements de prédilection portent sur l'informatique mathématique, l'algorithmique et l'aléatoire. Ces dernières années, j'ai enseigné les matières suivantes :

- Initiation à la programmation en python L1
- Mathématiques pour l'informatique L2
- Structures de données et algorithmique L2
- Programmation orientée objet L2
- Mathématiques discrètes (combinatoire analytique et analyse d'algorithmes) L3
- Probabilités et statistiques pour l'informatique M1
- Tests aléatoires (modélisation aléatoire, protocoles de tests) M1
- Structures aléatoires (graphes aléatoires, fonctions booléennes pour la cryptographie) M2
- Algorithmes probabilistes M2

Je suis bien sûr disposé à assurer d'autres types d'enseignements selon les besoins du département d'informatique. J'essaierai, si possible, de toujours enseigner du L1 jusqu'au M2, car cela permet une meilleure vision verticale de nos diplômes.

Je compte de plus m'investir davantage sur les projets des étudiants, car c'est à mon avis un point crucial pour améliorer la qualité de nos formations. En effet, cela permet d'une part aux étudiants de mettre en application efficacement les concepts vus en cours et cela produit d'autre part un lien naturel entre nos activités de recherche et les compétences que doivent acquérir les étudiants lors du master. J'ai ainsi encadré pendant l'année 2014-2015 quatre projets en M1 et M2 du master e-SECURE sur des thématiques liées à ma nouvelle équipe de recherche, trois projets en biométrie et un en authentification.

Activités pédagogiques à l'UCBN avant d'être maître de conférence

Diplôme	Enseignement	CM	TD	TP	Année
DEUG A 1ère année	Mathématiques		48		1993-1994
DEUG A 2ème année	Mathématiques		48		1993-1994
DEUG B 2ème année	Introduction à la programmation		24	24	1996-1998
CNAM	Programmation ADA		50		1998-1999
DEUG A 2ème année	Introduction au C++		15	15	1998-1999
DEUG LEA 2ème année	Environnement informatique		40	40	1998-1999
Licence LEA	Introduction au C		20	20	1998-1999

Activités pédagogiques à l'UCBN avant le LMD

Diplôme	Enseignement	CM	TD	TP	Année
DEUG A 2ème année	Introduction au C++		15	15	1999-2002
Licence d'informatique	Mathématiques pour l'informatique	25	25		1999-2003
DEUG SM	Algorithmique et programmation en pascal	19,5	13	14	1999-2003
DEUG A 2ème année	Programmation orientée en java		30	30	1999-2003
M1 neurosciences	Bases de données	12	15	15	1999-2003
Licence pro webmestre	Introduction à la sécurité	6	6		1999-2003
DESS NAPI	Algorithmique en java	6	4	11	1999-2003
DESS RADI	Transactions sécurisées	9	11		1999-2003
DESS RADI	Graphe du web	3	3		2000-2003

Activités pédagogiques à l'UCBN après le LMD

Diplôme	Enseignement	CM	TD	TP	Année
L1 Info	Introduction à la programmation		19,5	19,5	2006-2012
L2 Info	Mathématiques pour l'informatique	20	30		2005-2011
L2 Info	Structures de données et algorithmique		26	13	2012-2015
L2 Info	Programmation orientée objet		19,5	19,5	2013-2015
L3 Info	Mathématiques discrètes	20	30		2006-2014
M1 Info	Probabilités et statistiques	10	10		2011-2015
M1 Info	Tests et modèles aléatoires	20	20		2011-2015
M2 AMI	Structures aléatoires et analyse d'algorithmes	8			2006-2012
M2 e-SECURE	Outils algorithmiques pour la cryptographie	10			2013-2015

Chapitre 3

Activités d'administration

3.1 Activités liées à l'enseignement

Commission pédagogique

2000-2010 Vice-président de la commission pédagogique
2000-2005 Coordinateur des enseignements d'informatique en DEUG MIAS et SM
2003-2005 Participation à la mise en place de la licence d'informatique lors du passage au LMD

Responsabilités au sein du département d'informatique

- Directeur adjoint du département d'informatique
- Responsable des formations (remplace la commission pédagogique)
- Responsable de la licence d'informatique (directeur des études)
- Porteur des maquettes de la licence d'informatique 2012-2017 et 2017-2022

Conseil de l'UFR
Je suis fortement impliqué dans l'animation pédagogique à l'UFR sciences. J'ai été élu au conseil de l'UFR en mars 2011 et réélu en mai 2015. Je participe à tous les conseils (environ un par mois). J'ai participé à une réunion préparatoire à la fusion entre l'UFR de Sciences et l'IBFA avec des représentants des deux composantes. J'ai aussi participé à des discussions avec la direction du département de mathématiques car nos deux départements doivent se regrouper.

3.2 Activités liées à la recherche

Direction de thèses
co-encadrement à 50 % avec Jacques Madelaine de la thèse de Cyril Bazin intitulée ≪Tatouage de données géographiques et généralisation aux données devant préserver des contraintes≫, soutenue en janvier 2010. Après avoir occupé un postdoc à l'UCBN

dans le framework Sydony, il travaille depuis 2014 à DATEXIM, une start-up dans le traitement, l'analyse et la visualisation d'images médicales où il dirige l'équipe R&D.

Depuis mon arrivée début 2014 dans l'équipe Monétique et Biométrie, je co-encadre deux thèses en Biométrie avec Christophe Rosenberger et Christophe Charrier comme codirecteurs.

Je co-encadre à 33% la thèse de Benoit Vibert intitulée *Evaluation d'algorithmes de comparaison embarquée sur carte à puce (Match-On-Card)*. La soutenance est prévue fin 2016.

J'ai co-encadré à 33% la thèse de Zhigang Yao intitulée *Evaluation de la qualité des empreintes digitales*. Thèse soutenue en juillet 2015.

Collaborations nationales

2009-2013 membre de l'ANR programme blanc Boole, responsable du site de Caen
2004-2008 membre de l'ACI sécurité Tadorne, responsable du site de Caen
2002-2004 membre de l'AS, Nouveaux modèles de calcul
2001-2004 membre de l'ACI Cryptologie

Collaboration internationales

2013-2014 DYNALCO, projet de collaboration STIC-AmSud avec l'Argentine et l'Uruguay
2009-2012 projet de coopération ECOS-sud avec L'Uruguay

Groupes de recherche

- membre du GDR IM (Informatique mathématique)
- membre du groupe de travail Aléa du GDR IM
- membre du groupe de travail C2 (Codage et Cryptographie) du GDR IM
- membre du groupe international AofA (Analysis of Algorithms)

Evaluation de la recherche

- expertise d'une ANR programme blanc en 2012
- expertise d'une ACI en 2005
- rédaction de plus d'une trentaine de rapports pour des conférences et des journaux internationaux

Organisation de conférences

- membre du comité d'organisation de la conférence Numbers, Sequences, Lattices : Dynamical Analysis of Algorithms à Caen en juin 2010.
- co-organisateur des journées Aléa en 2005 au CIRM.
- membre du comité d'organisation des rencontres internationales d'analyse d'algorithmes AOFA2001 Tatihou, France
- membre du comité d'organisation de l'école Jeunes chercheurs en algorithmique et calcul formel à Caen (2000)

Organisation de séminaire

2001-2003 Responsable du séminaire hebdomadaire Algorithmique du GREYC.

Jurys de thèse

janvier 2010 membre du jury de la thèse de Cyril Bazin (co-direction à 50%)

juillet 2015 membre du jury de la thèse de Zhigang Yao (co-encadrement à 25%)

Délégation CNRS [septembre 2003- août 2005]

J'ai obtenu deux années de délégation CNRS au LIPN. Cela m'a permis de collaborer avec plusieurs membres du LIPN et d'élargir mes domaines de compétences en algorithmique et en combinatoire.

Implication dans le laboratoire

2007-2011 élu au conseil du laboratoire du GREYC.

2005-2008 membre extérieur de la commission de spécialistes du LIPN, université Paris XIII.

2004-2008 membre de la commission de spécialistes du GREYC.

Chapitre 4

Activités de recherche

4.1 Encadrement en recherche

L'encadrement d'étudiants de master, de doctorants ou de postdoctorants est une des activités les plus intéressantes d'un enseignant-chercheur. Plus que les résultats de recherches auxquelles je peux ainsi contribuer, c'est la formation d'un chercheur qui me semble le plus passionnant. Par exemple, pour un doctorant, il s'agit non pas de le diriger pour qu'il fasse le travail que l'on aurait fait, mais lui donner les moyens pour qu'il puisse réaliser la thèse correspondant à ses capacités et ses envies, tout en vérifiant bien sûr que les objectifs du sujet de thèse soient atteints.

4.2 Encadrements de thèse

Thèse de Cyril Bazin, soutenue en 2010
Ma première expérience avec Cyril Bazin fut à ce niveau particulièrement satisfaisante. Je participais à une ACI sécurité intitulée *Tadorne* sur le tatouage de données structurées notamment sur des données géographiques vectorielles. Ce projet impliqué les laboratoires *Cedric* du Cnam-Paris, le *COGIT* de l'IGN, le *Lamsade* de l'université Paris-Dauphine et le *GREYC*. C'était un projet très innovant, car contrairement aux images et aux données vidéos, peu d'études avaient été faites pour le tatouage de données structurées. Avec Jacques Madelaine, maître de conférences au GREYC, nous avons eu l'idée d'un sujet sur un tatouage aveugle (sans connaissance du document original) et robuste à des transformations naturelles comme la rotation et le découpage. J'ai défendu ce sujet devant la commission de la recherche du conseil régional, il a été classé premier sur l'ensemble des thèses proposées toutes disciplines confondues. C. Bazin a obtenu dès sa première année des résultats très intéressants en utilisant la triangulation de Delaunay sur des documents géographiques et en introduisant un biais statistique sur des caractéristiques de ces triangles. Il a ensuite eu plus de difficultés à formaliser son approche et à s'abstraire du type de données considéré afin d'avoir des résultats plus génériques. C'est sur ces deux aspects –la formalisation et la généralisation de son travail– que mon apport a été le plus important.

Thèse de Zhigang Yao, soutenue 21 juillet 2015
Depuis mon arrivée dans l'équipe Monétique et Biométrie en janvier 2014, je co-encadre à hauteur de 25% la thèse de Zhigang YAO avec C. Rosenberger et C. Charrier. Nous travaillons sur l'évaluation de mesures de la qualité des empreintes digitales. Cette évaluation s'effectue sur des templates (ensemble de minuties) sans avoir accès à la donnée biométrique originale, cela nécessite une notion de qualité différente de celle liée à la perception humaine. La mesure de qualité doit aussi être robuste, le plus possible invariante par rapport aux logiciels utilisés pour les expérimentations (enrôlement, matching). Certaines méthodes utilisent la triangulation de Delaunay, ce qui permet de faire le lien avec la thèse de Cyril Bazin où la notion de préservation de qualité était aussi un aspect très important. Depuis que je co-encadre cette thèse, j'ai cosigné plusieurs articles, j'ai également fait un exposé à la conférence ISBA à Hong Kong en mars 2015 pour présenter un de ses articles. Je participerai au jury de sa thèse.

Thèse de Benoit Vibert
Je co-encadre aussi depuis mon arrivée dans ma nouvelle équipe de recherche la thèse de Benoit Vibert avec C. Rosenberger et C. Charrier. Il travaille sur la sélection de minuties pour construire un template de taille fixée, limitation nécessaire par exemple pour MOC (Match On Card). Il s'intéresse également aux attaques. Enfin, il implémente les nouvelles fonctionnalités dérivées de son travail dans la plate-forme Evabio développée dans l'équipe. La thèse devrait être soutenue avant décembre 2016.

4.3 Encadrement de postdoctorants

Le terme postdoctorant signifie ici une personne ayant soutenue sa thèse et qui n'a pas encore obtenu de position permanente, typiquement une personne ayant un poste d'ATER ou ayant un contrat postdoctoral. Il ne s'agit pas de fournir un encadrement comme pour un étudiant en master ou en thèse, mais plutôt de faire partager mon expérience en recherche, communiquer mes méthodes de travail et les former à mes domaines de recherche. Ces expériences sont très enrichissantes, j'ai contribué non pas à initier à la recherche comme pour un doctorant, mais à compléter et à renforcer une formation de chercheur pour des docteurs ayant travailler dans d'autres domaines plus ou moins connexes.

Collaboration franco-uruguayéenne
Depuis 2009, dans le cadre d'un projet ECOS franco-uruguayen, nous co-encadrons, Alfredo Viola et moi, un étudiant uruguayen Nicolas Carrasco. Celui-ci a exposé nos travaux sur la génération aléatoire à la conférence internationale ITW au Brésil en octobre 2011. Carrasco est venu fin 2011 un mois à l'université de Caen pour travailler sur ces sujets. Nous avons publié tous les trois en 2013 un article dans le journal TCS (Theoretical Computer Science).

Pendant l'année 2014-2015, Alfredo Viola a proposé un projet annuel de deux étudiants, Sebastián Foncesa et Maria Cecilia Garcia pour programmer des méthodes

d'énumérations de fonctions sans corrélation de petit poids de Hamming en implémentant des algorithmes que j'ai définis.

Projets et stages de master
J'ai déjà encadré de nombreux projets et stages de master. Actuellement, les projets annuels en M1 et en M2 sont particulièrement bien adaptés pour initier les étudiants à mes activités de recherche. J'ai cette année 2014-2015 encadré quatre projets en M1 et M2 sur des sujets intéressant mon équipe de recherche et pour deux d'entre-eux –sur les triangulations de Delaunay pour les empreintes digitales– les résultats vont être utilisés pour les travaux de thèse de Benoit Vibert.

4.4 Publications

4.4.1 Choix des publications

Les habitudes de publications sont vraiment très différentes d'une discipline à une autre et même pour une même discipline, d'un domaine à un autre. Je sais, par exemple, qu'en mathématiques, les conférences ne jouent pas un rôle très important en terme de diffusion des résultats. En informatique, dans certains domaines (comme la cryptographie) les meilleurs publications se font dans des conférences. J'ai choisi la conférence LICS (Logic In Computer Science) pour mes deux meilleurs résultats en logique (théorie des modèles finis) ([16] et [17]). La conférence est de rang A* dans CORE (Computing Research and Education Association of Australasia) le site de classification des publications en informatique le plus réputé. Il n'existe pas de revue en logique plus réputée et avec une audience plus large. Grâce à ces publications, les meilleurs chercheurs du domaine comme Vardi, Gurevich, Kolaitis ont pu découvrir mes résultats et me signifier qu'ils appréciaient mon travail. C'est pourquoi j'estime que ces deux publications ont la même valeur que des revues de rang A*.

En revanche, d'autres conférences comme ISIT ([14]) et ITW ([11]) ont également une très large audience, mais le format (6 pages) est insuffisant pour développer les parties techniques. C'est pourquoi nous avons dans les deux cas publié une version étendue dans les revues Transaction in Information Theory ([4], classée rang A* par CORE) et Theoretical Computer Science ([2], classé rang A par CORE)

4.4.2 Sélection de cinq publications significatives

1. J-M. Le Bars. Counterexamples of the 0-1 law for fragments of existential second-order logic : an overview. Bulletin of Symbolic Logic, 9 :67–82, 2000. impact facteur 0,917

   Il s'agit d'un survey destiné à une large communauté en logique qui reprend les contre-exemples de lois 0-1 en logique existentielle du second-ordre, en particulier mes résultats principaux de thèse. Ceux-ci ont permis de résoudre les derniers problèmes ouverts sur le sujet et également de fournir un unique contre-exemple unifiant tous les résultats sur ce sujet. Le journal *Bulletin of Symbolic Logic* est dédié aux articles de haut niveau dont le sujet est susceptible d'intéresser un

large public en logique mathématique, en logique philosophique, en histoire de la logique et en philosophie des mathématiques. Il fait partie du TOP 5 des revues en logique.

2. J-M. Le Bars. The 0-1 law fails for frame satisfiability of propositional modal logic. In Proceedings of the 17th IEEE Symposium on Logic in Computer Science, 2002 , 225-234 Impact facteur 1,79 CORE : rang A*

   J'ai étendu mes résultats de contre-exemples de lois 0-1 à des logiques pour l'intelligence artificielle. J'ai ainsi réfuté un résultat connu établi par Halpern et Kapron en 1994.

3. C. Bazin, J-M. Le Bars et J. Madelaine A Blind, Fast and Robust Method for Geographical Data Watermarking ACM Symposium on Information, Computer and Communications Security (ASIACCS'07), Singapore, ACM SIGSAC, 2007, 265-272 Rang B

   Nous proposons dans cet article un algorithme de tatouage aveugle qui respecte la précision et la topologie des documents géographiques vectoriels . Cet algorithme découpe un document en sites définis à partir de la triangulation de Delaunay. Nous avons prouvé expérimentalement que cet algorithme résiste à des transformations naturelles telles que la rotation et le découpage.

4. J-M. Le Bars and A. Viola. Equivalence classes of Boolean functions for first-order correlation. IEEE Transactions on Information Theory, 56 (3), 1247 -1261, 2010. Impact factor 2,6 CORE Rang A*

   Cet article de journal reprend les premiers résultats que nous avons eus avec Viola sur les fonctions booléennes, celles-ci sont manipulées par classes d'équivalence, ce qui nous permet de compter et énumérer toutes les fonctions 1-résilientes jusqu'à 7 variables.

5. B Z. Yao, J.M. Le Bars, C. Charrier, C. Rosenberger, "Quality Assessment of Fingerprints with Minutiae Delaunay Triangulation", International Conference on Information Systems Security and Privacy (ICISSP), 2015 (taux de sélection : 20%).

   Je co-encadre (25 %) la thèse de Zhigang YAO avec C. Rosenberger et C. Charrier. Il travaille sur l'évaluation de la qualité d'empreintes digitales. Dans ce papier, nous proposons une méthode permettant de qualifier la qualité d'une empreinte digitale uniquement à partir de l'ensemble des minuties avec une représentation basée sur une triangulation de Delaunay. Cette méthode est la première méthode de l'état de l'art ne traitant que des minuties en entrée (sans avoir accès à l'image de l'empreinte digitale).

4.5 Liste des publications

Thèse

[1] J-M. Le Bars Probabilités asymptotiques et pouvoir d'expression des fragments de la logique du second ordre, UCBN, janvier 1998.

Revues internationales avec comité de lecture

[2] Yao, J-M Le Bars, C. Charrier and C. Rosenberger, A Literature Review of Fingerprint Quality Assessment and Its Evaluation, To IET Biometrics journal,

[3] J-M Le Bars et A. Viola. Enumerative encoding of correlation-immune. Theoretical Computer science, 487, 23-36, 2013

[4] J-M. Le Bars et A. Viola. Equivalence classes of Boolean functions for first-order correlation. IEEE Transactions on Information Theory, 56 (3), 1247 -1261, 2010.

[5] J-M. Le Bars. The 0-1 law fails for the monadic existential second-order logic on undirected graphs Information Processing Letters, 77/43-48, 2001.

[6] J-M. Le Bars. Counterexamples of the 0-1 law for fragments of existential second-order logic : an overview. Bulletin of Symbolic Logic, 9 :67–82, 2000.

Actes de colloques internationaux avec comité de lecture

[7] Z. Yao, J. Le bars, C. Charrier, C. Rosenberger, Pixel Pruning for Fingerprint Quality Assessment, NIST International Biometric Performance Testing Conference (IBPC), 2016.

[8] Z. Yao, J-M Le Bars, C. Charrier and C. Rosenberger, Fingerprint Quality Assessment With Multiple Segmentation, workshop on Biometric Security of the international conference on Cyberworlds, 7-8 october 2015..

[9] M. Barbier, J-M Le Bars, C. Rosenberger, Image Watermaking With Biometric Data For Copyright Protection, workshop MFESC of the International Conference on Availability, Reliability and Security (ARES) Toulouse, France, 24-28 august 2015

[10] B. Vibert, J-M Le Bars, C. Charrier, C. Rosenberger, Comparative study of minutiae selection algorithms for ISO fingerprint templates, SPIE electronic imaging, 2015

[11] B. Vibert, J-M Le Bars, C. Charrier, C. Rosenberger, EvaBio Platform for the Evaluation Biometric System - Application to the Optimization of the Enrollment Process for Fingerprints Devices, ICISSP 2015

[12] Z. Yao, J-M Le Bars, C. Charrier and C. Rosenberger, Quality Assessment of Fingerprints with Minutiae Delaunay Triangulation, International Conference on Information Systems Security and Privacy (ICISSP 2015)

[13] Z. Yao, J-M Le Bars, C. Charrier and C. Rosenberger, Fingerprint Quality Assessment Combining Blind Image Quality, Texture and Minutiae Features, INSTICC 2015.

[[14] N. Carrasco, J-M Le Bars and A. Viola. Enumerative encoding of correlation-immune Boolean functions. IEEE Information Theory Workshop, Paraty, Brésil, 2011, 643-647.

[15] C. Bazin, J-M. Le Bars et J. Madelaine A Novel Framework For Watermarking : The Data-Abstracted Approach International Workshop on Security, IWSEC, 2008, 201-217.

[16] C. Bazin, J-M. Le Bars et J. Madelaine A Blind, Fast and Robust Method for Geographical Data Watermarking ACM Symposium on Information, Computer

and Communications Security (ASIACCS'07), Singapore, ACM SIGSAC, 2007, 265-272.
[17] J-M. Le Bars and A. Viola. Equivalence classes of boolean functions for first-order correlation. 2007 IEEE International Symposium on Information Theory (ISIT 2007), 181-186.
[18] C. Banderier, J-M. Le Bars, V. Ravelomanana. Generating Functions For Kernels of Digraphs (Enumeration & Asymptotics for Nim Games) In Proceedings of the 16th Annual International Conference on Formal Power Series and Algebraic Combinatorics (VancouversBC, Canada, June 28 - July 2 2004), 91-105.
[19] J-M. Le Bars. The 0-1 law fails for frame satisfiability of propositional modal logic. In Proceedings of the 17th IEEE Symposium on Logic in Computer Science, 2002, 225-234.
[20] J-M Le Bars, Fragments of Existential Second-Order Logic without 0-1 Laws. LICS 1998, 525-536.

Conférence nationale avec comité de lecture

[21] B. Vibert, J.M. Le Bars, C. Charrier, C. Rosenberger, "Définition du type d'empreinte à partir d'un template ISO Compact Card II", Colloque COmpression et REprésentation des Signaux Audiovisuels (CORESA), 2016
[22] M. Barbier, J-M Le Bars, C. Rosenberger, Image Watermaking With Biometric Data For Copyright Protection, APVP 2015 (Atelier sur la Protection de la Vie Privée), juin 2015, Mosnes.

4.5.1 Prix et distinctions

Prix international Kleene Award, prix du meilleur article étudiant, à la conférence internationale Logic in Computer Science (LICS'98) pour l' article *Fragments of existential second-order logic without 0-1 laws* en 1998.

Prix national Le jury du Prix de Thèse SPECIF 1998, présidé par Gilles Kahn de l'Académie des Sciences, m'a attribué un des deux accessits pour ma thèse.

4.6 Projet de recherche

Mon projet de recherche porte sur l'extraction d'aléatoire sur des données réelles pour des applications de différents domaines comme la cryptographie, la biométrie ou le tatouage. La modélisation aléatoire de données est souvent extrêmement complexe et parfois irréalisable en pratique, mais heureusement elle n'est pas nécessaire pour la plupart des applications. En effet, on souhaite pouvoir mettre en évidence des parties aléatoires sans avoir à fournir une modélisation complète. Mon objectif est double :

– d'une part, de fournir un cadre général (framework) et de pouvoir formaliser ce nouveau type d'étude. Il s'agit, en particulier, d'identifier les aspects généraux et ceux qui dépendent des applications envisagées. La constitution de ce framework devra probablement nécessiter quelques années.

– d'autre part, diverses pistes pourront être explorées en proposant des recherches expérimentales, plus pratiques et ciblées, à des étudiants en thèse ou en master.

Troisième partie

Étude des noyaux dans les graphes aléatoires

Chapitre 1

Introduction

Il existe de nombreuses définitions d'un noyau sur les graphes. Nous considérons ici une définition issue de la théorie des graphes que l'on peut trouver dans des articles de Claude Berge, l'un des fondateurs de ce domaine [4, 5].

Un noyau est un sous-ensemble de l'ensemble des sommets d'un graphe orienté qui est à la fois stable et dominant.

Nous allons voir dans cette partie que cette propriété très simple à définir intervient pour des études dans des domaines différents et complémentaires de l'informatique mathématique, en particulier en algorithmique, en combinatoire et en complexité. Toutes les études proposées porteront sur des graphes construits aléatoirement avec différentes distributions.

1.1 Contexte

L'objectif de ce chapitre est de donner une présentation non technique des principaux résultats.

Le chapitre 2 apporte une contribution en logique, dans les domaines de la théorie des modèles finis et des logiques modales. Il reprend les variantes du noyau que j'ai utilisées comme contre-exemple de loi 0-1 lors de ma thèse et des années qui ont suivies. Je montre que cette propriété joue un rôle central sur ce sujet, j'ai, de plus, pu réfuter un résultat très connu en logiques modales.

La chapitre 3 apporte des détails sur les calculs asymptotiques en jeu. Il permet de comprendre comment les variantes de la propriété de noyau suivent des comportements asymptotiques différents. Outre leur utilisation pour les lois 0-1, ces variantes fournissent des résultats en transition de phase. Des résultats pour d'autres distributions sont également considérés, dans les graphes creux (nombres linéaires d'arcs par rapport au nombre de sommets).

Le chapitre 4 aborde les aspects algorithmique et de complexité et montre le lien avec la théorie des jeux. Il contient des algorithmes pour rechercher un noyau sur les arbres, les graphes orientés sans circuit (DAG) et les graphes aléatoires du modèle d'Erdös et Rényi. La proportion en moyenne de sommets dans le noyau pour les arbres étiquetés et non étiquetés et pour les graphes proches des arbres est calculé, on utilise pour cela des méthodes issues de la combinatoire analytique.

La complexité de la recherche d'un noyau que l'on a caché dans un graphe aléatoire est aussi envisagé. Une application en cryptographie est ensuite proposée, elle repose sur un protocole zero-knowledge de la connaissance d'un noyau dans un graphe.

1.2 Contributions

Le chapitre 2 contient mes premières études de la propriété de noyau abordés pendant ma thèse [41]. Je vais m'attacher à montrer la démarche que j'ai suivie à ce moment-là.

Je me suis intéressé pour la première fois à cette propriété pendant mon stage de DEA avec Etienne Grandjean. Le sujet portait sur les lois 0-1, une des thématiques majeures de la théorie des modèles finis.

Mon travail de stage de DEA était essentiellement de fournir un état de l'art sur ce sujet et de cerner les problèmes ouverts. J'ai poursuivi ensuite en thèse, toujours avec E. Grandjean, pour m'attaquer à ces problèmes.

En théorie des modèles finis, les structures sont finies, elles sont basées sur un domaine à n éléments, on dit alors qu'elles sont de taille n, où $n \in \mathbb{N}$. Elles sont de plus formées sur un vocabulaire, c'est-à-dire un ensemble de relations qui peuvent avoir différentes arités. De ce point de vue, un graphe orienté est une structure très simple où le domaine est l'ensemble des sommets et le vocabulaire contient une seule relation binaire sans boucle.

Pour une propriété $\mathcal{P}$ sur ces structures, on calcule, pour chaque $n \in \mathbb{N}$, $\mu_n(\mathcal{P})$, la probabilité qu'une structure de taille n tirée aléatoirement avec la distribution uniforme satisfasse la propriété $\mathcal{P}$ (on peut imaginer que l'on met dans une urne toutes les structures de taille n et que l'on tire une structure au hasard). $\mathcal{P}$ possède une probabilité asymptotique lorsque $\mu_n(\mathcal{P})$ a une limite $\mu(\mathcal{P})$ lorsque n tend vers l'infini. On distingue les valeurs 0 et 1 comme probabilité asymptotique. Nous dirons que $\mathcal{P}$ est *a.p.s.* (asymptotiquement presque sûrement) vraie lorsque sa probabilité asymptotique vaut 1 et $\mathcal{P}$ est *a.p.s.* fausse lorsque sa probabilité asymptotique vaut 0. Intuitivement, si une probabilité est *a.p.s.* vraie, nous avons toutes les chances de tirer dans l'urne une structure de taille n vérifiant la propriété (pour n assez grand) et, inversement, si une probabilité est *a.p.s.* fausse, nous avons toutes les chances de tirer dans l'urne une structure de taille n ne vérifiant pas la propriété.

Une logique $\mathcal{L}$ vérifie une loi 0-1 *lorsque toute propriété exprimable dans cette logique est soit a.p.s. vraie, soit a.p.s. fausse.*

Je me suis intéressé à des fragments très restreints de ESO, la logique existentielle du second ordre afin de déterminer des lois 0-1. Une première restriction est obtenue

en considérant uniquement des relations unaires pour la partie existentielle, cela correspond à la logique MESO, la logique monadique existentielle du second ordre. Un graphe vérifie une formule de MESO lorsqu'il existe un sous-ensemble des sommets qui vérifie une formule du premier ordre.

L'étude de la propriété de noyau s'est imposée car, d'une part, celle-ci s'exprime dans des fragments très restreints comme MESO_2, les formules de MESO n'ayant que deux variables au premier ordre ; d'autre part, le calcul de sa probabilité asymptotique s'avère particulièrement difficile à établir (nous verrons plus loin qu'elle est *a.p.s.* vraie).

Le résultat majeur de ma thèse a été de concevoir une variante de la propriété de noyau qui ne possède pas de probabilité asymptotique. Ce fut un résultat remarqué car, d'une part, cette variante fournit un contre-exemple de loi 0-1 aux fragments pour lesquels nous ne savions pas s'il y avait une loi 0-1 et, d'autre part, elle permet de retrouver les autres fragments sans loi 0-1. Notons que cette propriété n'est pas apparue par hasard. En effet, après avoir beaucoup étudié le pouvoir d'expression de MESO_2, je pense que seules des variantes de cette propriété pouvaient servir de contre-exemples de loi 0-1.

La démarche a surpris car elle était très différente de celle suivie lors des études précédentes. A chaque fois, il s'agit de relier des aspects logiques (connaître le pouvoir d'expression de ces logiques en définissant quels types de propriétés peuvent être exprimés dans ces logiques) à des aspects asymptotiques (calcul des probabilités asymptotiques). Cependant l'approche a été inversée ; en effet, auparavant les chercheurs du domaine recherchaient à exprimer dans ces logiques des propriétés qui n'avaient clairement pas de probabilité asymptotique, par exemple le fait que le nombre de sommets est pair (la probabilité asymptotique passe alternativement de 0 à 1 lorsque le nombre de sommets varie), alors qu'ici je suis parti d'un comportement asymptotique que je souhaitais atteindre et j'ai ensuite cherché une propriété ayant ce comportement asymptotique.

Voici en quelques mots le cheminement suivi. Soit n le nombre de sommets du graphe orienté, il existe un sous-intervalle I_n de $\mathcal{R}$ dépendant de n, mais dont la taille ne dépend pas de n, tel que presque tous les graphes vérifient la propriété suivante : il n'existe pas de noyau avec une taille en dehors des entiers de I_n et, pour tout entier de I_n, il existe au moins un noyau de cette taille. Partant de ce résultat, l'idée a été de concevoir une variante avec un intervalle J_n ayant les mêmes propriétés que I_n mais avec une taille inférieure à 1. On vérifie que selon la valeur de n, J_n contient soit 0 soit 1 entier et, par conséquent, nous avons soit 0 soit 1 taille possible pour les noyaux. Plus précisément, pour obtenir un résultat asymptotique, on montre qu'il existe une sous-suite de $\mathbb{N}$ pour laquelle J_n ne contient pas d'entier et une autre sous-suite de $\mathbb{N}$ pour laquelle J_n contient un entier. Il ne s'agit pas ici d'avoir une approche basée sur le pouvoir d'expression, car c'est le calcul asymptotique et non le sens donné à la propriété qui détermine la probabilité asymptotique. Je n'ai pas essayé de calculer la probabilité asymptotique d'un grand nombre de variantes, mais plutôt de façonner une variante jusqu'à obtenir le comportement asymptotique souhaité. On obtient de cette manière une propriété un peu artificielle, mais, à ma connaissance, il n'existe

pas de propriétés naturelles avec un tel comportement.

Le résultat a également surpris car il est a l'opposé de ce qui était attendu. En effet, de nombreuses spécialistes du domaine avaient essayé de montrer, sans succès évidemment, que MESO_2 avait une loi 0-1.

C'est ce qui a été remarqué par la communauté. En revanche, l'originalité de ce travail –le fait que les aspects logiques et aspects asymptotiques sont extrêmement intriqués– a été moins apprécié car les chercheurs du domaine –à quelques exceptions près– s'intéressent davantage à développer des techniques en logique qu'en asymptotique.

Ma contribution sur ce sujet ne s'est pas arrêtée à mes travaux de thèse. Valentin Goranko m'a proposé de travailler sur la validité des frames en logique modale. J'avais déjà lu leur article sur les frames en logique modale de Joseph Halpern et Bruce Kapron. Leur article contenait notamment un résultat de loi 0-1 pour la validité des frames. Celui-ci nécessitait une preuve compliquée et peu claire. C'est en travaillant avec Goranko que j'ai pu me familiariser un peu avec le domaine des logiques modales. Goranko avait discuté de cet article avec Moshe Vardi, spécialiste mondialement reconnu en théorie des modèles finis et des logiques modales. Cette logique étant assez proche de MESO_2, Vardi doutait, après que mon résultat ait été diffusé, de la justesse de leur résultat. A ce moment-là, je projetais de diversifier mes compétences en algorithmique et je souhaitais pour cela travailler dans d'autres domaines que les lois 0-1, mais Goranko m'a finalement convaincu de rechercher un contre-exemple. La principale difficulté a été de connaître les propriétés que je pouvais exprimer en logique modale, car les aspects logiques et asymptotiques sont fortement intriqués. Ce fut encore plus le cas ici que pour les contre-exemples obtenus en thèse, l'issue était vraiment très incertaine et je reconnais avoir eu de la chance d'y parvenir. Goranko m'a aussi apporté une aide efficace pour trouver de nouvelles variantes. Le processus qui mène à l'obtention d'un contre-exemple est assez difficile à décrire, c'est pour cette raison que dans mes articles la propriété choisie peut sembler tombée du ciel.

Le chapitre 3 reprend les méthodes de calculs nécessaire pour la propriété de noyau. Nous commençons par définir les modèles de graphes aléatoires. Erdös et Rényi sont les fondateurs de la théorie des graphes aléatoires, ils ont lancé en 1959 les bases de cette théorie dans leur célèbre article « On the evolution of random graphs » [16]. Ils considèrent dans cet article le modèle $\mathcal{G}(n, M)$ et étudient les fonctions seuil associées à diverses propriétés telles que l'apparition d'un sous-graphe fixé, la décomposition en arbres, le nombre de cycles, la croissance de la plus grande composante connexe.

Depuis de nombreux travaux ont été mené, le lecteur soucieux de parfaire sa connaissance dans ce domaine aura l'embarras du choix, il pourra en particulier consulter les livres suivants [6, 15, 30].

Edgar Gilbert fut le premier a introduire en 1959 [23] sur les graphes non-orientés le modèle $\mathcal{G}(n, p)$ pour l'étude des phénomènes de seuil pour la connexité.

Erdös et Rényi ont défini indépendamment et au même moment, toujours sur les graphes non-orientés, le modèle $\mathcal{G}(n, M)$ pour étudier le même problème.

Un graphe non-orienté G_n est constitué d'un ensemble V_n de sommets et d'un

– U est un sous-ensemble **stable** (on dit aussi ensemble indépendant) lorsque aucun couple de sommets U, (a, b) n'est un arc de D_n.
– U est **dominant** lorsque tout sommet du complémentaire de U dans V_n peut atteindre par un arc de D_n un sommet de U.
– U est un **noyau** lorsqu'il est à la fois un ensemble stable et dominant.

FIGURE 1.1 – Définition d'un noyau

ensemble E d'arêtes (on suppose pour nos études qu'il n'y a pas de boucle). Soit $p = p(n)$, $0 < p < 1$, G_n est construit avec le modèle $\mathcal{G}(n, p)$ en mettant une arête avec probabilité p pour chaque paire de sommets. La valeur $p = 1/2$ cela correspond à la distribution uniforme, ou équiprobabilité, où tout graphe a la même probabilité d'être construit. Généralement, nous obtenons des résultats très similaires lorsque nous prenons n'importe quelle constante p. En revanche, les résultats sont différents si l'on prend $p = p(n)$. Le modèle $\mathcal{G}(n, M)$ qui donne des résultats similaires. Pour ce modèle, on tire aléatoirement M arêtes parmi toutes les arêtes.

Nous pouvons reprendre ces modèles sur les graphes orientés. Soit $D_n = \langle V_n, A\rangle$ un graphe orienté où V_n est toujours l'ensemble des sommets et A l'ensemble des arcs. On remplace paire de sommet $\{a, b\}$ par couple de sommets (a, b), donc pour le modèle $\mathcal{D}(n, p)$ nous avons $n(n-1)$ essais de Bernoulli de paramètre p et pour $\mathcal{D}(n, M)$ les M arcs sont choisis parmi les $n(n-1)$ couples.

Beaucoup d'autres modèles peuvent être définis, en particulier lorsque nous ne voulons pas l'indépendance pour le placement des arêtes.

Nous considérerons dans nos études uniquement le modèle $\mathcal{D}(n, p)$ parmi ces deux modèles, mais des résultats similaires peuvent être obtenus avec le modèle $\mathcal{D}(n, M)$.

Pour un modèle fixé et une propriété $\mathcal{P}$, $\mu_n(\mathcal{P})$ est la probabilité de construire aléatoirement avec ce modèle un graphe satisfaisant $\mathcal{P}$. Contrairement à la définition donnée pour l'étude des lois 0-1 en théorie des modèles finis, toutes les structures n'ont pas forcément la même probabilité d'être tirée, cela dépend du modèle considéré. Comme précédemment, une logique admet une loi 0-1 lorsque toute propriété exprimable dans cette logique est soit *a.p.s.* vraie, soit *a.p.s.* fausse.

Les trois propriétés que j'ai considéré pour l'étude des noyaux, portent sur les sous-ensembles de V_n que nous allons étudier sont Stable, Dominant et Noyau.

Pour montrer qu'une propriété est *a.p.s.* vraie ou fausse, on utilise la méthode des moments. La méthode du premier moment permet de montrer que la propriété est *a.p.s.* fausse et la méthode du second moment qu'elle est *a.p.s.* vraie. En fusionnant les résultats de Wenceslas De la Vega [10] et de Ioan Tomescu [52] publiés indépendamment, j'ai montré [45] que la propriété NOYAU est *a.p.s.* vrai sur $\mathcal{D}(n, p)$, pour tout p constant. L'ensemble des tailles possibles est fini et dépend uniquement de la constante p.

En modifiant la définition de la propriété de NOYAU, on change le poids entre ces trois propriétés. C'est ce qui m'a permis d'obtenir des propriétés sans loi 0-1.

Ces études n'ont pas que des applications en lois 0-1, j'ai également introduit une variante de la propriété de NOYAU qui a une transition de phase avec un seuil abrupt [45].

J'ai plus récemment étudié avec Marco Illengo le modèle $\mathcal{D}(n, p(n))$, où $p(n) = c*n$, pour c fixé. Nous montrons que pour $c \neq e$, la probabilité asymptotique de la propriété NOYAU existe et est différente de 0. Concernant les tailles possibles des noyaux, le résultat est moins précis. On montre qu'il n'y a qu'une seule densité possible. La densité est définie asymptotiquement pour une famille d'ensemble U_n, comme la limite de $d(U_n) = \frac{|U_n|}{n}$. Par exemple, la densité vaut $1/4$ lorsque la proportion de sommets dans U_n tend vers $1/4$ lorsque n tend vers ∞.

Le chapitre 4 s'intéresse aux algorithmes énumérant les noyaux d'un graphe et à la difficulté de trouver un noyau dans un graphe aléatoire. Tout d'abord, notons que la propriété NOYAU est NP-complète [9].

La théorie des jeux offre une première application pour utiliser des algorithmes de recherche de noyau. En effet, le noyau intervient de manière centrale en théorie des jeux pour les jeux à deux joueurs, sans information cachée et où toute partie a un vainqueur (pas de match nul possible). Ernst Zermelo a prouvé en 1913 que pour un tel jeu, un des deux joueurs possède une stratégie gagnante. Ces travaux ont été ensuite repris par John von Neumann et Oskar Morgenstern [46]. Nous appellerons DAG un graphe orienté sans circuit. La relation d'arc d'un DAG forme un ordre partiel. Neumann et Morgenstern ont prouvé que tout DAG possédait un unique noyau. L'ensemble des positions gagnantes dans un jeu forment l'unique noyau d'un DAG dans le cas d'un jeu direct. De nombreux jeux appartiennent à cette famille de jeux à deux joueurs comme le jeu de Nim –qui a été rendu célèbre par le film d'Alain Resnais *L'année dernière à Marienbad*– et ses nombreuses variantes.

Dans un premier temps, nous nous intéressons aux graphes sans circuit (DAG) ou possédant peu de circuits. Ce sont des graphes avec un nombre linéaire d'arcs par rapport au nombre de sommets. Si le graphe est un arbre orienté ou un graphe sans circuit alors nous venons de voir qu'il ne possède qu'un seul noyau. En revanche, s'il possède des circuits il peut avoir $0, 1$ ou plusieurs noyaux. Je propose deux algorithmes linéaires en la taille du graphe (nombre de sommets + nombre d'arcs) qui sont adaptés pour ces graphes. Le premier algorithme nommé *ColoriageArbre* effectue un coloriage avec deux couleurs *noir* et *blanc*. Il colorie en noir tous les sommets atteignables à partir d'un puits (sommet qui ne possède pas d'arc sortant) qui ne sont pas connectés à un circuit. Le second algorithme nommé *ColoriageDAG* effectue un coloriage avec trois couleurs *rouge, vert* et *blanc*, *rouge* pour les sommets du noyau, *vert* pour les sommets hors du noyau et *blanc* pour les sommets qui n'ont pas encore été traités. Si le graphe est un arbre orienté ou un graphe sans circuit (un DAG) alors il ne possède qu'un seul noyau qui est trouvé par ce coloriage (à la fin il n'y a plus de sommet colorié en blanc). Les sommets coloriés correspondent aux positions gagnantes, pour gagner le joueur doit se déplacer ver un sommet colorié en rouge.

Les deux algorithmes *ColoriageArbre* et *ColoriageDAG* se font en plusieurs itérations et ils s'arrêtent lorsque soit tous les sommets sont colorés (il y a un noyau), soit aucun

sommet n'est colorié à la dernière itération (dans les deux cas, cela constitue un point fixe).

Le second algorithme peut aussi casser certains circuits lorsqu'ils ne sont pas trop nombreux, ni trop imbriqués les uns dans les autres.

J'ai travaillé en 2004 avec Cyril Banderier et Vlady Ravelomanana sur sur les graphes proches des arbres. En utilisant des techniques de la combinatoire analytique (utilisation de séries génératrices exponentielles), nous avons montré qu'environ 47% des arbres orientés étiquetés a une racine verte (et donc le premier joueur a une stratégie gagnante), qu'un graphe avec un seul cycle vérifie *a.p.s.* la propriété NOYAU et que $92,65\%$ des graphes ayant un seul circuit possède un noyau.

Nous avons également étudier le comportement de ces deux algorithmes sur les graphes aléatoires de $\mathcal{D}(n, p(n))$ en faisant varier la probabilité d'arc $p(n) = c/n$, où c est une constante.

Nous avons observé une transition sur le nombre de sommets coloriés, sans néanmoins obtenir de résultat théorique pouvant l'expliquer. Pour $c < 1$, l'algorithme COLORIAGEARBRE colore presque toujours tous les sommets et il faut attendre $c > e$ pour qu'une fraction non négligeable de sommets ne soient pas coloriés par COLORIAGEDAG.

Nous avons expliqué ce phénomène avec Marco Illengo lors de notre étude de la probabilité asymptotiquement de la probabilité de NOYAU sur $\mathcal{D}(n, p(n) = c/n)$. La preuve repose sur un système dynamique pour chacun de ces deux algorithmes qui analyse la densité de sommets non coloriés à chaque étape de l'exécution de l'algorithme.

Je me suis aussi intéressé au problème de retrouver un noyau caché dans un graphe aléatoire de $\mathcal{D}(n, p(n))$. La méthode consiste à insérer un noyau d'une des tailles les plus probables dans $\mathcal{D}(n, p(n))$. Ce travail est parti de l'article *Hiding Cliques for Cryptographic Security* d'Ari Juels and Marcus Peinado en 2000[31]. Les auteurs montraient comment cacher une clique dans un graphe aléatoire et proposaient des applications en cryptographie (fonction à sens unique et protocole zero-knowledge).

J'ai commencé à m'intéresser à ce problème dès 2003 et je souhaitais l'adapter au noyau. J'avais obtenu quelques résultats intéressants avec Fabrice Boudot qui venait de soutenir sa thèse sur les protocoles zero-knowledge, mais ce travail restait trop proche de celui de Juels et Peinado et il n'amenait pas vraiment un apport pour le domaine. Ce n'était pas clair de savoir quelle était la meilleure façon de cacher un noyau en restant le plus proche possible de la distribution initiale et il fallait aussi trouver des instances plus difficiles que celles pour les cliques.

Avec Eleonora Guerrini, Marco Illengo et Fabien Laguillaumie, nous avons mis en place un groupe de travail fin 2010 sur ce sujet : cacher un noyau dans un graphe aléatoire.

Ce groupe de travail a également entraîné l'étude de la propriété de noyau sur les graphes creux (graphes avec cn arcs, pour c fixé) pour lesquels nous conjecturons qu'ils fournissent des instances difficiles pour c suffisamment grand.

Soit $\mathcal{D}^*(n, p)$ la distribution obtenue à partir de $\mathcal{D}(n, p)$: on construit D_n de $\mathcal{D}(n, p)$ et on ajoute aléatoirement un noyau K pour obtenir D_n* de $\mathcal{D}^*(n, p)$. Nous montrons

que plus le nombre de noyaux de D_n est proche de la moyenne (espérance du nombre de noyaux sur $\mathcal{D}(n,p)$), plus la probabilité que K soit un noyau dans $\mathcal{D}^*(n,p)$ est proche de celle que K soit un noyau dans $\mathcal{D}(n,p)$. Ce résultat s'applique également aux cliques et apporte une meilleure compréhension du rapport entre les deux distributions que celle apportait par Juels et Peinado dans leur étude. Nous montrons aussi que s'il n'existe pas d'algorithme polynomial avec probabilité de succès significative sur $\mathcal{D}(n,p)$ alors nous avons le même résultat pour $\mathcal{D}^*(n,p)$. Cela permet de garantir la difficulté de retrouver le noyau en effectuant l'étude uniquement sur $\mathcal{D}(n,p)$. Nous introduisons enfin un protocole zero-knowledge de preuve de connaissance d'un noyau qui adapte (quoique plus complexe) celui proposé par Juels et Peinado sur les cliques.

Chapitre 2

Noyau et lois 0-1 en logique

2.1 Définitions et contexte

Cette section se concentre sur les aspects logiques. Je reviendrai sur le calcul des propriétés en jeu (stabilité, dominance et noyau) ainsi que sur les méthodes de calcul dans la partie suivante, je montrerai alors comment utiliser la méthode des moments.

Mes travaux de thèse concernent la logique mathématique et les fondements logiques de l'informatique, ils se situent plus précisément en théorie des modèles finis[41]. De nombreuses recherches ont été menées dans ce domaine pour relier différentes études comme la décidabilité, le pouvoir d'expression et les lois 0-1.

Le sujet de ma thèse était l'étude de propriétés exprimables dans des fragments de ESO (la logique existentielle du second ordre) et de MESO (la logique monadique existentielle du second ordre). Cette thématique est fortement liée à la complexité car l'ensemble des propriétés exprimables dans ESO forme exactement la classe NP. Une logique vérifie une loi 0-1 lorsque toute propriété exprimable dans cette logique possède une probabilité asymptotique (limite de la proportion des structures vérifiant la propriété lorsque la taille de ces structures tends vers l'infini) qui vaut soit 0, soit 1. La classe ESO n'admet pas de loi 0-1, mais certains de ses fragments admettent une loi 0-1. Mon principal résultat de thèse ont été de concevoir une variante de Noyau sans probabilité asymptotique et qui joue un rôle central de contre-exemple aux lois 0-1. En effet elle m'a permis de retrouver les fragments de ESO sans loi 0-1 déjà connus et aussi de résoudre les principaux problèmes restant ouverts [40, 42], en particulier un problème posé par Phokion Kolaitis et Moshe Vardi en 1992 [36] et un autre par Jörg Flum en 1994 [20].

Ma thèse contient également des résultats sur le pouvoir d'expression de fragments de ESO et une adaptation d'un résultat de Matt Kaufmann [32] pour montrer que MESO n'a pas loi 0-1 sur les graphes non-orientés [43], ce résultat n'est pas lié à la propriété de noyau, il adopte la méthode classique qui se base sur le pouvoir d'expression de la logique en exprimant une propriété qui n'a clairement pas de probabilité asymptotique. Ici il s'agit de la parité du nombre de sommets, propriété qui a été utilisée un grand nombre de fois. Mon article de survey [42] reprend les différents contre-exemples, celui de Kolaitis et Vardi de 2000 [37] intégre aussi mes résultats.

Structure finie
Le terme *domaine* désignera un ensemble M d'éléments tous distingués par un étiquetage, il peut être fini ou infini. Un *vocabulaire* désignera un ensemble fini de symboles de relation, nous n'aurons donc ni constante, ni fonction. Une structure $\mathcal{M}$ sur un vocabulaire $\mathcal{R}$ est composée d'un domaine M et d'une interprétation de $\mathcal{R}$, c'est-à-dire pour chaque relation d'arité k de $\mathcal{R}$, on précise quels sont les k-uplets de M qui appartiennent à $\mathcal{M}$. La structure se note $\mathcal{M} = \langle M, \mathcal{R}^M \rangle$. On parle de structure finie lorsque M est fini et lorsqu'il n'y a pas d'ambiguïté, on écrit $\mathcal{M} = \langle M, \mathcal{R} \rangle$.

Nous dirons qu'une propriété $\mathcal{P}$ portant sur des structures $\mathcal{M} = \langle M, \mathcal{R}^M \rangle$ est exprimable dans une logique $\mathcal{L}$ lorsqu'il existe un énoncé $\psi(\mathcal{R})$ de $\mathcal{L}$ formé sur le vocabulaire $\mathcal{R}$ tel que $\mathcal{M}$ vérifie $\mathcal{P}$ si et seulement si $\psi(\mathcal{R})$ est vraie sur $\mathcal{M}$.

Probabilité asymptotique et loi 0-1
$\mu_n(\mathcal{P})$ désignera la proportion de structures $\mathcal{M} = \langle M, \mathcal{R}^M \rangle$ vérifiant $\mathcal{P}$. Cela correspond à la la distribution uniforme ; $\mu_n(\mathcal{P})$ est la probabilité qu'une structure aléatoire formée de la manière suivante vérifie $\mathcal{P}$: pour toute variable de relation R de $\mathcal{R}$ d'arité k et tout k-uplet $(a_1, \ldots, a_k)$ de M, on choisit de mettre avec probabilité $1/2$ le k-uplet $(a_1, \ldots, a_k)$ dans R^M (essai de Bernoulli de paramètre $1/2$). Nous verrons dans la chapitre suivant comment définir d'autres distributions. Nous nous restreindrons dans la présente section à la distribution uniforme (sur toutes les structures) les chercheurs du domaine considèrent principalement cette distribution. Lorsque $\mu_n(\mathcal{P})$ possède une limite quand n tend vers l'infini, on note cette limite $\mu(\mathcal{P})$ et elle est appelée probabilité asymptotique de $\mathcal{P}$. Une propriété $\mathcal{P}$ est asymptotiquement presque sûrement vraie lorsque sa probabilité asymptotique vaut 1 et asymptotiquement presque sûrement vraie lorsqu'elle vaut 0. Pour simplifier l'écriture, nous écrirons par la suite *a.p.s.* au lieu de « asymptotiquement presque sûrement ».

Définition 2.1.1. *Une logique $\mathcal{L}$ vérifie une loi* 0-1 *lorsque toute propriété exprimable dans cette logique est soit a.p.s. vraie, soit a.p.s. fausse.*

Le premier résultat de loi 0-1 a été établi par Carnap en 1950 pour FO, la logique du premier ordre, mais sur un vocabulaire unaire, c'est-à-dire ne contenant que des relations unaires [8].

Structure dénombrable et axiomes d'extension
Haim Gaifman a montré [22] que pour tout vocabulaire fixé $\mathcal{R}$, il existe une unique structure dénombrable $\mathcal{M} = \langle V, \mathcal{R} \rangle$ (à isomorphisme prêt) satisfaisant un ensemble dénombrable d'axiomes d'extension EXT [22]. Richard Radó [49] et Paul Erdös et Alfred Rényi [14] ont effectué une étude similaire sur les graphes.

Ces axiomes d'extension permettent d'exprimer que toute sous-structure à k éléments peut être étendue en une structure à $k+1$ éléments de toutes les façons possibles. Pour k fixé, EXT_k désigne le sous-ensemble de EXT des axiomes d'extension à partir de k éléments.

Loi 0-1 pour FO
Glebskii, Kogan, Liogonki et Talanov ont montré en 1969 que FO avait une loi 0-1 quel que soit le vocabulaire [24], mais la diffusion de ce résultat est restée très confidentielle. En 1976, Ronald Fagin a obtenu indépendamment le même résultat avec une approche remarquable [18]. Il montre en effet que les trois assertions suivantes sont équivalents (transfert de Fagin) :

- φ est *a.p.s.* vraie.
- φ est vraie sur la structure dénombrable.
- φ est conséquence d'un nombre fini d'axiomes d'extension.

Etienne Grandjean a montré en 1992 [27] que si φ contient k variables, alors la deuxième assertion est équivalente à φ est une conséquence logique de $\textsc{Ext}_k$. Il a aussi montré que le problème de décider si une formule de FO est *a.p.s.* vraie et PSPACE-complet, ce qui contraste avec le résultat d'indécidabilité de Trachenbrot pour le problème de décider si une formule de FO est satisfaisable.

SO et MSO
SO est la logique existentielle du second ordre. On quantifie sur les variables de relation avec autant d'alternances que l'on souhaite. Cette logique possède un tel pouvoir d'expression que souvent on impose une restriction sur le nombre d'alternances. MSO est la sous-classe de SO obtenue avec des variables uniquement unaires. Kaufmann et Shelah ont montré en 1985 que MSO n'admettait pas de loi 0-1 [33].

La logique existentielle du second ordre
ESO, la logique existentielle du second-ordre, est constituée des énoncés (formules closes) de la forme

$$\psi = \exists S_1 \dots \exists S_k \varphi(R_1, \dots, R_l, S_1, \dots, S_k),$$

où $S_1, \dots S_k$ sont des variables de relation, $R_1, \dots, R_l$ sont des symboles de relation et
$\varphi(R_1, \dots, R_l, S_1, \dots, S_k)$ est un énoncé de FO, la logique du premier ordre, sur le vocabulaire $\{R_1, \dots, R_l, S_1, \dots, S_k\}$.

De la même manière, nous noterons USO, la logique universelle du second ordre, constituée des énoncés de la forme

$$\psi = \forall S_1 \dots \forall S_k \varphi(R_1, \dots, R_l, S_1, \dots, S_k),$$

Soit $\mathcal{L}$ une sous classe de FO, ESO($\mathcal{L}$) désigne le fragment de ESO constitué des formules $\exists S_1 \dots \exists S_k \varphi(R_1, \dots, R_l, S_1, \dots, S_k)$ où φ est un énoncé de $\mathcal{L}$.

Les logiques MESO et MUSO sont définies en imposant aux variables de relation d'être unaires. On définit de la même manière les classes MESO($\mathcal{L}$).

Classes préfixes et classe de Scott
Généralement $\mathcal{L}$ s'obtient par des contraintes syntaxiques portant sur le nombre de quantificateurs et l'alternance entre ces quantificateurs. Ainsi on définit une classe

préfixe en imposant aux formules φ d'avoir les quantificateurs en début de formule. Une classe préfixe est alors définie par un langage rationnel sur $\{\forall, \exists\}$. Les classes préfixes suivantes interviennent dans l'étude des lois 0-1 :

- classe d'Ackermann $\exists^*\forall\exists^*$
- classe de Bernays-Schönfinkel $\exists^*\forall^*$
- classe de Gödel $\exists^*\forall\forall\exists^*$
- classe de Gödel minimale $\forall^*\forall^*\exists$
- classe de Kahr-Moor-Wang $\forall\exists\forall$

On peut apporter d'autres restrictions, FO_2 est l'ensemble des énoncés du premier ordre comportant au plus 2 variables sans renommage des variables, la classe de Scott est l'ensemble des énoncés de la forme $(\forall x\forall y\varphi_1(x,y)) \bigwedge_{i\in\{1,\ldots,l\}}(\forall x\exists y\varphi_{2,i}(x,y))$, où φ_1 et les $\varphi_{2,i}$ sont sans quantificateur et contiennent uniquement les variables x et y. Nous écrirons aussi $\forall\forall \bigwedge \forall\exists$ la classe de Scott.

La classe de Scott minimale, notée également $\forall\forall\wedge\forall\exists$ est l'ensemble des énoncés de la forme $(\forall x\forall y\varphi_1(x,y)) \wedge (\forall x\exists y\varphi_2(x,y))$, ce sont les énoncés de la classe précédente avec $l = 1$.

Correspondance entre décidabilité et loi 0-1

Une logique $\mathcal{L}$ est décidable lorsque le problème de satisfaisabilité est décidable, c'est-à-dire lorsqu'il existe un algorithme décidant, pour chaque énoncé ψ de $\mathcal{L}$, s'il existe une structure vérifiant ψ. Les trois fragments maximaux décidables de FO sont les classes d'Ackermann, de Bernays-Schöfinkel et de Gödel sans l'égalité (le symbole $=$ n'apparaît pas dans les formules) [12].

Les résultats de loi 0-1 de Kolaitis et Vardi [34, 35, 36] et les contre-exemples de loi de 0-1 de Pacholski, Szwast, Vedo et tendera [47, 48, 54, 51] entraînent la correspondance suivante

Théorème 2.1.1. *Une classe préfixe avec l'égalité $\mathcal{L}$ est décidable si et seulement si MESO($\mathcal{L}$) admet une loi* 0-1*.*

Une classe est finie contrôlable lorsque tout énoncé satisfaisable est finiment satisfaisable (il existe une structure finie vérifiant cet énoncé). Comme l'ensemble des énoncés de FO valides est récursivement énumerable, toute classe finie contrôlable est évidemment décidable. Comme les trois fragments maximaux décidables de FO sont finis contrôlables, Kolaitis et Vardi [36] ont conjecturé que la réelle correspondance porte sur la finie contrôlabilité.

Définition 2.1.2. Contre-Exemples de lois 0-1

Un contre-exemple de loi 0-1 *d'une logique $\mathcal{L}$ est une propriété exprimable dans $\mathcal{L}$ qui soit a une probabilité asymptotique différente de* 0 *et* 1*, soit ne possède pas de probabilité asymptotique.*

La propriété Parité (le nombre d'éléments de M est pair) a permis d'établir la quasi-totalité des contre-exemples de loi 0-1.

Il est facile de voir que s'il existe une fonction f sur M vérifiant

$$f \circ f(a) = a \text{ et } f(a) \neq a, \forall a \in M,$$

alors la cardinalité de M est pair.

On peut remarquer qu'a priori cette propriété s'exprime uniquement sur le domaine, elle ne nécessite pas de relation dans le vocabulaire. On peut ainsi exprimer l'existence d'une telle fonction f dans les logiques MESO($\forall\forall\forall\exists$ avec l'égalité), MESO(Kahr-Moore-Wang avec l'égalité) sur un vocabulaire vide.

Pour les logiques MESO($\forall\forall\exists$ sans l'égalité) [35] et MESO(Kahr-Moore-Wang sans l'égalité) [54], on utilise un symbole de relation binaire pour exprimer *a.p.s.* la fonction f ; on définit une formule telle que si la structure satisfait certains axiomes d'extension de Ext alors la formule est équivalente à l'existence de f.

Pour MESO, Kaufmann [32] montre que l'on peut exprimer *a.p.s.* un ordre total et une relation de successeur. En utilisant la relation de successeur, il colorie les sommets alternativement en noir et blanc. Au final, il suffit de considérer la couleur du dernier élément et Parité est vérifiée si et seulement si celui-ci est colorié en blanc. Les structures considérées par Kaufmann contiennent plusieurs relations binaires, j'ai adapté ce résultat pour pouvoir exprimer la même formule sur les graphes non-orientés [43].

2.2 Comment construire une variante sans probabilité asymptotique

La propriété de noyau

Un graphe orienté $D_n = \langle V_n, A\rangle$ est constitué d'un ensemble de sommets $V_n = \{1, \ldots, n\}$ et d'un ensemble d'arcs A. Il peut être vu comme une structure finie avec une seule relation binaire dans la vocabulaire vérifiant l'axiome $\forall x \neg Axx$ (D_n n'a pas de boucle).

Un noyau est un sous-ensemble U de V_n vérifiant les deux propriétés suivantes :

- U est un **stable** : il n'y a pas d'arc entre deux sommets de U.
- K est **dominant** : pour tout sommet en dehors de U il existe un arc vers un sommet du U.

Un graphe vérifie la propriété Noyau lorsqu'il possède au moins un noyau.

Noyau est exprimable dans la logique MESO(Scott minimale sans l'égalité)

$$\exists U\big(\forall x\forall y((Ux \wedge Uy) \rightarrow \neg Axy)) \wedge \forall x\exists y(\neg Ux \rightarrow (Uy \wedge Axy)))\big)$$

mais aussi dans les logiques MESO(FO_2) et MESO(Gödel minimale sans l'égalité).

FO_2 étant finie contrôlable de nombreux chercheurs pensaient que MESO_2 admettait une loi 0-1.

Après des calculs préliminaires pour comprendre quelles structures finies satisfaisaient les formules de FO_2, mais aussi de MESO_2, j'ai eu la conviction pendant ma thèse qu'une variante de Noyau, toujours exprimable dans MESO(Scott minimale sans l'égalité), pouvait fournir un contre-exemple de loi 0-1.

Probabilité asymptotique de Noyau : un équilibre entre stabilité et dominance bien fragile

Nous verrons au chapitre suivant que Noyau est *a.p.s.* vraie. Pour être plus précis,

il existe une suite d'intervalles finis $(I_n)_{n\in\mathbb{N}}$ dont la taille ne dépend pas de n tel que pour toute suite $(r_n)_{n\in\mathbb{N}}$ de $(I_n)_{n\in\mathbb{N}}$ (pour tout $n \in \mathbb{N}, r_n \in I_n$) D_n possède *a.p.s.* un noyau de taille r_n et D_n ne possède *a.p.s.* de noyau de taille $r_n \notin I_n$.

Il est facile de voir que la probabilité d'avoir un noyau U de taille r dépend uniquement du nombre de sous-ensembles de taille r, de la probabilité qu'un sous-ensemble de taille r soit un stable et de la probabilité qu'un sous-ensemble de taille r soit dominant. Pour tout sous-ensemble U de V_n, plus le cardinal de U est grand, plus la probabilité qu'il soit un ensemble stable est faible. Inversement, plus l'ensemble U est petit plus la probabilité qu'il soit dominant est faible. La stabilité contraint donc U à ne pas être trop grand et la dominance à ce qu'il ne soit pas trop petit. Et la simultanéité des deux propriétés ne devient possible que pour quelques tailles de U. Le nombre d'ensemble de taille r intervient aussi,

c'est un équilibre entre ces trois parties (la stabilité, la dominance et le nombre d'ensembles de taille r) qui permet d'obtenir un intervalle I_n aussi petit. On montre que I_n est de taille inférieur à 4, donc il contient soit 3, soit 4 entiers selon les valeurs de $n \in \mathbb{N}$.

Notons que NOYAU exprimée sur les graphes avec boucles (c'est-à-dire sur les structures contenant une relation binaire quelconque) devient *a.p.s.* fausse. La stabilité se trouve renforcée car nous devons aussi vérifier que $(a, a) \notin R$, pour tout a de U. Pour obtenir le même comportement asymptotique sur les structures avec une relation binaire, il suffit d'ajouter la contrainte $x \neq y$ dans la partie stable

$(\forall x \forall y((Ux \wedge Uy \wedge x \neq y) \rightarrow \neg Rxy))$.

Cela illustre bien le fait que l'équilibre entre stabilité et dominance (qui permet l'existence de noyaux) apparaît comme bien fragile.

De manière générale, en modifiant la définition de la partie stabilité (la partie $\forall\forall$ de l'énoncé de la classe Scott minimale) ainsi que celle de la partie dominance (l'autre partie $\forall\exists$), nous pouvons soit renforcer soit réduire cet équilibre, selon que l'on souhaite avoir ou non des noyaux. Pour les lois 0-1, l'idée est de réduire l'équilibre afin que l'intervalle des tailles possibles soit un intervalle J_n de taille inférieur à 1. Cet intervalle contient donc soit 0, soit 1 entier. On définit alors deux sous-suites A et B de $\mathbb{N}$ tels que NOYAU est *a.p.s.* vraie sur les entiers n de A et *a.p.s.* fausse sur les entiers de B.

Transfert de Fagin sur FO

Soit $\mathcal{M}^{\mathbb{N}}$ le modèle dénombrable sur un vocabulaire $\mathcal{R}$. En utilisant un théorème de compacité, Kolaitis et Vardi ont montré en 1987 [34] que si un énoncé de USO $\forall\mathcal{S}\varphi(\mathcal{R}, \mathcal{S})$ est vrai sur $\mathcal{M}^{\mathbb{N}}$, alors il existe une formule du premier ordre ψ sur le vocabulaire $\mathcal{R}\cup\mathcal{S}$ tel que $\forall\mathcal{S}\varphi(\mathcal{R}, \mathcal{S})$ est conséquence logique de ψ, c'est-à-dire l'énoncé $\psi \rightarrow \forall\mathcal{S}\varphi(\mathcal{R}, \mathcal{S})$ est valide. Par conséquent, tout énoncé de USO vrai sur $\mathcal{M}^{\mathbb{N}}$ est *a.p.s.* vrai. Thierry Lacoste a proposé en 1997 [38] une preuve n'utilisant pas d'argument infini.

Nous venons de voir que NON NOYAU –la négation de NOYAU– est *a.p.s.* vraie sur les structures finies et pourtant elle n'est pas conséquence logique d'un énoncé du premier ordre ψ. Sinon il existerait $k \in \mathbb{N}$ tel que ψ serait conséquence logique d'un

sous-ensemble d'$\textsc{Ext}_k$ les axiomes d'extension à k variables. Or ceux-ci ne peuvent pas déterminer qu'un ensemble U de plus de k éléments est stable. Donc aucune formule du premier ordre avec seulement k variables ne peut distinguer une structure possédant un noyau de taille supérieure à k d'une structure ne possédant pas de noyau.

Les variantes de NOYAU *a.p.s.* fausses fournissent donc de bons contre-exemples au théorème de transfert de Fagin (voir dans la partie sur les logiques modales, les travaux de Goranko et Kapron).

Premier contre-exemple
Nous avons vu que les différentes classes du premier ordre étaient obtenues avec des contraintes sur le nombre de quantificateurs et sur les alternances de quantificateurs. En revanche, le nombre de symboles de relation des structures n'est pas fixé (le nombre de variables de relation non plus d'ailleurs). Cela peut surprendre mais cela vient du fait que ces restrictions ont peu d'incidence sur les autres études portant sur ces logiques comme la décidabilité. N'ayant pas de restriction à ce niveau, j'ai proposé un premier contre-exemple [41, 40, 42] avec un vocabulaire $\mathcal{R}$ de plus d'un demi-million de relations binaires, 524304 pour être précis !

$$\mathcal{R} = \{R_1, \ldots, R_{16}, S_1^1, \ldots, S_{16}^{2^{15}}, \ldots, S_1^{2^{15}}, \ldots, S_{16}^{2^{15}}\}.$$

On définit sur $\mathcal{R}$ les propriétés stabilité et dominance de la manière suivante.

stabilité U est stable lorsque pour tout couple (a, b) de U et pour tout $i \in \{1, \ldots 16\}$ $(a, b) \notin R_i$.

dominance U est dominant lorsque pour tout $j \in \{1, \ldots, 2^{15}\}$ et pour tout $a \in V_n \setminus U$, il existe $b \in U$ et $i \in \{1, \ldots 16\}$ tel que $(a, b) \in S_i^j$.

Une $\mathcal{R}$-structure $\mathcal{M}$ vérifie la propriété $\textsc{Noyau}_1$ lorsqu'il existe un sous-ensemble U stable et dominant.

Cette première variante $\textsc{Noyau}_1$ s'exprime avec l'énoncé de MESO(Scott avec égalité) suivant

$$\exists U\ \forall x \forall y \Big(Ux \wedge Uy \wedge x \neq y \Big) \rightarrow \bigwedge_{i \in \{1,\ldots 16\}} \neg R_i xy$$
$$\wedge \bigwedge_{j \in \{1,\ldots,2^{15}\}} \forall x \exists y \Big(\neg Ux \rightarrow \Big(Uy \wedge \neg \Big(\bigwedge_{i \in \{1,\ldots 16\}} \neg S_i^j xy \Big) \Big) \Big).$$

Malgré le grand nombre de relations binaires, l'énoncé reste relativement simple et proche au niveau logique de celui de NOYAU. En effet le regroupement de relations binaires sert à changer la probabilité d'arc. Comme seule la distribution uniforme sur toutes les structures finies de même taille est considérée, nous avons une probabilité de $1/2$ qu'un arc apparaisse entre deux sommets quelconque. En regroupant 16 relations nous obtenons la probabilité

$$\Pr \Big(\bigwedge_{i \in \{1,\ldots 16\}} \neg R_i xy \Big) = \frac{1}{2^{16}}.$$

D'autre part, nous prenons 2^{15} groupes de relation pour la partie dominance pour augmenter le poids de la partie dominance par rapport à la partie stabilité. Il est assez facile d'avoir une intuition de ce que l'on peut modifier, mais il faut vraiment rentrer dans le détail des calculs pour vérifier que les choix ont été judicieusement effectués afin de réduire l'intervalle des tailles J_n pour qu'il contienne soit 0, soit 1 entier (voir le chapitre suivant).

Second contre-exemple
Le vocabulaire $\mathcal{R}$ est constitué de 16 relations $R_1, \ldots, R_{16}$. Soit $\mathcal{M}_n$ une $\mathcal{R}$-structure sur le domaine V_n. On définit deux sous-ensembles W et X de V_n par

$$W = \{a \in V_n \mid (a,a) \notin R_1\} \text{ et } X = \{a \in V_n \mid (a,a) \notin R_i \text{ pour tout } i \in \{1,\ldots,16\}\}.$$

Clairement $X \subset W$.

On définit sur $\mathcal{R}$ les propriétés stabilité et dominance de la manière suivante :

stabilité U est stable lorsque d'une part $U \subset X$ et d'autre part pour tout couple (a,b) de U et pour tout $i \in \{1,\ldots 16\}$ $(a,b) \notin R_i$.

dominance U est dominant lorsque d'une part $U \subset X$ et d'autre part, pour tout $a \in X \subset U$, il existe $b \in U$ tel que $(a,b) \notin R_i$ pour tout $i \in \{1,\ldots,16\}$.
Une $\mathcal{R}$-structure $\mathcal{M}$ vérifie la propriété NOYAU$_2$ lorsqu'il existe un sous-ensemble U stable et dominant, cette propriété s'exprime avec l'énoncé suivant

$$\exists U\ \forall x \exists y \Big(\Big(Ux \to \bigwedge_{i=1\ldots16} \neg R_i xx \Big) \wedge (Ux \wedge Uy) \to \Big(\bigwedge_{i=1\ldots16} \neg R_i xy \Big) \Big)$$
$$\forall x \exists y \Big(\Big(\neg Ux \wedge \neg R_1 xx \Big) \to \Big(Uy \wedge \bigwedge_{i=1\ldots16} \neg R_i xx \Big) \Big).$$

Comme je l'ai indiqué précédemment, le nombre de relations a peu d'importance en théorie des modèles finis. Les améliorations pour ce second contre-exemple consiste essentiellement à passer de la classe de Scott $\forall\forall\bigwedge\forall\exists$ à la classe de Scott minimale $\forall\forall \wedge \forall\exists$. D'autre part, nous parvenons à supprimer le symbole $=$ dans la formule.

La propriété NOYAU$_2$ est donc exprimable dans la logique MESO(Minimale Scott sans l'égalité). Elle est par conséquent exprimable dans MESO(Gödel sans l'égalité) qui est un des trois fragments de FO maximaux décidables.

Au final,nous avons la correspondance suivante sur les classes préfixes. Soit $\mathcal{C}$ classe préfixe (avec ou sans l'égalité).

Théorème 2.2.1. *MESO ($\mathcal{C}$) admet une loi* 0-1 *si et seulement la propriété* NOYAU *n'est pas exprimable dans cette logique.*

2.3 La propriété de Noyau en logique modale

2.3.1 Introduction sur la logique modale

Frame, structure et logique modale

Alors que les deux contre-exemples précédents ont été obtenu pendant ma thèse, le travail réalisé sur la propriété de noyau en logique modale a commencé deux ans après celle-ci.

La logique modale est une extension naturelle de la logique propositionnelle [53]. Les sémantiques standard de la logique modale sont les sémantiques des mondes possibles. Un frame $\mathcal{F}$ est constitué d'une paire ordonnée $\langle S, R\rangle$, où S est l'ensemble des mondes ou états et R est une relation binaire encore appelée relation d'accessibilité. Une structure (de Kripke) $\mathcal{M}$ est basée sur un frame muni d'un assignement π sur un ensemble dénombrable de variables propositionnelles $p_1, p_2 \ldots$ pour chaque état, autrement dit π est une application de $S \times \{p_1, p_2, \ldots\} \to \{\textsc{Vrai}, \textsc{Faux}\}$.

Soit $\mathcal{M} = \langle S, R, \pi\rangle$ une structure de Kripke et s un état de S, on définit inductivement sur $(\mathcal{M}, s)$ l'ensemble des formules modales

- $(\mathcal{M}, s) \models p$, pour toute variable propositionnelle p, lorsque $\pi(s, p) = \textsc{Vrai}$.
- $(\mathcal{M}, s) \models \neg\varphi$ lorsque $(\mathcal{M}, s) \not\models \varphi$.
- $(\mathcal{M}, s) \models \varphi \wedge \psi$ lorsque $(\mathcal{M}, s) \models \varphi$ et $(\mathcal{M}, s) \models \psi$.
- $(\mathcal{M}, s) \models \Box\varphi$ lorsque $(\mathcal{M}, t) \models \varphi$, pour tout état t accessible ($(s, t) \in R$).

On définit aussi $\Diamond$ le dual de $\Box$ par

$$\Diamond\varphi \equiv \neg\Box\neg\varphi.$$

Les modèles de Kripke donnent une sémantique à la logique modale très facile à appréhender. Les formules vraies sont établies à partir d'un monde (ou état). Elles peuvent soit être vraies dans ce monde, ce sont alors des formules classiques de la logique propositionnelle, soit nécessairement vraie pour tous les mondes accessibles (formules $\Box\varphi$), soit encore vraie pour au moins un monde accessible (formules $\Diamond\varphi$). $\Box$ est appelé modalité du nécessaire et $\Diamond$ celle du possible.

Translation vers $\mathbf{MESO_2}$

Rappelons que MESO_2 désigne la logique monadique existentielle du second ordre à deux variables au premier ordre.

Il existe une translation très simple de la logique modale vers cette logique. À chaque structure de Kripke $\mathcal{M} = \langle S, R, \pi\rangle$, on associe une structure $\mathcal{A} = \langle S, \mathcal{R}\rangle$, sur le vocabulaire $\mathcal{R} = \{R\} \cup \mathcal{P}$, où $\mathcal{P} = \{P_1, P_2, \ldots\}$. Chaque P_i est une relation unaire codant l'ensemble des états où p_i est vrai ($\{s \in S \mid \pi(s, p_i) = \textsc{Vrai}\}$). Pour chaque formule modale φ on construit inductivement une formule $T(\varphi)$ ayant une variable libre x de la manière suivante

- $T(p) = P(x)$ pour toute variable propositionnelle p
- $T(\varphi \wedge \psi) = T(\varphi) \wedge T(\psi)$
- $T(\neg\varphi) = \neg T(\varphi)$
- $\Box\varphi = \forall y(Rxy \to T(\varphi)[x/y])$, où $T(\varphi)[x/y]$ est obtenue à partir de $T(\varphi)$ en remplaçant toutes les occurrences de x par y.

Les formules ainsi construites appartiennent clairement à FO_2, elle possède deux variables x et y et la variable x est libre (c'est-à-dire non quantifiée).

Nous avons l'équivalence entre la satisfaisabilité de φ sur $\mathcal{M}$ (il existe $s \in S$ tel que $\langle \mathcal{M}, s\rangle \models \varphi$) et la satisfaisabilité de $T(\varphi)$ sur $\mathcal{A}$ (il existe $s \in S$ tel que $\mathcal{A} \models T(\varphi)(s)$.

Une formule φ est satisfaisable sur les structures de Kripke lorsqu'il existe $\mathcal{M} = \langle S, R, \pi\rangle$ tel que $\mathcal{M} \models \exists x T(\varphi)(x)$. D'autre part φ est satisfaisable sur les frames de Kripke lorsqu'il existe $\mathcal{F} = \langle S, R\rangle$ tel que $\mathcal{F} \models \exists \mathcal{P} \exists x T(\varphi)(x)$.

Nous noterons MESO(∃ modale) ce fragment de $MESO_2$.

Validité sur les structures

Que cela soit sur les stuctures ou sur les frames, on peut remplacer le problème de satisfaisabilité par celui de la validité avec les formules respectives $\forall x T(\varphi)(x)$ et $\forall \mathcal{P} \forall x T(\varphi)(x)$.

Halpern et Kapron se sont intéressés en 1994 [28] à la validité sur les structures et les frames ils ont donc étudié la validité des formules $\forall x T(\varphi)(x)$ et $\forall \mathcal{P} \forall x T(\varphi)(x)$. La loi 0-1 pour la validité sur les structures provient de celle de FO, une formule modale φ est *a.p.s.* vraie si et seulement si $T(\varphi)$ est valide. Cette loi 0-1 sur la validité des structures est évidemment équivalente à celle sur la satisfaisabilité. Ils ont aussi fourni une axiomatisation des formules *a.p.s.* vraie sur les structures.

Validité sur les frames

Le cas de la validité sur les frames est moins clair. Leur preuve est très compliquée et donne peu d'indications logiques. Lorsque mes contre-exemples furent connus, la proximité de MESO(∃ modale) et de $MESO_2$ a conduit certains chercheurs du domaine à douter de la validité de leur résultat. Leur résultat utilise la notion de structures ε-spéciales et nous avons l'équivalence entre

(a) Si φ n'est pas satisfaite par une structure 0-spéciale alors φ est *a.p.s.* fausse.

(b) Si φ est satisfaite par une structure 0-spéciale alors φ est *a.p.s.* vraie.

La partie (b) est correcte, malheureusement la partie (a) est incorrecte. L'énoncé suivant est correct : si la probabilité asymptotique de φ est différente de 0 alors il existe un structure $\mathcal{M}$ qui est ε-spéciale pour tout $\varepsilon > 0$. Les auteurs utilisent ensuite un argument de continuité pour en déduire que cette structure est 0-spéciale.

J'ai transmis mon contre-exemple [44] à Halpern et Kapron et ceux-ci l'ont repris comme exemple pour expliquer pourquoi l'argument de continuité ne s'applique pas toujours. Ils ont ensuite publié un erratum qui est paru à Annals of Pure and Applied Logic, le journal où ils avaient publié leurs travaux sur les lois 0-1 en logiques modales [29].

La structure dénombrable ML^r

Toujours pour le problème de la validité, Goranko et Kapron ont étudié ML^r, la logique des énoncés vrais sur F^r, le frame de Kripke dénombrable (unique à isomorphisme près) et ML^{as} la logique des énoncés *a.p.s.* sur les frames finis. Ils donnent dans [26] une axiomatisation complète et consistante de ML^r et montrent qu'elle n'est pas finiment axiomatisable. Comme pour les autres fragments de MESO, ML^r est finiment

contrôlable et le problème de satisfaisabilité est EXPTIME. Le transfert de Fagin ne s'applique également pas à MESO_2 : il existe des énoncés de ML^{as} qui ne sont pas vrais sur le frame dénombrable. Par le théorème de compacité de Kolaitis et Vardi, nous avons donc $\mathrm{ML}^{r} \subset \mathrm{ML}^{as}$.

Ils proposent pour montrer la stricte inclusion une variante du noyau appelée DOUBLE NOYAU, cependant comme je leur ai précisé par la suite la simple propriété NOYAU fournit également un contre-exemple car NOYAU est *a.p.s.* fausse sur les structures avec une relation binaire.

2.3.2 Noyaux en logique modale

Comment exprimer Noyau sur les frames
Un frame satisfait la propriété DIAMÈTRE2 lorsque tout état est atteignable en exactement deux étapes, c'est-à-dire satisfait la formule $\forall x \forall y \exists z\ Rxz \wedge Rzy$. Cette formule est *a.p.s.* vraie sur les structures finies et vraie également sur F^r. Soit E et A respectivement les modalités existentielle et universelle. $\mathcal{F} \models \mathbf{E}\varphi$ (resp. $\mathcal{F} \models \mathbf{A}\varphi$) signifie qu'il existe un état s tel que $(\mathcal{F}, s) \models \varphi$ (resp. que tout état s vérifie $(\mathcal{F}, s) \models \varphi$). Tout frame de diamètre 2 vérifie

$$\mathbf{E}p \equiv \Diamond\Diamond p \text{ et } \mathbf{A}p \equiv \Box\Box p.$$

Nous allons maintenant exprimer NOYAU sur les frames satisfaisant DIAMÈTRE 2. Comme nous l'avons vu précédemment cette propriété est *a.p.s.* fausse sur les frames car la relation d'accessibilité est une relation binaire telle que $(a, a) \in R$ est possible.

On considère les frames avec une seule variable propositionnelle p codé par la relation unaire P. Le noyau U est alors l'ensemble des états s de S tel que $P(s) =$ VRAI.

Stabilité U est stable lorsqu'à partir de tout état de U, on ne peut pas atteindre un autre état de U, ce que l'on peut exprimer par la formule

$$p \to \Box\neg p.$$

Dominance Inversement, la dominance est assurée si tout état hors de U peut atteindre un état de U, ce qui s'exprime par la formule

$$\neg p \to \Diamond p.$$

En observant que les formules $p \to \Box\neg p$ et $p \to \Diamond p$ sont équivalentes, Goranko et Kapron en déduisent [26] que NOYAU s'exprime par la formule

$$A(p \to \neg\Diamond p) \wedge A(\neg p \to \Diamond p).$$

et donc que NON NOYAU s'exprime par une formule très simple

$$\mathbf{E}(p \iff \Diamond p).$$

Ce qu'ils auraient pu établir directement s'ils avaient pensé à exprimer NON NOYAU avec la formule

$$\forall U\ \exists x \left(Ux \iff \exists y Rxy \wedge Uy\right).$$

Le contre-exemple Noyau$_3$
L'élaboration du contre-exemple sur la satisfaisabilité sur les frames [44] a nécessité beaucoup plus de travail qu'une simple modification des contre-exemples précédents.

La véritable innovation a été d'exprimer une propriété faisant intervenir un élément distingué et de définir les notions de stabilité et de dominance à partir de cet état. C'est d'ailleurs assez naturel pour la logique modale car cela revient à partir d'un état fixé, contrairement à la propriété NOYAU qui ne distingue aucun sommet.

Soit $\mathcal{F} = \langle S, R\rangle$ un frame de Kripke satisfaisant DIAMÈTRE 2.

Soit s_0 un état de S, nous noterons S^{S_0} l'ensemble des états s accessibles à partir de s_0, autrement dit $S^{s_0} = \{s \in S \mid (s_0, s) \in R\}$. Soit $\mathcal{M} = \langle \mathcal{F}, \pi\rangle$ une structure basée sur $\mathcal{F}$ où seulement deux variables propositionnelles p et q sont utilisées, la première toujours pour définir le noyau et la seconde pour définir l'état distingué.

Stabilité (U, s_0) est stable lorsque $s_0 \notin S$ et pour tout couple (a, b) de $U \cup \{s_0\}$, $(a, b) \notin R$ (y compris pour le cas $a = b$).
Cette propriété s'exprime par la formule suivante où s_0 est un état où q est vrai

$$q \wedge \neg p \wedge A\Big((p \vee q) \rightarrow \neg\Diamond(p \vee q)\Big).$$

Dominance (U, s_0) est dominant lorsque chaque état a de S^{s_0} peut accéder à un état b de de U ($(a, b) \in R$). Ce qui s'exprime par la formule $\Box\Diamond p$.

Comme toujours (U, s_0) est un noyau s'il est à la fois stable et dominant.

On montre facilement que cette propriété sur le vocabulaire $\mathcal{R} = \{R\}$ s'exprime par la formule de MESO$_2$ suivante

$$\exists U \exists V \Big(\forall x \forall y \neg Vx \vee \neg Vy\Big) \wedge \Big(\exists x Vx \wedge \neg Ux\Big) \wedge \Big(\forall x \forall y (Ux \vee Vx) \wedge (Uy \vee Vy) \rightarrow \neg Rxy\Big)$$
$$\wedge \Big(\forall x \Big((\exists y Vy \wedge Ryx) \rightarrow (\exists y Uy \wedge Rxy)\Big)\Big).$$

2.4 Problèmes ouverts

Lois 0-1 pour un Vocabulaire restreint
Nous avons vu que NOYAU était une propriété incontournable pour obtenir des contre-exemples de loi 0-1 sur des fragments de MESO_2. Bien que cela n'ait pas été une priorité pour les problèmes abordés en théorie des modèles finis, il est naturel de chercher un contre-exemple avec le plus petit vocabulaire. Dans la conclusion de ma thèse [41], je conjecturais que MESO_2 avait une loi 0-1 pour un vocabulaire ne contenant qu'une relation binaire et qu'il fallait ajouter des relations unaires pour obtenir des contre-exemples. Ce qui est contredit par NOYAU_3 qui s'exprime avec un vocabulaire d'une seule relation binaire. Le cas des graphes restait cependant encore ouvert.

Proposition 2.4.1. *IL existe une variante de* NOYAU *qui est exprimable sur les graphes orientés dans la logique $MESO_2$ et qui n'admet pas de probabilité asymptotique.*

Preuve. Voir la variante NOYAU_4 dans le chapitre 3 □

Le cas des graphes non-orientés semble plus difficile. Pourtant contrairement aux conjectures que j'ai données dans ma thèse, je pense maintenant qu'il est possible d'obtenir une loi 0-1.

Conjecture 2.4.1. *Il existe une variante de* NOYAU *exprimable sur les graphes non-orientés dans la logique $MESO_2$ qui n'admet pas de probabilité asymptotique.*

Lois 0-1 faible Nous dirons qu'une logique $\mathcal{L}$ admet une loi 0-1 faible lorsque toute propriété exprimable dans $\mathcal{L}$ est
- soit *a.p.s.* vraie
- soit *a.p.s.* fausse
- soit n'admet pas de probabilité asymptotique

Et donc par définition toute logique qui admet une loi 0-1 admet une loi 0-1 faible.

Comme les seuls contre-exemples de loi 0-1 sur MESO_2 semblent être des variantes de NOYAU n'ayant pas de probabilité asymptotique, je propose la conjecture suivante. Il semble très difficile d'obtenir une preuve de cette conjecture, car il faudrait un argument général pour prouver que l'on ne peut avoir de probabilité asymptotique différente de 0 et de 1.

Conjecture 2.4.2. *$MESO_2$ admet une loi 0-1 faible.*

Comme les exceptions pour la correspondance pour les classes $\mathcal{L}$ de FO entre décidabilité et loi 0-1 pour $\text{MESO}(\mathcal{L})$ sont FO_2 et la classe de Gödel sans l'égalité qui sont toutes les deux finies contrôlable, je propose la conjecture suivante

Conjecture 2.4.3. *Soit $\mathcal{L}$ une classe « naturelle »de FO, MESO($\mathcal{L}$) admet une loi 0-1 faible si et seulement si $\mathcal{L}$ est finie contrôlable.*

Nous devons ajouter classe « naturelle »car il est possible de construire artificiellement une classe $\mathcal{L}$ qui est un contre-exemple. Il faut donc définir une classe par des contraintes sur les quantificateurs et leurs alternances comme c'est le cas pour les classes préfixes et pour FO_2.

Monadique NP vs Monadique co-NP

Nous avons vu que les variantes de NOYAU fournissent des bons contre-exemples du transfert de Fagin. Donc leur utilisation en théorie des modèles finis ne se limitent pas à l'étude des lois 0-1.

Fagin a prouvé que la logique MESO avait un pouvoir d'expression différent de celui de MUSO, la logique monadique universelle du second ordre [17]. Pour cela, il a montré que la connexité – le graphe est connexe – n'est pas exprimable dans MESO alors que cette propriété est facilement exprimable dans MUSO.

Rappelons que ESO (resp. USO) capture la classe NP (resp. co-NP). De même MESO (resp. MUSO) capture la classe Monadique NP (resp. Monadique co-NP). Cela répond par la négative à la question Monadique NP $\stackrel{?}{=}$ Monadique co-NP qui est une version affaiblie de la célèbre question NP $\stackrel{?}{=}$ co-NP.

On peut se poser la même question pour les propriétés qui sont définies *a.p.s.* sur des logiques. Deux logiques $\mathcal{L}_1$ et $\mathcal{L}_2$ sur un même vocabulaire $\mathcal{R}$ sont dites *a.p.s.* équivalentes lorsqu'il existe une suite $\mathcal{C}_n$ de $\mathcal{R}$-structures finies de taille n de mesure $1 - \lim_{n\to\infty} \mu_n(\mathcal{M} \in \mathcal{C}_n) = 1$ – telle que $\mathcal{L}_1$ et $\mathcal{L}_2$ définissent les mêmes propriétés sur $\mathcal{C}_n$ [39].

Michel de Rougemont qui a travaillé sur la séparation entre Monadique NP et Monadique co-NP pour des vocabulaires contenant des prédicats préconstruits [11], a fait la conjecture suivante

Conjecture 2.4.4. *Monadique NP et Monadique co-NP ne sont pas a.p.s. équivalentes.*

Nous avons essayé sans succès de montrer ce résultat après ma thèse en utilisant des variantes de NOYAU et des jeux d'Ehrenfeucht-Fraïssé. Je pense qu'il serait intéressant de reprendre ce travail. J'ai d'une part acquis depuis une plus grande maîtrise des graphes aléatoires, alors que j'avais peu de connaissances dans ce domaine lorsque j'ai recherché mon premier contre-exemple de loi 0-1 et, d'autre part, j'ai une meilleure connaissance des variantes possibles grâce au dernier contre-exemple NOYAU$_3$, ce qui devrait permettre une exploration plus approfondie.

Chapitre 3

Calculs asymptotiques

3.1 Construction d'un graphe aléatoire

Erdös et Rényi sont les fondateurs de la théorie des graphes aléatoires, ils ont lancé en 1959 les bases de cette théorie dans leur célèbre article « On the evolution of random graphs » [16]. Ils considèrent dans cet article le modèle $\mathcal{G}(n, M)$ et étudient les fonctions seuil associées à diverses propriétés telles que l'apparition d'un sous-graphe fixé, la décomposition en arbres, le nombre de cycles, la croissance de la plus grande composante connexe.

Depuis de nombreux travaux ont été mené, le lecteur soucieux de parfaire sa connaissance dans ce domaine aura l'embarras du choix, il pourra en particulier consulter les livres suivants [6, 15, 30].

Edgar Gilbert a introduit en 1959 [23] sur les graphes non-orientés le modèle $\mathcal{G}(n, p)$ pour l'étude des phénomènes de seuil pour la connexité.

Erdös et Rényi ont défini indépendamment et au même moment, toujours sur les graphes non-orientés, le modèle $\mathcal{G}(n, M)$ pour étudier le même problème.

Un graphe non-orienté sera noté $G_n = \langle V_n, E\rangle$, où $V_n = \{1, \ldots, n\}$ est l'ensemble des sommets et E est l'ensemble des arêtes (on suppose qu'il n'y a pas de boucle).

Modèle $\mathcal{G}(n, p)$
Soit $p = p(n)$, $0 < p < 1$, G_n est construit avec le modèle $\mathcal{G}(n, p)$ en mettant une arête avec probabilité p pour chaque paire de sommets $\{a, b\}$. Les arêtes sont mises de manière indépendante, cela correspond à $\binom{n}{2}$ essais de Bernoulli de paramètre p. La valeur $p = 1/2$ correspond à la distribution uniforme, ou équiprobabilité, où tout graphe a la même probabilité d'être construit. Généralement, nous obtenons des résultats très similaires lorsque nous prenons n'importe quelle constante p.

Modèle $\mathcal{G}(n, M)$
Soit $M = M(n)$ un entier compris entre 0 et $\binom{n}{2}$. G_n est construit avec le modèle $\mathcal{G}(n, M)$ en choisissant M arêtes parmi les $\binom{n}{2}$ paires de sommets avec la distribution uniforme. Ce modèle est parfois mieux adapté pour certaines démonstrations. Il est communément admis que les deux modèles sont équivalents – dans le sens où l'on

observe les mêmes phénomènes – lorsque $M \sim p\binom{n}{2}$ ($p\binom{n}{2}$ étant le nombre moyen d'arêtes d'un graphe de $\mathcal{G}(n,p)$) bien que le nombre exact d'arêtes ne soit connu que pour $\mathcal{G}(n,M)$.

Nous pouvons reprendre ces modèles sur les graphes orientés. Soit $D_n = \langle V_n, A\rangle$ un graphe orienté où V_n est toujours l'ensemble des sommets et A l'ensemble des arcs. On remplace paire de sommet $\{a,b\}$ par couple de sommets (a,b), donc pour le modèle $\mathcal{D}(n,p)$ nous avons $n(n-1)$ essais de Bernoulli de paramètre p et pour $\mathcal{D}(n,M)$ les M arcs sont choisis parmi les $n(n-1)$ couples.

Beaucoup d'autres modèles peuvent être définis, en particulier lorsque nous ne voulons pas l'indépendance pour le placement des arêtes.

Nous considérerons dans nos études uniquement le modèle $\mathcal{D}(n,p)$, mais des résultats similaires peuvent être obtenus avec le modèle $\mathcal{D}(n,M)$.

Probabilité asymptotique d'une propriété sur les graphes orientés
Soit $\mathcal{P}$ une propriété sur les graphes orientés. Soit D_n de $\mathcal{D}(n,p)$. On note $\mu_{n,p}(\mathcal{P})$ la probabilité que D_n vérifie $\mathcal{P}$. Si $\mu_{n,p}(\mathcal{P})$ admet une limite lorsque n tend vers ∞ cette limite est notée $\mu_p(\mathcal{P})$ et est appelée probabilité asymptotique de $\mathcal{P}$.

Lorsque $\mu_p(\mathcal{P}) = 1$, on dit que $\mathcal{P}$ est asymptotiquement presque sûrement (almost surely en anglais) vraie.

Inversement, lorsque $\mu_p(\mathcal{P}) = 0$, on dit que $\mathcal{P}$ est asymptotiquement presque sûrement fausse.

On notera souvent en abrégé *a.p.s.* pour asymptotiquement presque sûrement. Notons que certains chercheurs du domaine écrivent *avec forte probabilité* (with hight probability) à la place de asymptotiquement presque sûrement.

3.2 Méthode des moments

Nous allons considérer des propriétés $\mathcal{P}$ particulières, celles portant sur des sous-ensembles U de V_n. Donc la méthode que nous allons donner ici ne peut convenir à des propriétés comme la connexité ou l'hamiltonicité qui font intervenir tous les sommets du graphe.

Pour $U \subset V_n$, (D_n, U) vérifiera $\mathcal{P}$ lorsque D_n vérifiera une certaine propriété dépendant de U. La propriété $\mathcal{P}$ sera vérifié lorsqu'il existera au moins un tel U. Ce formaliste convient parfaitement pour l'étude des noyaux, D_n vérifie NOYAU lorsqu'il existe $U \subset V_n$ qui est un noyau.

Soit $\mathcal{P}$ une propriété portant sur les sous-ensembles de V_n. On définit pour tout $U \subset V_n$, la variable aléatoire élémentaire $X_U^{\mathcal{P}}$ telle que

$$X_U^{\mathcal{P}} = \begin{cases} 1 \text{ lorsque } (D_n, U) \text{ vérifie } \mathcal{P} \\ 0 \text{ sinon.} \end{cases}$$

Soient p et q tels $< p, q < 1$ et $0 \leq r \leq n$ (p et r peuvent être des fonctions de n). Nous allons maintenant considérer tous les ensembles U de taille r. Soit $X_r^{\mathcal{P}}$ la variable

aléatoire définie par

$$X_r^{\mathcal{P}} = \sum_{U,|U|=r} X_U^{\mathcal{P}}.$$

$X_r^{\mathcal{P}}$ nous donne le nombre de sous-ensembles U de taille r tel que (D_n, U) vérifie $\mathcal{P}$.

Les trois propriétés portant sur les sous-ensembles de V_n que nous allons étudier sont Stable, Dominant et Noyau.

Stable U est stable si pour tout couple de sommets (a, b) de U, (a, b) n'est pas un arc de D_n.

Dominant U est dominant lorsque pour tout sommet a de $V_n \setminus U$, il existe un sommet b de U tel que (a, b) est un arc de D_n.

Noyau U est un *noyau* lorsque U est à la fois stable et dominant.

Notons $S(q, r)$ (resp. $D(q, n, r)$ et $N(q, n, r)$) la probabilité qu'un ensemble U de taille r soit un ensemble stable (resp. un ensemble dominant, un noyau).

On montre facilement que l'on a

$$S(q, r) = q^{r(r-1)} \quad D(q, n, r) = (1 - q^r)^{n-r}.$$

Et comme la stabilité et la dominance font intervenir des ensembles de paires de sommets disjoints, il vient

$$N(q, n, r) = S(q, r)\ D(q, n, r).$$

On en déduit que les trois propriétés ont comme espérance

$$\begin{array}{lcl} E[X_r^{Stable}] & = & \binom{n}{r} q^{r(r-1)}, \\ E[X_r^{Dominant}] & = & \binom{n}{r} (1 - q^r)^{n-r}, \\ E[X_r^{Noyau}] & = & \binom{n}{r} q^{r(r-1)} (1 - q^r)^{n-r}. \end{array}$$

Méthode du premier moment

Pour montrer que nous n'avons *a.p.s.* pas de noyau de taille $r(n)$, on montre que l'espérance (le nombre moyen de noyaux de taille r) tends vers 0

$$\lim_{n\to\infty} E[X_r^{Noyau}] = 0,$$

comme la probabilité d'avoir un noyau de taille r est inférieure au nombre moyen de noyaux de taille r

$$\Pr(X_r^{Noyau} > 0) \leq E[X_r^{Noyau}] \qquad \text{(inégalité de Markov)},$$

on en déduit que $\lim_{n\to\infty} \Pr(X_r^{Noyau} > 0) = 0$.

Méthode du second moment
Pour montrer que nous avons *a.p.s.* un noyau de taille $r(n)$, on commence par montrer que l'espérance tend vers l'infini

$$\lim_{n\to\infty} E[X_r^{Noyau}] = \infty$$

et on montre ensuite que le moment factoriel d'ordre 2 est équivalent à l'espérance au carré

$$E[X_r^{Noyau}(X_r^{Noyau}-1)] \sim E[X_r^{Noyau}]^2.$$

Nous avons alors $\lim_{n\to\infty} \dfrac{Var(X_U^{Noyau})}{E[X_U^{Noyau}]^2} = \dfrac{1}{E[X_r^{Noyau}]} = 0.$

On utilise ensuite l'inégalité suivante qui nous donne une majoration de la probabilité de ne pas avoir de noyau de taille r

$$\Pr(X_U^{Noyau}=0) \leq \frac{Var(X_U^{Noyau})}{E[X_U^{Noyau}]^2} \qquad \text{(inégalité de Bienaymé-Chebychev)},$$

on en déduit que $\lim_{n\to\infty} \Pr(X_U^{Noyau} > 0) = 1$.

Remarque 3.2.1. *La méthode du premier et du second moment est très souvent utilisée pour beaucoup de propriétés portant sur des sous-ensembles V_n,. Elle nécessite néanmoins que l'on puisse fixer la taille pour ne s'intéresser qu'aux ensembles d'une même taille.*

Remarque 3.2.2. *Le moment factoriel d'ordre 2 demande le calcul, pour tout couple de sous-ensembles de taille r, de la probabilité que ces deux ensembles vérifient la propriété. Par exemple, pour le noyau, nous avons*

$$E[X_r^{Noyau}(X_r^{Noyau}-1)] = \sum_{\substack{U_1,|U_1|=r \\ U_2,|U_2|=r, U_2\neq U_1}} \Pr(U_1 \text{ et } U_2 \text{ sont des noyaux})$$

Et, de manière générale, cette probabilité varie selon la cardinalité de l'intersection entre les deux sous-ensembles.

3.3 Existence des noyaux dans les graphes denses

On parle de graphes orientés denses lorsque le nombre d'arcs est quadratique par rapport au nombre de sommets. Pour les graphes de $\mathcal{D}(n,p)$ cela correspond au cas où p est constant.

En 1990, Wenceslas De la Vega [10] et Ioan Tomescu [52] ont publié indépendamment (et dans le même journal!) des résultats très similaires sur les tailles possibles des noyaux. De la Vega a déterminé pour tout p constant une suite $m_p(n)$ telle qu'un graphe D_n de $\mathcal{D}(n,p)$ possède *a.p.s.* un noyau de cette taille. Tomescu a déterminé de son côté l'intervalle des tailles possibles dans le cas de la distribution uniforme ($p = 1/2$).

En étudiant leurs preuves [45], j'ai montré que l'on pouvait étendre leurs résultats qui se synthétise en un seul théorème.

Théorème 3.3.1. *Soient deux constantes $0 < p < 1$ et q tel que $p + q = 1$. Il existe deux réels k_1 et k_2 ne dépendant que de q tel que pour $\beta(n) = \log_{1/q} n - \log_{1/q}(\log_{1/q} n)$ et $I(p,n) = [\beta + k_1, \beta + k_2]$. Les noyaux de $D_n \in \mathcal{D}(n,p)$ satisfont*

1. *a.p.s. tous les noyaux ont une taille dans $r \in I(p,n)$;*
2. *pour toute suite $r(n) \in I(p,n)$, D_n possède a.p.s. un noyau de taille $r(n)$.*

Remarque 3.3.1. *La taille de l'intervalle $I(p,n)$ vaut $k_2 - k_1$, elle dépend uniquement de p et pas de n. On peut aussi vérifier que l'intervalle décroît lorsque p augmente.*

Sans donner tous les détails, nous pouvons expliquer comment est déterminé l'intervalle $I(p,n)$. Tout d'abord $\beta = \beta(q)$ désignera le réel satisfaisant $n = \beta q^{-\beta}$.

D'autre part, pour tout n et tout entier r proche de β, on associe le réel c tel que $n = e^c r q^{-r}$. On calcule les trois parties intervenant dans le noyau (stabilité, dominance et et nombre de sous-ensembles de taille r) :

$$\begin{aligned}
\ln S(q,r) &= r^2 \ln q - r \ln q. \\
\ln \binom{n}{r} &= r(c+1) - r^2 \ln q - \tfrac{1}{2} \ln r + \ln(\sqrt{2\pi r}) + o(1). \\
\ln D(q,n,r) = &= -r(e^c + o(1)).
\end{aligned}$$

On définit ensuite la fonction

$$g(c,q) = -e^c + c + 1 - \ln q.$$

Il vient finalement

$$E[X_r^{Noyau}] \sim \frac{e^{rg(c,q)}}{\sqrt{2\pi r}}.$$

Il existe deux réels $c_2 < c_1$ tels que $g(c_1) = g(c_2) = 0$ et $g(c) > 0$ si et seulement si $c_2 < c < c_1$. Soient k_1 et k_2 les deux réels vérifiant $n = e^{c_1}(\beta - k_1) b^{\beta - k_1}$ et $n = e^{c_2}(\beta - k_2) b^{\beta - k_2}$. On vérifie que pour tout p constant, $k_2 - k_1 > 1$, nous avons donc au moins une taille pour laquelle le nombre moyen de noyaux tend vers ∞.

On utilisant la méthode du premier moment, on voit facilement que les seules tailles qui peuvent être *a.p.s.* sont dans l'intervalle $I(p,n) = [\beta + k_1, \beta + k_2]$.

Le reste de la démonstration est plus délicate car il faut, d'une part, utiliser la méthode du second moment pour montrer que l'on a *a.p.s.* un noyau pour chacune des tailles de $I(p,n)$ et ensuite regrouper les autres tailles restantes pour les éliminer. En effet, l'ensemble des tailles possibles n'est pas borné et montrer que nous n'avons *a.p.s.* pas de noyau pour chacune des tailles ne permet pas de conclure pour l'ensemble total (voir le cas $p = c/n$). Je ne donnerai pas ici la démonstration complète.

Les variantes NOYAU$_1$, NOYAU$_2$ et NOYAU$_3$ ont été introduites dans la section 2.

La variante Noyau$_1$ On peut refaire les mêmes calculs pour la variante NOYAU$_1$. La probabilité d'arête devient $p = 1 - \left(\frac{1}{2}\right)^{16}$ et donc $\ln q = -16 \ln 2$. En reprenant les calculs précédents, on montre facilement que l'on obtient la fonction

$$g_1(c) = -2^{15} e^c + c + 1 + 16 \ln 2.$$

Comme précédemment, nous avons deux réels $c_2 < c_1$ tels que $g_1(c_1) = g_1(c_2) = 0$ et $g(c) > 0$ si et seulement si $c_2 < c < c_1$. Soient k_1 et k_2 sont les deux réels vérifiant $n = e^{c_1}(\beta - k_1)b^{\beta-k_1}$ et $n = e^{c_2}(\beta - k_2)b^{\beta-k_2}$. La variante a été conçue de sorte que $k_2 - k_1 = 0.22\ldots < 1$. Donc selon les valeurs de n, on peut avoir 0 ou 1 entier dans l'intervalle $[k_1, k_2]$. On construit deux sous-suites de $\mathbb{N}$, pour la première nous n'avons pas d'entier dans $[k_2(n), k_1(n)]$ et pour la seconde nous avons exactement un entier.

Définition 3.3.1. *Nous appellerons densité d'un sous-ensemble de V_n le rapport de sa cardinalité sur le nombre de sommets total, autrement dit la proportion de sommets appartenant à cet ensemble.*

La variante Noyau$_2$ Pour NOYAU$_2$, rappelons que W désigne l'ensemble

$$\{a \in V_n \mid (a, a) \notin R_1\}$$

et X désigne l'ensemble

$$\{a \in V_n \mid (a, a) \notin R_i \text{ pour tout } i \in \{1, \ldots, 16\}\}.$$

U est un stable de X et la dominance se fait de W vers U.

Soient x et w les densités respectives de X et W, nous avons *a.p.s.* $w \sim \frac{1}{2}$ et $x \sim \frac{1}{2^{16}}$ et donc $\frac{w}{x} \sim 2^{15}$. Le coefficient 2^{15} sert à amplifier le rapport entre dominance et stabilité, il était obtenue pour NOYAU$_1$ en introduisant un grand nombre de variables et ici nous l'avons avec le rapport entre la cardinalité de W et celle de X. On vérifie de plus que le fait de connaître seulement la densité de X et W et pas leur cardinalité exacte suffit pour effectuer les calculs.

La variante Noyau$_3$ Soit U un sous-ensemble de l'ensemble des états S et s_0 un état n'appartenant pas à U. On dit que (U, s_0) est stable lorsque pour tout couple (a, b) de $U \cup \{s_0\}$, $(a, b) \notin R$, y compris dans le cas $a = b$.

Soit L l'ensemble des états symétriques, autrement dit $L = \{s \in S \mid (s, s) \in R\}$ et n_L sa cardinalité. Pour choisir un ensemble $U \cup s_0$, nous devons fixer $r + 1$ éléments parmi les n_L éléments symétriques et ensuite choisir s_0. Le nombre de choix est donc

$$\binom{n_L}{r+1}(r+1) = \binom{n_L}{r}\frac{n_L - r}{r+1}(r+1) = \binom{n_L}{r}(n_L - r).^{1}$$

Notons $S_3(r + 1)$ la probabilité qu'un ensemble de taille $r + 1$ vérifie la stabilité. Il vient

$$S_3(r+1) = \binom{n_L}{r}(n_L - r)S(1/2, r+1) = \binom{n_L}{r}(n_L - r)2^{2r}S(1/2, r).$$

D'autre part, (U, s_0) est dominant lorsque chaque état a de S^{s_0} (l'ensemble des états accessibles à partir de s_0) peut accéder à un état b de de U (c'est-à-dire $(a, b) \in R$).

1. Il subsiste une coquille dans l'article [44], où il est écrit $\binom{n_L}{r+1}(r+1) = \binom{n_L}{r}n_L$, mais cela n'a pas d'incidence sur les calculs car $n_L \sim (n_L - r)$.

Soit n_{s_0} la cardinalité de S_0. La probabilité que (U, s_0) soit dominant vaut $D(1/2, n_{S_0}, r)$. Finalement, nous avons

$$E[X_r^{\text{NOYAU}_3}] = \binom{n_L}{r}(n_L - r)2^{-2r}N(1/2, n_{S_0}, r).$$

On montre que nous avons *a.p.s.*

$$n_L \sim n_{S_0} \sim \frac{n}{2}.$$

On reprend l'étude de NOYAU précédente en remplaçant n par $\frac{n}{2}$. On a alors $\beta(n) = \log_2(n/2) - \log_2(\log_2(n/2))$. Pour tout n, et tout entier r proche de β, on associe le réel c tel que $\frac{n}{2} = e^c r 2^r$.

Il vient

$$E[X_r^{\text{NOYAU}_3}] \sim \frac{n}{2}2^{-2r}\ E[X_r^{\text{NOYAU}}].$$

Par conséquent

$$E[X_r^{\text{NOYAU}_3}] \sim \frac{e^c r e^{r g_3(c)}}{\sqrt{2\pi r}}$$

où

$$g_3(c) = g(c) - \ln 2 = -e^c + c + 1.$$

En étudiant la fonction $g_3(c)$, on montre que la seule taille possible pour un noyau est $m(n) = \lceil \beta \rfloor$, lorsque β est proche d'une entier, c'est-à-dire $\beta(n) \sim m(n)$, on a alors

$$E[X_{m(n)}^{\text{NOYAU}_3}] \sim \frac{\sqrt{m(n)}}{\sqrt{2\pi}}.$$

On utilise ensuite le second moment pour montrer que nous avons *a.p.s.* des noyaux de taille $m(n)$.

C'est la seule taille pour laquelle l'espérance tant vers ∞. On définit ensuite deux suites A_n et B_n, pour A_n nous avons *a.p.s.* un noyau de taille $m(n) = \lceil \beta(n) \rfloor$ et pour la suite B_n nous avons *a.p.s.* pas de noyau.

La variante Noyau$_4$ La démonstration de ce contre-exemple n'apparaît dans aucun article. Nous allons donc donner la démonstration complète.

Définition 3.3.2. *Soit U un noyau de D_n avec la définition initiale de* NOYAU. *On ajoute la condition qu'il existe un sommet s_0 qui est connecté à tous les sommets de U dans les deux sens, autrement dit $(s_0, a) \in R$ et $(a, s_0) \in R$ pour tout $a \in U$. (U, s_0) est appelé un noyau$_4$.*

Nous sommes dans le cas de la distribution uniforme donc $p = q = 1/2$.

On obtient

$$\begin{aligned} E[X_r^{\text{NOYAU}_4}] &= \binom{n}{r}(n - r)2^{-2r}S(1/2, r)\ D(1/2, n-1, r). \\ &\sim n\ 2^{-2r}E[X_r^{\text{NOYAU}}]. \end{aligned}$$

Reprenons β le réel satisfaisant $n = \beta 2^\beta$. Les tailles possibles sont dans $I(p,n)$ et pour $r \in I(p,n)$,$n = e^c r 2^r$, il vient

$$E[X_r^{\text{NOYAU}_4}] \sim e^c r \, \frac{e^{rg(c,1/2)-\ln 2}}{\sqrt{2\pi r}}.$$

Pour tout $\varepsilon > 0$ et toute suite de $\mathbb{N}$ A tel que $|\lceil \beta \rfloor - \beta| > \varepsilon$, nous avons

$$\lim_{n\to\infty} E[X_r^{\text{NOYAU}_4}] = 0.$$

Soit $B = \{n = \beta 2^\beta | \beta \in \mathbb{N}\}$, nous avons $c = 0$ et comme $g(0,1/2) = \ln 2$, il vient $E[X_\beta^{\text{NOYAU}_4}] = \Theta(\sqrt{\beta})$. Calculons maintenant le moment d'ordre 2.

$$E[X_r^{Noyau_4}(X_r^{Noyau_4} - 1)] = \sum_{\substack{U_1, |U_1| = r \\ U_2, |U_2| = r \\ U_2 \neq U_1}} \sum_{\substack{s_1 \in V_n \setminus U_1 \\ s_2 \in V_n \setminus U_2}} \Pr((U_1, s_1) \text{ et } (U_2, s_2) \text{ sont des noyaux}_4\ )$$

Soient U_1 et U_2 tels que $|U_1| = |U_2| = \beta$ et $|U_1 \cap U_2| = l$. Si (U_1, s_1) et (U_2, s_2) sont des noyaux$_4$ alors s_1 et s_2 n'appartiennent pas à $U_1 \cup U_2$.

Nous avons par conséquent $(n - (2\beta - l))(n - (2\beta - l) - 1)$ couples (s_1, s_2) tels que s_1 et $s_2 \in V_n \setminus (U_1 \cup U_2)$ et avec $s_1 \neq s_2$ et $n - (2\beta - l)$ avec $s_1 = s_2$. Soit p_1 la probabilité que (s_1, u_1) et (u_1, s_1) soient des arcs de D_n pour tout $u_1 \in U_1$ et que (s_2, u_2) et (u_2, s_2) soient des arcs de D_n pour tout $u_2 \in U_2$ lorsque $s_1 \neq s_2$ et p_2 la même probabilité lorsque $s_1 = s_2$. On montre que l'on a $p_1 = 2^{-4\beta}$ $p_2 = 2^{-2(2\beta - l)}$. Nous avons

$$(n - (2\beta - l))(n - (2\beta - l) - 1)p_1 + (n - (2\beta - l))p_2 \sim n^2 2^{-4\beta}.$$

Ce qui implique

$$\begin{aligned} E[X_\beta^{Noyau_4}(X_\beta^{Noyau_4} - 1)] &\sim n^2 2^{-4\beta} E[X_\beta^{Noyau}(X_\beta^{Noyau} - 1) \\ &\sim n^2 2^{-4\beta} E[X_\beta^{Noyau}]^2. \\ &\sim E[X_\beta^{Noyau_4}]^2. \end{aligned}$$

On en déduit

$$\lim_{n\to\infty, n\in B} \Pr(X_r^{Noyau_4} > 0) = 1.$$

3.4 Transitions de phases sur les graphes denses

Rappelons que nous obtenons des graphes denses lorsque $D_n = \mathcal{D}(n,p)$ où p est constant. Pour les études précédentes sur les lois 0-1, nous avons toujours considéré la distribution uniforme, c'est-à-dire les cas $p = 1/2$. Nous avons tiré parti dans les différentes variantes de NOYAU de l'équilibre fragile entre la stabilité et la dominance en modifiant ces deux propriétés. Une autre étude naturelle consiste à changer la probabilité d'arc et de voir si cela modifie l'existence des noyaux. J'ai conçu une variante du noyau –TOURNOI$_1$– qui possède une transition de phase avec un seuil abrupt.

La propriété Tournoi$_1$

Définition 3.4.1. *d'un tournoi*
Un tournoi est une propriété définissable sur les graphes orientés $D_n = \langle V_n, A\rangle$, c'est un ensemble de sommets où toutes les paires de sommets sont connectées mais seulement dans un sens. C'est-à-dire $T \subset V_n$ est un tournoi lorsque pour toute paire $\{a, b\}$ de sommets de T,
- *soit $(a, b) \in A$ et $(b, a) \notin A$,*
- *soit $(b, a) \in A$ et $(a, b) \notin A$.*

Notons $T(q, r)$ la probabilité qu'un ensemble de cardinal r soit un tournoi.

$$T(q, r) = (2q(1-q))^{\binom{r}{2}}.$$

En prenant $q_T = \sqrt{2q(1-q)}$, il vient

$$T(q, r) = q_T^{r(r-1)} = S(q_T, r).$$

Définition 3.4.2. *d'un tournoi neutralisé*
Soit T un tournoi, nous dirons que T est neutralisé lorsque, pour tout $a \in V_n \setminus T$, il existe $b \in T$ tel que $(a, b) \in A$ et $(b, a) \in A$.

Soit $Neu(q, r)$ la probabilité qu'un tournoi de taille r soit neutralisé. Il vient

$$Neu(q, r, n) = (1 - (1 - p^2)^r)^{n-r}.$$

En posant $q_N = (1 - p^2)$, il vient

$$Neu(q, r, n) = (1 - q_N^r)^{n-r} = D(q_N, r, n).$$

Un graphe D_n vérifie la propriété Tournoi$_1$ lorsqu'il possède au moins un tournoi neutralisé.

Lorsque $q_T = q_N$, on retrouve exactement la propriété de Noyau sur un graphe de $\mathcal{D}(n, 1 - q_T)$.

$q_N = q_T$ correspond à l'équation $p^4 - 2p + 1 = 0$ qui admet une seule solution $\alpha = \frac{1}{3}\alpha' - \frac{2}{3}\frac{1}{\alpha'} - \frac{1}{3}$, où $\alpha' = (17 + 3\sqrt{33})^{\frac{1}{3}}$ dans l'intervalle $]0, 1[$, $\alpha \approx 0,5437$ [2].

J'ai montré en 2002 [45] que l'on a
- Tournoi$_1$ est *a.p.s.* vraie pour tout $p \in [\alpha, 1[$.
- Tournoi$_1$ est *a.p.s.* fausse pour tout $p \in]0, \alpha[$.

J'ai proposé la définition suivante pour un seuil croissant sur les graphes denses [45].

Définition 3.4.3. *Soit $\hat{p}$ une constante telle que $0 < \hat{p} < 1$. Nous dirons que $\hat{p}$ est une fonction de seuil croissante d'une propriété $\mathcal{P}$ lorsque*

$$\mu^p(\mathcal{P}) = \begin{cases} 0 \text{ pour } p < \hat{p} \\ 1 \text{ pour } p > \hat{p} \end{cases}$$

2. Des coquilles apparaissent pour le calcul de q_N, q_L et $Neu(q, r, n)$ dans la version finale, alors que les bonnes valeurs apparaissaient dans la version soumise aux referees

Nous avons la même définition pour un seuil décroissant en intervertissant 0 et 1.

Il est facile de voir que TOURNOI$_1$ admet, avec cette définition, une fonction de seuil croissante. La définition habituelle lorsque $\hat{p}$ n'est pas constant.

$$\mu^p(\mathcal{P}) = \begin{cases} i \text{ pour } p = o(\hat{p}) \\ j \text{ pour } \hat{p} = o(p) \end{cases}$$

Mais ici comme le seuil est une constante $\hat{p} = o(p)$ n'a pas de signification.

Soient $i \in]0,1[$ et p_i la probabilité définie par $\mu^p(\mathcal{P}) = i$. On note $\Delta_n(\varepsilon)$ le terme $p_{1-\varepsilon}(n) - p_\varepsilon(n)$. J'ai également montré que le seuil est abrupte, c'est-à-dire que $\Delta(\varepsilon) = o(\hat{p})$, plus précisément nous avons $\Delta(\varepsilon) = \Theta\left(\frac{1}{\ln n}\right)$.

Définition d'une transition de phase en probabilité

Cette partie est détaillée car elle ne fera pas l'objet d'une publication. L'objectif est de définir un cadre rigoureux pour définir une transition de phase pour une propriété non monotone.

Même si la définition proposée dans [45] du seuil sur les graphes denses est assez naturelle, elle a entraîné des objections chez certains chercheurs qui estiment que par définition une transition de phase ne peut être définie que pour une propriété monotone. Cependant leurs objections proviennent plus des méthodes de preuve qu'ils utilisent et qui fait intervenir de manière centrale la monotonie que d'une définition générale des transitions de phase. D'ailleurs la plupart du temps la définition de transition de phase est ad hoc, le choix de la taille et du poids apparaît comme évident (et non discuté) pour la propriété considérée.

J'ai donc recherché une définition générale qui puissent englober toutes les études (3-SAT, 3-COLORIAGE, TSP...). Il n'y a a priori pas de définition évidente car les paramètres varient d'un problème à un autre.

Je reprends ici la définition générale de Paul E. Dunne, Alan Gibbons et Michele Zito[13] qui me semble très convaincante et appropriée aux propriétés que j'ai étudiées.

On définit une fonction $f : I \to \{0,1\}$ qui est associée à un problème de décision,où I est l'ensemble des instances qui sont codées dans un alphabet contenant au moins deux symboles.

Plus précisément, on part d'une propriété $\mathcal{P}$ sur I et pour tout x de I et f est définie par $f(x) = 1$ si et seulement si x vérifie la propriété $\mathcal{P}$.

Soit $x \in I$, on note $|x|$ le nombre de symboles de x.

Ici I sera l'ensemble des graphes orientés.

Soit $D =< V, A >$, avec V l'ensemble des sommets et A l'ensemble des arcs. D peut être codé dans l'alphabet $\Sigma = \{0,1\}$ par la chaîne $x_{1,2}, \ldots, x_{n-1,n}$, où

$$x_{i,j} = \begin{cases} 1 \text{ si } \{i,j\} \text{ est un arc de } G. \\ 0 \text{ sinon} \end{cases}$$

Définition de la taille et du poids

Pour f fixée, on définit deux paramètres σ et τ.

$\sigma : I \to N$, la taille de l'instance
$\tau : I \to N$, le poids de l'instance

On impose les conditions suivantes sur ces deux paramètres :

1. $\exists \varepsilon > 0\ k \in N\ \forall x \in I |x|^{\varepsilon} \leq \sigma(x) \leq |x|^k$.
2. $\exists k \in N\ \forall x \in I\ \tau(x) \leq |x|^k$.
3. $\exists k \in N\ \forall n\ |\{m : \sigma(x) = n \text{ et } \tau(x) = m\}| \leq n^k$.

Les deux premières conditions signifient que la taille et le poids ne doivent être ni trop grands ni trop petits par rapport à la longueur de l'instance.
La dernière condition signifie qu'il ne doit pas y avoir trop d'instances de même poids.

Exemples

- Pour un graphe non-orienté (resp. orienté), on peut prendre comme taille le nombre de sommets et comme poids le nombre d'arêtes (resp. arcs).
- Pour une formule CNF, la taille est le nombre de variables et le poids le nombre de clauses.

Phénomène de seuil ou transition de phase en probabilité
Soit (f, σ, τ), I_n l'ensemble des instances de I de taille n et $I_{n,m}$ l'ensemble des instances de taille n et de poids m.

On définit $\mu_f(n, m)$ par

$$\mu_f(n, m) = \frac{|\{x \in I_{n,m} : f(x) = 1\}|}{|I_{n,m}|}.$$

Phénomène de seuil (transition de phase en probabilité faible)
(f, σ, τ), I_n vérifie un phénomène de seuil lorsqu'il existe une fonction strictement croissante $\varphi(n)$ telle que

$$\lim_{n \to +\infty} \mu_f(n, \psi(n)) = \begin{cases} 0 \text{ si } \psi(n) = o(\varphi(n)) \\ 1 \text{ si } \psi(n) = \omega(\varphi(n)). \end{cases}$$

Le seuil est dit abrupt (sharp) lorsqu'il existe une constante $c > 0$, telle que, pour tout $\varepsilon > 0$,

$$\lim_{n \to +\infty} \mu_f(n, \lfloor (c - \varepsilon)\ \varphi(n) \rfloor) = 0 \text{ (resp.1)}$$

$$\lim_{n \to +\infty} \mu_f(n, \lfloor (c + \varepsilon)\ \varphi(n) \rfloor) = 1 \text{ (resp.0)}$$

Remarque 3.4.1. *c peut être une quasi-constante (c oscille dans un petit intervalle). Par exemple,* $c = \lceil \log(n) \rceil - \log(n)$.

Pour tout $\varepsilon > 0$, on définit m_ε et $m_{1-\varepsilon}$ par

$$m_\varepsilon = Sup\ (\ m | \mu_f(n, m) \leq \varepsilon)$$

$$m_{1-\varepsilon} = Inf\ (m | \mu_f(n, m) \geq 1 - \varepsilon)$$

Posons $\Delta(\varepsilon) = \lim_{n\to+\infty} m_{1-\varepsilon} - m_\varepsilon$, lorsque cette limite existe.

Le seuil est dit doux (coarse) lorsque

$$\Delta(\varepsilon) = \Theta(\varphi(n)).$$

et abrupt(sharp) lorsque

$$\Delta(\varepsilon) = o(\varphi(n)).$$

Fonctions monotones

Soit un problème de décision $f : \{0,1\}^* \to \{0,1\}$.

- f est dite monotone croissante (resp. décroissante) lorsque pour tout $\alpha \in \{0,1\}^*$ et tout $\beta \in \{0,1\}^*$ obtenu en remplaçant un 0 (resp. 1) de α par un 1 (resp. 0), si $f(\alpha) = 1$ alors $f(\beta) = 1$.
- f est dite triviale lorsque $\forall x\ f(x) = 0$ ou $\forall x\ f(x) = 1$.
- Soit f un problème de décision sur les graphes. f est symétrique lorsque $f(G) = 1$ si et seulement si $f(H) = 1$ pour tout isomorphisme H de G.

Bollobás et Thomason ont montré [7] que si f est un problème de décision non trivial monotone alors elle admet une transition de phase faible.

De plus, Friedgut et Kalai ont montré [21] que tout problème de décision symétrique non trivial monotone sur les graphes f possède une transition de phase abrupt (sharp).

Exemples

1. (3-SAT, n, m) admet possède une transition de phase abrupte avec n nombre de variables, m nombre de clauses, $\varphi(n) = n$ et $c \approx 4,26$.
2. Notons HAM le problème de décision pour un graphe d'avoir un chemin hamiltonien pour un graphe non-orienté. (HAM, n, m) admet une transition de phase abrupte où n est le nombre des sommets et m est le nombre d'arêtes, $\varphi(n)$ est la fonction $n\log(n)$.

$$\lim_{n\to+\infty} \mu_f(n, \lfloor(0,5-\varepsilon)\ \varphi(n)\rfloor) = 0$$

$$\lim_{n\to+\infty} \mu_f(n, \lfloor(0,5+\varepsilon)\ \varphi(n)\rfloor) = 1$$

3. Soit $k \geq 2$, notons ISO_k le problème de décision pour un graphe d'avoir k sommets isolés. (ISO_k, n, m) admet possède une transition de phase abrupte où n nombre des sommets, m nombre d'arêtes, $\varphi(n) = n\log(n)$.

$$\lim_{n\to+\infty} \mu_f(n, \lfloor(0,5-\varepsilon)\ \varphi(n)\rfloor) = 0$$

$$\lim_{n\to+\infty} \mu_f(n, \lfloor(0,5+\varepsilon)\ \varphi(n)\rfloor) = 1$$

Comme le remarque Dunne et al, on peut obtenir une transition de phase abrupte pour une propriété non-monotone. Ils proposent ainsi ISO-HAM_k le problème de décision pour un graphe G d'avoir un sous-ensemble de sommets W tel que $|W| \geq n - k$ et le sous-graphe induit par W est hamiltonien. Il est facile de voir que cette propriété n'est pas monotone.

(ISO-HAM_k, n, m) admet possède une transition de phase abrupte où n est le nombre des sommets, m est le nombre d'arêtes et $\varphi(n)$ est la fonction $n\log(n)$.

Transition de phase pour Tournoi$_1$
Nous allons maintenant raffiner les définitions proposées dans [45]. On choisit pour n le nombre des sommets et pour m le nombre d'arêtes.

Soit f le problème de décision associé à (TOURNOI$_1$, n, m).
Nous avons pour tout $\varepsilon > 0$:

$$\lim_{n\to+\infty} \mu_f(n, \alpha - \varepsilon) = 0 \quad \lim_{n\to+\infty} \mu_f(n, \alpha) = 1 \quad \lim_{n\to+\infty} \mu_f(n, \alpha + \varepsilon) = 1$$

Malheureusement $\varphi(n) = \alpha$ ne convient pas pour la définition générale d'une transition de phase. $\varphi(n)$ doit être une fonction strictement croissante.

On choisit une autre définition pour le poids. $\tau(D_n) = m = \alpha\ n\ (n-1) - k$, où k est le nombre d'arcs.
On restreint notre étude à

$$0 \leq m \leq \frac{\alpha}{2}\ n\ (n-1).$$

C'est-à-dire

$$\frac{\alpha}{2}\ n\ (n-1) \leq k \leq \alpha\ n\ (n-1).$$

On définit ensuite

$$\varphi(n) = \frac{n(n-1)}{\ln n}.$$

On montre alors que

$$\lim_{n\to+\infty} \mu_f(n, \psi(n)) = 0 \text{ si } \psi(n) = o\ (\varphi(n)).$$

$$\lim_{n\to+\infty} \mu_f(n, \psi(n)) = 1 \text{ si } \psi(n) = \omega\ (\varphi(n)).$$

En effet, pour tout $i \in]0, 1[$, on note $\varphi_i(n)$ une fonction telle que

$$\mu_f(n, \varphi_i(n)) = i + o(1).$$

Pour tout $\varphi_i(n)$, on prouve que l'on a

$$\varphi_i(n) = \Theta\Big(\frac{n(n-1)}{\ln n}\Big) = \Theta\Big(\varphi(n)\Big).$$

Soit $\psi(n) = o(\varphi(n))$. On a alors $\mu_f(n, \psi(n)) = o(1)$.
Inversement si $\psi(n) = \omega(\varphi(n))$, on a $\mu_f(n, \psi(n)) = 1 + o(1)$.

Pour montrer que le seuil est abrupte, nous devons calculer de manière plus précise $\varphi_i(n)$. La taille de tournoi la plus probable est

$$m = \log_{1/q_N} n - \log_{1/q_N} \log_{1/q_N} n.$$

Je montre dans [45] qu'il existe une quasi-constante $\delta(n)$, $-1 \leq \delta(n) < 1$, telle que

$$\varphi_i(n) = \left(\delta(n) + \Theta\Big(\frac{1}{n}\Big)\right)\varphi(n).$$

Ce qui entraîne $\Delta(\varepsilon) = o(\varphi(n))$.

Transition de phase en complexité
En général, une transition de phase en probabilité pour une propriété NP-complète s'accompagne d'un changement de complexité pour les algorithmes décidant cette propriété.

L'étude de TOURNOI$_1$ dans [45] ne contenait que des résultats asymptotiques. Je présente ici une étude expérimentale de la complexité ; l'algorithme utilisé est une adaptation de COLORIAGEBACKTRACKING (voir le chapitre 4 sur la recherche de noyaux).

Les deux figures ci-dessous donnent sur 100 itérations respectivement le pourcentage de graphes $D_n \in \mathcal{D}(n,p)$, avec $n = 200$, qui possèdent un tournoi neutralisé et le temps moyen (en secondes) pour le décider.

FIGURE 3.1 – Probabilité d'obtenir un tournoi neutralisé

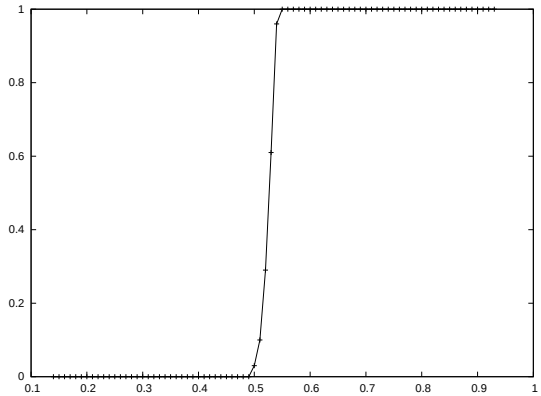
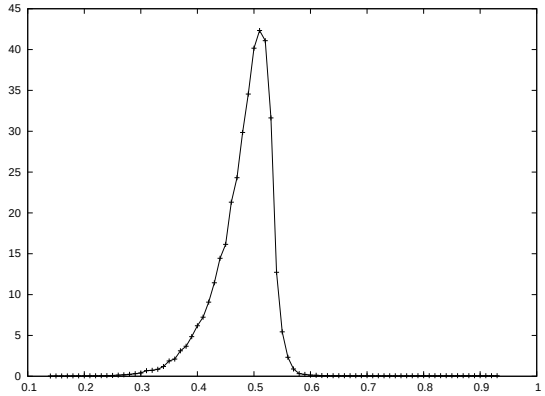


FIGURE 3.2 – Temps nécessaire pour trouver un tournoi neutralisé



Nous conjecturons dans le chapitre 4 que les algorithmes connus recherchant un noyau sont de complexité superpolynomial ($n^{\log n}$). Nous aurions donc la même complexité pour TOURNOI$_2$ lorsque p est proche de α.

3.5 Existence des noyaux dans les graphes creux

J'ai entamé cette étude avec Marco Illengo en 2011. Ce travail a donné lieu à un préprint, il n'a pas encore été publié.

On parle généralement de graphes creux (sparse en anglais) lorsque le nombre d'arêtes m est en $O(n)$. Ici nous considérerons le cas où $m = \Theta(n)$, cela correspond donc aux graphes aléatoires de $\mathcal{D}(n, p)$ où $p = c/n$,pour c constant.

Les résultats sont vraiment très différents de ceux du cas p constant. L'objectif initial était de trouver pour quels paramètres $p(n)$ il était difficile de trouver un noyau dans un graphe aléatoire (voir la section *Cacher un noyau dans un graphe aléatoire*).

Généralement la méthode du premier moment est relativement simple car elle ne fait pas intervenir de corrélation entre les sous-ensembles de sommets, alors qu'ici elle a nécessité des calculs aussi complexes que ceux que l'on rencontre d'habitude pour le second moment. C'est dû principalement au fait qu'aucune taille n'est probable et qu'il faut donc sommer sur plusieurs tailles l'espérance du nombre de noyaux.

Rappelons tout d'abord que l'on a

$$E[X_r^{Noyau}] = \binom{n}{r} S(r,q)\ D(n,r,q),$$

avec

$$S(q,r) = q^{r(r-1)} \quad D(q,n,r) = (1-q^r)^{n-r}.$$

Pour $p = c/n$ et $r = \mu n$, nous avons prouvés les asymptotiques suivants

$$\begin{array}{rcrcl} \binom{n}{\mu n}/n & \sim & \beta(c,\mu) & = & -\mu\log\mu - (1-\mu)\log(1-\mu), \\ S(\mu n, 1-c/n)/n & \sim & \iota(c,\mu) & = & -c\mu^2, \\ D(n,\mu n, 1-c/n)/n & \sim & \delta(c,\mu) & = & (1-\mu)(1-e^{-c\mu}), \end{array}$$

L'étude des tailles possibles s'effectue en trois étapes

3.5.1 Une seule densité possible

Théorème 3.5.1. *Tous les noyaux de D_n sont a.p.s. tous de taille*

$$r = \mu_c n + o(n),$$

où $c = -\frac{\log\mu_c}{\mu_c}$. Ou encore, $\mu_c = \frac{W(c)}{c}$, où W est la fonction de Lambert qui est la réciproque de la fonction f définie par $f(w) = w\exp(w)$.

3.5.1.1 Aucune taille probable

Théorème 3.5.2. *Pour toute suite $r(n)$, D_n n'a a.p.s. pas de noyau de taille r.*

3.5.2 Somme des espérances

Théorème 3.5.3. *L'espérance du nombre de noyaux de densité μ_c est supérieure à 1.*

Preuve. (Idée de la preuve)
Soient $\mu = \mu_c$, $f = o(n^{\frac{2}{3}})$ et $|x| \leq f$.
On considère l'espérance du nombre de noyaux de taille $r = \mu n + x$.

$$E[X_r^K] = \frac{e^{-\eta\frac{x^2}{n}}}{\zeta\sqrt{n}}(1 + o(1)),$$

où $\eta = \dfrac{(1-\log\mu)^2}{2\mu(1-\mu)}, \zeta = \mu\sqrt{2\pi\mu(1-\mu)}$.

On somme alors les espérances sur $f \in \omega(\sqrt{n})$,

$$\begin{array}{rcl} \lim\limits_{n\to\infty} \sum\limits_{|x|\leq|f|} E[X_r^K] & = & \dfrac{1}{\zeta}\displaystyle\int_{-\infty}^{\infty} e^{-\eta y^2} dy \\ & = & \dfrac{\sqrt{\pi}}{\zeta\sqrt{\eta}} \\ & = & \dfrac{\frac{1}{\mu}}{1+\log\frac{1}{\mu}} \\ & > & 1. \end{array}$$

□

Depuis Marco Illengo a affiné ce résultat en étudiant également le second moment. Il a obtenu le remarquable théorème suivant

Théorème 3.5.4. *Soit $D \in \mathcal{D}(n,p)$, avec $p = p(n) = c/n$, pour une certaine constante $c > 0$. Pour $c \neq e$, la probabilité asymptotique de la propriété* Noyau *existe et est différente de* 0.

Chapitre 4

Aspects algorithmiques et applications

4.1 Noyaux et jeux à deux joueurs

Le noyau intervient de manière centrale en théorie des jeux qui satisfont les conditions suivantes

- le jeu est fini, c'est-à-dire le nombre de coups maximal est connu au début
- le jeu se joue à deux joueurs
- le jeu est sans information cachée
- toute partie a un vainqueur (pas de match nul possible)

Ernst Zermelo a prouvé en 1913 que pour un tel jeu, un des deux joueurs possède une stratégie gagnante. Ces travaux ont été ensuite repris par John von Neumann et Oskar Morgenstern [46].

On peut modéliser l'ensemble des parties par un graphe orienté où les sommets sont les configurations successives au cours des parties et les arcs sont les transitions de configuration entre deux coups. Nous obtenons ainsi un DAG, un graphe orienté sans circuit. Il est facile de vérifier que l'existence d'un circuit contredirait le fait que le jeu est fini. Dans le cas d'un jeu direct, le joueur qui ne peut plus jouer –il se trouve sur un puits, un sommet de degré sortant égal à 0– a perdu. Dans le cas d'un jeu indirect, il faut au contraire forcer l'adversaire à jouer vers un puits. Neumann et Morgenstern ont prouvé que tout DAG possédait un unique noyau. L'ensemble des positions gagnantes dans un jeu forment l'unique noyau d'un DAG dans le cas d'un jeu direct.

Supposons que ce soit le joueur 1 qui possède une stratégie gagnante, alors celle-ci consiste à jouer à chaque tour vers une configuration du noyau. Richardson a ensuite montré [50] qu'un graphe sans circuit de longueur impair possédait également toujours au moins noyau. Mais il peut posséder plusieurs noyaux. Par exemple, on montre facilement que le graphe $D_4 = \langle V_4, A\rangle$, où $A = \{(1,2),(2,3),(3,4),(4,1)\}$, possède deux noyaux qui sont $\{1,3\}$ et $\{2,4\}$.

Un graphe qui possède des circuits de longueur impair peut ne pas avoir de noyau. Ainsi le graphe $D_3 = \langle V_3, A\rangle$, où $A = \{1,2),(2,3),(3,1)\}$, ne possède pas de noyau.

Cependant, en ajoutant un sommet 4 et un arc entre le sommet 1 et le sommet 4, le graphe possède alors un unique noyau.

Claude Berge a consacré [4] un chapitre à cette propriété et il a montré avec Pierre Duchet [5] une correspondance entre les graphes parfaits et les noyaux.

De nombreux jeux appartiennent à cette famille de jeux à deux joueurs comme le jeu de Nim et ses nombreuses variantes. Dans le jeu de Nim initial, on dispose d'un nombre fixé de tas finis d'allumettes et chaque joueur peut retirer 1, 2 ou 3 allumettes, le premier à retirer la dernière allumette a perdu. Il est facile pour ce jeu de déterminer la stratégie gagnante. Cependant, il faut bien voir que ce n'est pas parce qu'il existe toujours une stratégie gagnante pour un des deux joueurs que cette stratégie est facile à déterminer. Les capacités calculatoires des deux joueurs sont telles qu'elles excluent la possibilité d'explorer complètement ou même partiellement le DAG des configurations du jeu.

Nous verrons dans le chapitre 4 comment varie la difficulté de trouver un noyau dans un graphe aléatoire de $\mathcal{D}(n,p)$ en fonction de la valeur de p.

4.2 Noyaux sur les arbres, les DAG et les graphes ayant peu de circuits

4.2.1 Algorithme de recherche de noyau sur les arbres et les DAG

Les arbres et les DAG possèdent un seul noyau, nous pouvons donc proposer des algorithmes déterministes. La méthode consiste à colorier en rouge les sommets qui seront dans le noyau et en vert ceux qui seront hors du noyau. Le coloriage s'effectue à partir des puits (sommets qui n'ont pas d'arc sortant) qui sont les premiers sommets à être mis dans le noyau. On propage ensuite le coloriage avec des règles locales : un sommet en rouge a tous ses voisins coloriés en vert et un sommet qui a tous ses voisins sortant coloriés en vert est colorié en rouge. Ces deux règles suffisent pour effectuer un coloriage complet pour les DAG.

Le coloriage s'effectue en trois couleurs : blanc, rouge et vert.

BLANC Les sommets en blanc sont les sommets non traités.

ROUGE Les sommets en rouge sont les sommets que nous avons mis dans le noyau.

VERT Les sommets en vert sont les sommets que nous avons mis en dehors du noyau.

Pour maintenir la cohérence du coloriage, il suffit de vérifier des contraintes locales.

- il n'y a pas d'arc entre deux sommets coloriés en rouge.
- tout sommet colorié en vert peut atteindre un sommet colorié en rouge.

Avec une structure de données adaptée pour coder le graphe (tableau des listes des voisins sortants) et l'utilisation d'une file pour stocker les sommets blancs, nous obtenons un algorithme linéaire en la taille du graphe (nombre de sommets + nombre d'arcs) car chaque sommet est visité le nombre de fois correspondant à son degré sortant.

Notons $V^{in}(a) = \{b \in \text{BLANC} | (b,a) \in A\}$ et $V^{out}(a) = \{b \in \text{BLANC} | (a,b) \in A\}$.

Algorithme 1 *ColoriageDAG*

```
pour tout a ∈ BLANC faire
    enqueue(QUEUE,a)
tant que QUEUE ≠ ∅ faire
    a ← dequeue(QUEUE)
    si (a ∈ BLANC) et (V^out(a) ∩ WHITE = ∅) alors
        BLANC ← BLANC \ {a}
        ROUGE ← ROUGE ∪ {a}
        pour tout b ∈ V^in(a) faire
            si b ∈ BLANC alors
                BLANC ← BLANC \ {b}
                VERT ← VERT ∪ {b}
                pour tout c ∈ V^in(b) faire
                    enqueue(QUEUE,c)
```

4.2.2 Noyau dans les arbres non-étiquetés

Cette partie utilise les méthodes issues de la combinatoire analytique. Nous ne donnerons ici que les définitions nécessaires pour comprendre les résultats. Pour une bonne introduction à ce domaine, je conseille fortement la lecture du livre de Philippe Flajolet et Robert Sedgwick [19].

Nous allons tout d'abord nous intéresser aux arbres non étiquetés.

Une classe combinatoire $\mathcal{A}$ est un ensemble d'objets muni d'une fonction de taille $|\cdot|$. Nous devons aussi avoir un nombre fini d'objets de même taille.

La SGO (série génératrice ordinaire) de $\mathcal{A}$ vaut par définition

$$A(z) = \sum_{n\in\mathbb{N}} A_n z^n,$$

où A_n est le nombre d'objets de taille n.

En prenant comme taille le nombre de nœuds, notons $\mathcal{T}$ la classe des arbres planaires non-étiquetés enracinés ayant au moins un nœud. Sa SGO vaut

$$T(z) = \frac{1-\sqrt{1-4z}}{2} = \sum_{n\in\mathbb{N}} T_n z^n,$$

où T_n est égal au nombre de Catalan $C_{n-1} = \frac{\binom{2n-1}{n-1}}{n}$.

Soient $\mathcal{R}$ la classe des arbres de $\mathcal{T}$ dont la racine est coloriée en rouge et $\mathcal{V}$ la classe des arbres de $\mathcal{T}$ dont la racine est coloriée en vert. Soient $R(z)$et $V(z)$ les SGO de respectivement $\mathcal{R}$ et $\mathcal{V}$. Nous avons les équations ensemblistes suivantes liant ces trois classes combinatoires,

$$\begin{aligned} \mathcal{T} &= \mathcal{R} \cup \mathcal{V} \\ \mathcal{R} &= \{\cdot\} \times Seq(\mathcal{V}) \end{aligned}$$

$\cup$ désigne ici la réunion disjointe, $\cdot$ désigne l'arbre réduit à une racine, la première équation signifie que $(\mathcal{R}, \mathcal{V})$ réalise une partition de $\mathcal{T}$ et la seconde équation qu'un arbre de $\mathcal{R}$ est une racine reliée à une séquence d'arbres de $\mathcal{V}$.

Opération sur les classes				*SGO*		
Réunion	$\mathcal{A}$	$=$	$\mathcal{B}+\mathcal{C}$	$A(z)$	$=$	$B(z)+C(z)$
Produit cartésien	$\mathcal{A}$	$=$	$\mathcal{B}\times\mathcal{C}$	$A(z)$	$=$	$B(z)\cdot C(z)$
Séquence	$\mathcal{A}$	$=$	$SEQ(\mathcal{C})$	$A(z)$	$=$	$\dfrac{1}{1-C(z)}$

FIGURE 4.1 – Méthode du dictionnaire sur les classes non étiquetées

En utilisant la méthode du dictionnaire (voir figure 4.1 contenant les correspondances considérées) qui interprète les relations ensemblistes par des relations sur les séries génératrices, nous obtenons les équations fonctionnelles suivantes

$$T(z) \quad = \quad R(z)+V(z) \quad et \quad R(z) \quad = \quad \frac{z}{1-V(z)}$$

Et l'unique solution satisfaisant $V_1 = 0$ est

$$V(z) = \frac{3-\sqrt{1-4z}-\sqrt{2+12z+2\sqrt{1-4z}}}{4}.$$

On obtient

$$\frac{V_n}{T_n} = \frac{5+\sqrt{5}}{10} \quad \frac{R_n}{T_n} = \frac{5-\sqrt{5}}{10}.$$

Nous avons

$$\frac{V_n}{T_n} \approx 0,7236$$

et le premier joueur possède une stratégie gagnante si et seulement si la racine de l'arbre est verte (il peut alors jouer sur un sommet rouge). Ainsi, si l'on tire aléatoirement avec la distribution uniforme un arbre de taille n, nous avons 72% de chances que ce soit le joueur 1 qui ait une stratégie gagnante. Ce résultat n'était pas du tout évident au départ. Notons que V_n est aussi le nombre de façons de parenthéser $0+0+\ldots 0$ afin d'obtenir 1 avec, pour les règles

$$0+0=1 \quad 1+0=1 \quad 0+1=0 \quad 1+1=1.$$

Nous pouvons conduire la même étude sur les arbres binaires. La SGO des arbres binaires vaut

$$T(z) = \frac{1-\sqrt{1-4z}}{2z},$$

Les équations ensemblistes deviennent

$$\begin{array}{rcl} \mathcal{T} & = & \mathcal{R}\cup\mathcal{V} \\ \mathcal{R} & = & \{\cdot\}\times(\varepsilon+\mathcal{V}\times\varepsilon+\varepsilon\times V+\mathcal{V}\times\mathcal{V}), \end{array}$$

où ε désigne l'arbre vide. Avec la méthode du dictionnaire, on obtient les équations fonctionnelles suivantes

$$T(z) \quad = \quad R(z)+V(z) \quad et \quad R(z) \quad = \quad z(V(z)+1)^2.$$

Opération sur les classes				SGE		
Réunion disjointe	$\mathcal{A}$	=	$\mathcal{B}+\mathcal{C}$	$A(z)$	=	$B(z)+C(z)$
Composition	$\mathcal{A}$	=	$\mathcal{B}\star\mathcal{C}$	$A(z)$	=	$B(z)\times C(z)$
Séquence de longueur fixée	$\mathcal{A}$	=	$SEQ_k(\mathcal{C})$	$A(z)$	=	$C(z)^k$
Ensemble de taille non fixé	$\mathcal{A}$	=	$SET(\mathcal{C})$	$A(z)$	=	$\exp(C(z))$

FIGURE 4.2 – Méthode du dictionnaire sur les classes étiquetées

$R(z)$ est l'unique solution de l'équation

$$x^2 - (2zT(z) + \frac{1}{z})x + T(z)^2 = 0$$

telle que $R_1 = 1$ et nous obtenons comme rapport entre le nombre d'arbres verts sur le nombre total d'arbres binaires à n nœuds

$$\frac{V_n}{T_n} \approx 0,577.$$

Ce qui donne un jeu plus équilibré que le cas précédent. V_n est le nombre de parenthésages de $0+0+\ldots 0$ donnant 0 avec les règles

$$0+0=1 \quad 0+1=1+0=1+1=0$$

Dans les deux cas, V_n est répertoriée dans l'encyclopédie des suites de Sloane (respectivement A055113 et A055392).

4.2.3 Noyau dans les arbres orientés étiquetés

Pour une classe combinatoire étiqueté, un objet de taille n est composé d'atomes numérotés de 1 à n (ici les atomes seront les nœuds de l'arbre ou les sommets d'un graphe).

La SGE (série génératrice exponentielle) de $\mathcal{A}$ vaut par définition

$$A(z) = \sum_{n\in\mathbb{N}} \frac{A_n}{n!} z^n,$$

où A_n est le nombre d'objets de taille n.

Pour obtenir le nombre d'arbres non orientés de racine rouge ou vert étiquetés, il suffit de multiplier les nombres précédents par $n!$ (nombre d'étiquetages possibles).

J'ai étudié en 2004 avec Cyril Banderier et Vlady Ravelomanana les arbres orientés étiquetés [3].

La méthode du dictionnaire pour le cas étiqueté suit le même principe que dans le cas non étiqueté, la séquence est remplacée par l'ensemble (voir figure 4.2) contenant les correspondances considérées).

On montre facilement que la SGO des arbres orientés enracinés vaut

$$T(z) = C(2z)/2,$$

où $C(z)$ est la fonction de Cayley $C(z) = z\ exp(C(z))$. Comme nous voulons attacher des arbres orientés étiquetés entre-eux, nous allons considérer une orientation sur la racine pour savoir l'orientation de l'arc qui partira de ce sommet pour se connecter à un autre sommet, nous coderons cette orientation avec une flèche $\uparrow$ ou $\downarrow$. Soit T la classe des arbres orientés étiquetés, R la classe des arbres de racine rouge, V celle des arbres de racine verte. On suppose que l'on colorie T avec la flèche $\uparrow$, on considère les deux orientations pour R et V et on ajoute les deux orientations. Pour les arbres verts, on ajoute l'arbre $B^{\uparrow}$ qui est un arbre vert qui n'est pas encore connecté à un sommet rouge (il devra nécessairement être connecté à un sommet rouge). Un ensemble d'objet d'une classe A est codé par $SET(A)$. On note v et r un sommet colorié respectivement en vert et rouge.

On obtient le système

$$\left\{\begin{array}{lcl} T & = & V^{\uparrow} \cup R^{\uparrow} \\ V^{\uparrow} & = & v^{\uparrow} \times R^{\uparrow} \times Set(V^{\uparrow} \cup V^{\downarrow} \cup R^{\uparrow} \cup R^{\downarrow}) \\ V^{\downarrow} & = & v^{\downarrow} \times R^{\uparrow} \times Set(V^{\uparrow} \cup V^{\downarrow} \cup R^{\uparrow} \cup R^{\downarrow}) \\ R^{\uparrow} & = & r^{\uparrow} \times Set(V^{\downarrow} \cup V^{\uparrow}) \\ R^{\downarrow} & = & r^{\downarrow} \times Set(V^{\downarrow} \cup V^{\uparrow}) \\ B^{\uparrow} & = & v^{\uparrow} \times Set(V^{\uparrow} \cup V^{\downarrow} \cup R^{\uparrow} \cup R^{\uparrow}) \end{array}\right.$$

Soient $T(z), V^{\uparrow}(z), V^{\downarrow}(z), R^{\uparrow}(z), R^{\downarrow}$ et $B^{\uparrow}(z)$ les SGO respectives de $T, V^{\uparrow}, V^{\downarrow}, R^{\uparrow}, R^{\downarrow}, B^{\uparrow}$.

En utilisant la méthode du dictionnaire, nous obtenons

$$\left\{\begin{array}{lcl} T(z) & = & V^{\uparrow}(z) + R^{\uparrow}(z) \\ V^{\uparrow}(z) & = & V^{\downarrow}(z) = zR^{\uparrow}(z)\exp(R^{\uparrow}(z) + V^{\uparrow}(z) + V^{\uparrow}(z)) \\ R^{\uparrow}(z) & = & R^{\uparrow}(z) = z\exp(V^{\uparrow}(z) + B^{\uparrow}(z)) \end{array}\right.$$

Nous obtenons finalement sur les SGE non fléchées,

$$\left\{\begin{array}{lcl} T(z) & = & R(z) + V(z) \\ V(z) & = & z\ exp(2T(z)) - z\ exp(T(z) + V(z)) \\ R(z) & = & z\ exp(V(z) + T(z)) \end{array}\right.$$

L'unique solution de ce système nous donne

$$R(z) = -C(-C(2z)/2) \qquad V(z) = C(2z)/2 + C(-C(2z)/2).$$

Au niveau asymptotique, on montre que

$$\frac{1-\lambda}{1+\lambda} \approx 47,95\%$$

des arbres étiquetés orientés enracinés ont une racine verte, où λ est l'unique solution réelle de l'équation $2\lambda = exp(-\lambda)$.

Nous avons obtenu plusieurs autres résultats portant sur le coloriage de graphe ayant un seul circuit ou un seul cycle que nous appellerons respectivement graphes unicircuit et graphes unicycle.

La classe des graphes unicircuit a pour SGO

$$U(z) = T(z) - T(z)^2 - V(z) + \ln\left(\frac{1}{1-(V(z)+T(z)R(z))}\right)$$

Nous avons

$$\frac{1}{2} + \frac{1}{2\sqrt{5}} \approx 73,37\%$$

des sommets d'un graphe unicircuit qui possède un noyau sont verts. Donc si le joueur 1 tire aléatoirement un sommet pour déterminer la position de départ, il a $73,37\%$ de chances d'avoir une stratégie gagnante. D'autre part, $92,65\%$ des graphes unicircuit possède un noyau.

Pour les graphes unicycle, la probabilité de ne pas pouvoir le colorier est de $0,05n^{-1/2}$, donc un graphe unicycle possède *a.p.s.* un noyau.

Cette approche par la combinatoire analytique semble difficile à poursuivre lorsque l'on considère plusieurs cycles ou circuits. Banderier a obtenu d'autres résultats sur les arbres en considérant simultanément plusieurs paramètres : la taille du noyau, le *node independance number* (la taille maximum d'un stable de l' arbre) et le *path node-covering number* (le plus petit nombre de chemins couvrant l'arbre) [2].

4.2.4 Transitions de phase pour ColoriageDAG

Cet algorithme ColoriageDAG peut être utilisé pour tout graphe de manière préliminaire pour colorier une partie du graphe. Il est ainsi possible de casser certains circuits lorsqu'il n'y a pas trop de circuits.

On peut comparer le pourcentage de sommets coloriés par l'algorithme ColoriageDAG avec le pourcentage de sommets coloriés par l'algorithme ColoriageArbre suivant. On colorie en noir tous les sommets atteignant un puits et n'appartenant pas à un circuit. Au départ tous les sommets sont coloriés en blanc et tant qu'il reste un puits, on colorie tous les puits en noir et on élimine tous ces sommets du graphe.

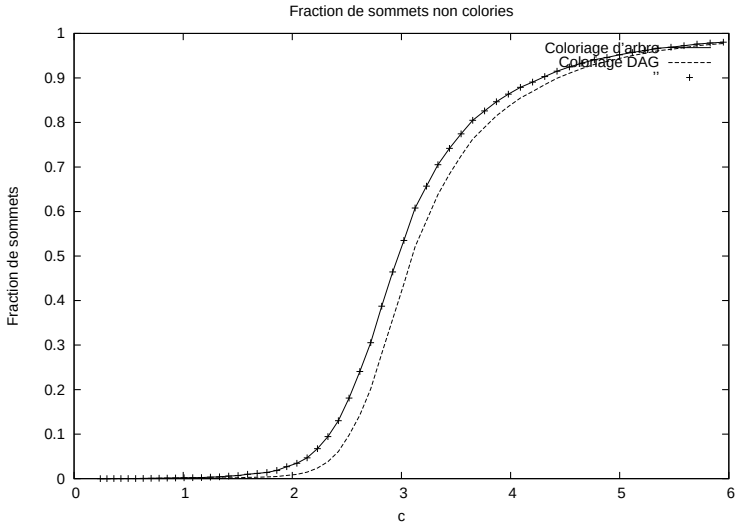
Algorithme 2 *ColoriageArbre*

pour tout $a \in$ Blanc **faire**
  enqueue(queue,a)
tant que queue $\neq \emptyset$ **faire**
  $a \leftarrow$ **dequeue**(queue)
  si $(a \in$ Blanc$) \bigwedge (V^{out}(a) \cap$ Vert $= \emptyset)$ **alors**
    Blanc $\leftarrow$ Blanc $\setminus \{a\}$
    Black $\leftarrow$ Black $\cup \{a\}$

Dès 2004, lorsque je travaillais avec C. Banderier et V. Ravelomanana, j'ai observé le comportement de ces deux algorithmes sur les graphes aléatoires en faisant varier la probabilité d'arc $p = c/n$, où c est une constante.

J'ai en particulier observé une transition sur le nombre de sommets coloriés, sans néanmoins obtenir de résultat théorique pouvant l'expliquer. Pour $c < 1$, l'algorithme ColoriageArbre colore presque toujours tous les sommets et il faut attendre $c > e$

FIGURE 4.3 – Coloriage d'arbres et de DAG sur $\mathcal{D}(n, c/n)$



pour qu'une fraction non négligeable de sommets ne soient pas coloriés par COLORIAGEDAG.

Plus récemment, en 2011, Marco Illengo a eu l'idée d'étudier comme un système dynamique w, la densité de sommets non coloriés pendant l'exécution des deux algorithmes. Les calculs effectués ne constituent une preuve rigoureuse, car nous faisons ici l'hypothèse qu'il est suffisant de manipuler des densités et non les cardinalités. Cependant ils sont conformes aux expérimentations réalisées et apportent une explication sur les deux phénomènes de seuil observés.

Coloriage d'arbre

Commençons par le coloriage d'arbre, notons B_i le nombre de sommets coloriés en noir et W_i le nombre de sommets coloriés en blanc. La densité de sommets noirs est $b_i = \frac{B_i}{n}$ et celle des sommets blancs est $w_i = \frac{n}{i}$.

Au départ $w_0 = 1$ et $b_0 = 0$. A l'étape $i + 1$, on colorie en noir tous les sommets de W_i qui ne peuvent atteindre aucun sommet de W_i. Nous avons

$$b_{i+1} \approx (1 - \frac{c}{bn})^{nw_i} \approx e^{-cw_i}.$$

Soit $\varphi(z) = 1 - e^{cz}$, nous avons donc la relation de récurrence

$$w_{i+1} = \varphi(w_i).$$

Les points fixes du système dynamique discret sont $z_1 = 0$ et z_2 dépendant de c.

– si $c < 1$ nous avons $z_1 > z_2$, avec z_1 attractif and z_2 répulsif ;

– si $c = 1$ nous avons $z_1 = z_2$ qui est faiblement attractif sur $\mathcal{R}^+$ et faiblement répulsif sur $\mathcal{R}^-$;
– pour $c > 1$ nous avons $z_1 < z_2$, avec z_1 répulsif et z_2 attractif.

On en déduit la limite de la densité w

$$w = \lim_i w_i = \begin{cases} x_1 = 0 & \text{si } c < 1 \\ x_2 > 0 & \text{si } c > 1 \end{cases}$$

Coloriage de DAG
Soient R_i, G_i, W_i le nombre de sommets coloriés respectivement en rouge, vert et blanc à l'étape i et r_i, g_i, w_i leur densité respective. On montre que l'on a à toute étape

– R_{i+1} est l'ensemble des sommets de V_n n'atteignant aucun sommet $V_n \setminus G_i$;
– G_{i+1} est l'ensemble des sommets de V_n atteignant un sommet de R_{i+1} ;
– $V_n \setminus G_{i+1}$ est l'ensemble des sommets de V_n n'atteignant aucun sommet de R_{i+1}.

Posons

$$\begin{cases} \psi(z) & = & e^{-cz}, \\ r_{i+1} & = & \psi(1 - g_i) \\ 1 - g_{i+1} & = & = \psi(r_{i+1}). \end{cases}$$

On obtient alors

$$r_i = \psi^{2i}(0) \text{ et } 1 - g_i = \psi^{2i+1}(0).$$

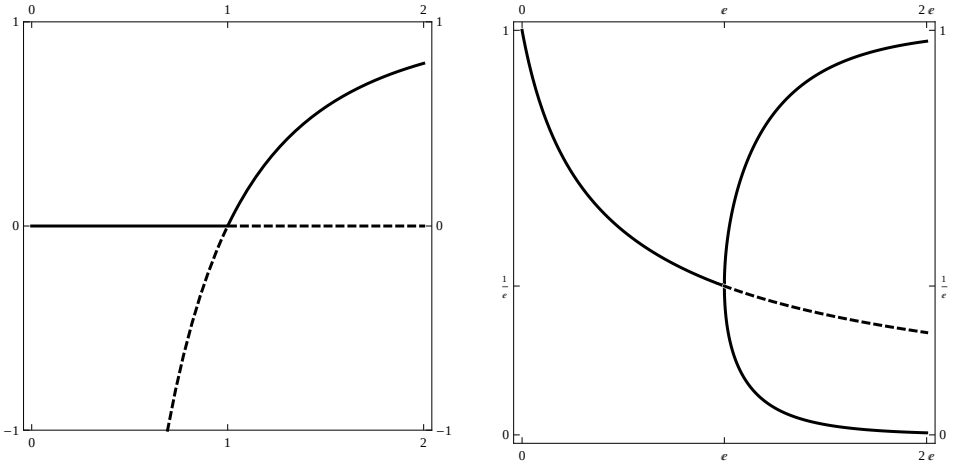
– si $c < e$ le point fixe z_0 est un attracteur ;
– si $c = e$ le point fixe z_0 est faiblement attracteur
– si $c > e$ le point fixe z_0 est un attracteur et il existe un 2-cycle (z_1, z_2) répulseur.

La densité limite des sommets blancs vaut

$$w = \lim_i w_i = \lim_i \left(\psi^{2i+1}(0) - \psi^{2i}(0)\right) = \begin{cases} z_0 - z_0 = 0 & \text{si } c < e \\ |z_1 - z_2| > 0 & \text{si } c > e \end{cases}$$

FIGURE 4.4 – Attracteurs et répulseurs des deux systèmes dynamiques

(a) $\varphi_c(z) = 1 - e^{-cz}$ (b) $\psi_c(z) = e^{-cz}$



En conclusion, pour $c < 1$, *a.p.s.* tous les sommets sont attachés à la partie arbre du graphe, pour $c < e$, *a.p.s.* tous les sommets ne sont pas contenus dans un circuit.

4.2.5 Recherche des noyaux d'un graphe aléatoire

4.2.5.1 Quelques algorithmes

Contrairement à l'algorithme précédent, ColoriageBacktracking, colorie tous les sommets pour n'importe quel graphe. Rappelons que le témoin d'un sommet hors du noyau est un sommet du noyau vers lequel il pointe.

L'algorithme effectue un coloriage avec quatre couleurs :

- BLANC sommets non traités
- ROUGE sommets dans le noyau
- BLEU sommets à l'extérieur du noyau sans témoin
- VERT sommets à l'extérieur du noyau avec témoin

Le principe du coloriage est très simple, tant que Blanc n'est pas vide, on prend un sommet a de Blanc, on énumère tous les noyaux contenant a et ensuite tous ceux ne contenant pas a.

Algorithme 3 ColoriageBacktracking(Blanc,Rouge,Vert,Bleu)

si Blanc= $\emptyset$ **alors**
  si Bleu= $\emptyset$ **alors**
    afficher Rouge
sinon
  Prendre a dans Blanc
  ColoriageBacktracking $\Big($ Blanc $\setminus\big(V^{out}(a) \cup V^{in}(a)\big)$,
    Rouge $\cup\{a\}$
    Vert $\cup(V^{in}(a) \cap$ Blanc),
    Bleu $\cup((V^{out}(a) \cap$ Blanc$)$ $\setminus V^{in}(a)))\Big)$
  ColoriageBacktracking$\Big($Blanc $\setminus\{a\}$,Rouge,Vert, Bleu $\cup\{a\}\Big)$

Remarque 4.2.1. *Cet algorithme énumère tous les noyaux du graphe, mais il est très facile d'adapter cet algorithme au problème de décision* Noyau*, il suffit pour cela de s'arrêter dès que l'on trouve un noyau.*

En utilisant l'algorithme ColoriageDAG de manière préliminaire, nous observons un pic de complexité pour $= c/n\,;\, c > e$.

4.2.5.2 Complexité des algorithmes

Mon objectif est d'analyser la complexité de la recherche d'un noyau sur un graphe aléatoire sur $\mathcal{D}(n,p)$ pour différentes fonctions $p(n)$. Je n'ai pas la prétention de proposer des algorithmes sophistiqués et performants. Il est certainement assez facile d'améliorer l'algorithme avec, par exemple, des heuristiques sur le choix de a ou en stoppant lorsque sommet de Bleu n'a pas de témoin dans Blanc (ce que nous avons d'ailleurs implémenté pour effectuer nos expérimentations).

Cas où les tailles possibles sont connues ou *a.p.s.* **connues**
Lorsque l'on connaît $\mathcal{I}$, l'intervalle des tailles possibles, nous disposons d'un algorithme naïf : on parcourt tous les sous-ensembles de taille $r \in \mathcal{I}$ et on teste si ce sous-ensemble est un noyau. Soit T_r le coût pour savoir si un ensemble de cardinal r est un noyau, on a $T_r \leq r\ n$.

La complexité de l'algorithme naïf est clairement majorée par

$$\sum_{r\in\mathcal{I}} \binom{n}{r} T_r.$$

La probabilité d'erreur ϵ de cet algorithme –on répond qu'il n'y a pas de noyau alors qu'il existe un noyau de taille $r \neq I$– est alors la probabilité qu'il existe un noyau de taille $r(n) \notin \mathcal{I}$.

Dans le cas p constant, nous obtenons ainsi un algorithme superpolynomial car

$$\binom{n}{\alpha \ln n} \sim n^{\alpha \ln n}.$$

L'erreur ϵ dépend de $\delta|\lceil\beta\rfloor - \beta|$. Pour $\delta =\sim 1$, $\epsilon < \frac{1}{n}$, sinon $\epsilon < \frac{1}{n^\alpha}$, avec $0 < \alpha < 1$.

Pour $p = c/n$, on choisit $\mathcal{I} = \{r = \mu_c n + x, x < n^{2/3}\}$, l'algorithme est alors exponentiel car pour tout α constant, $0 < \alpha < 1$,

$$\binom{n}{\alpha n} = 2^{n(H(\alpha)+o(1))},$$

où $H(\alpha)$ est la fonction d'entropy

$$H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha).$$

D'autre part, on montre que l'erreur vaut $\epsilon = e^{-\epsilon' n}$, pour un certain $\epsilon' > 0$.

Remarque 4.2.2. *Marco Illengo a aussi pensé à une autre distribution sur des graphes très dense, si l'on choisit $p = 1 - n^{-1/c}$, pour $c \in \mathcal{R}^+$, le graphe possède a.p.s. des noyaux qui sont uniquement de taille $r = \lceil c \rceil$, l'algorithme est alors polynomial, plus précisément n^r.*

On peut estimer la complexité de ColoriageBacktracking pour ces distributions de la manière suivante (nous avons utilisé la même méthode que Banderier et al [1] qui eux ont calculé la complexité de la recherche d'un stable maximum).

Soit $C_{p,n}(w)$ la complexité de l'algorithme lorsqu'il ne reste plus que $w-1$ sommets dans Blanc. La complexité de l'algorithme est donc $C_{p,n} = C_{p,n}(n+1)$.

Pour le premier appel récursif (a est colorié en rouge), on retire $1 + A_w$ sommets où A_w est la variable aléatoire du nombre de sommets connectés à a. Elle suit donc une distribution binomiale $\mathcal{B}(w, q^2)$. Pour le second appel récursif, (a est colorié en bleu) on retire toujours un seul sommet. On obtient donc en moyenne

$$C_{p,n}(w+1) = C_{p,n}(w) + E\left[C_{p,n}(A_w)\right] + c_{p,n},$$

où $c_{p,n}$ est le coût d'une étape.

L'estimation consiste à remplacer la moyenne de $C_{p,n}(A_w)$ par la complexité de la moyenne de A_w, ce qui nous donne

$$C_{p,n}(w+1) = C_{p,n}(w) + C_{p,n}(q^2 w) + c_{n,p}.$$

On peut alors calculer $C_{p,n}$ et retrouve les mêmes complexités, superpolynomiale pour p constant et exponentiel pour $p = c/n$.

Cependant, pour une étude plus rigoureuse, le modèle $\mathcal{D}(n,p)$ ne convient pas. Il faut reprendre l'analyse avec le modèle $\mathcal{D}(n,M)$ où $M = \lfloor p\binom{n}{2}\rceil$. Nous obtenons la récurrence suivante pour le coût moyen (qui semble difficile à exploiter)

$$C_{M,n}(w+1) = C_{M-1,n}(w) + \sum_{l=0}^{w} \frac{\binom{w}{l}\binom{\binom{w}{2}-l}{M-l}}{\binom{\binom{w}{2}}{M}} C_{M-l,n}(w-l).$$

4.3 Cacher un noyau dans un graphe aléatoire

Ce travail est parti de l'article *Hiding Cliques for Cryptographic Security* d'Ari Juels and Marcus Peinado en 2000[31]. Les auteurs montraient comment cacher une clique dans un graphe aléatoire et proposaient des applications en cryptographie (fonction à sens unique et protocole zero-knowledge).

J'ai commencé à m'intéresser à ce problème dès 2003 et je souhaitais l'adapter au noyau. J'avais obtenu quelques résultats intéressant avec Fabrice Boudot qui venait de soutenir sa thèse sur les protocoles zero-knowledge, mais ce travail restait trop proche de celui de Juels et Peinado et il n'amenait pas vraiment un apport pour le domaine. Ce n'était pas clair de savoir quelle était la meilleure façon de cacher un noyau en restant le plus proche possible de la distribution initiale et il fallait aussi trouver des instances plus difficiles que celles pour les cliques.

Avec Eleonora Guerrini, Marco Illengo et Fabien Laguillaumie, nous avons mis en place un groupe de travail fin 2010 sur ce sujet : cacher un noyau dans un graphe aléatoire.

Ce groupe de travail a également entraîné l'étude de la propriété de noyau sur les graphes éparses pour lesquels nous conjecturons qu'ils fournissent des instances difficiles (voir Section 3.5).

Dans toutes les études qui suivent, la taille du noyau sera fixée et dépendra du graphe aléatoire considéré. D'après la section précédente, le mieux est de choisir $p = c/n$ et de prendre un noyau de taille $k = \mu_c n + o(n)$, la recherche exhaustive naïve est alors exponentielle.

Définissabilité et problèmes NP-complets

Nous allons maintenant reprendre des notations de théorie des modèles finis introduites dans la section 2.

Le problème de décider si un graphe D_n (orienté ou non) vérifie une propriété $\mathcal{P}$ sur les graphes est dans la classe NP lorsqu'il existe un ensemble de symboles de relation $\mathcal{S}$ et une formule close $\varphi(E, \mathcal{S})$ tel que nous avons l'équivalence entre les deux assertions :

- $G_n \models \exists \mathcal{S} \varphi(E, \mathcal{S})$.
- G_n vérifie la propriété $\mathcal{P}$.

On peut montrer que les propriétés exprimables sur les graphes ne requiert que des symboles de relation unaire ou binaire. Nous nous concentrerons au cas où $\mathcal{S}$ ne contient que des symboles de relation unaire (formules de Meso). La partie $\exists \mathcal{S}$ correspond donc à une partition des sommets. Nous détaillerons le cas du noyau, mais cela peut être généralisé à toute propriété de MESO.

La partition est $a \in K$ ou $a \notin K$ et $\varphi(K, A)$ est une formule du premier ordre qui vérifie que K est à la fois stable et dominant

$$\varphi(K, A) = \forall x \forall y (Kx \wedge Ky \wedge x \neq y) \Rightarrow \neg Axy) \wedge \forall x \exists y \neg Kx \Rightarrow Ky.$$

Notons qu'une propriété peut être définie par plusieurs formules logiques. Nous verrons dans la sous-section 4.3 consacrée aux protocoles zero-knowledge que le choix de la formule exprimant la propriété influe fortement sur l'efficacité du protocole.

Comment cacher un noyau de façon optimale
La méthode proposée par Juels et Peinado consiste à prendre un graphe de $\mathcal{G}(n,p)$ et ensuite à insérer une clique de grande taille. Cependant cette méthode n'est optimale que dans le cas très particulier où la propriété est monotone. C'est le cas ici, car si l'on ajoute des arêtes, les cliques déjà présentes sont préservées.

Dans le cas des noyaux, la meilleure façon consiste à choisir un sous-ensemble K de V_n de taille r. On ne met aucun arc entre les sommets de K et, pour tout $a \notin K$, on choisit un témoin b dans K avec la distribution uniforme (on met un arc entre a et b). Notons que cette méthode semble généralisable à toute propriété de MESO.

On ajoute ensuite avec probabilité p tous les arcs qui peuvent être ajoutés (qui ne sont pas à l'intérieur du noyau). Nous noterons $\mathcal{D}^*(n,p)$ cette nouvelle distribution. On retrouve la distribution de départ $\mathcal{D}(n,p)$ conditionné au fait que K soit un noyau de D_n. Pour tout K, nous obtenons la même distribution et comme le tirage de K se fait avec la distribution uniforme on obtient un graphe D_n^* de $\mathcal{D}(n,p)^*$ obtenu à partir de $\mathcal{D}(n,p)$ en rejetant les graphes sans noyau.

Proposition 4.3.1.
$$\frac{\Pr(D_n^* = D)}{\Pr(D_n = D)} = \frac{k}{\mu},$$
où k est le nombre de noyau de D et μ est l'espérance du nombre de noyaux sur $D(n,p)$.

Preuve. Soit $D_n \in \mathcal{D}(n,p)$ et D un graphe à n sommets ayant e arcs. Notons $N = n(n-1)$ le nombre d'arcs possibles. La probabilité que D_n soit égal à D dépend uniquement du nombre d'arcs dans D.

$$\Pr(D_n = D) = p^e(1-p)^{N-e}(1).$$

Soit $r \in \{1, \ldots, n-1\}$. On considère $\mathcal{K}_r$ l'ensemble des parties de V_n de cardinal r. Nous avons donc $card(\mathcal{K}_r) = \binom{n}{r}$. Pour tout $K \in \mathcal{K}_r$, on associe la variable aléatoire X_K telle que $X_K = 1$ si K est un noyau de D_n et 0 sinon.

Notons $\mathcal{D}_K$ l'ensemble des graphes ayant K comme noyau.

$$\Pr(D_n \in \mathcal{D}_K) = \sum_{D' \in \mathcal{D}_K} \Pr(D_n = D') = E[X_K].$$

Soit X le nombre de noyaux de taille r.

$$X = \sum_{K \in \mathcal{K}_r} X_K.$$

L'espérance $E[X] = \mu$ est le nombre moyen de noyaux de taille r sur $\mathcal{D}(n,p)$. Par symétrie, $\Pr(D_n \in \mathcal{D}_K)$ ne dépend pas de K, donc

$$\Pr(D_n \in \mathcal{D}_K) = \frac{\mu}{\binom{n}{r}}.$$

Supposons maintenant que D possède k noyaux $E = \{K_1, \ldots, K_k\}$. Soit K' un de ces noyaux et K le noyau tiré aléatoirement dans la construction de D_n^*.

$$\Pr(D_n = D) = \Pr(D_n = D | D_n \in \mathcal{D}_K) \Pr(D_n \in \mathcal{D}_k).$$

Si $D_n \in \mathcal{D}_K$ alors il a un témoin (un arc de b vers a pour un a de K) pour chaque sommet b en dehors de K, soit $n - r$ arcs. L'emplacement des témoins ne change pas la probabilité, seulle nombre d'arcs intervient dans (1). Nous n'avons pas besoin de regarder les $r(r-1)$ couples (a, b) dans K car nous travaillons dans $\mathcal{D}_K$.

Donc

$$\Pr(D_n = D | D_n \in \mathcal{D}_K) = p^e (1-p)^{N-(n-r)-r(r-1)}.$$

$$\Pr(D_n^* = D) = \Pr(D_n^* = D | K \in E) \Pr(K \in E).$$

D'où

$$\Pr(K \in E) = \frac{k}{\binom{n}{r}}.$$

Il reste maintenant à calculer $\Pr(D_n^* = D | K \in E)$. Comme $K \in E$, nous n'avons à nouveau pas besoin de regarder les couples (a, b) dans K. Nous aussi avons par construction de D_n^* un témoin pour chaque b en dehors de K. D'où

$$Pr(D_n^* = D | K \in E) = p^e (1-p)^{N-(n-r)-r(r-1)} = \Pr(D_n = D | D_n \in \mathcal{D}_K).$$

Nous avons donc la probabilité que D_n^* soit égal à D sachant que D possède K comme noyau est égal à la probabilité que D_n soit égal à D sachant que D_n possède K comme noyau.

On obtient finalement

$$\begin{aligned} \Pr(D_n^* = D) &= \frac{\Pr(D_n = D)}{\mu / \binom{n}{r}} \frac{k}{\binom{n}{r}}. \\ &= \Pr(D_n = D) \frac{k}{\mu}. \end{aligned}$$

□

Mauvais graphes

Les deux distributions sont proches sur les graphes avec un nombre de noyaux proche de la moyenne. Pour les autres, la différence entre les deux distributions est d'autant plus grande que le nombre de noyaux s'éloigne de la moyenne. Pour les algorithmes de recherche, les mauvais graphes sont les graphes ayant beaucoup plus de noyaux que la moyenne. Dans $\mathcal{D}(n, p)$, ils sont d'autant plus rares que le nombre de noyaux est élevé. Il faut vérifier que les graphes sont rares aussi dans $\mathcal{D}(n, p)^*$ afin de montrer qu'il n'est pas plus facile de trouver un noyau dans $\mathcal{D}(n, p)^*$ que dans $\mathcal{D}(n, p)$.

Préservation des algorithmes polynomiaux

Nous voulons montrer la propriété de préservation polynomiale suivante : s'il existe un algorithme déterministe $\mathcal{A}$ qui trouve un noyau dans $D_n^* \in \mathcal{D}(n,p)$ en temps polynomial avec une probabilité de succès $\geq n^{-j}$, pour un certain $j \in \mathbb{N}$ (donc avec une probabilité non négligeable), alors ce même algorithme trouve un noyau dans $D_n \in \mathcal{D}(n,p)$ avec une probabilité de succès n^{-k}, pour un certain $k \in \mathbb{N}$.

Cette propriété garantit que s'il n'existe pas d'algorithme polynomial avec probabilité de succès significative (*i.e.* non négligeable) sur $\mathcal{D}(n,p)$ alors nous avons le même résultat pour $\mathcal{D}^*(n,p)$.

Proposition 4.3.2. *Notons $Y = \frac{X}{\mu}$. Si $1 \leq \mu < n^k$ et $E(Y^2)$ est bornée alors la propriété de préservation polynomiale est vérifiée.*

Preuve. Supposons qu'il existe $c > 0$ tel que $E(Y^2) < c$, pour n assez grand. Soit $\lambda > 0$.

$$\Pr(Y > \lambda) < \frac{E(X^2)}{\lambda^2\mu^2} = \frac{E(Y^2)}{\lambda^2} \leq \frac{c}{\lambda^2}.$$

Supposons maintenant qu'il existe un algorithme déterministe A qui trouve un noyau en temps polynomial pour $D_n^* \in \mathcal{D}_n^*(n,p)$ avec une probabilité de succès $\geq n^{-j}$, pour un certain $j \in \mathbb{N}$.

Soit Z l'ensemble des graphes pour lesquels A trouve un noyau, nous avons donc

$$\sum_{D \in Z} \Pr(D_n^* = D) \geq n^{-j}.$$

Soient λ un polynôme en n et $(Z_i)_{i \geq 0}$ une partition de Z avec Z_0 contenant les graphes de Z ayant moins de $\lambda\mu$ noyaux et Z_i, pour $i \geq 1$, les graphes ayant k noyaux où

$$\mu^i\lambda \leq k < \mu^{i+1}\lambda.$$

$$\begin{array}{rcl} \sum_{D \in Z_i} \Pr(D_n^* = D) & = & \sum_{D \in Z_i} \left(\Pr(D_n = D)\frac{k}{\mu} \right) \\ & < & \frac{\lambda\mu^{i+1}}{\mu} \sum_{D \in Z_i} \Pr(D_n = D) \\ & < & (\lambda\mu^i) \Pr(Y > \lambda\mu^{i-1}) \\ & < & c\mu^{-i}\lambda^{-1}. \end{array}$$

Il vient

$$\begin{array}{rcl} \sum_{i \geq 1} \sum_{D \in Z_i} \Pr(D_n^* = D) & \leq & c\lambda^{-1}\frac{\mu^{-1}}{1-\mu^{-1}} \\ & \leq & c\mu^{-1}\lambda^{-1}. \end{array}$$

$$\begin{array}{rcl} \sum_{i \geq 1} \sum_{D \in Z_i} \Pr(D_n^* = D) & > & \frac{1}{\mu} \Pr(Y < \lambda) \\ & > & \frac{1}{\mu}(1 - \frac{c}{\lambda^2}). \end{array}$$

Pour $\lambda = n^{-j-1}$, nous avons donc

$$\sum_{i \geq 1} \sum_{D \in Z_i} \Pr(D_n^* = D) << \sum_{D \in Z_0} \Pr(D_n^* = D).$$

D'autre part,

$$\begin{array}{lll} \sum_{D\in Z_0} \Pr(D_n = D) & \geq & \frac{\mu}{\lambda\mu} \sum_{D\in Z_0} \Pr(D_n* = D) \\ & \geq & \lambda^{-1} n^{-j} \\ & \geq & n^{-2j} \\ & \geq & n^{-k}, \text{ pour } k = 2j. \end{array}$$

□

Il faut bien voir que la démonstration ne fait intervenir à aucun moment la propriété. Le résultat est donc valable pour toute propriété définissable dans MESO vérifiant la proposition 4.3.2.

Dans le cas du noyau, on montre que l'on a $E(Y^2) \sim 1$, nous sommes donc dans les conditions d'application de la proposition 4.3.2.

4.4 Preuve de connaissance zero-knowledge

Nous savons que toute propriété NP-complète possède une preuve de connaissance zero-knowledge [25]. Le schéma présenté ici peut se généraliser à n'importe quelle formule de MESO. Le protocole est un peu plus coûteux (en termes d'engagements) que celui proposé par Juels et Peinado pour une raison très simple : la clique est une propriété monotone. Pour prouver son existence, il suffit d'exhiber des arêtes. Pour le noyau, c'est plus compliqué car il faut à la fois prouver la présence de certains arcs et l'absence d'autres arcs. Ainsi le schéma de Juels et Peinado ne requiert un engagement que sur les arêtes, alors qu'ici il faut un engagement pour tous les couples de sommets, soit $n(n-1)$ engagements. Cependant, les distributions $p(n)$ pour lesquelles les auteurs conjecturent qu'il n'existe pas d'algorithme polynomial avec une probabilité de succès significative trouvant une clique de grande taille appartiennent à l'intervalle $]p_0, 1 - n^{1/4+\gamma}[$, où $0 < p_0 < 1$ est une constante et $\gamma > 0$. Pour toutes ses distributions le nombre d'arcs est quadratique, le coût du protocole que nous proposons est donc bien de même ordre de grandeur.

Le protocole suivant est répété autant de fois que nécessaire pour convaincre Bob. Tout d'abord Alice colorie en rouge tous les sommets du noyau K et en vert tous les autres sommets.

Déroulement du protocole
Alice et Bob ont tous les deux en leur possession le graphe D contenant un noyau K.

1. Alice génère aléatoirement une permutation π sur V_n. Elle envoie comme engagements les couples $(\pi(a), C(\pi(a)))$, où $C(\pi(a))$ est la couleur de $\pi(a)$ et les triplets $(\pi(a), \pi(b), estUnArc)$, où estUnArc est un booléen qui vaut Oui si et seulement si $(a, b) \in A$. Les couples sont donc de la forme (a, Vert) ou (a, Rouge) et les triplets (a, b, oui) ou (a, b, non).
2. Bob jette une pièce de monnaie et envoie le résultat R.

3. Si Alice reçoit PILE, elle renvoie à Bob la permutation π et tous les désengagements des triplets $(\pi(a), \pi(b), estUnArc)$. Si elle reçoit FACE, elle renvoie les désengagements de $(\pi(a), \text{ROUGE})$ et les désengagements d'un témoin b de a, $(\pi(a), \text{VERT})$ et $(\pi(a), \pi(b), estUnArc)$, pour chaque sommet a du noyau, ainsi que les désengagements de $(\pi(a), \pi(b), estUnArc)$ pour chaque couple de sommets (a, b) de K.
4. Si $R = \text{PILE}$, Bob vérifie que π est une permutation de V_n et que $(\pi(a), \pi(b), \text{OUI})$ si et seulement si $(a, b) \in A$. Si $R = \text{FACE}$, Bob vérifie d'une part que $(\pi(a), \pi(b), estUnArc) = (\pi(a), \pi(b), \text{NON})$ pour tout $(\pi(a), \text{ROUGE})$ et $(\pi(b), \text{ROUGE})$ et $(\pi(a), \pi(b), estUnArc) = (\pi(a), \pi(b), \text{oui})$ lorsque $\pi(a)$ et $\pi(b)$ appartiennent à $(\pi(a), \text{ROUGE})$ et $(\pi(b), \text{VERT})$.

Cette méthode d'engagement de la couleur est plus forte que de simplement désengager les couples (a, b) pour vérifier que l'on a un noyau car elle force Alice à s'engager sur les sommets qui seront dans le noyau et hors du noyau. De ce fait, il est aussi possible de cacher la taille du noyau car on peut effectuer une vérification partielle. Pour cela, on fixe un paramètre $\alpha > 0$ qui donnera le pourcentage de sommets engagés. Pour chaque permutation π et les engagements correspondants, Bob choisit un ensemble C de αn couples $(\pi(a), C(\pi(a)))$. Si $R = \text{PILE}$, le protocole est inchangé et si $R = \text{FACE}$, Alice envoie les mêmes désengagements que précédemment mais uniquement pour l'ensemble de sommets $S = \{\pi(a) \mid (\pi(a), C(\pi(a))) \in C\}$. Bob vérifie que les sommets $(\pi(a), \text{ROUGE})$ forment un noyau sur D restreint à S.

Nous avons vu que les instances les plus difficiles pour les distributions étudiées étaient obtenues avec $p(n) = c/n$, où $c > e$. Dans ce cas, les tailles possibles des noyaux ont une densité μ_c, où $c = -\frac{\log \mu_c}{\mu_c}$. Nous avons conjecturé que l'algorithme COLORIAGEBACKTRACKING était exponentiel. Si Alice cache un noyau de densité μ_c, avec le protocole ci-dessus, Bob peut déterminer la densité mais pas la taille exacte. Ce qui augmente la difficulté pour Bob de retrouver le noyau (l'espace de recherche reste cependant exponentiel même pour une seule taille).

De manière générale, pour toute propriété de MESO, le fait de cacher la cardinalité des parts de la partition semble pertinent, sinon on apporte une information supplémentaire sur la solution. Cela n'est pas pris en compte dans les études existantes, par exemple pour la 3-coloriabilité, la propriété de référence pour les protocoles zero-knowledge [25]. Ici Bob ne peut qu'estimer la densité, le problème de chercher une méthode qui ne donnerait aucune information sur la solution reste un problème ouvert. Je conjecture que l'on ne peut faire mieux que de donner la densité, mais il faudrait une étude plus approfondie pour s'en convaincre. Une des principales difficultés pour effectuer une telle preuve, vient du fait que le protocole dépend fortement de la formule $\exists \mathcal{S} \varphi(E, \mathcal{S})$ et qu'il n'est pas facile de caractériser toutes les propriétés exprimable dans cette logique.

En conclusion de cette partie, nous pouvons remarquer que nous sommes partis d'un problème de logique sur les loi 0-1 et que nous sommes à nouveau confrontés pour cette dernière étude à un problème de logique.

Références de la partie 1

[1] Cyril Banderier, Hsien-Kuei Hwang, Vlady Ravelomanana, and Vytas Zacharovas. Analysis of an Exhaustive Search Algorithm in Random Graphs and the $n^{c \log n}$-asymptotics. *SIAM J. Discrete Math.*, 28(1) :342–371, 2014.

[2] Cyril Banderier, Markus Kuba, and Alois Panholzer. Analysis of three graph parameters for random trees. *Random Structures and Algorithms*, 35(1) :42–69, 2009.

[3] Cyril Banderier, Jean-Marie Le Bars, and Vlady Ravelomanana. Generating Functions For Kernels of Digraphs (Enumeration & Asymptotics for Nim Games). In *Formal Power Series and Algebraic Combinatorics (FPSAC'04)*, pages 91–105, Vancouver, Canada, 2004. University of British Columbia.

[4] Claude Berge. *Graphs*, volume 6 (1) of *North-Holland Mathematical Library*. North-Holland, 3rd revised edition, 1991.

[5] Claude Berge and Pierre Duchet. Recent problems and results about kernels in directed graphs. *Discrete Mathematics*, 86(1-3) :27–31, 1990.

[6] B. Bollobas. *Random Graphs*. Cambridge University Press, 2001.

[7] B. Bollobás and A. Thomason. Threshold functions. *Combinatorica*, 7(1) :35–38, January 1987.

[8] Rudolf Carnap. *Logical Foundations of Probability*. Chicago]University of Chicago Press, 1962.

[9] Nadia Creignou. *Computer Science Logic : 6th Workshop, CSL '92 San Miniato, Italy, September 28 – October 2, 1992 Selected Papers*, chapter The class of problems that are linearly equivalent to satisfiability or a uniform method for proving NP-completeness, pages 115–133. Springer Berlin Heidelberg, 1993.

[10] W.Fernandez de la Vega. Kernels in random graphs. *Discrete Mathematics*, 82(2) :213 – 217, 1990.

[11] Michel De Rougemont. Second-order and Inductive Definability on Finite Structures. *Mathematical Logic Quarterly*, 33(1) :47–63, 1987.

[12] B. Dreben and W.D. Goldfarb. *Decision Problem : Solvable Classes of Quantificational Formulas*. Addison-Wesley Longman, Incorporated, 1980.

[13] Paul E. Dunne, Alan Gibbons, and Michele Zito. Complexity-theoretic models of phase transitions in search problems. *Theor. Comput. Sci.*, 249(2) :243–263, 2000.

[14] P. Erdős and A. Rényi. Asymmetric graphs. *Acta Mathematica Academiae Scientiarum Hungarica*, 14(3) :295–315, 1963.

[15] R Erdos and J.H. Spencer. *The Probabilistic Method in combinatorics*. A Series of Monographs and Textbooks. Academic Press, 1974.

[16] P. Erdős and A Rényi. On the Evolution of Random Graphs. In *Publication of the mathematical institute of the hungarian academy of sciences*, pages 17–61, 1960.

[17] Ronald Fagin. Monadic generalized spectra. *Mathematical Logic Quarterly*, 21(1) :89–96, 1975.

[18] Ronald Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41 :50–58, 3 1976.

[19] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[20] Jorg Flum. Problems in finite model theory collected in oberwolfach, 1994.

[21] Ehud Friedgut and Gil Kalai. Every monotone graph property has a sharp threshold. *PROC. AMER. MATH. SOC*, 124 :2993–3002, 1996.

[22] Haim Gaifman. Concerning measures in first order calculi. *Israel Journal of Mathematics*, 2(1) :1–18, 1964.

[23] E. N. Gilbert. Random Graphs. *Ann. Math. Statist.*, 30(4) :1141–1144, 12 1959.

[24] Yu. V. Glebskii, D. I. Kogan, M. I. Liogon'kii, and V. A. Talanov. Range and degree of realizability of formulas in the restricted predicate calculus. *Cybernetics*, 5(2) :142–154, 1969.

[25] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs That Yield Nothing but Their Validity or All Languages in NP Have Zero-knowledge Proof Systems. *J. ACM*, 38(3) :690–728, July 1991.

[26] Valentin Goranko and Bruce Kapron. The modal logic of the countable random frame. *Archive for Mathematical Logic*, 42(3) :221–243, 2003.

[27] Etienne Grandjean. Complexity of the first-order theory of almost all finite structures. *Information and Control*, 57(2–3) :180 – 204, 1983.

[28] Joseph Y. Halpern and Bruce Kapron. Zero-One Laws for Modal Logic. *Annals of Pure and Applied Logic*, 69(2-3) :157–193, 1994.

[29] Joseph Y. Halpern and Bruce M. Kapron. Erratum to Zero-One Laws for Modal Logic [ann. pure appl. logic 69 157–193]. *Annals of Pure and Applied Logic*, 121(2-3) :281–283, 2003.

[30] S. Janson, T. Luczak, and A. Rucinski. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. Wiley, 2011.

[31] Ari Juels and Marcus Peinado. Hiding Cliques for Cryptographic Security. *Designs, Codes and Cryptography*, 2(3) :269–280, 2000.

[32] Matt Kaufmann. Counterexample to the 0-1 law for existential monadic second-order logic. Technical report, 1987.

[33] Matt Kaufmann and Sharon Shelah. On random models of finite power and monadic logic. In *Discrete mathematics*, volume 54, pages 285–293, 1985.

[34] P. Kolaitis and M. Vardi. The Decision Problem for the Probabilities of Higher-order Properties. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 425–435, New York, NY, USA, 1987. ACM.

[35] Phokion G. Kolaitis and Moshe Y. Vardi. Special Issue : Selections from 1988 IEEE Symposium on Logic in Computer Science 0–1 Laws and decision problems for fragments of second-order logic. *Information and Computation*, 87(1) :302 – 338, 1990.

[36] Phokion G. Kolaitis and Moshe Y. Vardi. *Logic from Computer Science : Proceedings of a Workshop held November 13–17, 1989*, chapter 0–1 Laws for Fragments of Second-Order Logic : An Overview, pages 265–286. Springer New York, New York, NY, 1992.

[37] Phokion G. Kolaitis and Moshe Y. Vardi. *Mathematical Foundations of Computer Science 2000 : 25th International Symposium, MFCS 2000 Bratislava, Slovakia, August 28 – September 1, 2000 Proceedings*, chapter 0–1 Laws for Fragments of Existential Second-Order Logic : A Survey, pages 84–98. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.

[38] Thierry Lacoste. Finitistic proofs of 0–1 laws for fragments of second-order logic. *Information Processing Letters*, 58(1) :1 – 4, 1996.

[39] Kerkko Luosto Lauri Hella, Phokion G. Kolaitis. Almost Everywhere Equivalence of Logics in Finite Model Theory. *The Bulletin of Symbolic Logic*, 2(4) :422–443, 1996.

[40] Jean-Marie Le Bars. Fragments of Existential Second-Order Logic without 0-1 laws. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 525–536, 1998.

[41] Jean-Marie Le Bars. *Probabilités asymptotiques et pouvoir d'expression des fragments de la logique du second ordre.* PhD thesis, Université de Caen, 1998.

[42] Jean-Marie Le Bars. Counterexamples of the 0-1 law for fragments of existential second-order logic : an overview. *Bulletin of Symbolic Logic*, 6(1) :67–82, 2000.

[43] Jean-Marie Le Bars. The 0-1 law fails for monadic existential second-order logic on undirected graphs. *Inf. Process. Lett.*, 77(1) :43–48, 2001.

[44] Jean-Marie Le Bars. The 0-1 law fails for frame satisfiability of propositional modal logic. In *Logic in Computer Science, 2002. Proceedings. 17th Annual IEEE Symposium on*, pages 225–234, 2002.

[45] Jean-Marie Le Bars. *Mathematics and Computer Science II : Algorithms, Trees, Combinatorics and Probabilities*, chapter A Sharp Threshold for a Non-monotone Digraph Property, pages 197–211. Birkhäuser Basel, Basel, 2002.

[46] John Von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[47] Leszek Pacholski and Wieslaw Szwast. Asymptotic Probabilities of Existential Second-Order Gödel Sentences. *The Journal of Symbolic Logic*, 56(2) :427–438, 1991.

[48] Leszek Pacholski and Wieslaw Szwast. A Counterexample to the 0-1 Law for the Class of Existential Second-Order Minimal Gödel Sentences with Equality. *Information and Computation*, 107(1) :91 – 103, 1993.

[49] R Rado. Universal graphs and universal functions. *Acta Arith.*, 9 :331–340, 1964.

[50] M Richardson. Solutions of irreflexive relations. *Ann. of Math.*, pages 573–590, 1954.

[51] Lidia Tendera. A note on asymptotic probabilities of existential second-order minimal classes : the last step. *Fundamenta Informaticae*, 20 :277–285, 1997.

[52] Ioan Tomescu. Almost all digraphs have a kernel. *Discrete Mathematics*, 84(2) :181 – 192, 1990.

[53] J.F.A.K. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.

[54] Anne Vedø. Asymptotic Probabilities for Second-Order Existential Kahr-Moore-Wang Sentences. *J. Symb. Log.*, 62 :304–319, 1997.

Quatrième partie

Énumération et génération aléatoire de fonctions booléennes

Chapitre 1

Introduction

1.1 Contexte

Les fonctions booléennes ne constituent pas à proprement parler un domaine de recherche. Comme les arbres, les graphes ou les mots, se sont des objets de base pour diverses études en mathématique et en informatique. Des chercheurs de compétences variées – combinatoire énumérative, combinatoire algébrique, corps finis, complexité des circuits booléens, implantation hardware de ces circuits, satisfaction de contraintes, cryptographie symétrique – ont étudié ces objets. Le livre *Boolean Models and Methods in Mathematics*, édité par Yves Crama et Peter L. Hammer, offre un bon panorama de ces différentes études [69], mais ces 700 pages ressemblent plus à une compilation de plusieurs livres.

Il serait difficile de proposer une présentation générale des fonctions booléennes car il existe souvent peu d'interactions entre les différents domaines, il suffit pour cela de voir l'intersection vide des bibliographies des publications entre deux domaines. Cela explique que le vocabulaire est très différent pour parler souvent des mêmes propriétés, mais avec un point de vue et des objectifs vraiment différents.

De mon côté, j'ai surtout lu des articles portant sur les fonctions booléennes pour la cryptographie, bien que je trouve les autres sujets aussi intéressants. En particulier, j'ai étudié la partie de *Boolean Models and Methods in Mathematics* consacrées aux fonctions booléennes et aux codes correcteurs, *Boolean Functions for Cryptography and Error Correcting Codes.* de Claude Carlet [61].

Les fonctions booléennes jouent un rôle central dans la sécurité des systèmes de chiffrements à flots ou par blocs (ces derniers regroupent les deux grandes catégories de schémas de chiffrement à clef secrète). Ce sont des primitives cryptographiques qui assurent la confusion et la diffusion de la fonction de chiffrement, selon les concepts introduits par Shannon. Pour cela, elles doivent vérifier des critères cryptographiques essentiels portant par exemple sur la résilience, le degré algébrique ou la non-linéarité. Ceux-ci permettent de quantifier le niveau de résistance aux attaques connues des cryptosystèmes implémentant de telles fonctions. Lorsque l'on considère plusieurs critères, il est souvent impossible d'obtenir des fonctions optimales pour tous ces critères. Le travail d'une majorité des chercheurs du domaine consiste à définir les

meilleurs compromis pour ces critères et à proposer des constructions de fonctions très spécifiques réalisant ces compromis.

Il existe cependant un inconvénient majeur à ces constructions, en général on construit ainsi une faible proportion de ces fonctions et nous ne pouvons donc pas savoir si elles sont représentatives, ni si elles génèrent par ailleurs une structure particulière qui pourrait être exploitée par un attaquant.

Mon objectif n'était pas de devenir spécialiste du domaine, mais de voir s'il était possible de capturer d'un point de vue combinatoire et algorithmique des classes entières de fonctions booléennes selon des critères cryptographiques. Généralement, les études effectuées donnent des constructions récursives très spécifiques qui produisent qu'une fraction négligeable d'une classe de fonctions. Je montre dans le chapitre 2 que c'est parfaitement réalisable pour les fonctions sans corrélation d'ordre 1 où nous sommes capables de compter, énumérer et générer aléatoirement les fonctions booléennes d'une classe en utilisant une méthode que nous avons appelé *méthode des classes*. Dans le chapitre 3, je montre la difficulté d'énumérer en considérant à la fois le poids de Hamming et le degré algébrique, ce qui limite l'efficacité de la méthode des classes, car ces deux critères interviennent souvent simultanément. Ce chapitre contient également une étude de la classe des fonctions coïncidentes –fonctions invariantes par la transformée de Möbius– dont les propriétés pourraient permettre de contrôler (énumération et génération aléatoire) des sous-classes avec des critères pertinents pour la cryptographie.

1.2 Contributions

Le chapitre 2 porte sur la corrélation d'ordre k. Cette propriété est souvent considérée en cryptographie car c'est une propriété de base que doivent avoir les fonctions booléennes utilisée pour combiner plusieurs LFSR (registres à décalage à rétroaction linéaire). Je cherchais avec Alfredo Viola à capturer la classe de l'ensemble des fonctions sans corrélation d'ordre k, en considérant le cas $k = 1$ dans un premier temps. L'objectif initial était d'obtenir des séries génératrices en décomposant une fonction à n variables en fonction ayant moins de variables.

Le problème de l'énumération de classe de fonctions booléennes est relativement ancien, W.K. Clifford fut le premier à s'y intéressé en 1877 [65], mais sa terminologie et ses méthodes étaient trop ardues pour être facilement reprises.

L'article *Balancing the n-cube* de Palmer, Read and Robinson [74] semble le seul article antérieur à nous travaux proposant une méthode directe pour compter le nombre de fonctions sans corrélation d'ordre 1. Ils proposent pour cela deux méthodes issues de la combinatoire algébrique, l'énumération de Polya [76] et la superposition [77]. Ils obtiennent le nombre de fonctions 1-résilientes à 6 variables. Cet article n'est pas connu par les chercheurs travaillant sur les fonctions booléennes pour la cryptographie. Par exemple, il n'est pas cité dans la monographie de Claude Carlet sur les fonctions booléennes [61] qui est pourtant une référence très complète des fonctions booléennes en cryptographie. Cet ouvrage extrait d'un livre plus général sur les fonctions booléennes *Boolean Models and Methods in Mathematics* [69], m'a souvent été

très utile pour connaître les propriétés des fonctions booléennes. L'inconvénient de l'approche de Palmer et al est que la complexité de leurs algorithmes n'est pas claire, on ne sait pas si leur méthode permet de construire des algorithmes optimisés afin d'obtenir le nombre de fonctions 1-résilientes à 7 ou 8 variables. D'autre part leurs méthodes s'appliquent au comptage et à l'énumération, mais elles ne semblent pas appropriées à la génération aléatoire.

Nous avons décidé avec Alfredo Viola dès 2004 d'étudier une propriété très simple, la 1-résilience (une fonction booléenne est 1-résiliente lorsqu'elle est sans corrélation d'ordre et équilibrée). L'objectif initial était de comprendre la structure de l'ensemble de la classe de ces fonctions. En particulier, nous souhaitions utiliser des méthodes issues de la combinatoire analytique. Nous verrons plus loin comment définir la série génératrice des fonctions 1-résilientes et celle des fonctions sans corrélation d'ordre 1, nous verrons également les limites de cette approche.

L'originalité de notre méthode – à la fois combinatoire et algorithmique – que nous avons appelée méthode des classes, a été de classifier l'ensemble des fonctions booléennes en fonction de leur écart avec les fonctions 1-résilientes. Nous formons de cette manière des classes d'équivalence qui définissent un demi-treillis –le maximum de ce treillis étant la classe des fonctions 1-résilientes– l'ordre partiel de ce treillis préserve la cardinalité, c'est-à-dire pour deux classes ω_1 et ω_2, si $\omega_1 < \omega_2$ alors $|\omega_1| < |\omega_2|$. Les classes à n variables se décomposent récursivement en deux fonctions à $n-1$ variables en utilisant la décomposition de Shannon, jusqu'à arriver aux fonctions constantes 0 et 1. L'énumération et la génération aléatoire des fonctions d'une de ces classes est rendue efficace grâce à une seconde relation d'équivalence qui tient compte des permutations entre les variables x_i et x_j et également des échanges entre les valuations avec $x_i = 0$ et celles avec $x_i = 1$ (ces deux transformations préservent la cardinalité des classes). Cette seconde relation d'équivalence produit des classes normalisées, le stockage de la cardinalité de ces classes suffit pour l'énumération et la génération aléatoire. Comme leur nombre est considérablement réduit, cela nous permet d'obtenir des algorithmes très performants pour énumérer toutes les fonctions 1-résilientes à au plus 8 variables et à effectuer la génération uniforme avec la distribution uniforme sur ces fonctions booléennes.

J'ai présenté aux journées C2 2005 une première approche que j'avais eu avec Thomas Leduc, ingénieur de recherche au CERMA. La méthode n'était alors pas très efficace et ne permettait l'énumération que jusqu'à 6 variables (comme celle de Palmer et al.).

Dans [72, 73], nous avons Viola et moi effectué l'énumération jusqu'à 7 variables en $50s$. Carrasco, un étudiant de Viola, a optimisé un programme pour aller jusqu'à 8 variables, celui-ci nécessite 15 jours de calculs.

Après l'énumération des fonctions 1-résiliente, nous nous sommes intéressés à la génération aléatoire de ces fonctions. Nous avons conçu une méthode très efficace pour générer aléatoirement une fonction 1-résiliente avec la distribution uniforme [62, 63]. Pour 8 variables, nous sommes en mesure de générer en $5,4$ secondes en moyenne une fonction parmi un espace des 10^{68} fonctions 1-résilientes.

Le programme d'énumération précédent n'a besoin d'être exécuté qu'une seule

fois car la cardinalité des classes en jeu sont stockées en mémoire. Pour générer des fonctions 1-résilientes, nous avons en effet besoin de connaître la cardinalité des classes normalisées de $k < n$ variables, celles-ci sont donc stockées en mémoire.

Ce travail a été réalisé avec Viola et Carrasco qui a implémenté les algorithmes. C'est un travail essentiellement algorithmique et de programmation et Carrasco s'est particulièrement attaché à ce que le maximum de calculs puissent être effectué dynamiquement.

Nous avons déjà vu que la méthode des classes permet de compter et d'énumérer l'ensemble des fonctions d'une classe de corrélation. Nous avons proposé une méthode pour générer aléatoirement avec la distribution uniforme une fonction booléenne d'une classe ω. L'idée est d'avoir deux bijections réciproques l'une de l'autre, la première associe un numéro de 0 à $|\omega| - 1$ à chaque fonction de ω et la seconde associe une fonction à chaque numéro de 0 à $|\omega| - 1$. On appelle codage énumératif la donnée de deux algorithmes réalisant ces deux fonctions. La performance d'un codage énumératif dépend à la fois de la mémoire et du temps de calcul. Nous allons voir que la méthode des classes qui permet de décomposer une fonction en plusieurs fonctions de moins de variables donne directement un codage énumératif. La véritable difficulté a été de concevoir des algorithmes les plus efficaces possibles. Il faut pour cela trouver un bon compromis pour séparer en deux les informations, d'un côté, celles que l'on doit nécessairement mémoriser et, de l'autre, celles que l'on peut calculer dynamiquement lors de la construction d'une fonction.

Notre méthode de génération par décomposition récursive correspond à la *méthode récursive* formalisée dans [66] par Flajolet, Zimmerman et Van Cutsem et appelée *méthode récursive*. Comme nous l'avons vu au chapitre III, la méthode du dictionnaire permet de faire correspondre des relations sur les ensembles vers des relations sur les séries génératrices. La méthode récursive associe aux relations sur les ensembles des algorithmes de codage énumératif. On retrouve les relations sur les ensembles les plus usuels : réunion, produit cartésien, séquence.

Notons que les auteurs de [66] considèrent bien la génération aléatoire uniforme, mais ils ne mentionnent pas le codage énumératif bien qu'il soit sous-jacent. Un avantage pratique d'effectuer cette génération aléatoire par l'intermédiaire d'un codage énumératif est que nous pouvons concentrer tous les aléas en entrée plutôt qu'au fur et a mesure des choix des décompositions récursives.

J'ai participé à l'ANR Boole (209-2013) qui était composée de membres des équipes du Prisme de L'université de Versailles, de l'INRIA de Rocquencourt, de l'ENS, du LIF de l'université de Luminy et du GREYC.

L'objectif de cette ANR était d'élaborer des cadres permettant l'étude des structures booléennes (circuits, systèmes de preuve, formules et fonctions booléennes). Il s'agissait de mesurer, modéliser et prédire les propriétés probabilistes de celles-ci. Dans cette perspective, quatre cadres booléens ont été constitués : Circuits booléens et formes normales booléennes, Fonctions booléennes et cryptographie, Satisfaisabilité et Logiques quantitatives.

Ces quatre cadres s'appuient sur deux grands axes méthodologiques que sont les

méthodes combinatoires et analytiques et les méthodes probabilistes.

Un des grands mérites de cette ANR était aussi d'associer des chercheurs ayant des points de vue différents comme Claude Carlet et Sihem Mesnager (fonctions booléennes pour la cryptographie), Jean Vuillemin (circuits booléens, OBDD), Philippe Flajolet Danièle Gardy et Antoine Genitrini (combinatoire analytique), Brigitte Chauvin et Nicolas Pouyanne (probabilités sur les arbres booléens), Nadia Creignou et Hervé Daudé (fonctions booléennes monotones et transitions de phase). Nous avons pu mieux apprécier les spécificités dans chaque domaine et surtout faire ressortir les similarités à la fois sur les études et les méthodes.

Nous montrons dans le chapitre 3 pourquoi il est difficile de considérer simultanément le poids de Hamming et le degré algébrique pour l'énumération et la génération aléatoire. Nous avons A. Viola et moi chercher à appliquer la méthode des classes à d'autres propriétés cryptographiques des fonctions booléennes. Nous avons notamment étudié la non-linéarité et l'immunité algébrique (avec également Antoine Genitrini et Julien Clément). Malheureusement nous n'avons pas réussi à obtenir des systèmes d'équation pour décomposer les fonctions booléennes selon ces propriétés aussi simples que ceux qui apparaissent pour les fonctions 1-résilientes.

J'ai acquis la conviction que ce qui freinait l'application de la méthode des classes aux autres propriétés était que le poids de Hamming et le degré algébrique interviennent conjointement pour ces propriétés. Or le poids de Hamming est naturellement maîtrisé avec la décomposition de Shannon, alors que le degré algébrique requiert la décomposition de Reed-Müller. Par exemple, il est très facile énumérer les fonctions booléennes selon leur poids de Hamming en utilisant la décomposition de Shannon et aussi de les énumérer selon leur degré algébrique en utilisant la décomposition de Reed-Müller. Cependant, à ma connaissance personne ne sait énumérer les fonctions booléennes en considérant à la fois le poids de Hamming et le degré algébrique. Une telle énumération sur des sous-classes des fonctions booléennes reste cependant envisageable.

Le chapitre 3 s'intéresse aussi aux fonctions coïncidentes, il s'agit de fonctions qui font coïncider la représentation sous forme de table de vérité et celle avec la FAN (forme algébrique normale). J'ai entamé pendant l'ANR Boole une étude sur la transformée de Möbius et les fonctions coïncidentes début 2012 avec Hayat Cheballat et Morgan Barbier qui étaient à ce moment-là postdoctorants dans l'équipe AMACC. Notre innovation méthodologique consiste à effectuer des opérations directement sur des polynômes formels (qui ont des indéterminées et non des variables) au lieu de le faire sur des représentations des fonctions booléennes pour lesquelles le nombre de variables doit être fixé. Cela nous permet de faire varier le nombre de variables sans changer de polynôme. On montre que l'énumération et la génération aléatoire des fonctions coïncidentes est facile à réaliser, mais elle nécessite l'utilisation de la transformée de Möbius. On montre aussi expérimentalement que les fonctions coïncidentes aléatoires possèdent des caractéristiques proches des fonctions booléennes aléatoires pour différents critères (poids de Hamming, degré algébrique, non-linéarité).

Chapitre 2

Fonctions sans corrélation d'ordre 1

2.1 Énumération et comptage

2.1.1 La méthode des classes

Notons $\mathcal{BF}_n$ l'ensemble des fonctions booléennes à n variables. La table de vérité d'une fonction $f(x_1, \ldots, x_n)$ de $\mathcal{BF}_n$ peut être vue comme un mot binaire de longueur 2^n que nous noterons $T(f)$. Il existe plusieurs façons naturelles de coder ainsi la fonction par un mot $b_1 \ldots b_{2^n}$ en choisissant un ordre entre les variables. L'ordre qui va être pertinent pour nos décompositions récursives est le suivant :

$$f(a_1, \ldots, a_n) = b_k, \text{ où } k = \sum_{i=1}^{n} a_i \, 2^{i-1}.$$

Nous noterons par la suite, $w = u \star v \in \mathbb{F}_2^{2^n}$ la concaténation de deux mots $u, v \in \mathbb{F}_2^{2^{n-1}}$, d'autre part, nous écrirons f au lieu de $(f(x_1, \ldots, x_n)$.

Par exemple, la fonction f suivante correspond au mot $T(f) = 10100110$:

f	1	0	1	0	0	1	1	0
x_3	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_1	0	1	0	1	0	1	0	1

Soit $\epsilon \in \mathbb{F}_2$, $f \mid_{x_i=\epsilon}$ désignera la restriction de f conditionné à la valuation $x_i = \epsilon$. Il est clair que $f \mid_{x_i=\epsilon}$ est une fonction booléenne à $n-1$ variables et elle est donc codée par un mot binaire de taille 2^{n-1}. Il vient

$$T(f) = T(f \mid_{x_n=0}) \star T(f \mid_{x_n=1}). \tag{2.1}$$

On peut décomposer récursivement f jusqu'à arriver aux feuilles contenant les constantes 0 et 1 ou s'arrêter aux quatre classes à 1 variables, nous ferons l'un ou l'autre choix selon les algorithmes que nous implémenterons.

Le fait de décomposer une fonctions à n variables en deux fonctions à $n-1$ variables n'est pas nouveau, cette méthode a été proposée en premier par Shannon [78] et utilisée par la suite par Bryant [56] afin d'obtenir des structures de données et des algorithmes efficaces, appelé *Ordered Binary Decision Diagrams (OBDD).*

Notons que la décomposition de Shannon

$$f = (1 \oplus x_i) f \mid_{x_i=0} \oplus x_i f \mid_{x_i=1} \tag{2.2}$$

est valable pour toute variable x_i. Néanmoins, seule la décomposition

$$f = (1 \oplus x_n) f \mid_{x_n=0} \oplus x_n f \mid_{x_n=1}$$

peut être interprétée comme la concaténation de (2.1).

Remarque 2.1.1. *Il faut bien séparer les aspects sémantiques et syntaxiques, la décomposition de Shannon a une interprétation sémantique qui ne dépend pas du choix de la structure de données utilisée pour coder la fonction, mais nous pourrons avoir une interprétation syntaxique en choisissant une représentation de cette fonction comme (2.1).*

Notons $w_H(f)$ le poids de Hamming de f, c'est-à-dire le nombre d'occurrences de 1 dans le mot $T(f)$ et

$$\delta_i(f) = w_H(f \mid_{x_i=0}) - w_H(f \mid_{x_i=1}).$$

Définition 2.1.1. *f est équilibrée lorsque $w_H(f) = 2^{n-1}$, autrement dit lorsque la fonction prend autant de fois la valeur 0 que 1.*

Définition 2.1.2. *f est sans corrélation d'ordre 1 lorsque $\delta_i(f) = 0$, pour tout $i \in \{1, \ldots, n\}$.*

Définition 2.1.3. *f est 1-résiliente lorsqu'elle est sans corrélation d'ordre 1 et équilibrée.*

Définition 2.1.4. Classes d'équivalence pour la corrélation d'ordre 1
Deux fonctions booléennes à n variables f et g sont équivalentes lorsqu'elles satisfont le système

$$f \, \mathcal{R}_{Cor1} \, g \Leftrightarrow \left\{ \begin{array}{rcl} w_H(f) & = & w_H(g) \\ \delta_i(f) & = & \delta_i(g), \quad 1 \leq i \leq n. \end{array} \right. \tag{2.3}$$

Chaque fonction booléenne f appartient à la classe de corrélation

$$\omega = \langle w_H(f), \delta_n(f) \ldots \delta_1(f) \rangle,$$

définie par l'opérateur $\omega = \Omega(f)$. En posant

$$\Omega_n^m = \{\omega \mid \exists f \; \Omega(f) = \omega \text{ et } w_H(f) = m\},$$

il vient $\Omega_n = \bigcup_{m=0}^{2^n} \Omega_n^m$. Nous avons quatre classes de corrélation à 1 variable

$$\Omega(00) = \langle 0, 0 \rangle \quad \Omega(01) = \langle 1, -1 \rangle \quad \Omega(10) = \langle 1, 1 \rangle \quad \Omega(11) = \langle 2, 0 \rangle.$$

Définition 2.1.5. Classe valide
Nous dirons qu'une classe $\omega = \langle m, \delta_n, \ldots, \delta_n \rangle$ *est valide lorsqu'il existe au moins une fonction* f *telle que* $\Omega(f) = \omega$.

Il est important de noter que nous ne connaissons pas d'algorithme efficace pour vérifier directement qu'une classe est valide.

Il est facile de voir que le poids de Hamming des fonctions sans corrélation d'ordre 1 est pair. Notons $Cor_n^m = \langle 2m, 0, \ldots, 0 \rangle$ la classe de corrélation des fonctions sans corrélation d'ordre 1 de poids de Hamming $2m$ et Res_n^1 l'ensemble des fonctions 1-résilientes à n variables.

Le théorème ci-dessous est le point de départ central de notre méthode des classes. C'est grâce à celui-ci que nous allons pouvoir obtenir des algorithmes efficaces pour construire les classes de corrélation.

Théorème 2.1.1. *Let* $\omega^0 = \langle p, \delta_{n-1}^0, \ldots, \delta_1^0 \rangle \in \Omega_{n-1}^p$, $\omega^1 = \langle q, \delta_{n-1}^1, \ldots, \delta_1^1 \rangle \in \Omega_{n-1}^q$, *avec* $p, q \in \{0, \ldots 2^{n-1}\}$. *Alors* $\omega = \omega^0 \star \omega^1$, *où* $\omega = \langle m, \delta_n, \ldots, \delta_1 \rangle \in \Omega_n^m$ *satisfait le système*

$$\begin{cases} m &= p + q, \\ \delta_n &= p - q \\ \delta_i &= \delta_i^0 + \delta_i^1, \text{ pour tout } i \in \{1, \ldots, n-1\}. \end{cases} \tag{2.4}$$

Définition 2.1.6. Décomposition
Une paire (ω^0, ω^1) *qui satisfait la relation*

$$\omega^0 \star \omega^1 = \omega$$

est appelée une décomposition of ω.

La classe $\omega^0 \star \omega^1$ *contient toutes les fonctions* f *vérifiant*

$$f = f^0 \star f^1, \text{ avec } \Omega(f^0) = \omega^0 \text{ et } \Omega(f^1) = \omega^1.$$

Ce système peut être initialisé avec les classes de corrélation à 0 variable $\langle 0 \rangle$ et $\langle 1 \rangle$ correspondant respectivement aux constantes 0 et 1.

Définition 2.1.7. *Produit cartésien*
Soient ω^0 *et* $\omega^1 \in \Omega_{n-1}$. *le produit cartésien* $\omega^0 \times \omega^1$ *est l'ensemble*

$$\{f \mid f = f^0 \star f^1, \ \Omega(f^0) = \omega^0, \ \Omega(f^1) = \omega^1\}.$$

Le théorème d'énumération suivant prouve que si l'on sait effectuer l'énumération des fonctions des classes de corrélation à $n-1$ variables, alors on peut énumérer les fonctions d'une classe à n variables.

Théorème 2.1.2. *(***Énumération***) Soit* $\omega \in \Omega_n$. *L'ensemble des fonctions de* ω *est donné par la formule*

$$\omega = \bigcup_{\omega^0 \star \omega^1 = \omega} \omega^0 \times \omega^1.$$

Par définition du produit cartésien, nous avons $|\omega^0 \times \omega^1| = |\omega^0| \times |\omega^1|$. On déduit directement la cardinalité de ω.

Corrolaire 2.1.1. *(**Comptage**) Soit $\omega \in \Omega_n$ et $|\omega|$ sa cardinalité. Il vient*

$$|\omega| = \sum_{\omega^0 \star \omega^1 = \omega} |\omega^0| \times |\omega^1|.$$

Définition 2.1.8. *Classe miroir*
Soit $\omega = \langle m, \delta_n, \ldots, \delta_1 \rangle \in \Omega_n^m$. La classe miroir de ω est la classe

$$\overline{\omega} = \langle m, -\delta_n, \ldots, -\delta_1 \rangle.$$

Les classes miroir permettre de simplifier le théorème 2.1.2.

Corrolaire 2.1.2. *Soit $0 \leq m \leq 2^{n-1}$, nous avons*

$$\mathcal{C}or_n^m = \bigcup_{\omega^0 \in \Omega_{n-1}^m} \omega^0 \times \overline{\omega}^0.$$

Corrolaire 2.1.3. *Soit $0 \leq m \leq 2^{n-1}$, nous avons*

$$|Cor_n^m| = \sum_{\omega^0 \in \Omega_{n-1}^m} |\omega^0|^2 \qquad Res_n^1 = \sum_{\omega^0 \in \Omega_{n-1}^{2^{n-1}}} |\omega^0|^2.$$

Notons que seules les classes à $n-1$ variables équilibrées sont nécessaires pour calculer la cardinalité de la classe Res_n^1, en revanche on montre que toutes les classes à $n-2$ variables interviennent.

Les idées essentielles pour appliquer la méthode des classes sont d'or et déjà exposées.

Souvent la confusion est faite entre compter et énumérer. Pourtant il s'agit de deux objectifs en principe différents. L'énumération demande un itérateur sur les objets et le temps de calcul entre deux itérations est un paramètre important pour décider de l'efficacité d'une méthode. Le comptage peut également s'effectuer sans connaissance des objets et de leur structure. Il n'y a pas a priori un des deux problèmes qui soit plus difficile que l'autre. Parfois l'énumération nécessite peu de mémoire (peu d'information est nécessaire pour passer d'un objet à un autre car ceux-ci ont une structure proche) alors que le comptage demande davantage de mémoire.

Ici les deux objectifs se rejoignent car il ne semble pas possible de compter les objets d'une classe sans capturer celle-ci. D'autre part, le stockage en mémoire du grand nombre de classes de corrélation de moins de n variables forme l'écueil majeur pour obtenir des algorithmes efficaces pour n grand que ce soit pour l'énumération ou le comptage.

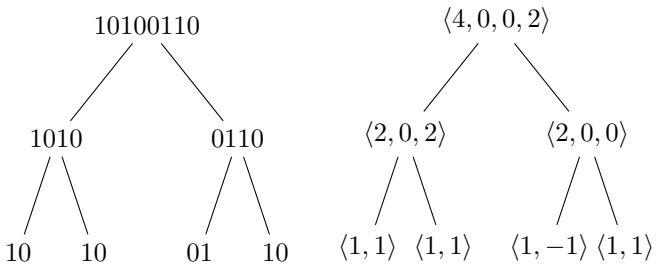


FIGURE 2.1 – Arbre de décomposition (à gauche) et arbre de corrélation (à droite) de f

2.1.2 Comptage et énumération sans le poids

Il est possible de définir des classes de corrélation sans poids

$$\alpha = \langle \delta_n, \ldots, \delta_1 \rangle.$$

Nous noterons U_n l'ensemble des classes sans poids à n variables. Comme précédemment, une classe à n variables peut se décomposer en deux classes à $n-1$ variables, mais ici la décomposition est moins naturelle et ne suit pas la décomposition de Shannon.

Définition 2.1.9. Opérateur de classe $\circ$
Soient $\alpha^0 = \langle \delta^0_{n-1}, \ldots, \delta^0_1 \rangle \in U_{n-1}$ et $\alpha^1 = \langle \delta^1_{n-1}, \ldots, \delta^1_1 \rangle \in U_{n-1}$. L'opérateur $\circ$ est défini par

$$\alpha^0 \circ \alpha^1 = \alpha,$$

où $\alpha = \langle \delta_n, \ldots, \delta_1 \rangle$ satisfait le système

$$\left\{ \begin{array}{rcl} \delta_n & = & \delta^0_{n-1} - \delta^1_{n-1} \\ \delta_i & = & \delta^0_i + \delta^1_i, \text{ pour tout } i \in \{1, \ldots, n-1\}. \end{array} \right. \tag{2.5}$$

Définition 2.1.10. *Soient f^0_{n-1} et f^1_{n-1} deux fonction booléennes à $n-1$ variables et $f^{00}_{n-2}, f^{01}_{n-2}, f^{10}_{n-2}, f^{11}_{n-2}$ les quatre fonctions booléennes à $n-2$ variables vérifiant*

$$\begin{array}{rcl} f^0_{n-1} & = & f^{00}_{n-2} \star f^{01}_{n-2} \\ f^1_{n-1} & = & f^{10}_{n-2} \star f^{11}_{n-2} \end{array}$$

Soit f_n la fonction $f^{00}_{n-2} \star f^{11}_{n-2} \star f^{10}_{n-2} \star f^{01}_{n-2}$. Nous écrirons

$$f_n = f^0_{n-1} \circ f^1_{n-1}.$$

Proposition 2.1.1. *Soient f^0_{n-1}, f^1_{n-1} et f_n telles que $f_n = f^0_{n-1} \circ f^1_{n-1}$. Les classes sans poids α^0 et α^1 de f^0_{n-1} et f^1_{n-1} vérifient $f_n \in \alpha^0 \circ \alpha^1$.*

L'opérateur de classe $\circ$ joue le même rôle que l'opérateur $\star$ pour les classes avec poids, mais il était plus difficile (moins naturel) à déterminer. On retrouve ainsi les mêmes résultats de décomposition et de comptage.

Théorème 2.1.3. *(***Décomposition***)*
Soit $\alpha \in U_n$.

$$\alpha \ = \bigcup_{\alpha^0 \star \alpha^1 = \alpha} \alpha^0 \otimes \alpha^1.$$

Corrolaire 2.1.4. *(***Comptage***)*
Soit $\omega \in \Omega_n$.

$$|\alpha| = \sum_{\alpha^0 \star \alpha^1 = \alpha} |\alpha^0| \ \times \ |\alpha^1|.$$

Définition 2.1.11. *Soit* $\alpha = \langle m, \delta_n, \ldots, \delta_1 \rangle \in U_n$. *La classe miroir de* α *vaut*

$$\overline{\alpha} = \langle -\delta_n, \ldots, -\delta_1 \rangle.$$

Théorème 2.1.4. *(***Décomposition de la classes de fonctions sans corrélation d'ordre** 1*)*
Soit $n \in \mathbb{N}$.

$$\mathcal{C}or_n = \bigcup_{\alpha^0 \in U_{n-1}} \alpha^0 \times \overline{\alpha^0}.$$

Corrolaire 2.1.5.

$$|\mathcal{C}or_n| = \sum_{\alpha^0 \in U_{n-1}} |\alpha^0|^2.$$

2.1.3 Séries génératrices

La série génératrices des classes de corrélation se définit inductivement de la manière suivante :

$$\left\{ \begin{array}{lcl} \varphi_0(x_0) & = & 1 + x_0, \\ \varphi_n(x_0, x_n, \ldots, x_1) & = & \varphi_{n-1}(\mathbf{x_0 x_n}, x_{n-1}, \ldots, x_1) \times \varphi_{n-1}(\mathbf{x_0 / x_n}, x_{n-1}, \ldots, x_1). \end{array} \right. \tag{2.6}$$

Rappelons que $[z^n]A(z)$ désigne le coefficient A_n, où $A(z)$ est la série génératrice ordinaire

$$\sum_{n \geq 0} A_n z^n.$$

En généralisant cette définition à plusieurs symboles formels $x_0, \ldots, x_n,$,

$$[x_0^m \ x_n^{\delta_n} \ \ldots x_1^{\delta_1}]\varphi_n(x_0, x_n, \ldots, x_1)$$

donne le nombre de fonctions de la classe $\langle m, \delta_n, \ldots, \delta_1 \rangle$.

Par exemple, $[x_0^{2^{n-1}} \ x_n^0 \ \ldots x_1^0]\varphi_n(x_0, x_n, \ldots, x_1)$ donne le nombre de fonctions 1-résilientes.

Notons que l'induction (2.6) peut s'écrire directement

$$\varphi_n(x_0, x_n, \ldots, x_1) = \prod_{\varepsilon_i \in \{-1,1\}} \left(1 + x_0 \prod_{i=1}^{n} x_i^{\varepsilon_i} \right). \tag{2.7}$$

Un simple algorithme maple en 7 lignes permet de calculer le nombre de fonctions 1-résilientes jusqu'à 6 variables. Il nécessite tout de même un peu moins de 26 mn de calcul, il n'est donc pas envisageable de l'utiliser pour $n > 6$.

---

f := proc(n) option remember ;
if n = 0 then 1 + x[0]
else $subs(x[0] = x[0] * x[n], f(n-1)) * subs(x[0]y = x[0]/x[n], f(n-1))$
end if end proc
$g := proc(n)$ options remember ;
$subs(x[0] = x[0] * x[n], f(n-1))$ end ;
convert(map($x \to x*x$, [coeffs(coeff(expand(collect(g(n),x[0])), $x[0], 2\hat{}(n-2))$)]),'+') ;

Pour les classes sans poids, nous avons le schéma d'induction

$$\begin{cases} \psi_1(x_1) & = & (1+x_1)(1+x_1^{-1}) = 2 + x_1 + x_1^{-1} \\ \psi_n(x_1, \ldots, x_n) & = & \psi_{n-1}(x_1, \ldots, \mathbf{x_{n-1}/x_n})\ \ \psi_{n-1}(x_1, \ldots, \mathbf{x_{n-1}x_n}) \end{cases} \tag{2.8}$$

On retrouve $\psi_n(x_1, \ldots, x_n) = \varphi_n(1, x_n, \ldots, x_1)$. Notons que si en posant $x_0 = 1$, on passe directement de la SGO des classes avec poids aux classes sans poids, l'opérateur $\circ$ est indispensable pour obtenir directement la cardinalité d'une classe $\alpha = \langle \delta_n, \ldots, \delta_1 \rangle$ sans passer par la relation

$$\langle \delta_n, \ldots, \delta_1 \rangle = \bigcup_{m \in \{0, \ldots, 2^n\}} \langle m, \delta_n, \ldots, \delta_1 \rangle$$

qui ne permet pas d'avoir des algorithmes plus efficaces que pour les classes avec poids.

L'algorithme maple suivant donne le nombre de fonctions sans corrélation d'ordre 1 pour $n = 6$ en 10 secondes.

---

f := proc (n) option remember ;
if n = 1 then 2+x[1]+1/x[1]
else subs(x[n-1] = x[n-1]*x[n], f(n-1))*subs(x[n-1] = x[n-1]/x[n], f(n-1))
end if end proc
convert(map(x → x*x end proc, coeffs(coeff(expand(collect(f(5), x[1])), x[1], 0))]), '+')

Remarque 2.1.2. *C'est en observant la série génératrice des classes sans poids que nous avons recherché l'opérateur $\star$ et une interprétation de celui-ci sur les fonctions booléennes.*

2.1.4 Classes normalisées

Le nombre de classes de corrélation est rapidement trop important pour pouvoir stocker en mémoire la cardinalité des classes. Pour diminuer le nombre de classes à mémoriser, on définit une seconde relation d'équivalence sur les classes. Les nouvelles classes obtenues, appelées classes normalisées, capturent deux transformations naturelles préservant la corrélation d'ordre 1 (et plus généralement la corrélation d'ordre k) :

(a) Échange entre les variables x_i et x_j.

(b) Échange entre les valuations $x_i = 0$ et $x_i = 1$.

L'ensemble des transformations engendrées par ces deux transformations élémentaires forment le sous-groupe des isomorphismes affines appelé *renaming* (renommage) par Strazdins [79].

Interprétation sur les fonctions booléennes Soit $f \in \mathcal{B}F_n$. On considère deux types de bijection $b_{ij} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ Dans les deux cas, nous obtenons une fonction booléenne g de $\mathcal{BF}_n$ telle que $g(a) = f(b)$, où $b = b_{ij}(a)$.

(a) Soit $a = (a_1, \ldots, a_n) \in \{0,1\}^n$. Alors $b = b_{ij}(a)$ avec $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ tel que $b_l = a_l$, pour tout $l \in \{1, \ldots, n\} \setminus \{i, j\}$, $b_i = a_j$ et $b_j = a_i$.

(b) Soit $a = (a_1, \ldots, a_n) \in \{0,1\}^n$. Alors $b = b_{ij}(a)$ avec $b = (b_1, \ldots, b_n) \in \{0,1\}^n$ tel que $b_l = a_l$, pour tout $l \in \{1, \ldots, n\} \setminus \{i\}$, $b_i = a_i \oplus 1$.

Interprétation sur les classes de corrélation Soit $\omega_1 = \langle m, \delta_n, \ldots, \delta_1 \rangle$.

(a) Soit $\omega_2 = \langle m, \alpha_n, \ldots, \alpha_1 \rangle$, telle que $\alpha_l = \delta_l$, pour $l \in \{1, \ldots, n\} \setminus \{i, j\}$, $\alpha_i = \delta_j$ et $\alpha_j = \delta_i$.

(b) Soit $\omega_2 = \langle m, \alpha_n, \ldots, \alpha_1 \rangle$, où $\alpha_l = \delta_l$, pour $l \in \{1, \ldots, n\} \setminus \{i\}$, $\alpha_i = -\delta_j$.

Dans les deux cas, $b_{i,j}$ forme une bijection entre les fonctions de ω_1 et celles de ω_2. Par conséquent tout renommage R préserve la cardinalité des classes.

Définition 2.1.12. *Une classe de corrélation $\omega = \langle m, \delta_n, \ldots, \delta_1 \rangle$ est appelée classe normale*

$$0 \leq \delta_n \leq \delta_{n-1} \leq \ldots \leq \delta_1.$$

On montre pour tout $\omega \in \Omega_n$, il existe une unique classe normalisée Θ obtenue par renommage. Nous noterons $N(\omega)$ cette classe normalisée. Notons qu'il peut exister plusieurs renommages R tel que $R(\omega) = \Theta$, mais que l'on peut définir un renommage canonique (voir la partie génération aléatoire).

FIGURE 2.2 – Nombre de classes normalisées

n	Fonctions booléennes	Classes	Classes normalisées
1	4	4	3
2	16	15	6
3	256	153	19
4	65536	5817	118
5	4294967296	936545	1755
6	18446744073709551616	632587361	73524

2.1.5 Ordre partiel entre les classes normalisées de même poids

Soient $\Theta_1 = \langle m, \delta_n^0, \ldots, \delta_1^0 \rangle$ et $\Theta_2 = \langle m, \delta_n^1, \ldots, \delta_1^1 \rangle$ deux classes normalisées de poids m. Nous dirons que Θ_1 est plus petite que Θ_1 et nous noterons $\Theta_1 \leq \Theta_2$ lorsque

$$\delta_i^0 \leq \delta_i^1, \text{pour tout } i \in \{1, \ldots, n\}.$$
$$\delta_j^0 < \delta_j^1, \text{pour au moins un } j \in \{1, \ldots, n\}.$$

On montre que si Θ_2 est valide –elle contient au moins une fonction booléenne– et $\Theta_1 \leq \Theta_2$ alors Θ_1 est valide. Nous avons de plus le théorème suivant

Théorème 2.1.5. *Si* $\Theta_1 \leq \Theta_2$ *alors* $|\Theta_1| < \Theta_2$.

On montre que pour tout m, il existe un maximum (toutes les classes sont plus petites que cet élément), il s'agit de la classe sans corrélation d'ordre 1 $\langle m, 0, \ldots, 0 \rangle$ lorsque m est pair et de la classe $\langle m, 1, \ldots, 1 \rangle$ lorsque m est impair (les δ_i ont toujours la même parité que m).

Une classe est un minimum local s'il n'existe pas plus petite qui est valide. L'ensemble des classes normalisées de poids m fixé forme donc un demi-treillis. On notera $\delta(\Theta)$ la longueur du chemin pour aller de la classe Θ au maximum et de manière plus générale $\delta(\omega)$ vaudra $\delta(N(\omega))$, pour toute classe ω.

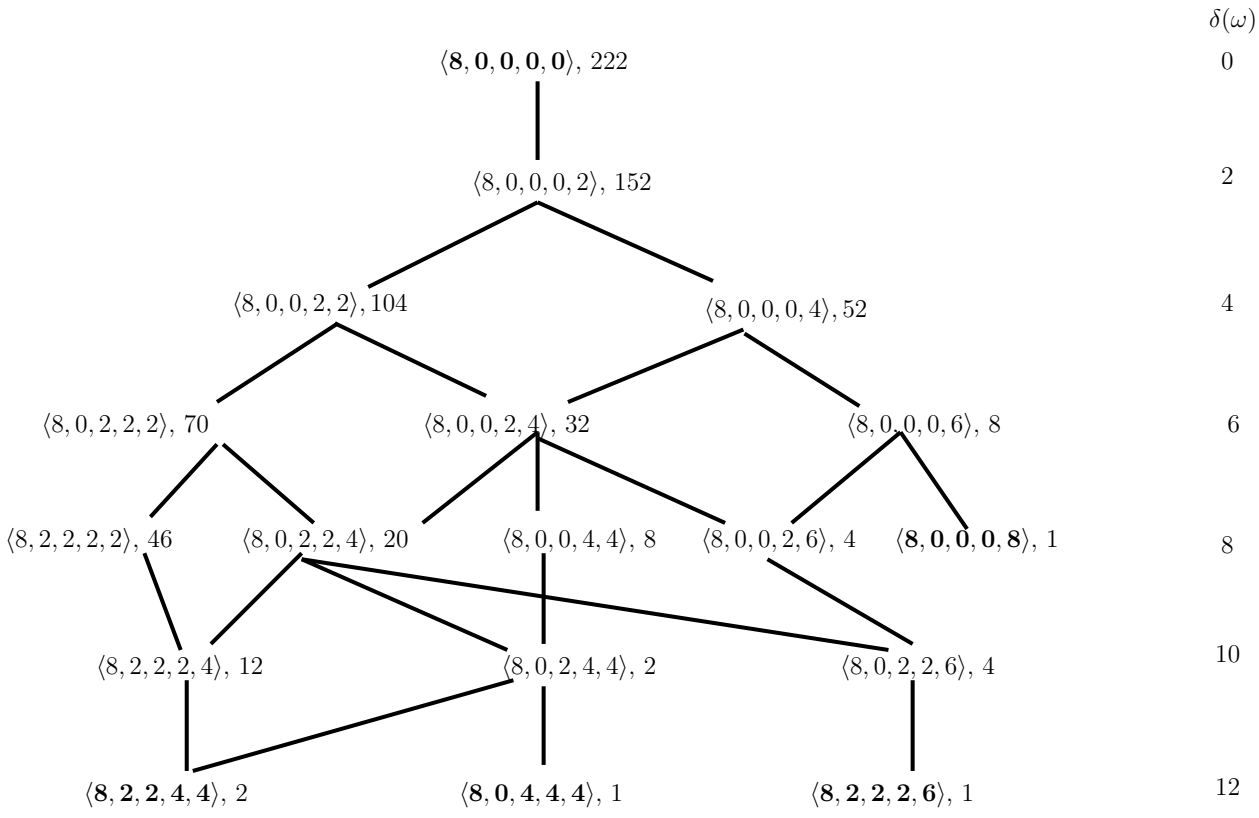


FIGURE 2.3 – Demi-treillis formé par les classes normalisées à 4 variables équilibrées

2.1.6 Nombre de fonctions 1-résilientes

n	$\mathrm{Res}^1{}_n$
5	807980
6	95259103924394
7	2347801575478885443949762268929 6
8	52198620942407957076910735856809911895553771388490307930972455149942

Notons qu'il est possible d'avoir une approximation des classes de corrélation d'ordre k par des méthodes asymptotiques [55, 58]. En particulier Eric Bach [55] a obtenu $2,347810\ 10^{31}$ comme approximation de Res_7^1.

2.1.7 Construction de fonction k-résilientes

Nous pouvons étendre la notion de corrélation et de résilience à plusieurs variables.

Définition 2.1.13. *Soit $n \in \mathbb{N}, f \in \mathcal{BF}_n$ et $k < n$. Nous dirons que f est sans corrélation d'ordre k lorsque, pour tout $(\epsilon_1, \ldots, \epsilon_k) \in \mathbb{F}_2^n$, il existe $m \leq 2^n$ tel la fonction $f\mid_{x_{i_1}=\epsilon_1,\ldots,x_{i_k}=\epsilon_k}$ est de poids m, pour tout $\{i_1, \ldots, i_k\} \subset \{1, \ldots, n\}$.*

Définition 2.1.14. *f est k-résiliente lorsqu'elle est sans corrélation d'ordre k et équilibrée.*

Il est possible d'utiliser les classes de corrélation afin d'obtenir des fonctions k-résilientes. Par exemple, soient ω une classes de corrélation à n variables et g et h deux fonctions de cette classe. Alors la fonction $f = g \star h \star \overline{g} \star \overline{h}$ est une fonction 2-résiliente à $n + 2$ variables, où $\overline{g} = 1 \oplus g$.

De manière générale, on obtient une fonction k-résiliente f_n avec la récurrence suivante

1. $f_{m+2} = g \star h \star \overline{g} \star \overline{h}$.
2. $f_{m+i+1} = g_{m+i-1} \star h_{m+i-1} \star h_{m+i-1} \star g_{m+i-1}$, avec $f_{m+i} = g_{m+i-1} \star h_{m+i-1}$.

Nous pouvons comparer cette construction avec celle de Maroiana-MacFarland.

Soient n, k et $r \in \mathbb{N}$ tels que $0 \leq k < r < n$ et φ une application de $\mathbb{F}_2^{n-r}$ vers $\mathbb{F}_2^r$ telle que pour tout u de $\mathbb{F}_2^{n-r}$, $w_H(\varphi(u)) > k$ et g une fonction booléenne équilibrée à $n - r$ variables. Alors la fonction

$$f_{\varphi,g}(x, y) = x \cdot \varphi(y) + g(y)$$

est k-résiliente.

C'est à ma connaissance, la construction générale (valable pour tout n) permettant d'obtenir le plus de fonctions k-résilientes. Elle donne la borne inférieure suivante

$$\sum_{r=k+1}^{n-1} \left(\left(\sum_{i=k+1}^{r} \binom{r}{i} \right)^{2^{n-r}} + \binom{2^{2^{n-r}}}{2^{2^{n-r-1}}} \right).$$

Notre méthode de construction permet de construire autant de fonctions que nous pouvons construire de fonctions 1-résilientes à $n-k+1$. Nous obtenons ainsi pour $n=10$:

k	Notre borne inférieure	Borne inférieure de Maroiana-MacFarland
2	5.13×10^{115}	5.4×10^{44}
3	4.04×10^{67}	8.0×10^{24}
4	2.34×10^{31}	3.4×10^{13}
5	9.52×10^{13}	1.9×10^{7}
6	807980	8912
7	222	117

2.2 Codage énumératif et génération aléatoire

2.2.1 Introduction

Nous allons présenter dans cette section une méthode très efficace pour générer aléatoirement une fonction 1-résiliente avec la distribution uniforme. Pour 8 variables, nous sommes en mesure de générer en $5,4$ secondes en moyenne une fonction parmi un espace de 10^{68} fonction.

Ce travail a été réalisé toujours avec Viola et un de ses étudiant, Nicolas Carrasco qui a réalisé le programme. C'est un travail essentiellement algorithmique et Carrasco s'est particulièrement attaché à ce que le maximum de calculs puissent être effectués dynamiquement.

Nous avons déjà vu que la méthode des classes permet de compter et d'énumérer l'ensemble des fonctions d'une classe de corrélation. Nous allons maintenant proposer une méthode pour générer aléatoirement avec la distribution uniforme une fonction booléenne d'une classe ω. L'idée est d'avoir deux bijections réciproques l'une de l'autre, la première associe un numéro de 0 à $|\omega|-1$ à chaque fonction de ω et la seconde associe une fonction à chaque numéro de 0 à $|\omega|-1$. On appelle codage énumératif la donnée de deux algorithmes réalisant ces deux fonctions. La performance d'un codage énumératif dépend à la fois de la mémoire et du temps de calcul. Nous allons voir que la méthode des classes qui permet de décomposer une fonction en plusieurs fonctions de moins de variables donne directement un codage énumératif. La véritable difficulté a été de concevoir des algorithmes les plus efficaces possibles. Il faut pour cela trouver un bon compromis pour séparer en deux les informations, d'un côté, les informations que l'on doit nécessairement mémoriser et, de l'autre, celles qu'il faut calculer dynamiquement lors de la construction d'une fonction.

La décomposition en somme, produit cartésien et séquence a déjà été formalisée dans [66] et appelée *méthode récursive*. Notons les auteurs considèrent bien la génération aléatoire uniforme, mais ils ne mentionnent pas le codage énumératif bien qu'il soit sous-jacent. L'avantage pratique du codage énumératif est que nous pouvons concentrer tous les aléas en entrée plutôt qu'au fur et a mesure des choix des décompositions.

2.2.2 Idée de la méthode

Soit ω une classe de corrélation. Nous allons implanter deux fonctions **Retrieve**$[\omega]$ et **Rank**$[\omega]$ satisfaisant les conditions suivantes

1. **Retrieve**$[\omega]$ est une bijection de $\{0, \ldots, |\omega| - 1\}$ vers ω.
2. **Rank**$[\omega]$ est une bijection de ω vers $\{0, \ldots, |\omega| - 1\}$.
3. **Retrieve**$[\omega](r) = f \iff$ **Rank**$[\omega](f) = r$.
4. **Retrieve**$[\omega]$ et **Rank**$[\omega]$ sont calculées de manière efficace.

Pour pouvoir distinguer les deux fonctions et leur implantation, nous noterons Retrieve$[\omega]$ et Rank$[\omega]$ les algorithmes que nous avons définis pour implanter respectivement **Retrieve**$[\omega]$ et **Rank**$[\omega]$. Rappelons qu'une décomposition (ω^0, ω^1) de ω vérifie

$$\omega^0 \star \omega^1 = \omega.$$

À toute classe ω, on associe la liste de ses décompositions $(\omega^{0(1)}, \omega^{1(1)}), (\omega^{0(2)}, \omega^{1(2)}), (\omega^{0(3)}, \omega^{1(3)}), \ldots$.

Pour chaque classe ω^R, nous allons associer un arbre conceptuel appelé arbre des calculs qui contiendra toutes les décompositions possibles à chaque étape. Toutes les algorithmes de comptage, d'énumération, ainsi que Retrieve$[\omega]$ et Rank$[\omega]$ peuvent être expliqués sur cette arbre (voir Figure 2.4).

Retrieve$[\omega]$ et Rank$[\omega]$ s'effectuent en trois étapes :

1. Décomposition. Retourne la bonne décomposition $(\omega^{0(i)}, \omega^{1(i)})$ de ω.
2. Appels récursifs. Deux appels récursifs sur les nœuds $\omega^{0(i)}$ and $\omega^{1(i)}$.
3. Composition. Composition des deux résultats obtenus par les deux appels récursifs.

L'arbre des calcul de ω^R contient trois types de nœuds,

- **Nœuds rectangulaires** représentant une classe. Ses fils sont les décompositions possibles (ω^0, ω^1) telles que $\omega^0 \star \omega^1 = \omega$.
- **Nœuds ovales** représentant une décomposition (ω^0, ω^1). Ses fils sont deux nœuds rectangulaires contenant ω^0 et ω^1.
- **Feuille**, classe à 1 variable. Chaque feuille correspond à une unique fonction booléenne.

Les nœuds ovales contiennent des information dépendant de l'algorithme. Par exemple, pour le comptage (Corollaire 2.1.1), ils contiennent le produit des cardinalités des deux fonctions (voir Figure 2.4).

Le cas de base –les fonctions ont une variable– est facile à traiter car les quatre classes ω à une variable ne possède qu'une fonction, donc il n'existe qu'une implantation possible de **Retrieve**$[\omega]$ et **Rank**$[\omega]$. Pour les autres classes, on introduit une fonction auxiliaire *Cumulative* telle que Cumulative$[i]$ retourne le nombre de fonctions construites avec l'une des $i - 1$ premières décompositions.

$$\begin{array}{lcll} \text{Cumulative}[1] & = & 0, & \\ \text{Cumulative}[i] & = & \sum_{j=1}^{i-1} |\omega^{0(j)}| \times |\omega^{1(j)}|, & \text{pour tout } i > 1. \end{array}$$

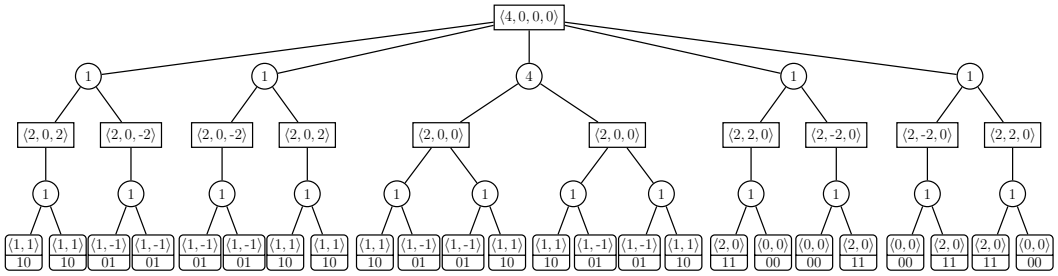


FIGURE 2.4 – Arbre des calculs de la classe $\langle 4, 0, 0, 0\rangle$.

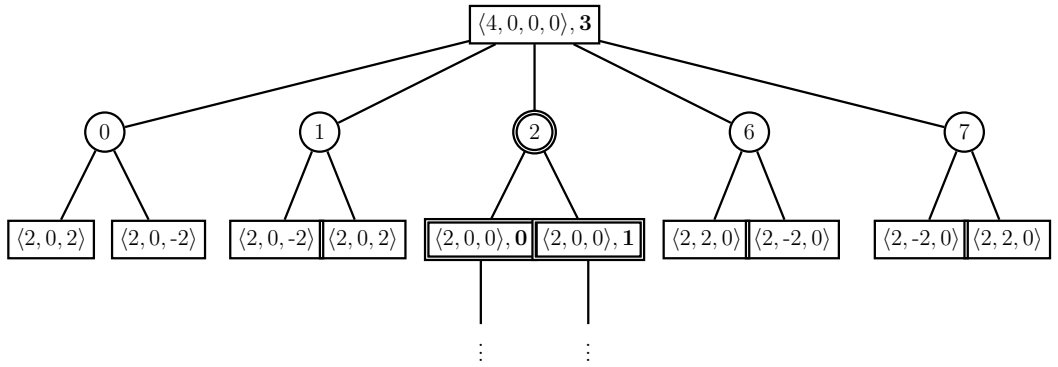


FIGURE 2.5 – Premier étape de RETRIEVE$[\langle 4, 0, 0, 0\rangle](3)$. Les nœuds rectangulaires sont étiquetés par le classe et le rang de sa fonction.

2.2.3 L'algorithme RETRIEVE$[\omega]$

1. DÉCOMPOSITION. On retourne le plus grand i tel que $[i] \leq r$.
   Ainsi chaque décomposition est choisie par exactement $n_i = |\omega^{0(i)}| \times |\omega^{1(i)}|$ différents index r_ω.
2. APPELS RÉCURSIFS. Pour chaque r_ω, on calcule $m = r_\omega -$ Cumulative $[i]$.
   r^0 and r^1 sont alors les index respectifs de $\omega^{0(i)}$ et $\omega^{1(i)}$:

   $$r^0 = m \text{ div } |\omega^{1(i)}|, \quad r^1 = m \bmod |\omega^{1(i)}|.$$

   On effectue ensuite les deux appels récursifs :

   $$f^0 = \text{RETRIEVE}[\omega^{0(i)}](r^0), \quad f^1 = \text{RETRIEVE}[\omega^{1(i)}](r^1).$$

3. COMPOSITION. Le résultat final est

   $$\text{RETRIEVE}[\omega](r_\omega) = \text{RETRIEVE}[\omega^{0(i)}](r^0) \star \text{Retrieve}[\omega^{1(i)}](r^1).$$

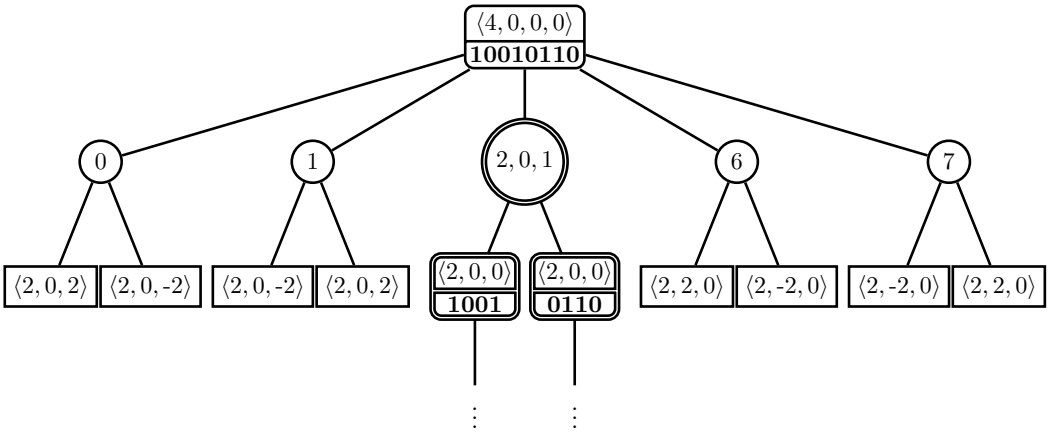


FIGURE 2.6 – Première étape de RANK$[\langle 4,0,0,0\rangle](10010110)$.

2.2.4 L'algorithme RANK$[\omega]$

Les nœuds rectangulaires vont contenir une classe ω et une fonction booléenne f de cette classe et les nœuds ovals, un triplet (n_i, r^0, r^1), où $n_i = |\omega^{0(i)}| \times |\omega^{1(i)}|$, et r^0 et r^1 sont les index calculés par les appels récursifs. Les trois étapes de l'algorithme sont :

1. DÉCOMPOSITION : Soit i tel que $\Omega(f^0) = \omega^{0(i)}$ et $\Omega(f^1) = \omega^{1(i)}$.
2. APPELS RÉCURSIFS.

$$r^0 = \text{RANK}[\omega^{0(i)}](f^0), \quad r^1 = \text{RANK}[\omega^{1(i)}](f^1).$$

3. COMPOSITION. Soit M =Cumulative $[i]$. On calcule

$$\text{RANK}[\omega](f) = M + r^0|\omega^{1(i)}| + r^1.$$

2.2.5 Classes normalisées et permutations signées

Pour retarder l'explosion combinatoire du nombre de classes, nous avons effectué le codage énumératif en mémorisant uniquement les classes normalisées. Nous avons défini une bijection entre les fonctions d'une classe et celles de sa classe normalisée. Une permutation signée permet de passer d'une classe à l'autre.

Définition 2.2.1. *Soit σ une permutation sur $\{1,\ldots,n\}$ et $\varepsilon = (\varepsilon_1,\ldots,\varepsilon_n) \in \{-1,1\}^n$. $\sigma_\varepsilon = (\varepsilon_1\sigma(1),\ldots,\varepsilon_n\sigma(n))$ sera appelée permutation signée de $\{1,\ldots,n\}$. On définit également son inverse σ_ε*

$$\sigma_\varepsilon^{-1} = \left(\varepsilon_{\sigma^{-1}(1)}\,\sigma^{-1}(1),\ldots,\varepsilon_{\sigma^{-1}(n)}\,\sigma^{-1}(n)\right).$$

Définition 2.2.2. *On associe à toute permutation signée σ_ε, une fonction $\mathcal{C}_{\sigma_\varepsilon} : \Omega_n \mapsto \Omega_n$ vérifiant*

$$\mathcal{C}_{\sigma_\varepsilon}(\langle m, \delta_n,\ldots,\delta_1\rangle) = \langle m, \varepsilon_{\sigma(n)}\delta_{\sigma(n)},\ldots,\varepsilon_{\sigma(1)}\delta_{\sigma(1)}\rangle.$$

Nous avons alors l'équivalence

$$\mathcal{C}_{\sigma_\varepsilon}(\omega_1) = \omega_2 \iff \mathcal{C}_{\sigma_\varepsilon^{-1}}(\omega_2) = \omega_1.$$

Rappelons qu'une classe normalisée $\Theta = \langle m, \delta_n, \ldots, \delta_1 \rangle$ vérifie

$$0 \leq \delta_n \leq \delta_{n-1} \leq \ldots \leq \delta_1.$$

Pour toute classe ω il existe une permutation signée σ_ε et une classe normalisée Θ telle que $\mathcal{C}_{\sigma_\varepsilon}(\omega) = \Theta$. Il peut exister plusieurs permutations signées σ_ε telles que $\mathcal{C}_{\sigma_\varepsilon}$, dans notre implémentation nous avons fixé une permutation signée canonique [71].

Nous avons comment passé d'une classe à une autre, il faut également pouvoir la permutation équivalente sur les fonctions booléennes.

Définition 2.2.3. *Soit σ_ε une permutation, à chaque ε_i on fait correspondre η_i tel que $\varepsilon_i = (-1)^{\eta_i}$. On définit la bijection $F_{\sigma_\varepsilon} : \mathcal{B}F_n \mapsto \mathcal{B}F_n$*

$$F_{\sigma_\varepsilon}(f) = g,$$

telle que $g(a_1, \ldots, a_n) = f(a_{\sigma(1)} \oplus \eta_{\sigma(1)}, \ldots, a_{\sigma(n)} \oplus \eta_{\sigma(n)})$.

Pour tous f et $g \in \mathcal{B}F_n$, nous avons l'équivalence

$$F_{\sigma_\varepsilon}(f) = g \iff F_{\sigma_\varepsilon^{-1}}(g) = f,$$

Nous allons maintenant voir comment implémenter le codage énumératif avec les classes normalisées et les permutations signées.

Retrieve 2) Appels récursifs. Soient $\theta^0 = N(\omega^{0(i)})$ et $\theta^1 = N(\omega^{1(i)})$.
Soient r^0 et r^1, les index respectifs de θ^0 et θ^1 par $r^0 = m \text{ div } |\theta^0|$ et $r^1 = m \bmod |\theta^1|$.
On effectue les appels récursifs suivants

$$g^0 = \textsc{Retrieve}[\theta^0](r^0),\ g^1 = \textsc{Retrieve}[\theta^1](r^1).$$

3) Composition. Calculer les permutations signées canoniques telles que $\sigma^0_{\varepsilon^0}$ and $\sigma^1_{\varepsilon^1}$ telles que

$$\theta^0 = \mathcal{C}_{\sigma^0_{\varepsilon^0}}(\omega^{0(i)}) \text{ et } \theta^1 = \mathcal{C}_{\sigma^1_{\varepsilon^1}}(\omega^{1(i)}).$$

Le résultat final s'obtient par $F_{\left(\sigma^0_{\varepsilon^0}\right)^{-1}}\left(g^0\right) \star F_{\left(\sigma^1_{\varepsilon^1}\right)^{-1}}\left(g^1\right)$.

Rank 2) Recursive calls. Soient $\theta^0 = N(\omega^{0(i)})$ et $\theta^1 = N(\omega^{1(i)})$. On calcule les permutations signées canoniques $\sigma^0_{\varepsilon^0}$ et $\sigma^1_{\varepsilon^1}$ telles que

$$\theta^0 = \mathcal{C}_{\sigma^0_{\varepsilon^0}}(\omega^{0(i)}) \text{ et } \theta^1 = \mathcal{C}_{\sigma^1_{\varepsilon^1}}(\omega^{1(i)}).$$

Soient $g^0 = F_{\sigma^0_{\varepsilon^0}}(f^0)$ et $g^1 = F_{\sigma^0_{\varepsilon^1}}(f^1)$. On applique les appels récursifs suivants

$$r^0 = \textsc{Rank}[\theta^0](g^0), \quad r^1 = \textsc{Rank}[\theta^1](g^1).$$

3) Composition. Soit M =Cumulative $[i]$. On retourne

$$\textsc{Rank}[\omega](f) = M + r^0|\Theta^1| + r^1.$$

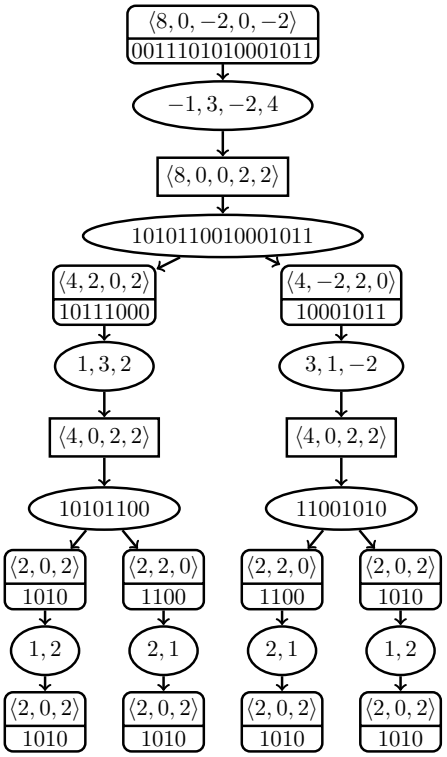


FIGURE 2.7 – Classes normales et la fonction Retrieve

Calcul dynamique de l'index de la décomposition
Les algorithmes sont réellement efficaces car nous ne conservons pas en mémoire les cardinalités des différentes décompositions. La fonction Cumulative est calculée dynamiquement. De plus nous avons une méthode plus efficace pour la calculer lors du premier appel récursif sur les classes équilibrées à $n-1$ variables.

2.3 Méthode des classes pour les fonctions k-résilientes

La méthode des classes peut être étendue aux fonctions sans corrélation d'ordre k et aux fonctions k-résilientes. Le système d'équation est obtenu à partir de la transformée de Fourier.

2.3.1 Transformée de Fourier et méthode des classes

La transformée de Fourier de f_n est définie pour tout $u \in \{0,1\}^n$, par

$$\hat{f}_n(u) = \sum_{x \in \{0,1\}^n} f_n(x)\ (-1)^{x \cdot u},$$

où $x = x_1 \dots x_n$, $u = u_1 \dots u_n$ (x et u sont vus comme une suite de n bits), $x \cdot u$ désigne le produit cartésien $x_1 u_1 \oplus \dots \oplus x_n u_n$.

Remarque 2.3.1. $w_H(f) = \hat{f}_n(0 \dots 0)$.

Proposition 2.3.1. *f est sans corrélation d'ordre k si et seulement si $\hat{f}_n(u) = 0$, pour tout mot u de poids inférieur ou égal à k.*

Soient f^0 et $f^1 \in \mathcal{BF}_n$ tels que

$$f(x) = (1 \oplus x_n) f^0(y) \oplus x_n f^1(y).$$

Soient $u = u_1 \dots u_n \in \mathbb{F}_2^n$ et $v = u_1 \dots u_{n-1}$, on montre que l'on a

$$\begin{aligned} \hat{f}_\chi(u) &= \hat{f}^0_\chi(v) + \hat{f}^1_\chi(v), \text{ si } u = v0. \\ \hat{f}_\chi(u) &= \hat{f}^0_\chi(v) - \hat{f}^1_\chi(v), \text{ si } u = v1. \end{aligned}$$

Définition 2.3.1. *On définit $\omega_k(f) = \langle \delta(u) \rangle$, où $\delta(u) = \hat{f}_n(u)$ et $w_H(u) \leq k$ (notons que $\omega_k(f)$ est une séquence, il faut donc définir un ordre entre les $\delta(u)$).*

Soient $\omega_k^0 = \langle \delta^0(v) \rangle$ et $\omega_k^1 = \langle \delta^1(v) \rangle$ les classes de f^0 et f^1.

L'opérateur sur les classes $*$ vérifiant

$$\omega_k = \omega_k^0 \star \omega_k^1.$$

est défini par le système suivant lorsque $n - 1 \geq k$

$$\begin{aligned} \delta(v0) &= \delta^0(v) + \delta^1(v), \text{ pour tout } v \in \mathbb{F}_2^{n-1}, w_H(v) \leq k \\ \delta(v1) &= \delta^0(v) - \delta^1(v), \text{ pour tout } v \in \mathbb{F}_2^{n-1}, w_H(v) \leq k-1 \end{aligned}$$

et lorsque $n - 1 = k$ par

$$\begin{aligned} \delta(v0) &= \delta^0(v) + \delta^1(v), \text{ pour tout } v \in \mathbb{F}_2^{n-1}, w_H(v) \leq k-1 \\ \delta(v1) &= \delta^0(v) - \delta^1(v), \text{ pour tout } v \in \mathbb{F}_2^{n-1}, w_H(v) \leq k-1 \end{aligned},$$

2.3.2 Construction de fonctions k-résilientes

Les résultats obtenus pour la 1-résilience ont requis une implémentation très conséquente pour pouvoir compter et générer jusqu'à 8 variables. La définition 2.3.1 permet de compter et d'énumérer, mais pour que cela soit efficace, il faut ajouter des optimisations. En particulier, il faut éviter au maximum l'explosion combinatoire en introduisant des relations d'équivalences comme pour les classes normalisées.

Nombre de fonctions 2 résilientes
Avec la définition 2.3.1, j'ai pu calculer (sans optimisation) le nombre de fonctions 2-résilientes jusqu'à 6 variables. Nous avons ainsi 16750860 fonctions 2-résilientes.

Chapitre 3

Représentation et décomposition des fonctions booléennes

3.1 Décomposition selon le poids de Hamming et le degré algébrique

3.1.1 Table de vérité et forme algébrique normale

Monômes et mintermes
Soient $x = (x_1, \ldots, x_n)$ et $u = (u_1, \ldots, u_n)$ une valuation des x_i sur $\mathbb{F}_2^n$, on notera x^u le monôme $x_1^{u_1} \ldots x_n^{u_n}$

M_u désignera le minterme $\prod_{i=1}^{n}(x_i \oplus u_i \oplus 1)$, la fonction booléenne prenant la valeur 1 uniquement sur la valuation u.

Forme algébrique normale FAN
Une fonction booléenne peut être vue comme une somme de monômes

$$f = \bigoplus_{u \in \mathbb{F}_2^n} \alpha_u x^u,$$

où $\alpha_u \in \mathbb{F}_2$. Nous noterons $A(f)$ la représentation de f comme un mot de longueur 2^n codant les monômes apparaissant dans la FAN de f, c'est-à-dire

$$A(f) = a_1 \ldots a_{2^n},$$

où $a_k = \alpha_u$, avec $k = k(u) = \sum_{i=1}^{n} u_i \, 2^{i-1}$.

Table de vérité et mintermes
Une fonction booléenne peut aussi être vue comme une somme de mintermes

$$f = \bigoplus_{u \in \mathbb{F}_2^n} \beta_u M_u,$$

où $\beta_u \in \mathbb{F}_2$. De la même manière, nous noterons $T(f)$ –comme cela a déjà été fait dans la section précédente– la représentation de f comme un mot de longueur 2^n codant les minermes apparaissant dans la table de vérité de f, c'est-à-dire

$$T(f) = t_1 \dots t_{2^n},$$

où $t_k = \beta_u$, avec $k = k(u) = \sum_{i=1}^{n} u_i \, 2^{i-1}$.

Notons que dans les deux cas, nous faisons une distinction entre une définition de la fonction booléenne (somme de monômes et somme des mintermes) et une représentation (ici par un mot de longueur 2^n).

3.1.2 Décomposition de Shannon vs décomposition de Reed-Müller

Décomposition de Shannon
Rappelons la décomposition de Shannon (2.2) introduite à la section précédente et qui a servi de point de départ à la méthode des classes. Pour toute fonction f à n variables, il existe deux fonctions uniques à $n-1$ variables f_S^0 et f_S^1 vérifiant

$$f = (1 \oplus x_i) f_S^0 \oplus x_i f_S^1.$$

Nous avons vu dans la section précédente que cette décomposition correspondait pour $i = 1$ à une concaténation de deux mots de longueur 2^{n-1} (2.1)

$$T(f) = T(f_S^0) \star T(f_S^1).$$

$T(f_S^0)$ (resp. $T(f_S^1)$) contient tous les mintermes M_u apparaissant dans f avec $u_1 = 0$ (resp. $u_1 = 1$).

Décomposition de Reed-Müller
Il existe deux fonctions uniques à $n-1$ variables f_R^0 et f_R^1 vérifiant

$$f = f_R^0 \oplus x_i f_R^1.$$

Comme précédemment nous avonc la concaténation

$$A(f) = A(f_R^0) \star A(f_R^1).$$

$A(f_S^0)$ (resp. $A(f_S^1)$) contient tous les monômes x^u de la FAN de f, avec $u_1 = 0$ (resp. $u_1 = 1$).

On en déduit immédiatement $f_S^0 = f_R^0$ et $f_R^1 = f_S^0 \oplus f_S^1$.

La décomposition de Reed-Müller est très souvent utilisée sur les codes de Reed-Müller [70] et les circuits booléens et BDD [80]. Elle est moins utilisée que lé décomposition de Shannon pour les fonctions booléennes pour la cryptographie car généralement on souhaite maitriser le poids de Hamming.

3.1.3 Enumération selon le poids de Hamming

De la décomposition de Shannon, nous déduisons immédiatement une relation sur les poids de Hamming :

$$w_H(f) = w_H(f_S^0) + w_H(f_S^1).$$

Posons $w_H(f) = m$, $w_H(f_S^0) = m^0$ et $w_H(f_R^1) = m^1$.

Soit $\mathcal{C}(n, m)$ la classe des fonctions à n variables de poids de Hamming m.

Enumération

$$\mathcal{C}(n,m) = \begin{cases} \bigcup\limits_{m^0 \in \{0,\ldots,m)\}} \mathcal{C}(n-1,m^0) \times \mathcal{C}(n-1,m-m^0), \text{ pour } m \leq 2^{n-1} \\ \bigcup\limits_{m^0 \in \{m-2^{n-1},\ldots,2^{n-1})\}} \mathcal{C}(n-1,m^0) \times \mathcal{C}(n-1,m-m^0), \text{ pour } m > 2^{n-1} \end{cases} \tag{3.1}$$

Comptage

$$|\mathcal{C}(n,m)| = \begin{cases} \sum\limits_{m^0 \in \{0,\ldots,m)\}} |\mathcal{C}(n-1,m^0)||\mathcal{C}(n-1,m-m^0)|, \text{ pour } m \leq 2^{n-1} \\ \sum\limits_{m^0 \in \{m-2^{n-1},\ldots,2^{n-1})\}} |\mathcal{C}(n-1,m^0)||\mathcal{C}(n-1,m-m^0)|, \text{ pour } m > 2^{n-1} \end{cases} \tag{3.2}$$

3.1.4 Enumération selon le degré algébrique

Degré algébrique

Soit $f \in \mathcal{BF}_n$, $d(f)$ est la taille du plus monôme de la FAN de f, c'est-à-dire

$$d(f) = \max\{w_H(u) | u \in \mathbb{F}_2^n \text{ et } \alpha_u = 1\}.$$

Soient f_R^0 et f_R^1 tels que $f = f_R^0 \oplus x_i f_R^1$, nous avons la relation

$$d(f) = sup(d(f_R^0), d(f_R^1) + 1).$$

Soit $\mathcal{D}(n, d)$ (resp. $\mathcal{D}(n \leq d)$) la classe des fonctions booléennes à n variables de degré d (resp. $\leq d$) (pour $n < d$, $\mathcal{D}(n, d) = \emptyset$).

Tête d'une fonction

Pour toute fonction booléenne f, on appelle tête de f –que l'on note $H(f)$– l'ensemble des monômes de plus haut degré.

Enumération

$$\mathcal{D}(n,d) = \big(\mathcal{D}(n-1,d) \times \mathcal{D}(n-1,\leq d)\big) \cup \big(\mathcal{D}(n-1,\leq d-1) \times \mathcal{D}(n-1,d-1)\big). \tag{3.3}$$

Comptage

$$|\mathcal{D}(n,d)| = \big(|\mathcal{D}(n-1,d)|\ |\mathcal{D}(n-1,\leq d)|\big) + \big(|\mathcal{D}(n-1,\leq d-1)|\ |\mathcal{D}(n-1,d-1)|\big). \quad (3.4)$$

3.1.5 Enumération selon le poid de Hamming et le degré algébrique

Avec la décomposition de Shannon
$f = (1 \oplus x_i) f_S^0 \oplus x_i f_S^1$.

$$\begin{array}{lll} m = w_H(f) & m^0 = w_H(f_S^0) & m^1 = w_H(f_S^1) \\ d = d(f) & d^0 = d(f_S^0) & d^1 = d(f_S^1 \end{array}$$

On définit $\mathcal{E}(n,m,d)$ (resp. $\mathcal{E}(n,m,\leq d)$) la classe des fonctions booléennes de poids de Hamming m et de degré d (resp. $\leq d$). d, d^0, d^1 leur degré algébrique respectif et m, m^0 et m^1 leur poids de Hamming respectif. Nous avons les 4 décompositions possibles

E_1 $d^0 = d-1$ et $d^1 \leq d-2$.

E_2 $d^0 \leq d-2$ et $d^1 = d-1$.

E_3 $d^0 = d^1 = d-1$ et $H(f_{n-1}^0) \neq H(f_{n-1}^1)$.

E_4 $d^0 = d^1 = d$ et $H(f_{n-1}^0) = H(f_{n-1}^1)$.

Soit $\mathcal{E}(n,m,d,H)$ la classe des fonctions booléennes de $\mathcal{E}(n,m,d)$ de tête H.

On obtient donc

$$\mathcal{E}(n,m,d) = \bigcup_{m^0=0}^{inf(m,2^{n-1})} E_1 \cup E_2 \cup E_3 \cup E_4.$$

$$\begin{aligned} E_1 &= \mathcal{E}(n-1,m^0,d-1) \times \mathcal{E}(n-1,m-m^0,\leq d-2) \\ E_2 &= \mathcal{E}(n-1,m^0,\leq d-2) \times \mathcal{E}(n-1,m-m^0,d-1) \\ E_3 &= \textstyle\bigcup_{H^0} \mathcal{E}(n-1,m^0,\leq d-1,H^0) \times \big(\mathcal{E}(n-1,m-m^0,d-1) \setminus \mathcal{E}(n-1,n-m^0,d-1,H^0)\big). \\ E_4 &= \textstyle\bigcup_{H^0} \mathcal{E}(n-1,m^0,\leq d,H^0) \times \big(\mathcal{E}(n-1,m-m^0,d,H^0). \end{aligned}$$

Cette décomposition ne peut malheureusement pas être utilisée pour un algorithme, en effet, cela fait intervenir un grand nombre de termes, nous avons $2^{\binom{n}{d}} - 1$ têtes possibles de degré d pour n variables.

Avec la décomposition de Reed-Müller
$f = f_R^0 \oplus x_i f_R^1, d = d(f), d^0 = d(f_R^0), d^1 = d(f_R^1)$.

Calcul du degré

1. $d^0 = d$ et $d^1 \leq d-1$.
2. $d^0 \leq d-1$ et $d^1 = d-1$.

Soit $(a_1,\ldots,a_n) \in \{0,1\}^n$ de $(x_1,\ldots,x_n)$.

Calcul du poids de Hamming
$f_n(a_1, \ldots, a_n) = 1$ dans les trois cas suivants

1. $f_R^0(a_1, \ldots, a_{n-1}) = 1$ et $a_n = 0$.
2. $f_R^0(a_1, \ldots, a_{n-1}) = 1$, $a_n = 1$ et $f_R^1(a_1, \ldots, a_{n-1}) = 0$.
3. $f_R^0(a_1, \ldots, a_{n-1}) = 0$, $a_n = 1$ et $f_R^1(a_1, \ldots, a_{n-1}) = 1$.

La décomposition de Reed-Müller ne semble pas du tout adaptée pour calculer le poids de Hamming de f_n.

3.2 Transformée de Möbius sur les polynômes

Transformée de Möbius
La transformée de Möbius est une fonction très utilisée sur les fonctions booléennes. Elle a été étudiée intensivement dans la thèse de P. Guillon [67].

Nous avons vu que le poids de Hamming est calculé avec les mintermes présents dans la table de vérité et que le degré algébrique s'obtient avec les monômes présents dans la FAN. La transformée de Möbius permet de passer de la table de vérité à la FAN.

$$\begin{array}{rcl} \mu : \mathcal{BF}_n & \longleftrightarrow & \mathcal{BF}_n \\ f & \longmapsto & \mu(f), \end{array}$$

telle que , pour tout $f \in \mathcal{BF}_n$

$$f = \bigoplus_{u \in \mathbb{F}_2^n} \mu(f)(u) x^u.$$

La dualité entre mintermes et monômes apparaît avec les relations suivantes
Soit $u \in \mathbb{F}_2^n$,

$$\left\{ \begin{array}{rcl} x^u & = & \bigoplus_{u \preceq v} M_v \\ M_u & = & \bigoplus_{u \preceq v} x^v \end{array} \right. \tag{3.5}$$

Relation entre monômes et mintermes
Soit $u \in \mathbb{F}_2^n$, $\mu(x^u) = M_u$ et $\mu(M_u) = x^u$.

Transformée de Möbius, table de vérité et FAN
Soient f et $g \in \mathcal{BF}_n$, les trois assertions suivantes sont équivalentes

1. $\mu(f) = g$.
2. $\mu(g) = f$.
3. $T(f) = A(g)$.
4. $A(f) = T(g)$.

Notons que les deux premières assertions impliquent que μ est une involution ($\mu^2(f) = f$).

Proposition 3.2.1. $f(u) = \bigoplus_{v \preceq u} \mu(f)(v)$.

Preuve. On reprend la définition 3.3.1 et on observe que pour tous v et $u \in \mathbb{F}_2^n$, $v^u = 1$ si et seulement si $u \prec v$. □

3.2.1 Manipulation de polynômes

Nous allons voir dans cette partie comment définir récursivement la transformée de Möbius en manipulant directement des polynômes.

Polynôme et indéterminées
Nous allons considérer les polynômes de manière formelle, c'est-à-dire sans évaluation (comme on le fait pour les séries formelles). Les x_i désignerons ainsi des indéterminées au lieu de variables. On définit un polynôme P à k indéterminées $x_{i_1}, \ldots, x_{i_k}$ comme une somme de monômes sur les indéterminées $x_{i_1}, \ldots, x_{i_k}$, où $i_1, \ldots, i_k \in \mathbb{N}$. Nous noterons $\mathcal{P}(k, n)$ l'ensemble de ces polynômes où $i_1, \ldots, i_k \leq n$ et $\mathcal{P}(n)$ pour $k \leq n$ non fixé. Donc n donne une borne maximale des indices des indéterminées de P, $P \in \mathcal{P}(n)$ lorsque toutes les indéterminées x_i de P vérifient $i \leq n$.

Remarque 3.2.1. *$(\mathcal{P}(n)_{n\in\mathbb{N}})$ ne forme pas une partition de l'ensemble des polynômes à un nombre fixé d'indéterminées. En particulier*

$$\mathcal{P}(n) \subset \mathcal{P}(n+1),$$

pour tout $n \in \mathbb{N}$.

Nous noterons $\mathcal{P} = \cup_{n\in\mathbb{N}}\mathcal{P}(n)$. Soit $P \in \mathcal{P}$, on définit $d(P)$ comme étant le petit n tel que $P \in \mathcal{P}(n)$. Autrement dit, x_n est une indéterminée de P et toutes les autres indéterminées x_i vérifient $i < n$.

Remarque 3.2.2. *Nous écrirons les polynômes en utilisant le symbole $+$ au lieu du $\oplus$ pour les fonctions, les coefficients restant toujours sur $\mathbb{F}_2$. Cela permettra de repérer directement si l'on manipule un polynôme ou une fonction booléenne.*

Notons $[n] = \{1, \ldots, n\}$ et $\mathcal{P}([n])$ l'ensemble des parties de $[n]$. $P \in \mathcal{P}(n)$ si et seulement s'il existe $E \subset \mathcal{P}([n])$ tel que

$$P = \sum_{u\in E} x^u,$$

où $u \in E$ est un abus d'écriture pour $u = (u_1, \ldots, u_n) \in \mathbb{F}_2^n$ et $u_i = 1$ si et seulement si $i \in E$.

On supposera que chacune de ces indéterminées apparaît dans au moins un monôme. La différence entre variable et indéterminée vient de l'aspect formel d'une indéterminée, elle n'est pas associée à une valuation. D'autre part, une variable peut ne pas apparaître explicitement dans la FAN d'une fonction, ainsi $f = x_1x_2 \oplus x_1$ n'aura pas la même table de vérité s'il s'agit d'une fonction à 2 variables ou à 3 variables, mais elle est associée au même polynôme $x_1x_2 + x_1$.

Définition 3.2.1. *Soit $f \in \mathcal{BF}_n$, on considère sa FAN $f = \bigoplus_{u\in\mathbb{F}_2^n} \alpha_u x^u$. On définit la fonction π_n qui à une fonction f associe le polynôme $P \in \mathcal{P}(n)$ tel que*

$$P = \sum_{u\in\mathbb{F}_2^n \ \alpha_u=1} x^u.$$

Remarque 3.2.3. *On vérifie facilement que π_n est bijection de $\mathcal{BF}_n$ vers $\mathcal{P}(n)$ et π_n^{-1} permet de passer de $P \in \mathcal{P}(n)$ vers $f \in \mathcal{BF}_n$.*

Pour $P \in \mathcal{P}$, $\pi_n^{-1}(P)$ est défini pour tout $n \geq d(p)$.

Définition 3.2.2. *Soit $P \in \mathcal{P}(n)$, on définit récursivement la fonction*

$$\mu_n : \mathcal{P}(n) \to \mathcal{P}(n)$$

1. $\mu_1(1) = 1 \oplus x_1$, $\mu_1(0) = 0$, $\mu_1(x_1) = x_1$, $\mu_1(1 \oplus x_1) = 1$.
2. *Pour $n > 1$,*
$$\mu_n(P) = (1 \oplus x_n)\mu_{n-1}(P^0) \oplus x_n\mu_{n-1}(P^1),$$
où $P = P^0 + x_nP^1$, c'est-à-dire P^0 est la somme des monômes de P qui ne contiennent pas l'indéterminée x_n et x_nP^1 est la somme des monômes de P qui contiennent l'indéterminée x_n. Notons que P^0 et P_1 correspondent à la décomposition de Reed-Müller sur les polynômes.

Proposition 3.2.2. *Soit $n \in \mathbb{N}^*$, f et $g \in \mathcal{BF}_n$.*

$$g = \mu(f) \iff \pi_n(g) = \mu_n(\pi_n(f)).$$

Autrement dit μ_n est la transformée de Möbius sur les polynômes de $\mathcal{P}(n)$ pour les fonctions booléennes à n variables. Nous l'appellerons transformée de Möbius d'ordre n.

On obtient le diagramme commutatif suivant

$$\begin{array}{ccccc} & f & \xrightarrow{\mu} & g & \\ \pi_n & \downarrow & & \downarrow & \pi_n \\ & P & \xrightarrow{\mu_n} & P' & \end{array}$$

Preuve. On montre que l'on a pour tout polynôme $P \in \mathcal{P}(n-1)$,

$$\begin{array}{ll} \mu(\pi_n^{-1}(P)) = (1 \oplus x_n)\mu(\pi_{n-1}^{-1}(P)) & \mu_n(P) = (1 \oplus x_n)\mu_{n-1}(P) \\ \mu(\pi_n^{-1}(x_nP)) = x_n\mu(\pi_{n-1}^{-1}(P)) & \mu_n(x_nP) = x_n\mu_{n-1}(P) \end{array}$$

□

Proposition 3.2.3. *Les applications μ et μ_n sont linéaires.*
Soient f_1 et $f_2 \in \mathcal{BF}_n$ et P_1 et $P_2 \in \mathcal{P}(n)$,

$$\begin{array}{lcl} \mu(f_1 \oplus f_2) & = & \mu(f_1) \oplus \mu(f_2). \\ \mu_n(P_1 + P_2) & = & \mu_n(P_1) + \mu_n(P_2). \end{array}$$

On montre que μ_n peut être calculé avec l'opérateur linéaire $\otimes$ suivant.

Définition 3.2.3. *On définit l'opérateur $\otimes$ –que nous nommerons produit spécial– sur les monômes de la manière suivante :*
Soient u et $v \in \mathbb{F}_2^n$.

$$x^u \otimes x^v = \begin{cases} x^{u \vee v} \text{ si } u \wedge v = (0, \ldots, 0) \\ 0 \text{ sinon} \end{cases}$$

On étend alors cette définition à tout polynôme. Soient P_1 et $P_2 \in \mathcal{P}(n)$,

$$P_1 = \sum_{u \in E_1} x^u, P_2 = \sum_{u \in E_2} x^u.$$

$$P_1 \otimes P_2 = P_3 = \sum_{w \in E_3} x^u,$$

où $w \in E_3$ si et seulement si le nombre de couples $(u, v) \in E_1 \times E_2$ tel que $u \vee v = w$ et $u \wedge v = (0, \ldots, 0)$ est pair.

Proposition 3.2.4. *Soit $P \in \mathcal{P}(n)$.*

$$\mu_n(P) = P \otimes \prod_{i=1}^{n} (1 + x_i).$$

Remarque 3.2.4. *Comme $\prod_{i=1}^{n}(1 + x_i) = \sum_{u \in \mathbb{F}_2^n} x^u$, la transformée de Möbius d'ordre n correspond au produit spécial d'un polynôme de $\mathcal{P}(n)$ par le polynôme contenant tous les monômes sur les indéterminées $x_1, \ldots, x_n$.*

Exemple Prenons $P = x_1x_2 + x_2$.

$$\begin{array}{lcl} P & = & x_1x_2 + x_1 \\ \mu_2(P) & = & x_1 \\ A(\pi_2^{-1}(P)) & = & 0101 \\ A(\pi_2^{-1}(\mu_2(P))) & = & 0100 \\ \mu_3(P) & = & (1 + x_3)\mu_2(P) = x_1x_3 + x_1 \\ A(\pi_3^{-1}(P)) & = & 01010000 \\ A(\pi_3^{-1}(\mu_2(P))) & = & 01000100 \\ \mu_4(P) & = & (1 + x_4)\mu_3(P) = x_1x_2x_3 + x_1x_3 + x_1x_4 + x_1 \\ A(\pi_2^{-1}(P)) & = & 0101000000000000 \\ A(\pi_2^{-1}(\mu_2(P))) & = & 0100010001000100 \end{array}$$

On montre facilement que lorsque l'on augmente le nombre de variables par rapport aux nombre d'indéterminées de P, on effectue des rembourrages avec des 0 sur $A(\pi_n^{-1}(P))$ et des recopies (duplications) sur $A(\pi_n^{-1}(\mu_2(P)))$.

3.2.2 Complexité des algorithmes calculant la transformée de Möbius

Algorithme papillon
Le meilleur algorithme calculant la transformée de Möbius est appelé algorithme papillon (butterfly algorithm). Si la fonction booléenne est donnée en entrée sous forme

d'une table de vérité $T = t_1 \dots t_{2^n}$, il est de complexité quasi-linéaire, plus précisément avec $n\ 2^{n-1}$ opérations $\oplus$ (voir par exemple l'introduction de la monographie de C. Carlet [61]).

L'algorithme initial est récursif, mais il possède une version itérative suivante

Algorithme 4 Algorithme papillon itératif

Entrée $T = t_1 \dots t_{2^n}$ table de vérité d'une fonction booléenne f
Sortie $T' = t'_1 \dots t'_{2^n}$ table de vérité de la fonction booléenne $\mu(f)$

pour $i = 1$ à n **faire**
  pour $k = 0$ à $2^{n-i} - 1$ **faire**
    pour $l = 0$ à 2^{i-1} **faire**
      $T[k * 2^i + 2^{i-1} + l] = T[k * 2^i + 2^{i-1} + l] \oplus T[k * 2^i + l]$
retourner T

Remarque 3.2.5. *Du fait de la dualité entre monômes et mintermes dans la transformée de Möbius, nous pouvons dans l'algorithme papillon remplacer en entrée* $T = T(f)$ *par* $A = A(f)$.

Algorithme 5 Calcul de la transformée de Möbius avec $\otimes$

Entrée $P \in \mathcal{P}(n)$ donné sous la forme d'une liste de monômes
Sortie $P' \in \mathcal{P}(n)$ donné sous la forme d'une liste de monômes tel que $\mu_n(P) = P'$

$P_0 \leftarrow P$
pour $i = 1$ à n **faire**
  $P_i \leftarrow P_{i-1} \otimes (1 + x_i)$
retourner P_n

L'algorithme fonctionne quelque soit l'ordre des monômes donné en entrée. Comme l'opérateur est linéaire, l'opération $P_i \leftarrow P_{i-1} \otimes (1 + x_i)$ se décompose en $P_i \leftarrow P_{i-1} + x_i \otimes P_{i-1}$. On recopie donc les monômes obtenus à l'étape précédente et on ajoute ou retire selon qu'ils soient présents dans P_{i-1} les monômes calculés par $x_i \otimes P_{i-1}$. Pour cela il faut vérifier si un monôme apparaît ou non, ce qui ne peut pas se faire en coût constant s'il n'y a pas d'ordre dans les monômes, on obtient donc un algorithme de très mauvaise complexité. Pour remédier à ce problème, on peut coder les P_i par un mot $A = a_1 \dots a_{2^n}$ où $a_i = 1$ si x^u est un monôme de P_i pour $k = \sum_{i=1}^{n} u_i\ 2^{i-1}$ et $a_i = 0$ sinon. On retrouve alors exactement l'algorithme papillon sur les monômes (avec $A(f)$ en entrée).

Même si l'algorithme papillon semble optimal dans le cas général, nous pouvons obtenir un algorithme meilleur pour certaines fonctions, en particulier, lorsque la FAN possède peu de monômes.

Notons $C(P)$ le coût du calcul de $\mu_n(P)$ en travaillant monôme par monôme. Supposons que la sortie soit un mot $A = a_1 \dots a_{2^n}$. Soit $u \in \mathbb{F}_2^n$, $l = w_H(u)$. Comme $\mu_n(x^u) = \sum_{v \succeq u} x^v$, $C(x^u) = 2^{n-l} \leq 2^{n-1}$ pour $u \neq 1_n$. Soit m le nombre de monômes

de P, si $m < n$ alors l'algorithme sera toujours plus efficace que l'algorithme papillon. D'autre part si les monômes sont de grands degré, nous aurons également une faible complexité. Avec $2^{\frac{n}{2}}$ monômes de taille $2^{\frac{n}{2}}$, on obtient une complexité 2^n. Notre approche avec les polynômes doit permettre de retrouver avec une vision simplifiée des résultats de calcul optimisé de la transformée de Möbius dans certains cas [68].

Un problème plus simple consiste à calculer le poids de Hamming d'une fonction. il suffit de compter le nombre de monômes de la transformée, mais dans certains cas il n'est pas nécessaire de calculer cette transformée, en particulier lorsque certains monômes sont isolés (ne contiennent que des variables qui n'apparaissent pas dans les autres monômes) [57].

3.2.3 Décomposition de Shannon et de Reed-Müller et transformée de Möbius

3.2.3.1 Transformée de Möbius et décomposition de Reed-Müller

Soient $f \in \mathcal{BF}_n$ et $f_R^0, f_R^1 \in \mathcal{BF}_{n-1}$ tels que $f = f_R^0 \oplus x_n f_R^1$. La transformée de Möbius se calcule de manière récursive avec la relation suivante :

$$\mu(f) = (1 \oplus x_n)\mu(f_R^0) \oplus x_n \mu(f_R^1).$$

3.2.3.2 Transformée de Möbius et décomposition de Shannon

Soient $f \in \mathcal{BF}_n$ et $f_S^0, f_S^1 \in \mathcal{BF}_{n-1}$ tels que $f = (1 \oplus x_n) f_S^0 \oplus x_n f_S^1$. La transformée de Möbius se calcule de manière récursive avec la relation suivante :

$$\mu(f) = \mu(f_S^0) \oplus x_n \mu(f_S^1).$$

Dans les deux cas, la preuve s'effectue facilement sur les polynômes en utilisant la proposition 3.2.2 et la définition 3.2.2, la définition récursive de μ_n.

3.3 Fonctions coïncidentes

La notion de fonction coïncidente a été introduite par Pieprzyk, Wang et Zhang [75]. Nous reprenons ici des résultats qu'ils ont déjà obtenus , mais en les présentant avec des polynômes.

3.3.1 Définitions

Soit $f \in \mathcal{BF}_n$, f est une fonction coïncidente lorsque

$$\mu(f) = f, \text{ ou encore } f = \bigoplus_{u \in \mathbb{F}_2^n} f(u) x^u.$$

Il existe plusieurs autres définitions équivalentes.

Proposition 3.3.1. *Les propriétés suivantes sont équivalentes à la définition précédente.*

– $T(f) = A(f)$.
– *Soit $u \in \mathbb{F}_2^n$, M_u est un minterme de la table de vérité de f si et seulement si x^u est un monôme de la FAN de f.*

Définition 3.3.1. *Soit $P \in \mathcal{P}(n)$. Nous dirons que P n-coïncident lorsque $\mu_n(P) = P$.*

Nous avons bien sûr l'équivalence entre f est coïncidente et $\pi_n(f)$ est n-coïncident.

La proposition 3.2.1 permet d'avoir la définition suivante d'une fonction coïncidente sur le treillis $\mathcal{L}_n$.

Proposition 3.3.2. *Soit $f \in \mathcal{BF}_n$, f est coïncidente si et seulement si*

$$\bigoplus_{v \prec u} f(v) = 0, \text{ pour tout } u \in \mathbb{F}_2^n.$$

Autrement dit, le nombre de $v \in \mathbb{F}_2^n$ tels que $v \prec u$ est pair, pour tout $u \in \mathbb{F}_2$.

Soit $u \in \mathbb{F}_2^n$, $E = \{i \in [n] | u_i = 1\}$, par abus d'écriture, nous noterons M_u le polynôme de $\mathcal{P}(n)$ valant $\prod_{i \in E} x_i \prod_{i \in [n] \subset E}(1 + x_i)$ (on identifie le minterme avec son polynôme associé).

Définition 3.3.2. *Notons φ_n la fonction de $\mathcal{P}(n) \to \mathcal{P}(n)$ qui vérifie*

$$\varphi_n(P) = P + \mu_n(P).$$

Autrement dit,

$$\varphi_n(P) = \sum_{x^u \in P \text{ et } x^u \notin \mu_n(P)} x^u + \sum_{x^u \notin P \text{ et } x^u \in \mu_n(P)} x^u.$$

Proposition 3.3.3. *Soit $u \in \mathbb{F}_2^n$ and $C_u = x^u + M_u$. C_u est n-coïncident et vérifie*

$$C_u = \sum_{u \prec v} x^v = \sum_{u \prec v} M_v.$$

Preuve. C'est une implication directe de la proposition 3.2 ($C_u = \varphi_n(x^u)$). □

Proposition 3.3.4. *Pour tout $P \in \mathcal{P}(n)$, $\varphi_n(P)$ est n-coïncident.*

Preuve. Découle directement des propositions 3.2.3 et 3.3.3. □

Proposition 3.3.5. *Soit $\mathcal{C}_n$ l'ensemble des polynômes de $\mathcal{P}(n)$ n-coïncident.*

Il existe un isomorphisme ψ_{n-1} de $\mathcal{C}_n$ vers $\mathcal{P}(n-1)$ tel que, pour tout $P \in \mathcal{P}(n-1)$, $\varphi_n(P) = C$ si et seulement si $\psi_{n-1}(C) = P$.

Preuve. Soit $C \in \mathcal{P}(n)$, il existe P_R^0 et $P_R^1 \in \mathcal{P}(n-1)$ tels que $C = P_R^0 + x_n P_R^1$. Il vient $\mu_n(C) = \mu_{n-1}(P_R^0) + x_n(\mu_{n-1}(P_R^0) + \mu_{n-1}(P_R^1))$, comme $\mu_n(C) = C$, $\mu_{n-1}(P_R^0) = P_R^0$ (et donc $P_R^0 \in C_{n-1}$) et $P_R^1 = P_R^0 + \mu_{n-1}(P_R^1)$ et $P_R^0 = \varphi_{n-1}(P_R^1)$.

Soit $P = \mu_{n-1}(P_R^1)$, nous avons

$$P \oplus \mu_n(P) = \varphi_{n-1}(P_R^1) + x_n P_R^1 = P_R^0 + x_n P_R^1.$$

Donc $C = \varphi_n(P)$, nous avons de plus,

$$C = \varphi_{n-1}(P) + x_n \mu_{n-1}(P).$$

D'où,

$$\varphi_n(P) = \varphi_{n-1}(P) + x_n(P + \varphi_{n-1}(P))$$

D'autre part, la fonction ψ_n est définie par

$$\psi_n(C) = P_R^0 + P_R^1.$$

□

Algorithme 6 Génération aléatoire uniforme d'une fonction coïncidente

Entrée n, le nombre de variables
Sortie $C \in \mathcal{C}_n$
$P \leftarrow$ uniforme($\mathcal{BF}_{n-1}$)
$C \leftarrow \varphi_n(P)$
retourner C

Où uniforme(E) retourne avec la distribution uniforme (l'équiprobabilité) un élément de l'ensemble E.

Définition 3.3.3 (Relation d'équivalence $\equiv_C$ sur $\mathcal{BF}_n$)**.**
Soient P et $Q \in \mathcal{P}(n)$. Nous écrirons $P \equiv_C Q$ lorsque $\varphi_n(P) = \varphi_n(Q)$. Une classe d'équivalence sera notée $C_n(C)$, où $C_n(C) = \{P \in \mathcal{P}(n) \mid \varphi_n(P) = C\}$.

Proposition 3.3.6. *Let $C \in \mathcal{C}_n$.*

$$C_n(C) = \{\psi_{n-1}(C) + C' \mid C' \in \mathcal{C}_n\}.$$

Preuve. Soient C_1et $C_2 \in \mathcal{C}_1$ et $P_1 \in \mathcal{C}_n(C_1)$. $\varphi_n(P_1 + C_2) = \varphi_n(P_1) + \varphi_n(C_2) = C_1$, donc toutes les classes $C_n(C)$ ont la même cardinalité $2^{2^n}/2^{2^{n-1}} = 2^{2^{n-1}}$.

D'autre part, soit $P = \psi_{n-1}(C)$ et $C \in \mathcal{C}_n$.

$$\varphi_n(P + C) = \varphi_n(P) + \varphi_n(C) = C + 0 = C.$$

En conclusion, la relation d'équivalence $\equiv_C$ réalise donc une partition sur $\mathcal{BF}_n$ de $2^{2^{n-1}}$ classes d'équivalence contenant chacune $2^{2^{n-1}}$ fonctions booléennes, avec en particulier $C_n(0_n) = C_n$. □

Algorithme 7 Énumération d'une classe coïncidente

Entrée n, le nombre de variables, $C \in \mathcal{C}_n$
  $P = \psi(n, C)$
  pour tout $C' \in \mathcal{C}_n$ **faire**
    Énumérer $P + C'$

3.3.2 Énumération de fonctions coïncidentes

On considère pour ces algorithmes d'énumération que les polynômes sont représentés par des mots binaires contenant $a_1 \dots a_{2^n}$, où $a_k = 1$, avec $k = k(u) = \sum_{i=1}^{n} u_i \, 2^{i-1}$, lorsque x^u apparaît dans P.

On suppose implémenter la fonction $\psi(n, C)$ qui renvoie $P = \psi_n(C)$. On suppose aussi que l'on a l'ensemble $\mathcal{C}_n$ ou une énumération de cet ensemble. L'algorithme 7 énumère les fonctions d'une classe $C_n(C)$.

Remarque 3.3.1. *$P = \psi_{n-1}(C) = P_R^0 + P_R^1$, pour $C = P_R^0 + x_n P_R^1$, d'où un coût de 2^{n-1} au début de l'énumération.*

Ensuite, le délai d'énumération de l'algorithme 7 est de 2^{n-1}. car $P + C'$ a un coût de 2^{n-1} (comme $C' \in \mathcal{P}(n-1)$, le calcul ne se fait que sur les monômes ne contenant pas x_n).

Algorithme 8 Énumération des fonctions coïncidentes

Entrée n, le nombre de variables, $\mathcal{C}_{n-1}$
  pour tout $C \in \mathcal{C}_{n-1}$ **faire**
    pour tout $P \in C_{n-1}(C)$ **faire**
      Énumérer $C + x_n P$

L'algorithme 8 permet d'énumérer les classes de $\mathcal{C}_n$ à partir d'une énumération de $\mathcal{C}_{n-1}$.

Remarque 3.3.2. *Le délai d'énumération de l'algorithme 8 est de 2^{n-2} car $C + x_n P$ ne nécessite aucun calcul et le délai provient uniquement de l'énumération de P qui est de 2^{n-2} (voir le délai d'énumération de l'algorithme 7).*

L'intérêt des fonctions coïncidentes pour l'énumération et la génération aléatoire était que pour une fonction coïncidente la présence d'un monôme était équivalente à la présence du minterme correspondant. Les algorithmes ci-dessus n'exploite pas cette propriété. Pour effectuer l'énumération selon le nombre de monômes, il suffit de considérer les classes $\mathcal{C}_{n,k}$ des fonctions coïncidentes à n variables et k monômes et effectuer la même sous-partition sur $C_n(C)$, $C_{n,k}(C)$.

3.3.3 Fonctions coïncidentes symétriques

Dans [60], Canteaut et Videau proposent une étude approfondie des fonctions symétriques en considérant des critères cryptographiques tels que le degré, la corrélation-immunité et la non-linéarité.

Nous allons présenter dans cette partie un algorithme pour énumérer le $2^{\lfloor \frac{n}{2} \rfloor + 1}$ fonctions coïncidentes symétriques.

Définition 3.3.4. *Soit $k \leq n$, Σ_k^n désignera la fonction booléenne f telle que $\pi_n^{-1}(f) = \sum_{u \in \mathbb{F}_2^n | w_H(u)=k} x^u$. Une fonction symétrique est définie par*

$$f = \sum_{k=0}^{n} \lambda_k \Sigma_k^n,$$

où $(\lambda_0, \ldots, \lambda_n) \in \mathbb{F}_2^n$ satisfait

$$\lambda_i = \begin{cases} 1, & \text{si } \pi_n^{-1}(f) \text{ contient tous les monômes de degré } i \\ 0, & \text{sinon.} \end{cases}$$

Nous noterons $\lambda(f) = (\lambda_0, \ldots, \lambda_n)$.

Comme à chaque $\lambda \in \mathbb{F}_2^{n+1}$ correspond une unique fonction booléenne telle que $\lambda(f) = \lambda$, nous avons 2^{n+1} fonctions symétriques.

Une fonction symétrique est invariante par permutation des variables, soit $a = (a_1, \ldots a_n) \in \mathbb{F}_2^n$,

$$f(a_1, \ldots, a_n) = f(a_{\sigma(1)}, \ldots, a_{\sigma(n)}),$$

pour toute permutation σ of $\{1, \ldots n\}$. On peut donc coder f avec le vecteur $v(f) = (v_0, \ldots, v_n)$, où $v_i = 1$ signifie que $f(a) = 1$ pour tout $a \in \mathbb{F}_2^n$ avec $w_H(a) = i$.

Nous dirons qu'un polynôme P de $\mathcal{P}(n)$ est n-symétrique lorsque f est symétrique pour $\pi_n^{-1}(P) = f$, où $n = d(P)$. Nous écrirons aussi $\lambda(P) = (\lambda_0, \ldots, \lambda_n)$, où $\lambda_i = 1$ si et seulement si P contient tous les monômes de degré i.

Une fonction symétrique f est coïncidente si et seulement si $\lambda(f) = v(f)$. De manière similaire, P est n-coïncident si et seulement si $\lambda(P) = \lambda(\mu_n(P))$.

Proposition 3.3.7. *Soit $P = P_R^0 + x_n P_R^1$. P est symétrique si et seulement si P_R^0 et P_R^1 sont symétriques. De plus, soit*

$$\begin{array}{rcl} \lambda(P) & = & (\lambda_0, \ldots, \lambda_n); \\ \lambda(P_R^0) & = & (\lambda_0^0, \ldots, \lambda_{n-1}^0); \\ \lambda(P_R^1) & = & (\lambda_0^1, \ldots, \lambda_{n-1}^1); \end{array}$$

nous avons le système suivant

$$\begin{cases} \lambda_i & = & \lambda_i^0 = \lambda_{i-1}^1, \text{ for any } i \in \{1, \ldots, n-1\}, \\ \lambda_n & = & \lambda_{n-1}^1. \end{cases}$$

Les coefficients de Lucas interviennent de manière très naturelle pour l'étude des fonctions symétriques [60].

Définition 3.3.5 (Coefficients de Lucas)**.**
Soient k et $j \in \mathbb{N}$, on pose $p(k, j) = \binom{k}{j}$ mod 2.

Notation 3.3.1. *Soient $k = \sum_{i\in\mathbb{N}} k_i\ 2^i$ et $j = \sum_{i\in\mathbb{N}} j_i\ 2^i$ la représentation 2-adic de k et j. Nous écrirons $j \preceq k$ lorsque $j_i = 1$ implique $k_i = 1$, pour tout $i \in \mathbb{N}$.*

Théorème 3.3.1 (de Lucas)**.**
Soient k et $j \in \mathbb{N}$, $p(k, j) = 1$ si et seulement si $j \preceq k$.

Proposition 3.3.8. *Nous avons la relation entre $\lambda(f)$ et $v(f)$ suivante (voir par exemple [60]) :*

$$v_j = \sum_{k=0}^{j} \lambda_k\ p(k, j).$$

Proposition 3.3.9. *Soit $P \in \mathcal{P}(n)$ un polynôme symétrique, $\lambda(P) = (\lambda_0, \ldots, \lambda_n)$. P est coïncident si et seulement si*

$$\bigoplus_{k<j} \lambda_k\ p(j, k) = 0,\ \textit{for any } j \in \{0, \ldots, n\}.$$

Preuve. P est coïncident si et seulement si $\lambda(P) = \lambda(\mu_n(P))$.
Posons $\lambda(\mu_n(\Sigma_n^k)) = (v_0^k, \ldots, v_n^k)$, nous avons le système

$$\left\{ \begin{array}{lcl} v_j^k & = & 0,\ \text{for } j < k \\ v_k^k & = & 1 \\ v_j^k & = & p(j, k),\ \text{for } k < j \leq n. \end{array} \right. \tag{3.6}$$

□

On conclut avec $\lambda(\mu_n(P)) = (\sum_{i=0}^n v_i^0, \ldots, \sum_{i=0}^n v_i^k)$.

Proposition 3.3.10 (Enumération des fonctions coïncidentes symétriques)**.** *Notons CS_n l'ensemble des fonctions coïncidentes symétriques. Les éléments de CS_n seront codés par $(\lambda_0, \ldots, \lambda_n)$.*

CS_n admet le schéma d'induction suivant

1. *Pour $n = 1$, $CS_1 = \{(0, 0), (0, 1)\}$.*
2. *Si n est pair, alors pour tout $(\lambda_0, \ldots, \lambda_{n-1}) \in CS_{n-1}$,*

$$(\lambda_0, \ldots, \lambda_{n-1}, 0)\ \textit{et}\ (\lambda_0, \ldots, \lambda_{n-1}, 1) \in CS_n.$$

Si n est impair, pour tout $(\lambda_0, \ldots, \lambda_{n-1}) \in CS_{n-1}$, si

$$\lambda_0\ p(n, n) \oplus \lambda_2\ p(n, n-2) \ldots \oplus \lambda_{n-2}\ p(n, 2) = \lambda_{n-1}$$

alors $(\lambda_0, \ldots, \lambda_{n-1}, 0)$ et $(\lambda_0, \ldots, \lambda_{n-1}, 1) \in CS_n$.

Preuve. Soit P un polynôme n-coïncident symétrique ($\pi_n^{-1}(P) \in CS_n$). Posons $P = P_R^0 + P_R^1$. Nous avons vu que P_R^0 et P_R^1 étaient symétriques et que de plus P_R^0 était n-coïncident. Nous en déduisons que P_R^0 est n-coïncident symétrique. En reprenant le système de (3.6), il reste à montrer

$$\bigoplus_{k<n} \lambda_k\ p(n, k) = 0.$$

Si n est pair alors $p(n,1) = 0$ et $p(n,k) = 0$ pour k impair et $p(n,k) = p(n-2,k-2)$, pour k pair. Il reste à vérifier

$$\bigoplus_{k<n-2} \lambda_k \, p(n-2,k) = 0,$$

qui est déjà vérifié car P_R^0 est n-coïncident symétrique. Nous pouvons donc choisir $\lambda_n = 0$ or 1. Si n est impair, alors $p(n,1) = 1$ et nous avons

$$\lambda_0 \, p(n,n) \oplus \lambda_2 \, p(n,n-2) \ldots \oplus \lambda_{n-2} \, p(n,2) = \lambda_{n-1}.$$

En considérant le cas précédent appliqué à $n-1$, nous savons que la moitié des polynômes $n-1$-coïncidents symétriques vérifient $\lambda_{n-1} = 0$. □

Corrolaire 3.3.1.

$$|CS_n| = 2^{\lfloor \frac{n}{2} \rfloor + 1}.$$

3.3.4 Propriétés des fonctions coïncidentes aléatoires

Les fonctions coïncidentes ne sont pas encore utilisées en cryptographie pour des applications pratiques. Je pense qu'elle peuvent permettent des constructions de fonctions efficaces grâce au rôle similaire que jouent les monômes et les mintermes. Cependant les fonctions booléennes utilisées en pratique doivent satisfaire des compromis entre les différents critères qui interviennent dans la sécurité de ces fonctions. Nous devons donc pouvoir construire des fonctions coïncidentes avec autant de variétés dans les critères que pour l'ensemble des fonctions booléennes.

Nous avons généré respectivement 1000 fonctions booléennes aléatoires et 1000 fonctions coïncidentes à 20 variables aléatoires avec le générateur de l'algorithme 6 et comparés les distributions pour ces critères.

3.3.4.1 Hamming weight distribution

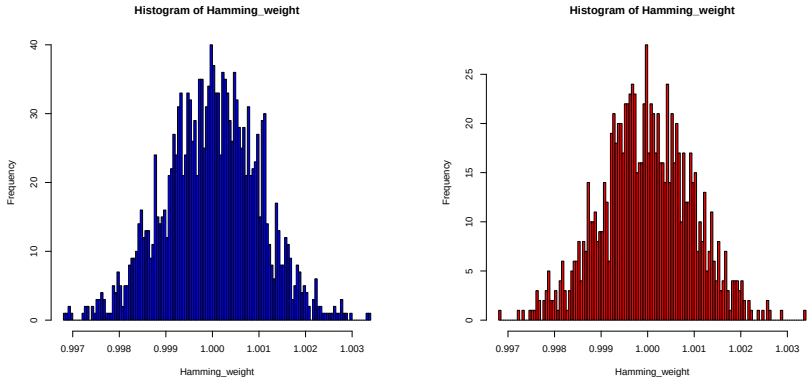
La figure 3.1 donne la distribution des poids des 1000 fonctions aléatoires de $\mathcal{C}_n$ et$\mathcal{BF}_n$. Le poids de Hamming w est normalisé par sa moyenne 2^{n-1} et l'abscisse est donc $w' = w/2^{n-1}$. Les deux distribution sont vraiment similaires.

3.3.4.2 Distribution des degrés des monômes

Soient f et g deux fonctions générées aléatoirement sur $\mathcal{C}_n$ et $\mathcal{BF}_n$. On compte le nombre de monômes de degré d pour $d \in \{0,\ldots,n\}$. Celui-ci est d'espérance $\binom{n}{d}/2$ pour g. Nous avons utilisée plusieurs fois le test statistique de Kolmogorov-Smirnov pour montrer que les deux distributions étaient très proches.

3.3.4.3 Non-linéarité

Le critère de non-linéarité d'une fonction booléenne est la distance de Hamming de celle-ci avec les fonctions affines. Il a été montré [61] qu'une fonction booléenne doit



Random functions over $\mathcal{C}_{20}$ Random functions over $\mathcal{BF}_{20}$

FIGURE 3.1 – Distribution du poids de Hamming pour les fonctions booléennes et les coïncidentes aléatoires

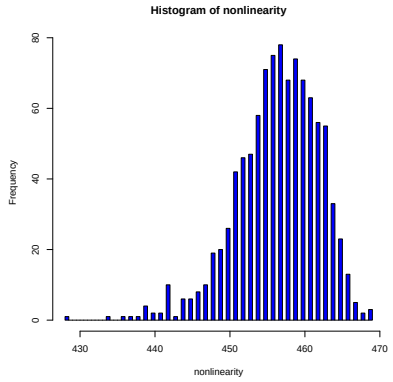
avoir une haute non-linéarité pour être utilisée en cryptographie, en particulier pour éviter les attaques par corrélation([59, 64]). Les fonctions qui atteignent la meilleure non-linéarité sont appelées fonctions courbes, elles n'existent que pour n pair et ont une non-linéarité $2^{n-1} - 2^{n/2-1}$. Dans le cas où n est impair, la meilleure non-linéarité est strictement meilleure pour $n > 7$ que celle des fonctions quadratiques qui est de $2^{n-1} - 2^{\frac{n-1}{2}}$ (voir [61] pour une bonne introduction à la non-linéarité et aux fonctions courbes).

Notons $\mathcal{A}$ l'ensemble des polynômes affines (éléments de $\mathcal{P}$ de degré algébrique inférieur ou égal à 1) et $\mathcal{A}_n$ les polynômes affines de $\mathcal{P}(n)$. Nous avons vu (proposition 3.3.5) que les polynômes n-coïncidents et les polynômes de $\mathcal{P}(n-1)$ étaient en bijection, à chaque un polynôme n-coïncident C correspond un unique $P' \in \mathcal{P}(n-1)$ tel que $C = (1 + x_n)\varphi_{n-1}(P') + x_n P'$. Soit $A \in \mathcal{A}_n$. Soit $A \in \mathcal{A}_n$, $A = (1 + x_n)A' + x_n A'$ ou $(1 + x_n)A' + x_n(A' + 1)$, pour $A' \in \mathcal{A}_{n-1}$. Donc $d_H(\pi_n^{-1}(P), \pi_n^{-1}(A)) = d_H(\pi_{n-1}^{-1}(\varphi_{n-1}(P')), \pi_{n-1}^{-1}(A')) + d_H(\pi_{n-1}^{-1}(P'), \pi_{n-1}^{-1}(A'))$ ou $d_H(\pi_n^{-1}(P), \pi_n^{-1}(A)) = d_H(\pi_{n-1}^{-1}(\varphi_{n-1}(P')), \pi_{n-1}^{-1}(A')) + d_H(\pi_{n-1}^{-1}(P' + 1), \pi_{n-1}^{-1}(A'))$. Pour avoir les mêmes non-linéarités que pour une fonction booléenne aléatoire, il faut donc qu'il n'y est pas de corrélation entre les distances $d_H(\pi_{n-1}^{-1}(\varphi_{n-1}(P')), \pi_{n-1}^{-1}(A'))$ et

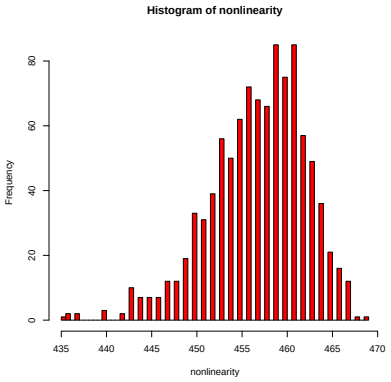
$d_H(\pi_{n-1}^{-1}(P'), \pi_{n-1}^{-1}(A'))$. On observe expérimentalement que c'est le cas. La figure 3.2, contient la fréquence des non-linéarité pour 1000 fonctions de C_n et $\mathcal{BF}_n$ pour $n = 10$ et $n = 11$. Notons que les fonctions courbes de $\mathcal{BF}_{10}$ ont une non-linéarité de 496. Que se soient sur C_n ou $\mathcal{BF}_n$, aucune fonctions courbe n'est construite, la meilleur non-linéarité approche 470.

3.3.5 Perspectives

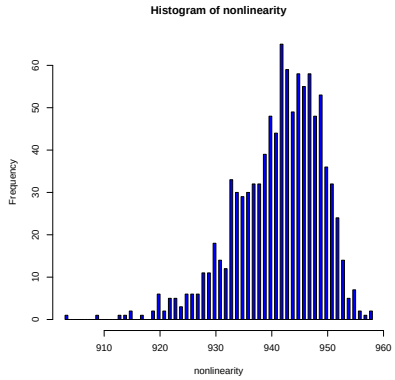
Nous avons vu l'intérêt de manipuler des polynômes à la place des fonctions, notamment pour la décomposition de Reed-Müller. Les fonctions coïncidentes aléatoires sont très proches de fonctions booléennes aléatoires. La génération aléatoire des fonc-



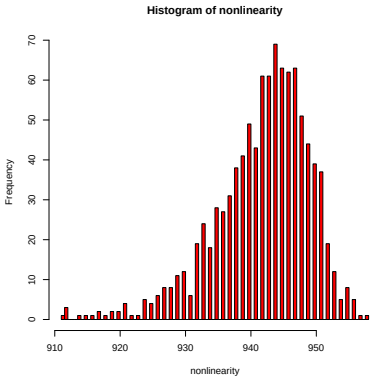
Fonctions aléatoires de $\mathcal{C}_{10}$



Fonctions aléatoires de $\mathcal{BF}_{10}$



Fonctions aléatoires de $\mathcal{C}_{11}$



Fonctions aléatoires de $\mathcal{BF}_{11}$

FIGURE 3.2 – Distribution de la non-linéarité pour 10 et 11 variables.

tions coïncidentes nécessitent malheureusement un calcul de la transformée de Möbius, il serait donc intéressant de pouvoir les générer directement. D'autre part, le problème de l'énumération et de la génération aléatoire pour un poids de Hamming et un degré algébrique fixés semble abordable pour cette classe de fonctions.

Références de la partie 2

[55] Eric Bach. Improved Asymptotic Formulas for Counting Correlation Immune Boolean Functions. *SIAM J. Discrete Math.*, 23(3) :1525–1538, 2009.

[56] Randal E Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24 :293–318, 1992.

[57] Çagdas Çalik and Ali Doganaksoy. Computing the Weight of a Boolean Function from Its Algebraic Normal Form. In *Sequences and Their Applications - SETA 2012 - 7th International Conference, Waterloo, ON, Canada, June 4-8, 2012. Proceedings*, pages 89–100, 2012.

[58] E. Rodney Canfield, Zhicheng Gao, Catherine Greenhill, Brendan D. McKay, and Robert W. Robinson. Asymptotic enumeration of correlation-immune boolean functions. *Cryptography and Communications*, 2(1) :111–126, 2010.

[59] Anne Canteaut and Michaël Trabbia. Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 573–588, 2000.

[60] Anne Canteaut and Marion Videau. Symmetric Boolean functions. *IEEE Trans. Information Theory*, 51(8) :2791–2811, 2005.

[61] Claude Carlet. Boolean Functions for Cryptography and Error Correcting Codes. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics*, pages 257–397. Cambridge University Press, 2010.

[62] Nicolás Carrasco, Jean-Marie Le Bars, and Alfredo Viola. Enumerative encoding of correlation immune boolean functions. In *2011 IEEE Information Theory Workshop, ITW 2011, Paraty, Brazil, October 16-20, 2011*, pages 643–647, 2011.

[63] Nicolás Carrasco, Jean-Marie Le Bars, and Alfredo Viola. Enumerative encoding of correlation-immune boolean functions. *Theor. Comput. Sci.*, 487 :23–36, 2013.

[64] Vladimir Chepyzhov and Ben Smeets. *On A Fast Correlation Attack on Certain Stream Ciphers*, pages 176–185. Springer Berlin Heidelberg, 1991.

[65] W.K. Clifford. On the types of compound statement involving four classes. *Mem. Lit. Soc. Manchester*, 16 :88–101, 1877.

[66] Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1) :1 – 35, 1994.

[67] Philippe Guillon. *Fonctions courbes et transformation de Mobius*. PhD thesis, Université de Caen, 1999.

[68] K. C. Gupta and P. Sarkar. Computing Partial Walsh Transform From the Algebraic Normal Form of a Boolean Function. *IEEE Transactions on Information Theory*, 55(3) :1354–1359, March 2009.

[69] Marcin Kamiński. Review of Boolean Models and Methods in Mathematics, Computer Science, and Engineering by Yves Crama and Peter L. Hammer. *SIGACT News*, 44(1) :21–24, March 2013.

[70] T. Kasami and N. Tokura. On the Weight Structure of Reed-Muller Codes. *IEEE Trans. Inf. Theor.*, 16(6) :752–759, September 2006.

[71] Donald E. Knuth. *The Art of Computer Programming, Volume 3 : (2Nd Ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[72] Jean-Marie Le Bars and Alfredo Viola. Equivalence classes of Boolean functions for first-order correlation. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 181–185, June 2007.

[73] Jean-Marie Le Bars and Alfredo Viola. Equivalence classes of Boolean functions for first-order correlation. *IEEE Trans. Information Theory*, 56(3) :1247–1261, 2010.

[74] E.M. Palmer, R.C. Read, and R.W. Robinson. Balancing the n-Cube : A Census of Colorings. *Journal of Algebraic Combinatorics*, 1(3) :257–273, 1992.

[75] Josef Pieprzyk, Huaxiong Wang, and Xian-Mo Zhang. Mobius transforms, coincident boolean functions and non-coincidence property of boolean functions. *Int. J. Comput. Math.*, 88 :1398–1416, 2011.

[76] George Pólya. Sur les types des propositions composées. *Journal Symbolic Logic*, 5 :98–103, 1940.

[77] Ronald C Read. The enumeration of locally restricted graphs i. *J. London Maths Soc*, pages 417–436, 1959.

[78] C. E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1) :59–98, Jan 1949.

[79] I. Strazdins. Universal Affine Classification of Boolean Functions. *Acta Applicandae Mathematica*, 46(2) :147–167, 1997.

[80] Jean Vuillemin and Fredéric Béal. *ASIAN 2004*, chapter On the BDD of a Random Boolean Function, pages 483–493. Springer Berlin Heidelberg, 2005.

Cinquième partie

Aléatoire des données pour le tatouage et la biométrie

Chapitre 1

Introduction

Cette partie a une structure moins formelle que les deux précédentes. Elle reprend les travaux de thèses de trois étudiants, dont deux ayant déjà soutenu leur thèse. Leurs travaux s'inscrivent dans deux domaines de la sécurité informatique, le tatouage et la biométrie. Plutôt que de présenter l'intégralité de leurs travaux, j'ai souhaité ici aborder leurs études sous un angle subjectif. Pour conserver une cohérence générale du manuscrit d'HDR, j'ai choisi de mettre en évidence les parties qui sont en lien avec mes thèmes de recherche. A chaque fois, le sujet et le contexte de la thèse sont brièvement exposés. J'espère que c'est suffisant pour suivre les études développées, cependant, je recommande recommande la lecture de leur manuscrit de thèse pour une présentation complète de leur sujet.

La thèse de Cyril Bazin propose une méthode de tatouage de données géographiques rapide, aveugle et robuste à la rotation et à la translation. La méthode consiste à introduire un biais statistique sur des petites parties du document appelés sites.

Zhigang Yao et Benoît Vibert travaillent tous les deux en biométrie sur les empreintes digitales. Yao s'intéresse à la qualité de ces empreintes. Il travaille sur l'évaluation de la qualité des empreintes digitales et sur l'impact de la qualité sur la performance des systèmes biométriques. Vibert considère la biométrie embarquée sur élément sécurisé (technologie MOC pour match on card). L'empreinte de référence est stockée sous forme numérique sur le MOC par un template de minuties (points remarquables). Il travaille sur la sélection de minuties et sur les attaques pour s'identifier indûment auprès d'un élément sécurisé.

1.1 Contributions

Le chapitre 2 est consacré au tatouage de documents géographiques. Ce travail s'inscrit dans le cadre du projet TADORNE de l'ACI Sécurité 2004 –dirigé par David Gross-Amblard– auquel cinq laboratoires ont participé : le Cedric (Cnam-Paris), le Lamsade (université de Paris-Dauphine), le Le2i (université de Bourgogne), le COGIT (IGN) et le GREYC. Le logiciel de tatouage de données contraintes watermill a été développé dans le cadre de cette ACI [98]. Les membres de l'ACI Tadorne ont mené des travaux sur les données géographiques (tatouage de bâtiments [96, 97]), sur les bases

de données (Thèse de Julien Lafaye [95]) et la complexité des schémas de tatouage [94]. Dans [89], d'autres types de données sont considérés comme les bases de données ou les documents xml.

Les différentes rencontres que nous avons eu nous ont permis à Jacques Madelaine et moi, tous deux membres du GREYC participant à ce projet, de bien comprendre la problématique du tatouage sur les documents géographiques, en particulier sur les aspects juridiques de la propriété intellectuelle, sur les traitements appliqués aux données géographiques et sur les usages des utilisateurs. Comme Gross-Amblard, nous souhaitions travailler sur des données géographiques vectorielles sur lesquels Madelaine possède une grande expertise. Il a en effet participé au développement de ThemaMap qui est un outil de cartographie thématique multi plateforme, distribué en tant que logiciel libre et accessible à l'url *https ://themamap.greyc.fr/.*

Plutôt que de définir dès la départ un cadre théorique, nous avons souhaité partir d'un contexte applicatif très précis (le tatouage d'une carte du Calvados) et de ne chercher qu'ensuite à en dégager des principes généraux. Cyril Bazin a débuté une thèse [83] sous notre direction sur ce sujet en 2006.

Le fait de partir d'un cadre applicatif précis ne signifie pas qu'il s'agit de développer un logiciel à but commercial. Ici l'approche a été résolument exploratoire et le sujet s'y prêtait bien car peu d'études existaient déjà. Cyril Bazin a soigné la mise en œuvre des logiciels de marquage et de détection, plus pour avoir une vision complète de la faisabilité des méthodes testées que pour délivrer un produit commercialisable. Nous n'avons pas souhaité non plus déposer un brevet –le travail réalisé s'y prêtait également bien– afin d'avoir plus de liberté sur la manière de diffuser nos résultats, nous préférions une diffusion classique par le biais des publications. De plus, nous n'étions pas sûr que le schéma globale tel qu'il était proposé convienne aux distributeurs de données géographiques. Par exemple, l'IGN, bien que reconnaissant la nécessité de la protection de la propriété intellectuelle (l'IGN a constaté des infractions à son encontre venant d'autres pays), ne voyait pas le tatouage comme une priorité en termes d'investissements financiers.

On s'aperçoit avec le recul que nous avons maintenant que l'apport de notre travail a été double. D'une part, nous avons proposé un schéma de tatouage extrêmement efficace pour les documents géographiques vectoriels [84], mais aussi généralisable à d'autres types de données structurées et contraintes [85] D'autre part, nous comprenons mieux quelle partie du schéma de tatouage est commune à tous ces types de données et quelle partie doit être spécifique à chacun de ces types et requiert de ce fait un expert de ces données.

Le chapitre 3 concerne le domaine de la biométrie. J'ai intégré l'équipe Monétique & Biométrie en janvier 2014. J'ai souhaité me former aux thématiques de ma nouvelle équipe afin de pouvoir participer aux activités de recherche de celle-ci. En particulier, je me suis initié au domaine de la Biométrie. J'ai découvert un domaine en pleine émergence pour lequel la qualité des données collectées est essentielle. De plus, la validation des méthodes et algorithmes étudiés passe par des tests conséquents qui demande une méthodologie rigoureuse (évaluation de la performance, comparaison

entre les métriques mesurant la performance).

J'ai découvert avec intérêt le sujet phare du thème Biométrie, la dynamique de frappe [88]. La méthode consiste à identifier une personne en fonction de son comportement lorsque, par exemple, elle tape son mot de passe, en analysant différentes mesures telles que le temps entre deux touches, la durée pendant laquelle une touche est appuyée. L'analyse se base sur des caractéristiques très différentes des caractéristiques physiques plus classiquement utilisées comme les empreintes digitales, les veines de la main, les traits du visage ou la forme de l'iris. J'ai appris également comment s'effectue l'évaluation des systèmes biométriques.

Je me suis surtout intéressé aux empreintes digitales utilisées comme moyen d'identification et d'authentification d'une personne. Cette modalité est une des plus étudiées en biométrie [91, 99].

Les templates de minuties (ensembles de points singuliers extraits de l'image de l'empreinte) permettent l'identification des empreintes digitales. Certains traitements peuvent s'effectuer uniquement avec un template de minuties, sans connaissance de l'image de l'empreinte digitale : mesure de la qualité d'un template (l'enrôlement a bien été réalisé), sélections des meilleures minuties pour la réduction de template, construction de faux templates pour attaquer un élément sécurisé.

Après avoir appris les bases du domaine, j'ai pu participer à des études en cours. Christophe Rosenberger m'a proposé de co-encadrer avec Christophe Charrier et lui-même deux thèses sur les empreintes digitales. D'une part, Zhigang Yao travaillait sur la mesure de la qualité des empreintes digitales. L'objectif de sa thèse [109] était d'évaluer l'impact du comportement d'un système biométrique lorsque les données acquises sont de mauvaise qualité. D'autre part, Benoît Vibert s'intéresse à la sélection des empreintes digitales pour des applications sur cartes à puce (MOC pour Match On Card) et aux attaques dans le cadre de ces applications en essayant, par exemple, de forger des empreintes pour s'identifier indûment auprès du MOC.

J'ai également participé avec Christophe Rosenberger et Morgan Barbier à la conception d'une amélioration d'une méthode de tatouage d'image à base de biométrie révocable qu'ils avaient tous les deux proposée auparavant.

Chapitre 2

Tatouage de documents structurés

2.1 Introduction

2.1.1 Problématique du tatouage

Le tatouage est un moyen de contrôler la copie et la diffusion illicite de documents. Il s'effectue en deux étapes : dans la première étape, on dissimule de l'information dans le document en insérant une marque ou filigrane et dans la seconde étape on détecte la marque. Le marquage et la détection s'effectuent à l'aide d'une clef, ce qui permet d'identifier la marque, car seul quelqu'un en possession de cette clef peut réaliser la détection. On identifie ainsi l'auteur du document ou son propriétaire licite. La marque doit être transparente ; C'est-à-dire qu'elle ne doit pas être visible et qu'elle ne doit pas dégrader la qualité du document,

Le principe est donc radicalement différent de celui de la cryptographie. La cryptographie comprend également deux étapes, le chiffrement et le déchiffrement. Une fois que le document est chiffré, il n'est plus utilisable, il n'a même pas de ressemblance avec le document original et une fois déchiffré on retrouve celui-ci sans modification. Pour le tatouage, le document tatoué est dégradé mais reste exploitable, en effet, on s'assure que la marque ne nuit pas à son utilisation. Notons aussi qu'une modification légère du document ne retirera pas la marque, inversement, si on altère un document chiffré, il ne sera plus déchiffrable ; ce qui garantit l'intégrité qui est un fondement de la cryptographie avec l'authenticité et la confidentialité. Nous retrouverons cette différence essentielle dans le cas de la biométrie, car une marque biométrique est soumise à une variabilité lors de l'enrôlement qui ne doit pas gêner l'authentification.

La détection ne retire pas la marque, elle permet juste de vérifier si la diffusion du document est licite. La signature numérique –qui combine généralement une clef et une fonction de hachage– se rapproche du tatouage car elle ne dégrade pas le document, mais elle correspond à une métadonnée qui peut se retirer aisément. Dans le cas de documents numériques diffusés sur internet, l'utilisateur ne se préoccupe généralement pas de l'origine ou de l'auteur du document, il est donc préférable que la signature

soit insérée dans le document pour être sûr qu'elle soit présente lors de la diffusion du document.

2.1.2 Principaux critères

La majeure partie des travaux dans ce domaine concerne le tatouage de documents multimédias (image, vidéo, audio). Nous avons vu que le marquage devait préserver la qualité du document. La différence principale avec les documents structurés porte sur la notion de qualité. Dans le cas des documents multimédias, celle-ci se base essentiellement sur la perception humaine (il existe tout de même des méthodes pour formaliser cela), alors que la dégradation pour les documents structurés provient de contraintes (métriques, topologiques, ...) qui doivent être préservées ; par exemple, nous devons avoir les mêmes réponses pour un ensemble de requêtes. On peut donc formaliser (avec des formules logiques) les contraintes à préserver.

Cependant pour définir convenablement la notion de qualité, il est nécessaire de connaître les usages qu'auront les utilisateurs de leurs documents. Nous avons eu des discussions très intéressantes sur ce sujet avec Anne Ruas –alors directrice au COGIT, le laboratoire d'informatique de l'IGN, et participante du projet TADORNE– en particulier sur les transformations légitimes pour des données géographiques. Nous parlerons de transformations légitimes en opposition à des transformations conçues par un attaquant pour faire disparaître la marque. D'après A. Ruas, il ressort que les clients de l'IGN ne souhaitent pas préciser l'utilisation qu'ils feront de leurs données. Il faut donc essayer que le tatouage ne dégrade pas le document pour un maximum d'usages, sans garantir que cela ne sera pas le cas pour une utilisation très particulière. Mais il n'est pas possible de garantir que le tatouage ne nuira pas à l'utilisation des documents tatoués car les usages peuvent être extrêmement variés.

Nous verrons plus loin que les méthodes de tatouage dépendent fortement du type de données tatouées et qu'il semble donc irréalisable d'unifier tous les travaux sur ce domaine. Cependant certaines idées peuvent être reprises. Nous avons ainsi repris le travail sur la triangulation de Delaunay d'Obushi et al [100, 101] pour les maillages 3D que nous avons ensuite adapté sur des données géographiques vectorielles. Notre méthode –appelée méthode des sites– a une certaine ressemblance avec la méthode des patchworks, bien que la sélection d'un site soit locale, alors que celle d'un patchwork se fasse sur le document entier.

Notre méthode a été validée sur les documents géographiques et nous avons ensuite montré comment l'appliquer à tout document structuré et devant préserver des contraintes. Par contre, nous pensons qu'elle a peu de chance de pouvoir s'appliquer à des données multimédias, à cause de cette différence importante quant à la notion de qualité qui est centrale dans notre méthode.

Voici les principales caractéristiques de notre méthode.

Tatouage aveugle On parle de tatouage non aveugle ou informé lorsque la détection nécessite d'avoir le document original, la marque s'extrait alors en superposant les

deux documents. En revanche, pour le tatouage aveugle, la détection se fait uniquement avec le document tatoué. Nous avons choisi le second type de tatouage qui est beaucoup plus pratique pour les applications envisagées. En effet, un grand nombre de versions d'un document géographique peut être distribué et il ne semble pas réaliste de pouvoir conserver toutes ces versions (pour un problème de mémoire, mais aussi d'archivage).

Tatouage rapide La plupart des algorithmes de tatouage de données contraintes telles que les maillages 3D sont basés sur l'analyse spectrale. Notre méthode ne manipule que des parties locales du document. L'algorithme le plus coûteux pour le marquage comme pour la détection est le calcul de la triangulation de Delaunay qui est linéaire avec l'hypothèse que le nombre de voisins dans le Delaunay est borné ; ce qui a toujours été vérifié pour les documents testés. Tous les autres algorithmes se font également en temps linéaire.

Tatouage robuste Le tatouage doit résister à différentes transformations du document. On distingue deux types de transformations : celles qui sont légitimes (nous dirons aussi naturelles), l'utilisateur va les effectuer pour un meilleur usage de son document et celles propres à des attaques, l'utilisateur va les effectuer dans le but de supprimer la marque. Nous nous sommes essentiellement concentrés sur les transformations légitimes. Je reviendrai plus loin sur la difficulté de prévoir les attaques possibles. Les transformations les plus naturelles pour un document géographique est le découpage et les transformations géométriques (translation et rotation). Notre méthode est conçue pour résister parfaitement à ces modifications. Pour la simplification (suppression de points ou de polygones), l'algorithme est relativement résistant si celle-ci ne s'effectue pas systématiquement sur l'ensemble du document.

Tatouage 0-bit Pour pouvoir insérer une métadonnée comme le nom du propriétaire et l'heure et le lieu du marquage, il faut un tatouage n-bits. Nous avons alors n bits d'information qui sont insérées dans le document. Dans notre cas, la marque se fait localement et sans ordre entre les sommets, afin de mieux résister au découpage et à la rotation. Ce qui nous a conduit à un tatouage 0-bit. L'algorithme de détection répond, en possession du document et de la clef, oui ou non selon que le document est tatoué ou pas. C'est à partir de la clef que l'on retrouve les informations nécessaires à l'authentification (propriétaire du document, identité de l'utilisateur, date...). Notre méthode pourrait néanmoins être modifiée pour insérer quelques bits, mais pas pour insérer un long message.

2.2 Présentation du schéma de tatouage

La principale originalité de notre schéma de tatouage est d'avoir une approche entièrement locale. Le marquage et la détection vont se faire sur des petites parties du document appelées *sites*. Nous allons définir une propriété sur ces sites et c'est en modifiant la distribution de cette propriété sur certains sites que nous allons introduire

un biais statistique. Pour mettre en évidence ce biais statistique, il faut posséder la clef secrète qui a permis de sélectionner ces sites.

La première étape de l'algorithme de marquage consiste à construire la triangulation de Delaunay [87] formée par les sommets du document géographique original. Une triangulation de Delaunay est une triangulation ayant comme propriété que le cercle circonscrit à tout triangle ne contient pas d'autre point que les trois points de ce triangle. Cette triangulation favorise l'équilatéralité (les trois côtés ont des longueurs proches), sauf au niveau de l'enveloppe convexe. Nous obtenons ainsi un graphe avec des sommets possédant des coordonnées et deux sortes d'arêtes, celles provenant des polygones ou polylignes du document géographique et celles ajoutées par la triangulation. En théorie, il peut y avoir plusieurs triangulations de Delaunay possibles. En pratique, nous en avons une seule et il est toujours possible de concevoir un algorithme donnant toujours la même triangulation.

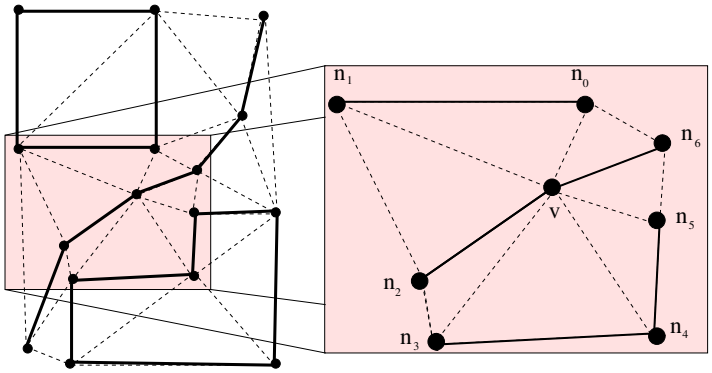
Nous avons choisi pour la notion de qualité pour les documents tatoués une définition relative au document original et non basée sur des critères absolus. Il s'agit donc plutôt d'une préservation de la qualité que de qualité proprement dite. Nous avons choisi de préserver deux contraintes très différentes. Une contrainte topologique –le document tatoué doit donner la même triangulation de Delaunay– et une contrainte métrique– la perte de précision (la distance maximale autorisée pour déplacer un sommet) est fixée au départ. Notre schéma de tatouage laisse cependant la possibilité d'ajouter d'autres contraintes à préserver si cela s'avère nécessaire.

FIGURE 2.1 – Carte vectorielle des routes du Calvados, 44613 objets et 159800 sommets

Définition d'un site L'idée est de travailler sur des petites parties du documents, appelés sites. Un site (voir figure 2.2) est composé d'un sommet central (le centre du site), de tous ses voisins dans la triangulation de Delaunay, de tous les sommets miroirs par rapport à chacune de ces faces adjacentes dans la triangulation et de toutes les arêtes entre le centre et ses voisins dans le document original. Dans un premier temps, nous n'avions pas considéré les sommets miroirs, mais ils sont nécessaires pour vérifier que le Delaunay est inchangé (toutes les vérifications de préservation de qualité

doivent s'effectuer en ayant à notre disposition uniquement le site).

FIGURE 2.2 – Définition d'un site



Sélection des sites Á chaque site correspond un codage dépendant des relations entre le centre et ses voisins.

Le codage est une fonction C de $\mathcal{S}$,l'ensemble des sites, vers $\mathbb{N}$.

Notons qu'il ne dépend pas d'un ordre entre les sommets, il est donc invariant par transformations géométriques et il est uniquement topologique (pas métrique). Tant que la triangulation est préservée (ce qui peut se vérifier localement) le codage reste inchangé. Nous obtenons ainsi une première partition –la partition de codage– que tout le monde peut calculer.

Une seconde partition en p parties va regrouper les sites en les numérotant de 0 à $p-1$, où p est un paramètre fixé au départ. On calcule le numéro d'un site en appliquant une fonction de hachage à la concaténation du codage du site et d'une clef secrète. Plus précisément soit *hash* la fonction de hachage utilisée, $C(s)$ la codage du site s et k la clef, le numéro du site vaut

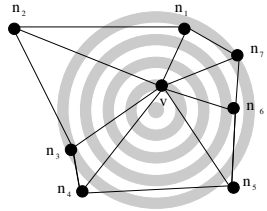
$$P_p(s,k) = hash(C(s),k) \mod p.$$

Modification des sites Nous avons défini une propriété métrique sur un site. On construit des disques concentriques coloriés alternativement en blanc et noir autour du barycentre des voisins du centre (figure 2.3). Le site vérifie la propriété ϕ si le centre est dans un disque noir. Le codage se faisait uniquement sur un critère topologique, alors qu'ici la propriété est seulement métrique.

Le marquage s'effectue en forçant les sites de la partie 0 à vérifier la propriété ϕ et en forçant les sites de la partie 1 à ne pas vérifier ϕ.

Biais statistique et borne de Chernov On observe expérimentalement qu'un site vérifie la propriété ϕ avec une probabilité μ proche de $1/2$. Divers tests ont été effectué en prenant des échantillons de diverses tailles (n sites, où n varie) et de différentes manières (sites contigus ou espacés). Des tests du Chi-deux nous permettent de valider l'hypothèse d'essais de Bernoulli de paramètre μ (succès si ϕ est vérifiée et échec sinon).

FIGURE 2.3 – Modification de la propriété ϕ



Avec cette hypothèse d'essais de Bernoulli, on peut majorer la probabilité d'avoir une certaine fraction de sites qui vérifie la propriété, car le nombre de succès suit une loi binomiale de paramètre μ.

Soient n le nombre de sites considérés, le nombre de succès moyen vaut donc $n\mu$. Soit m entre 0 et n et E la variable aléatoire du nombre de succès, en utilisant la borne de Chernov, la probabilité que l'écart (en valeur absolu) entre E et la moyenne s'écarte de plus que $|m - n\mu|$ est majorée par

$$\Pr(|E - n\mu| > |m - n\mu|) \leq 2e^{-2n(\frac{m}{n}-\mu)^2}.$$

Détection de la marque Le fait de forcer une partie des sites à vérifier et une autre partie à ne pas vérifier ϕ induit un biais statistique sur le document. Ce biais statistique est appliqué aux parties 0 et 1.

Soit n_0 et n_1 le nombre de sites des parties respectivement 0 et 1. Soit m_0 et m_1 le nombre de sites que l'on observe dans le document tatoué vérifiant la propriété ϕ. Lors du tatouage, on s'arrange pour que m_0 soit le plus grand possible et que m_1 soit le plus faible possible.

On utilise la borne de Chernov ci-dessus pour majorer la probabilité que le document soit non tatoué Soit $P(n_0, n_1, m_0, m_1) = 4e^{-2n((\frac{m_0}{n_0}-\mu)^2+(\frac{m_1}{n_1}-\mu)^2}$.

$$\Pr(\text{document non tatoué}) \leq P(n_0, n_1, m_0, m_1).$$

Comme la seconde partition dépend de la clef, une personne qui ne possède pas celle-ci ne saura pas quels sites ont été modifiés et donc devra déplacer tous les sites pour essayer de retirer la marque. Ce qui ne lui assure pas de retirer la marque, car le biais statistique est résistant au bruit (déplacement des sommets selon une certaine distribution).

Algorithme de marquage On extrait tous les sites un à un dans un certain ordre. Pour chacun de ces sites, on calcule sa partie (numéro attribué avec la fonctions de hachage). Si c'est la partie 0, on modifie le centre pour que le site vérifie ϕ, si c'est la partie 1, on le modifie pour qu'il ne vérifie pas ϕ. Pour les autres parties, on effectue aucune modification. On vérifie ensuite si la triangulation de Delaunay est préservée et on répercute la modification sur le document uniquement si c'est le cas. Notons qu'à ce niveau l'ordre a une importance, car si l'on change d'ordre nous n'aurons pas exactement les mêmes sites qui seront modifiés, car un sommet apparaît dans

plusieurs sites. Le fait de ne pas effectuer la modification si une des deux contraintes n'est pas préservée n'est pas préjudiciable à la performance de la méthode que si cela ne concerne qu'une faible fraction des sommets.

Algorithme de détection On fixe un seuil λ. Le début de l'algorithme est le même que pour le marquage, on extrait les sites un à un et on calcule leur partie. On calcule n_0, n_1, m_0, m_1. Si $P(n_0, n_1, m_0, m_1) \leq \lambda$ alors on répond que le document est tatoué, sinon qu'il n'est pas tatoué.

2.3 Évaluation du schéma de tatouage

2.3.1 Tests de tatouage

Nos tests ont été effectué sur deux corpus –deux cartes géographiques vectorielles provenant de l'IGN– une contenant les tronçons de routes du Calvados (figure 2.2) et une autre contenant les limites des communes du Calvados.

Le fait de disposer de documents de très grande taille –par exemple, les tronçons de routes du Calvados comportent 159800 sommets et 170748 arêtes– permet de valider l'efficacité réelle d'un logiciel (passage à l'échelle) et pas uniquement de tester un prototype. On dispose ainsi d'un corpus en effectuant des découpages de différentes tailles, cela permet d'obtenir un grand nombre de documents tatoués et non tatoués et de tester intensivement les limites de la méthode de tatouage.

Faux positif (ou fausse acceptation) Nous obtenons un faux positif lorsqu'un document non tatoué est déclaré tatoué.

Faux négatif (ou faux rejet) Nous obtenons un faux négatif lorsqu'un document tatoué est déclaré non tatoué.

Le choix du paramètre λ utilisé dans l'algorithme de détection est très important. Si λ est trop grand, nous allons favoriser les faux positifs et inversement, s'il est trop petit,nous aurons plus de faux négatifs.

La perte de précision autorisée est aussi un paramètre important. Elle détermine la largeur des disques concentriques intervenant dans la propriété ϕ.

Pour nos tests, nous avons choisi cinq valeurs pour λ, $10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ et 10^{-5} et pour la perte de précision $1, 3$ et 5 mètres, plusieurs clefs ont été utilisées.

Par exemple, pour $p = 4$, la détection fonctionne parfaitement pour tous les documents de plus de 100 sommets pour la perte de précision de 1 mètre avec $\lambda = 10^{-2}$, c'est-à-dire qu'aucun faux positif et aucun faux négatif n'est observé. Lorsque p augmente ($12, 14$ ou 16), il faut considérer des documents plus gros, car une fraction plus faible de site est impliquée dans le marquage.

Robustesse aux transformations Par définition du schéma de tatouage, celui-ci résiste à la rotation, la translation et le changement d'ordre des objets (les sommets peuvent apparaître dans n'importe quel ordre). Notons que cette dernière propriété

est difficile à garantir pour un tatouage n bits car l'information nécessite un ordre, même lorsque l'on utilise des codes correcteurs pour avoir une certaine redondance.

Robustesse au découpage Pour tester la robustesse au découpage, nous avons découper la carte en 16 documents qui ont été tatoués et ensuite découpés en documents de différentes tailles. Pour $p = 4$ et une perte de précision de 1 mètre , on s'aperçoit qu'il reste des faux négatifs pour des documents ayant jusqu'à 400 sommets.

Robustesse au retatouage Le tatouage avec successivement deux clefs sur des documents de divers tailles avec $p = 4$ et une perte de précision d'un mètre a été effectué. Nous arrivons à discriminer les documents tatoués et non tatoués à partir de 200 sommets.

Corrélations entre les clefs La seconde partition regroupe les classes de codage en fonction de la clef. Donc quelle que soit la clef, deux sites qui ont le même codage subiront le même traitement, c'est-à-dire soit ne seront pas traités, soit forcés à vérifier ϕ ou soit forcés à ne pas vérifier ϕ. Donc nous avons des corrélations entre deux partitionnements pour deux clefs différentes. En revanche, nous avons testé par un test du Chi-deux l'indépendance entre deux partitions si nous considérons un seul sommet par classe. Les tests effectués montrent que le nombre de classes de codage est assez important pour ne modifier qu'un seul site par classe de codage. Par exemple, pour des documents de moins de 3000 sommets, le rapport entre nombre de classes sur le nombre de sommets est en général supérieur à 60%.

2.4 Schéma générique

2.4.1 Critères à définir

Comme cela est indiqué dans le préambule, ce travail exploratoire sur le tatouage de documents géographiques a deux objectifs majeurs complémentaires, d'une part de concevoir un logiciel complet qui permet le passage à l'échelle et n'omet aucun aspect technique pour une mise en œuvre efficace ; d'autre part, de pouvoir séparer la partie propre au type de documents considérés et une autre partie générale pour tout type de données structurées devant préserver des contraintes. La conception du schéma de tatouage demande donc à la fois une expertise sur le tatouage de documents numériques et une expertise sur le type de documents à tatouer.

Le schéma générique est conçu pour formaliser cette séparation. Nous allons voir quels sont les critères à vérifier.

Qualité du document et d'un site Il faut définir $\mathcal{D}$, l'ensemble des documents possibles et une relation $Q_{\mathcal{D}}$ sur les documents, si deux documents D_1 et D_2 sont tels que $Q_{\mathcal{D}}(D_1, D_2)$ est vraie alors on dira que la qualité du document est préservée lorsque l'on passe de D_1 à D_2. La relation sera réflexive et symétrique, mais pas forcément transitive (par exemple, la préservation de précision métrique n'est pas transitive).

On définit une relation de qualité du site $Q_{\mathcal{S}}$ sur $\mathcal{S}$, l'ensemble des sites possibles telle que si deux sites $Q_{\mathcal{S}}(s_1, s_2)$ est vraie pour deux sites s_1 et s_2, alors $Q_{\mathcal{D}}(D_1, D_2)$ est vraie si D_2 est le document obtenu à partir de D_1 en replaçant le site s_2 à partir de s_1.

Que se soit pour la qualité du document ou pour la qualité du site, il s'agit plus précisément d'une préservation de qualité, une définition formelle de qualité à partir d'un seul document ou d'un seul site n'a pas vraiment de signification.

Codage de sites On définit une fonction de codage $C : \mathcal{S} \to \mathbb{N}$. Le codage d'un site doit s'effectuer sur des caractéristiques significatives du site, c'est-à-dire sur des données sur lesquelles est basée la qualité du site.

Propriétés ϕ_0 et ϕ_1 Les propriétés ϕ_0 et ϕ_1 doivent être vérifiées par les sites des parties respectivement 0 et 1. Elles doivent porter sur des critères « orthogonaux »à ceux du codage. Pour notre étude, le codage était basé sur la topologie (Delaunay) et ϕ_0 et ϕ_1 sur un critère métrique.

Dans notre schéma précédent, ϕ_0 et ϕ_1 étaient une propriété et sa négation, mais on peut envisager d'autres possibilités (voir l'application sur les bases de données).

Biais statistique Nous avons vu que la modification d'un seul site par classe de codage augmente la sécurité du tatouage, il faut donc vérifier que le nombre de classes est suffisamment important. Il faut aussi s'assurer par des tests statistiques que ϕ a une bonne distribution.

2.4.2 Application à une base de données

Ce problème a été proposé par David Gross-Amblard [90]. Il s'agit de tatouer une base de données tout en préservant une requête de somme sur un attribut pour un ensemble d'enregistrements de la base. Pour effectuer les tests, on simplifie le problème en considérant qu'un enregistrement est un couple (i, b) où i est un identifiant robuste (un attribut qui ne peut être modifié) et b est un bit modifiable de l'attribut à sommer.

Les contraintes à préserver –qui vont servir à définir la notion de qualité du document– sont, d'une part, de ne pas modifier les identifiants et, d'autre part, de préserver la somme totale.

Un site s sera formé par deux couples $(i_1, b_1), (i_2, b_2)$, la préservation de qualité de site correspond à ce que $b_1 + b_2$ reste inchangé.

le codage est formé par les attributs qui sont des identifiants (ici i_1 et i_2 pour notre simplification) et s vérifiera ϕ_0 lorsque $b_1 = 0$ et $b_2 = 1$ ϕ_1 lorsque $b_1 = 1$ et $b_2 = 0$. Avec l'hypothèse que 25% des sites vérifient ϕ_0 et 25% vérifient ϕ_1, nous pouvons calculer les biais statistiques.

A priori le nombre de sites est quadratique par rapport à n, le nombre d'enregistrements. Il est possible en regroupant les sites d'avoir autant de sites que d'enregistrements. Les algorithmes de marquage et de détection deviennent alors linéaire

par rapport à n. Notons que l'algorithme de marquage de Gross-Amblard était quasi-linéaire car dans une première étape les enregistrements étaient triés, alors qu'ici cette étape n'est pas nécessaire.

2.5 Bilan et perspectives

Nous avons montré que la méthode des sites était particulièrement efficace pour les documents numériques géographiques que ce soit en temps, en robustesse –transformations géographiques– ou parce qu'il est aveugle.

Nous avons vu que le codage des sites et la propriété à modifier sur les sites devaient porter sur des critères différentes. Il semble intéressant de voir si la méthode des sites peut s'appliquer à d'autres types de documents.

Données textuelles Ce travail peut s'appliquer à des données textuelles. L'idée est d'insérer l'identité d'un auteur dans le texte en effectuant des choix (par exemple, en prenant un mot parmi un ensemble de synonymes) en fonction de l'auteur. Ces modifications ne changerait pas le sens du texte, ni sa construction grammaticale.

La définition et le codage de sites semblent assez facile à définir, mais il semble plus délicat de d'obtenir une propriété à modifier orthogonale au site.

Modèle de l'attaquant Pour donner une preuve qu'un schéma est résistant aux attaques (transformations illicites), il faut définir un modèle de l'attaquant. On définit ainsi plusieurs niveaux d'objectifs et plusieurs niveaux de moyens d'un attaquant. Nous avions envisagé de travailler sur ce sujet, malheureusement le domaine du tatouage –particulièrement pour des données non-multimédia– est assez immature , contrairement à la cryptographie qui possède des fondements théoriques plus solides. Il semble donc peu probable de proposer très prochainement des modèles d'attaquant complets faisant l'unanimité.

Chapitre 3

Biométrie – empreintes digitales et templates de minuties

3.1 Empreintes digitales et template des minuties

Il existe d'autres modalités de la biométrie pour la reconnaissance d'une personne comme la reconnaissance du visage, de l'iris, des veines de la main. Mais l'empreinte digitale est l'une des modalités les plus utilisées et également des plus étudiées.

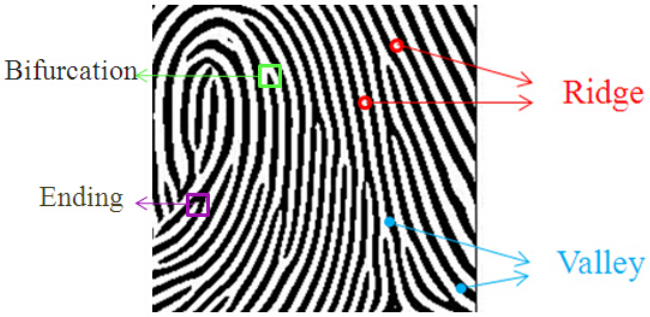
Bien que les sujets de thèse de Yao et Vibert aient des objectifs très différents, ils portent sur les mêmes données : les templates de minuties, points remarquables des empreintes digitales extraits lors de l'enrôlement de l'empreinte digitale.

L'empreinte digitale possède des lignes parallèles appelées stries ou crêtes. Ce sont ces lignes qui caractérisent la forme des empreintes digitales. On regroupe ces empreintes en plusieurs grandes familles (ou types) (figure 3.1).

FIGURE 3.1 – Types d'empreintes digitales

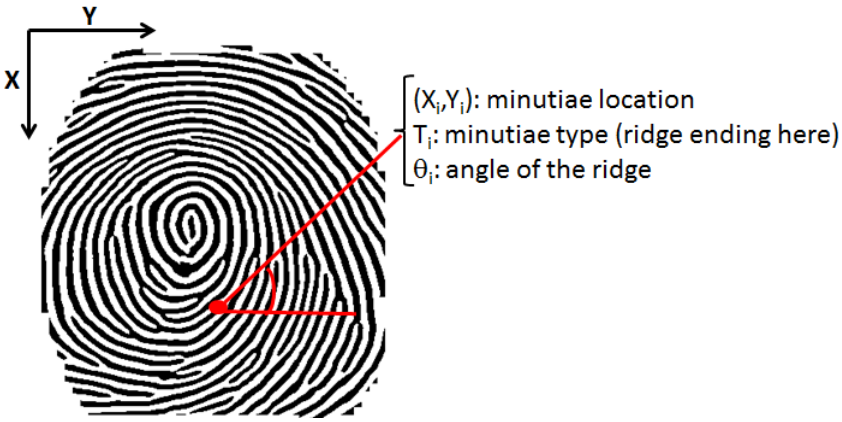
Plutôt que de stocker l'empreinte digitale, il est souvent préférable de ne stocker que les coordonnées de points singuliers locaux appelés minuties, ce sont des points d'irrégularité se trouvant sur les lignes papillaires (terminaisons, bifurcations, îlots-assimilé à deux terminaisons, lacs) (figure 3.2). Le stockage d'une minutie comprend ses coordonnées et un angle pour l'orientation (figure 3.3).

FIGURE 3.2 – Types de minuties



La triangulation de Delaunay appliqués à un template de minuties permet d'avoir des informations sur ce template en effectuant des statistiques sur les aires, les périmètres, les longueurs des triangles (figure 3.4).

FIGURE 3.3 – Extraction des minuties



3.1.1 Qualité des empreintes digitales

Z. Yao a effectué sa thèse sur l'évaluation de la qualité des empreintes digitales et sur l'impact de la qualité sur la performance des systèmes biométriques. Une partie importante des études en biométrie concerne l'évaluation de la performance [86].

La figure 3.5 montre différentes acquisitions d'empreintes de qualité différente. Il semble évident que la performance des algorithmes d'identification va baisser si l'enrôlement est de mauvaise qualité. Il faut donc être capable de mesurer cette qualité, pour par exemple recommencer l'enrôlement.

Je n'ai encadré Yao que pour sa dernière année –qu'il a soutenu le 21 juillet 2015– de ce fait mon apport reste modeste.

Je me sentais moins concerné par certaines de ses études qui portaient sur les images des empreintes digitales, car j'ai peu de connaissances en imagerie et je ne pouvais donc pas juger finement de la pertinence de ses choix. J'ai cependant joué

FIGURE 3.4 – Exemples de triangulations de Delaunay

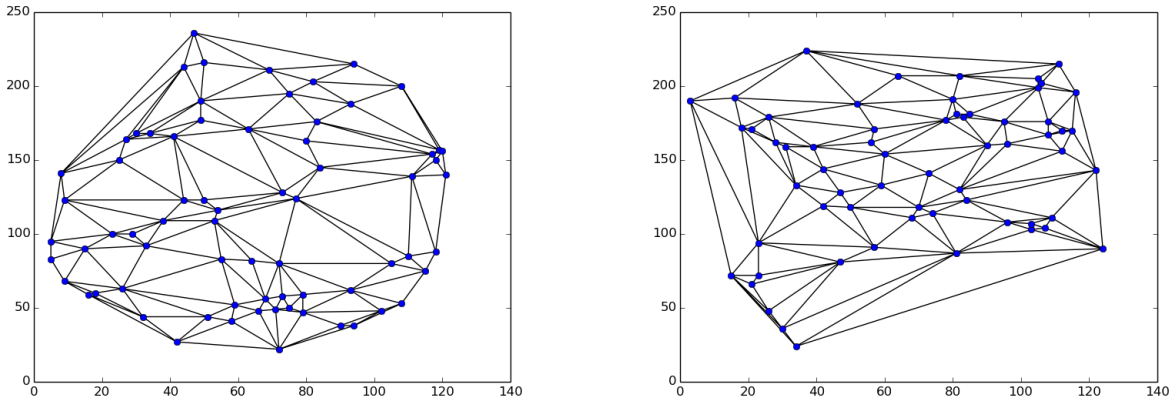
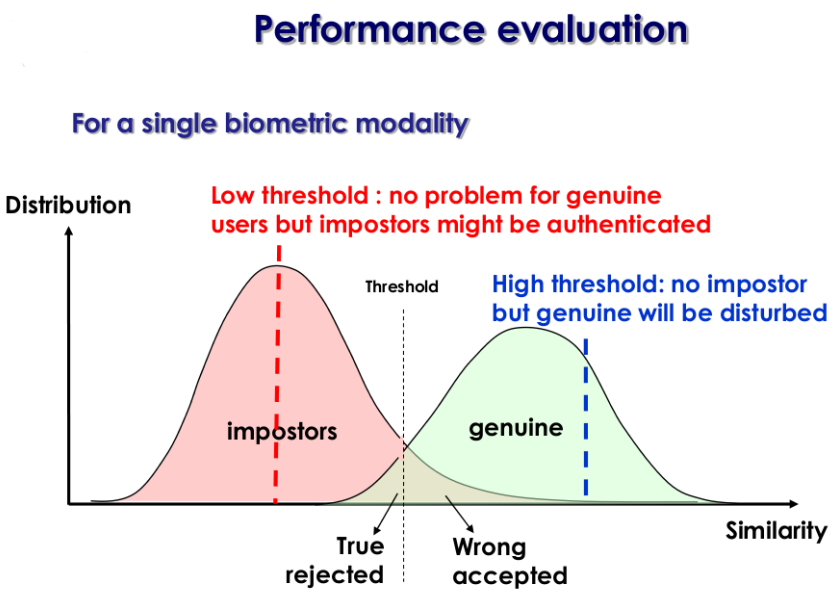


FIGURE 3.5 – Relevés d'empreintes digitales

mon rôle de co-encadrant en le conseillant pour la rédaction et la présentation de ses travaux que ce soit pour les articles soumis en conférence ou pour son manuscrit de thèse. Yao a eu plusieurs contributions de nature différents dans sa thèse. Il a proposé un framework générique pour valider la mesure de qualité des systèmes biométriques [110], il a montré comment combiner des mesures de qualités [108]. La figure 3.6 illustre l'évaluation de la performance lorsque l'on modifie le seuil pour départager les empreintes provenant de la même personne et celles provenant de deux personnes différentes. Selon la valeur du seuil, on va privilégier les fausses acceptations (deux empreintes sont de personnes différentes et l'on décide qu'elles sont de la même personne) ou les faux rejets (deux empreintes sont de la même personne et l'on décide qu'elles sont de personnes différentes). Les deux types d'erreur sont appelées respectivement False Match Rate (FMR) and False Non-Match Rate (FNMR). On peut décider que la meilleure méthode est d'obtenir la valeur la plus faible lorsque

ces deux erreurs atteignent la même valeur. On peut également choisir comme critère l'aire sous la courbe (courbe ROC, pour Receiver Operation Characteristic).

FIGURE 3.6 – Évaluation de la performance



La partie qui m'a le plus intéressé utilise une triangulation obtenue à partir du template de minuties d'une personne. Les données biométriques collectées sont conservées sous forme numérique (template de minuties) et dans la suite des traitements, nous n'avons plus accès aux images initiales. Ainsi Yao a étudié [107] la mesure de la qualité d'un template de minuties à partir de critères issus de la triangulation de Delaunay, la même structure que celle qui a joué un rôle central pour le tatouage de données géographiques.

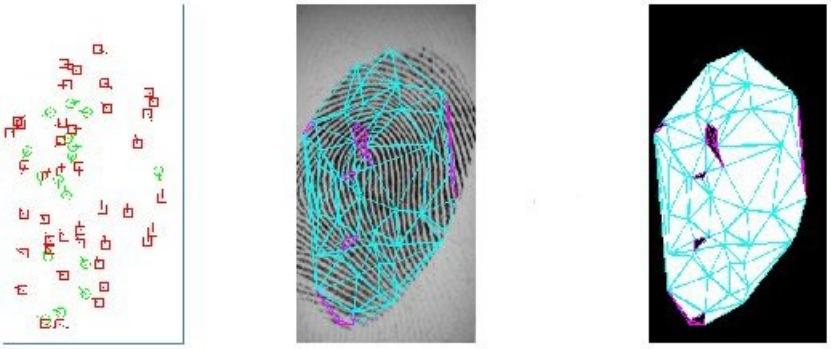
La mauvaise qualité d'un template est principalement due à l'absence de certaines minuties lors de l'extraction du template (il peut aussi y avoir de fausses minuties détectées, mais Yao ne l'a pas pris en compte).

La méthode de Yao consiste à calculer la surface intérieure à l'enveloppe convexe de la triangulation de Delaunay. Cela correspond à la somme des aires des triangles de la triangulation. Il retire ensuite les triangles qu'il juge improbable (ils apparaissent parce que des minuties n'ont pas été extraites de l'image), il appelle ces triangles des *mauvais triangles*. Il a établi trois seuils pour le périmètre, l'aire et le rapport périmètre/aire (qui est fortement lié aux angles). L'aire des triangles dépassant un de ces trois seuils est retirée. La qualité est mesurée en fonction du pourcentage de surface restante.

Les images de la figure 3.7 montrent de gauche à droite, le templates de minuties d'une empreinte digitale, la triangulation de Delaunay effectuée sur les minuties et les zones retirées pour le calcul de la qualité.

Les choix qu'il a effectués donnent des résultats convaincants, mais ils ne sont justifiés qu'a posteriori, après des tests effectués sur des bases d'empreintes digitales.

FIGURE 3.7 – Qualité d'un template de minuties



Il faut noter que les tests jouent un rôle très important dans ce domaine, ils sont réalisés sur des bases conséquentes provenant de plusieurs compétitions FVC (Fingerprint Verification Competition). D'autres part, les bases sont constituées d'acquisitions d'empreintes digitales par différentes modalités : optique, thermique, construit avec le logiciel SFinGe (logiciel générant de fausses empreintes).... Il est donc normal que dans la rédaction des articles les expérimentations et leurs interprétations occupent une place importante. Cependant je trouve indispensable de pouvoir justifier les choix effectués, de les discuter avec d'autres approches, surtout pour le manuscrit de thèse.

Je n'ai pas pu approfondir et formaliser avec lui cette partie (et ce n'est pas faute d'avoir insisté), car il a malheureusement privilégié une exploration en largeur de son sujet, en proposant de nombreuses études variées, au détriment d'une exploration plus en profondeur sur quelques études plus ciblées. Je partage le point de vue des membres du jury de la thèse qui estime que la partie sur la triangulation de Delaunay doit être approfondie. Il reste encore à effectuer des calculs statistiques poussés sur les distributions des aires, angles et périmètres, afin de déterminer quelles sont les caractéristiques les plus pertinentes pour mesurer la qualité d'un template. J'espère que l'équipe poursuivra d'autres études sur la mesure de la qualité des templates par la triangulation de Delaunay.

3.1.2 Sélection et attaques sur les empreintes digitales

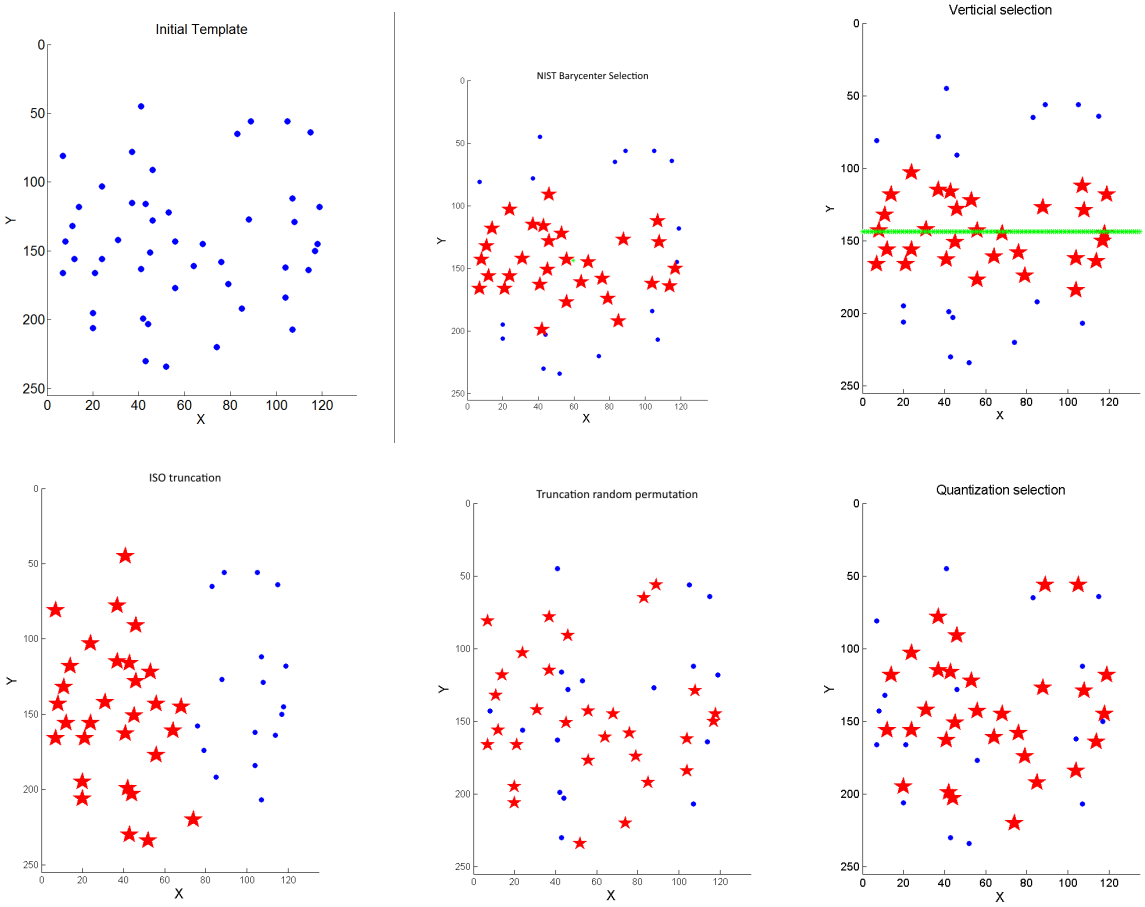
Je co-encadre la thèse de B. Vibert pour les 2/3 de sa durée (juillet 2014–décembre 2016). La thèse de Vibert porte sur la biométrie embarquée sur élément sécurisé (SE). Une partie de son travail concerne l'évaluation des systèmes biométriques (performance, sécurité, usage) [103, 104], en particulier il est en charge du développement de la plateforme EvaBio dédiée à l'évaluation de systèmes biométriques [105, 106]. Il travaille également sur la réduction de templates de minuties [102] et sur les attaques pour s'authentifier auprès d'un MOC (Match On Card).

Comme pour la qualité des empreintes digitales, je m'intéresse plus particulièrement aux propriétés des triangulations de Delaunay des templates de minuties, mais pour des études différentes. Nous sommes toujours dans un contexte où nous ne dispo-

sons pas de l'image de l'empreinte digitale, nous avons seulement les coordonnées des minuties et leur orientation. Il n'est donc pas possible d'utiliser des techniques d'imagerie pour appliquer des traitements sur l'empreinte. La triangulation de Delaunay s'avère spécialement pratique pour manipuler les minuties. Pour certaines applications –typiquement la technologie MOC– le template de minuties est comparé à celui d'une empreinte entré en référence dans une carte numérique. Le format impose de limiter le nombre de minuties. Il importe donc de trouver un moyen de sélectionner les minuties qui donneront la meilleure performance lors de l'authentification.

B. Vibert a comparé les différentes manières de sélectionner les minuties [102]. Les méthodes usuelles consistent à retirer les minuties les plus loin du core de l'empreinte (qui n'est qu'estimé car on ne dispose pas de l'image). Il a proposé une stratégie complètement différente qui consiste à répartir dans l'espace les minuties choisies. Plus précisément, il a utilisé k-means, une méthode de partitionnement. La figure3.8 montre la répartition spatiale des minuties retenues par les différentes méthodes de sélection (la méthode k-means est appelée quantization method).

FIGURE 3.8 – Répartition spatiale des minuties sélectionnées



Il a montré que cette nouvelle méthode était plus performant dans certains cas. Malheureusement, elle est beaucoup plus coûteuse en temps que les autres méthodes. Je lui ai proposé de tester des méthodes utilisant la triangulation de Delaunay et qui effectuent une répartition similaire des minuties sélectionnés. Pour la première, on retire de manière incrémentale la minutie apparaissant dans le plus grand nombre de triangles et on recalcule la triangulation. La méthode peut être implémentée avec une faible complexité car il n'est pas nécessaire de reconstruire toute la triangulation, il suffit de reformer les triangles proche de la minutie retirée. Pour la seconde méthode, on retire toujours incrémentalement la minutie dont la somme des aires des triangles contenant cette minutie est minimale. D'autres méthodes peuvent être testées, par exemple la combinaison pondérée de ces deux méthodes.

J'espère fortement que Vibert aura le temps avant la fin de sa thèse d'effectuer ces expérimentations qui permettront de compléter et renforcer son travail sur la sélection de minuties.

La triangulation de Delaunay est aussi utile pour les attaques, où plusieurs problématiques sont envisageables. Si on se place du côté de l'attaquant, peut-on créer un template de minuties dont la triangulation de Delaunay ressemble à celui d'une empreinte digitale ? Peut-on construire de faux templates augmentant significativement les chances de s'identifier indûment ? Du côté de la protection des données biométriques, peut-on à partir d'une triangulation de Delaunay déterminer si elle provient d'une empreinte digitale réelle ? L'objectif à long terme est d'obtenir une modélisation des templates de minuties permettant de discerner les différences entre deux templates pour l'identification et en même temps de mettre en avant les caractéristiques communes entre toutes les empreintes. Une telle modélisation permettrait de répondre aux questions ci-dessus. Des études statistiques sur des caractéristiques tels que les angles, les périmètres ou encore l'aire, nous permettent de déterminer les propriétés les plus pertinentes pour la construction de faux templates.

Le premières études semblent montrer que les statistiques sur les triangles sont assez stables, y compris d'un type à un autre, en revanche les statistiques sur les angles de l'orientation sont suffisantes pour classifier les types. D'autre part, le logiciel SFinge produit des templates avec des statistiques sur les triangles proches des véritables empreintes, il pourrait donc être utilisé pour effectuer des attaques.

3.2 Biométrie révocable pour le tatouage d'image

Avec Christophe Rosenberger et Morgan Barbier, nous avons proposé un schéma de tatouage d'images basé sur des données biométriques révocables.

Notons qu'une des différences majeures entre la biométrie et la cryptographie comme outil pour sécuriser des données est la variabilité d'une donnée biométrique. Les méthodes d'identification (ou d'authentification) doivent tenir compte de cela en s'assurant que l'identification est préservée lorsque la donnée varie un peu.

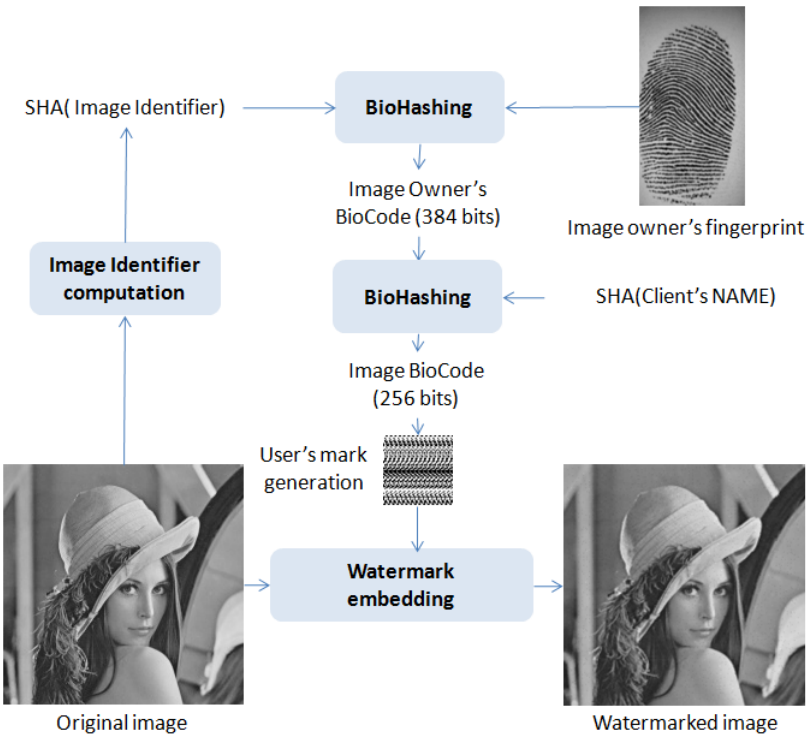
Le biocode permet cela mais d'autres méthodes existent comme l'engagement flou (fuzzy commitment) [92, 93] qui assure un engagement pour des données proches de celle de départ en utilisant des codes correcteurs d'erreur et qui est réalisé

indépendamment de la partie biométrique.

A partir d'une empreinte digitale de l'utilisateur décrite par un vecteur de paramètres appelé FingerCode, l'algorithme de BioHashing transforme le vecteur à valeurs réelles de taille n représentant en un vecteur binaire appelé BioCode de taille $m \leq n$ fixé.

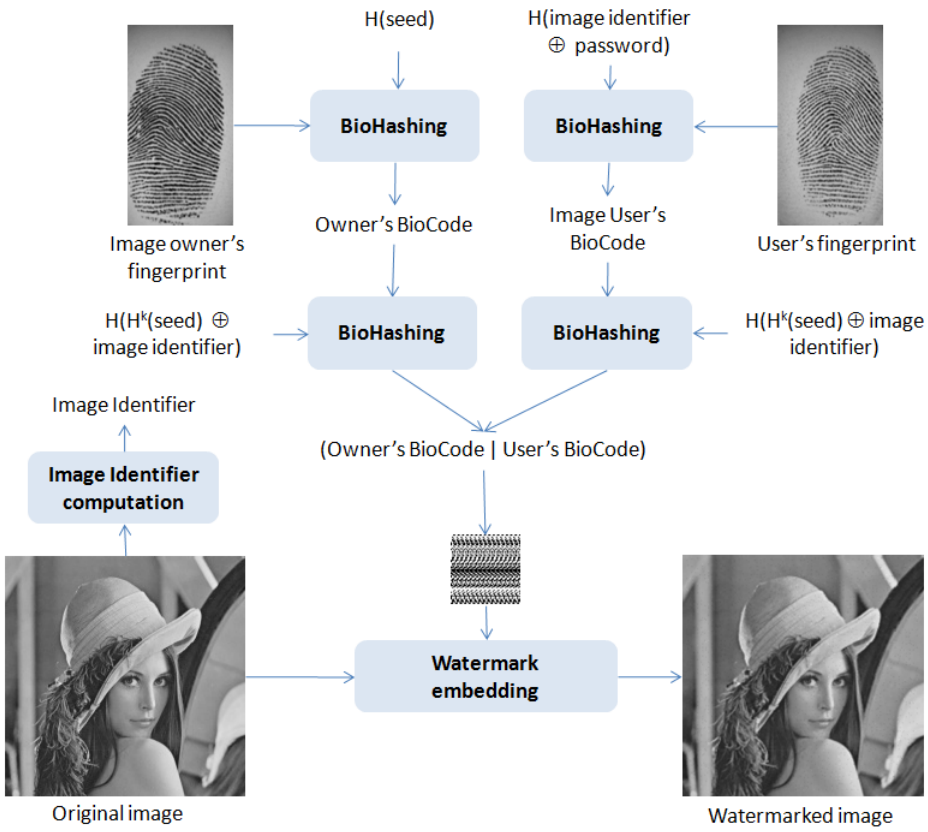
Pour permettre la révocabilité, la marque contient un biocode obtenu à partir d'une empreinte digitale et d'un aléa sur lesquels on applique le biohashing. L'algorithme de biohashing permet une transformation isométrique non inversible des données biométriques, ce qui permet d'obtenir plusieurs biocodes pour une même personne non corrélés et révocables. La figure 3.9 expose le premier schéma qui montre comment identifier une personne à l'aide de son biocode. Ce schéma est générique, il n'est pas dédié à une application particulière.

FIGURE 3.9 – Insertion du biocode dans une image



La dernière version de notre schéma (figure 3.10) produit un tel biocode à la fois du côté du propriétaire et de l'utilisateur et c'est la concaténation de ces deux biocodes qui est insérée dans l'image. On peut noter deux innovations concernant ce schéma. Premièrement, l'aléa utilisé (qui est combiné à l'identifiant de l'image) constitue un engagement biométrique qui joue un rôle similaire aux engagements que l'on trouve en cryptographie (par exemple, pour les protocoles zero-knowledge). Deuxièmement, le schéma ne nécessite pas de tiers de confiance. Le propriétaire peut directement saisir un juge s'il trouve une image sur un site d'une personne à qui il n'a pas vendu cette image.

FIGURE 3.10 – Insertion du biocode du vendeur et du client



Ce travail a déjà fait l'objet de plusieurs publications [82, 81, **?**], mais certaines améliorations peuvent être apportées. D'une part, le code correcteur utilisé est un simple code à répétitions, un code plus élaboré renforcerait ce schéma. D'autre part, l'identification de l'utilisateur par son empreinte digitale nécessite son adhésion. Il faudrait donc s'assurer que le juge puisse l'empêcher de s'y soustraire. Une définition plus fine du cadre applicatif permettrait de vérifier si c'est légalement envisageable.

Références de la partie 3

[81] Morgan Barbier, Jean-Marie Le Bars, and Christophe Rosenberger. Image Watermarking with Biometric Data for Copyright Protection. In *10th International Conference on Availability, Reliability and Security, ARES 2015, Toulouse, France, August 24-27, 2015*, pages 618–625, 2015.

[82] Morgan Barbier and Christophe Rosenberger. Tatouage d'images avec des données biométriques révocables pour la preuve de propriété. In *SAR-SSI 14*, 2014.

[83] Cyril Bazin. *Tatouage de donnees geographiques et generalisation aux donnees devant preserver des contraintes.* PhD thesis, Université de caen Basse-Normandie, 2010.

[84] Cyril Bazin, Jean-Marie Le Bars, and Jacques Madelaine. A Blind, Fast and Robust Method for Geographical Data Watermarking. In *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*, ASIACCS '07, pages 265–272, New York, NY, USA, 2007. ACM.

[85] Cyril Bazin, Jean-Marie Le Bars, and Jacques Madelaine. A Novel Framework for Watermarking : The Data-Abstracted Approach. In *Advances in Information and Computer Security, Third International Workshop on Security, IWSEC 2008, Kagawa, Japan, November 25-27, 2008. Proceedings*, pages 201–217, 2008.

[86] Christophe Charrier, Mohamad El-Abed, and Christophe Rosenberger. *Evaluation of Biometric Systems*, chapter 7, page 22. InTech, 2012.

[87] Boris Delaunay. Sur la sphère vide. A la mémoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS*, 6 :793–800, 1934.

[88] Romain Giot, Mohamad El-Abed, Baptiste Hemery, and Christophe Rosenberger. Unconstrained keystroke dynamics authentication with shared secret. *Computers and Security*, 30(6–7) :427 – 445, 2011.

[89] David Gross-Amblard. *Tatouage des bases de données.* Hdr, Université de Bourgogne, December 2010.

[90] David Gross-Amblard. Query-preserving Watermarking of Relational Databases and xml Documents. *ACM Trans. Database Syst.*, 36(1) :3 :1–3 :24, March 2011.

[91] A. K. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1) :4–20, Jan 2004.

[92] Ari Juels and Madhu Sudan. A Fuzzy Vault Scheme. *Designs, Codes and Cryptography*, 38(2), 2006.

[93] Ari Juels and Martin Wattenberg. A Fuzzy Commitment Scheme. In *Proceedings of the 6th ACM Conference on Computer and Communications Security*, CCS '99, 1999.

[94] Julien Lafaye. On the Complexity of Obtaining Optimal Watermarking Schemes. In *IWDW*, 2007.

[95] Julien Lafaye. *Tatouage des bases de données avec préservation de contraintes*. PhD thesis, CNAM, Paris, CEDRIC Laboratory, Paris, France, 2007.

[96] Julien Lafaye, Jean Beguec, David Gross-Amblard, and Anne Ruas. Invisible Graffiti on your Buildings : Blind and Squaring-proof Watermarking of Geographical Databases. In *BDA*, 2007.

[97] Julien Lafaye, Jean Beguec, David Gross-Amblard, and Anne Ruas. Blind and squaring-resistant watermarking of vectorial building layers. *GeoInformatica*, 16 :245–279, 2012.

[98] Julien Lafaye, David Gross-Amblard, Camelia Constantin, and Meryem Guerrouani. Watermill : An Optimized Fingerprinting System for Databases under Constraints. *IEEE Trans. Knowl. Data Eng.*, 20 :532–546, 2008.

[99] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2nd edition, 2009.

[100] Ryutarou Ohbuchi, Akio Mukaiyama, and Shigeo Takahashi. A frequency domain approach to watermarking 3d shapes. *Comput. Graph. Forum*, 21(3) :373–382, 2002.

[101] Ryutarou Ohbuchi, Shigeo Takahashi, Takahiko Miyazawa, and Akio Mukaiyama. Watermarking 3d Polygonal Meshes in the Mesh Spectral Domain. In *Proceedings of Graphics Interface 2001*, pages 9–17, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.

[102] B Vibert, Christophe Charrier, Jean-Marie Le Bars, and Christophe Rosenberger. Comparative Study of Minutiae Selection Algorithms for ISO Fingerprint Templates. In *IS&T/SPIE Electronic Imaging,*, San Francisco, United States, February 2015.

[103] B. Vibert, C. Rosenberger, and A. Ninassi. Security and performance evaluation platform of biometric match on card. In *Computer and Information Technology (WCCIT), 2013 World Congress on*, pages 1–6, June 2013.

[104] Benoît Vibert, John Lebouteiller, Felix Keita, and Christophe Rosenberger. Biometric Sensor and Match-On-Card Evaluation platform. In *International Biometric Performance Testing Conference (IBPC)*, pages –, gattersburg, United States, April 2014.

[105] Benoit Vibert, Zhigang Yao, Sylvain Vernois, Jean-Marie Bars, Christophe Charrier, and Christophe Rosenberger. *Information Systems Security and Privacy :*

First International Conference, ICISSP 2015, Angers, France, February 9-11, 2015, Revised Selected Papers, chapter EvaBio a New Modular Platform to Evaluate Biometric System, pages 234–250. Springer International Publishing, Cham, 2015.

[106] Benoît Vibert, Zhigang Yao, Sylvain Vernois, Jean-Marie Le Bars, Christophe Charrier, and Christophe Rosenberger. Evabio Platform for the Evaluation Biometric System - Application to the Optimization of the Enrollment Process for Fingerprints Devices. In *ICISSP 2015 - Proceedings of the 1st International Conference on Information Systems Security and Privacy, ESEO, Angers, Loire Valley, France, 9-11 February, 2015.*, pages 329–335, 2015.

[107] Z. Yao, J. Le Bars, C. Charrier, and C. Rosenberger. Quality Assessment of Fingerprints with Minutiae Delaunay Triangulation. In *Proceedings of the 1st International Conference on Information Systems Security and Privacy*, pages 315–321, 2015.

[108] Z. Yao, J. M. Lebars, C. Charrier, and C. Rosenberger. Fingerprint quality assessment combining blind image quality, texture and minutiae features. In *International Conference on Information Systems Security and Privacy (ICISSP)*, (Angers, France), Feb. 2015.

[109] Zhigang Yao. *Digital Fingerprint Quality Assessment.* Theses, Université de Caen, July 2015.

[110] Zhigang Yao, Jean-Marie Le Bars, Christophe Charrier, and Christophe Rosenberger. A Literature Review of Fingerprint Quality Assessment and Its Evaluation. *IET journal on Biometrics*, February 2016.

Sixième partie

Projet de recherche

Introduction

La sécurité informatique couvre une large partie des domaines de l'informatique. De plus, elle offre des problématiques et des cadres applicatifs variés et elle sera amenée à jouer un rôle de plus en plus crucial pour la société.

Mon projet pour mes recherches futures porte sur l'étude et l'exploitation de l'aléatoire des données numériques avec principalement des applications dans le domaine de la sécurité informatique, notamment la cryptographie, le tatouage et la biométrie.

Je souhaite organiser un framework autour de ces thématiques en mettant en place des dispositifs qui se complètent :

1. en apportant des contributions concrètes à des problèmes clairement identifiés, spécialement par le biais de thèses académiques (allocations ministérielles) ou industrielles (dispositif Cifre, bourse région), avec des co-encadrements dans mon équipe ou hors de l'équipe (partenariat avec d'autres équipes, entreprise avec un département R& D).
2. en proposant à des post-doctorants ou des ingénieurs d'études de maintenir des boîtes à outils afin de faciliter l'expérimentation et le partage des résultats obtenus.
3. en participant à des projets de recherche (ANR, FUI...) sur des thématiques moins centrées sur ma recherche, mais avec une composante pour laquelle mon expertise sera utile.
4. en essayant de comprendre en profondeur les résultats obtenus, en formalisant les problèmes traités et en précisant les limites de ces études (identifier ce qui est atteignable, ce qui est améliorable...).

Je suis intéressé par une telle démarche de recherche formant un cercle vertueux. Je suis persuadé que le domaine de la sécurité permet une activité de recherche intégrant tous ces aspects complémentaires.

Projets de recherche dans le prolongement de mes travaux

A cours terme, j'envisage de poursuivre mes travaux parmi les études suivantes pour lesquels je suis complètement opérationnel car elles sont dans le prolongement de mes travaux d'HDR.

Partie 1 – Cacher une propriété
Pour la partie sur le noyau dans les graphes aléatoires, le fait de cacher une propriété dans une donnée me semble très intéressante (avec un protocole zero-knowledge ou un autre dispositif). Malheureusement, il manque un cadre applicatif précis. Pour imposer un nouveau protocole cryptographique, il faut soit convaincre que la méthode est nettement plus efficace que celles utilisées ou proposer de nouvelles fonctionnalités dont l'utilité est avérée. Je poursuivrais dans ce domaine si jamais une opportunité se présente. Je ne suis pas persuadé que le fait de construire et partager un large

graphe aléatoire soit facilement implémentable. La propriété peut plus simplement être insérée dans une donnée réelle (réseaux sociaux, bases de données...), ce qui nous rapproche du problème de l'insertion de la marque en tatouage.

Partie 2 – Génération aléatoire pour la cryptographie et les codes correcteurs d'erreur

Les fonctions booléennes sont vues comme des primitives cryptographiques et nous avons vu que l'énumération et la génération selon des critères cryptographiques sont très difficiles à réaliser. Je souhaiterai mieux comprendre quelles sont les critères pour lesquels ce problème demeure abordable. D'autre part, certains fonctions booléennes peuvent être vu comme des mots d'un code correcteur d'erreur. En fait, tout code de longueur 2^n, peut être représenté par un ensemble de fonctions booléennes à n variables. Par exemple, le code de Reed-Müller d'ordre r est l'ensemble des fonctions de degré algébrique inférieur ou égal r. J'aimerai étudier les liens entre ces deux domaines et voir si mes méthodes pour l'énumération et la génération aléatoire peuvent être utiles pour des problèmes sur les codes correcteurs d'erreur.

Partie 3 – Tatouage de données textuelles

Comme nous l'avons montré dans la partie 2, la méthode des sites –conçue par Bazin et qu'il a testée en profondeur sur les documents géographiques– possède une partie générique indépendante. Cette approche générique permet qu'une grande partie du programme soit développé indépendamment du type de documents envisagé. A l'opposé, l'implémentation fine : choix des sites, codage des sites, propriété sur un site, nécessite la participation d'un expert du domaine. Je souhaiterai appliquer ce travail aux documents textuels en travaillant par exemple avec des membres du GREYC travaillant sur les documents textuels. L'idée consiste à insérer la signature d'un auteur dans des textes non littéraires, typiquement des articles de journaux, des dépêches. La difficulté majeure est de déterminer dans les textes des données non corrélés pour effectuer, d'une part, le codage des sites et, d'autre part, définir une propriété d'un site.

Partie 3 – Tatouage d'applications sur mobile

La méthode des sites est peu coûteuse en temps et pourrait s'adapter pour une application online sur mobile, par exemple la recherche de services sur une carte vectorielle. La carte est tatouée par l'application et avant de rafraîchir les informations, une vérification est faite pour authentifier le propriétaire du mobile. Comme pour tout tatouage, les modifications sont opérées sur les bits de poids faible et elles seront faites de sorte de ne pas nuire au fonctionnement de l'application. La détection du tatouage aura pour objectif de vérifier à tout instant que la personne utilisant l'application est bien la personne autorisée à le faire.

Partie 3 – Triangulation de Delaunay pour la biométrie

Je souhaiterai poursuivre les études sur la triangulation de Delaunay entamées par Yao et Vibert pendant leur thèse. Bien que les études puissent paraître très différentes (évaluation de la qualité des empreintes digitales, réduction du template de minuties, création de faux templates de minuties pour s'authentifier auprès d'un MOC), il s'agit essentiellement d'identifier quelles sont les parties qui servent à identifier le plus efficacement une personne. Deux pistes me semblent particulièrement intéressantes à explorer plus en profondeur. D'une part, de savoir si les informations peuvent s'abstraire ou non de la répartition spatiale (on étudie des statistiques sur des critères portant sur les triangles comme l'aire, le périmètre ou l'angle sans utiliser l'emplacement relatif des triangles les uns par rapport aux autres). D'autre part, de définir la meilleure granularité selon les études dans la triangulation. Par exemple, Vibert a observé que la répartition des aires, périmètres et angles de la triangulation de Delaunay d'un template de minuties permettait bien de différencier cette triangulation de celle d'un ensemble de points ne provenant pas d'une empreinte digitale, mais que les répartitions restaient proches d'un template à un autre. Les regroupement des triangles par clusters semble donc nécessaire. Cependant il faut s'assurer que ces clusters sont robustes à la sélection de minuties ou aux templates de qualité non optimale (lorsque l'image de l'empreinte ne permet pas la détection de toutes les minuties).

Quantifier et caractériser l'aléatoire des données

Je compte donner une priorité sur l'étude de l'aléatoire des données numériques, abordée dans la troisième partie. J'envisage des résultats plutôt à moyen ou long terme car je dois renforcer mes compétences sur ce sujet et il reste encore du travail de formalisation afin d'identifier et lever les verrous sous-jacents.

Contrairement aux données obtenues par des générateurs aléatoires (qui sont plutôt en général des générateurs pseudo-aléatoires) pour lesquelles nous pouvons connaître leurs propriétés à partir de leur modélisation aléatoire, on ne peut extraire ou mettre en évidence l'aléatoire des données numériques qu'en effectuant des statistiques sur un type de données. Toute étude comportera donc nécessairement une partie expérimentale.

Deux thématiques me semblent particulièrement intéressantes. D'une part être capable de distinguer le type de données selon leur aléatoire et, d'autre part, d'avoir des mesures du niveau d'aléatoire des données.

Plus généralement, je souhaiterai développer un framework autour de ces études afin de formaliser à la fois les problèmes sous-jacent et mettre en perspective les résultats que l'on obtient.

Classification de l'aléatoire des données

Je me suis intéressé à cette thématique en suivant les premiers travaux de thèse de Thomas Gougeon qui est encadré par Morgan Barbier, Patrick Lacharme et Chris-

tophe Rosenberger dans notre équipe et Gildas Avoine qui est à l'IRISA à Rennes. Certaines données appelées *dumps* sont accessibles à partir de cartes de paiement (cartes, de crédit, carte de transport, passeport électroniques...). Ces dumps sont constitués de blocs de taille variables soit d'informations en clair (mais codé), soit de données chiffrées. L'objectif du travail de Gougeon est de classer les bits du dumps selon ce deux types de données afin de retrouver et exploiter les parties non chiffrées. Son approche –qui s'est avère très efficace– consiste à adapter des tests statistiques à des séquences courtes et de choisir les meilleurs combinaisons de tests à l'aide de machine learning (boosting).

Ce thème de recherche est vraiment novateur car cette classification apparaît très différente de celle que j'ai pu étudier entre les données aléatoires et celles pseudo-aléatoires.

J'aimerai travailler sur cette thématique de classifications de types de données pour les dumps, mais aussi pour d'autres données qui peuvent intervenir en sécurité informatique. Différentes approches peuvent être testées. En complexité algorithmique des données, on cherche la taille d'un programme construisant une donnée. J'aimerai adapter des études telles que l'on trouve en complexité de Kolmogorov afin de concevoir une méthode simple à mettre en œuvre.

On peut aussi modifier légèrement la donnée et regarder si les algorithmes de traitement donnent les mêmes résultats, on peut espérer une différence significative entre les types de données.

Mesurer le niveau d'aléatoire des données

Avec P. Lacharme, je devrais encadrer dès septembre 2016 une thèse sur les mots de passe (le sujet a été classé prioritaire au GREYC). La sécurité des systèmes d'information reste encore fortement liée à la robustesse des mots de passe utilisés lors de l'authentification. Les mesures de robustesse traditionnelles telles que l'entropie de Shannon ou de Rényi donnent des résultats très éloignés de ce que l'on observe lors des attaques réelles. L'objectif de la thèse est de mesurer finement la robustesse des mots de passe crées par les utilisateurs. Le doctorant explorera divers pistes : modélisation probabiliste (modèles de Markov, méthodes de Monte-Carlo...), attaques réelles (logiciels john the ripper, hashcat...), dictionnaires contextuels, complexité algorithmique (complexité de Kolmogorov). On peut envisager plusieurs mesures selon le contexte d'attaque : attaque sur une personne, sur une base de données de mots de passe, connaissance d'informations sur la personne (création d'un dictionnaire contextuel). La question qui m'intrigue le plus est de savoir s'il est illusoire de demander à l'utilisateur de construire un mot de passe facile à mémoriser et difficile à trouver. S'il s'avère nécessaire de prendre des mots de passe aléatoires, d'autres problèmes surgiront : comment vérifier que le mot de passe est bien aléatoire ? Comment stocker éventuellement ses mots de passe si l'on n'arrive pas à le mémoriser ?

La robustesse des mots de passe forme un bon exemple d'étude de la mesure de l'aléatoire, mais j'espère avoir aussi la possibilité d'étendre ce travail à d'autres sujets.

Table des figures

Résumé

L'objectif de cette présentation est de montrer le rôle central de l'aléatoire dans des domaines de recherche en combinatoire, algorithmique et complexité. Nous verrons également qu'il intervient dans des domaines d'application comme la cryptographie, le tatouage et la biométrie. Selon les études proposées, ce sont des aspects différents de l'aléatoire qui entrent en jeu comme la modélisation aléatoire et le calcul de probabilités asymptotiques, la génération aléatoire ou encore l'extraction de l'aléatoire des données.

Une première partie sera consacrée à l'étude des noyaux dans les graphes aléatoires. Un noyau est un ensemble de sommets qui vérifie deux propriétés très connues en théorie des graphes : la stabilité et la dominance, ces deux propriétés s'opposent et limitent les tailles possibles des noyaux. Nous verrons comment modifier ces tailles soit en proposant des variantes de la propriété de noyau (résultats de contrexemples de lois 0-1 en théorie des modèles finis et en logiques modales, obtention de transitions de phase), soit en changeant la distribution de probabilités (graphes sans circuit, graphes creux, graphes denses). Nous verrons quel impact que cela entraîne sur la complexité des algorithmes de recherche de noyau.

La seconde partie porte sur les classes de fonctions booléennes définies selon des propriétés utiles pour la cryptographie symétrique. L'objectif est l'énumération et la génération aléatoire uniforme de ces classes de fonctions. Nous verrons qu'il est possible d'énumérer et générer efficacement les fonctions 1-résilientes jusqu'à 8 variables. L'originalité de notre méthode – à la fois combinatoire et algorithmique – que nous avons appelée méthode des classes, a été de classifier l'ensemble des fonctions booléennes en fonction de leur écart avec les fonctions 1-résilientes.

Nous nous intéressons dans la troisième partie à l'étude de l'aléatoire des données, ce travail s'inscrit dans des co-encadrements de thèse. Il s'agit d'éviter une modélisation aléatoire difficile et peu fidèle et de déterminer les parties aléatoires de ses données. La thèse de Cyril Bazin (soutenue en 2010) propose une méthode de tatouage de données géographiques vectorielles qui est rapide, aveugle et robuste à la rotation et à la translation. La méthode consiste à introduire un biais statistique sur des petites parties du document appelés sites. Les thèses de Zhigang Yao (soutenue en 2015) et Benoît Vibert (en cours) portent sur des données biométriques –les empreintes digitales– plus précisément sur les minuties extraites de ces empreintes. Nous verrons comment mesurer la qualité d'une empreinte et comment sélectionner les minuties les plus pertinentes.

Nous proposerons finalement un projet de recherche sur la quantification et la classification de l'aléatoire des données provenant de transactions numériques.