Habilitation à diriger des recherches
soutenue par
Sophie LAPLANTE

devant L'Université Paris-Sud

Applications de la complexité de Kolmogorov à la complexité classique et quantique

Composition du jury

Rapporteurs	Harry	BUHRMAN, Professeur, CWI et Université d'Amsterdam
	Bruno	DURAND, Professeur, LIF, Université de Provence
	Alexander	SHEN, Directeur de Recherche CNRS, LIF, Université de Provence
Examinateurs	Pascal	KOIRAN, Professeur, LIP, Ecole Normale Supérieure de Lyon
	Miklos	SANTHA, Directeur de Recherche CNRS, LRI, Université Paris-Sud XI
	Uwe	SCHÖNING, Professeur, Universität Ulm
	Paul	VITÁNYI, Professeur, CWI et Université d'Amsterdam

Habilitation préparée au Laboratoire de Recherche en Informatique,
UMR CNRS Université Paris-XI 8623

9 décembre 2005

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction

1.1 La complexité du calcul

La complexité du calcul se penche sur la question des ressources nécessaires (par exemple, le temps, l'espace-mémoire, la quantité de communication) pour calculer des fonctions et décider des langages. Son objectif est de ranger les problèmes dans des classes de complexité selon leur degré de difficulté algorithmique, et plus généralement, de comprendre la structure de la hiérarchie des classes de complexité.

Les classes de complexité regroupent les problèmes qui ont un degré de difficulté semblable, où ce niveau est établi en fonction de la quantité de ressources nécessaire pour résoudre le problème dans un modèle de calcul bien spécifié. Par exemple, la classe P regroupe les problèmes qui peuvent être calculés en temps polynomial sur une machine de Turing. On peut aussi considérer d'autres ressources comme l'espace, la communication, et ainsi de suite. Les modèles de calcul peuvent être déterministes, nondéterministes, probabilistes, quantiques, ils peuvent être les arbres de décision, des circuits booléens, etc.

Si les premiers modèles théoriques de calcul remontent aux années 1930 à 1950 avec machines de Turing [Tur36] les circuits booléens [Sha49], et le lambda calcul de Church [Chu41], l'idée de prendre en compte le temps de calcul et de regrouper les problèmes dans des classes de complexité en fonction de la quantité asymptotique de ressources nécessaires pour exécuter un calcul est attribuable à Hartmanis et Stearns en 1983 [HS65]. Depuis l'énoncé du problème $P \stackrel{?}{=} NP$ par Edmonds [Edm65], la question de séparer ou de montrer l'égalité de classes de complexité occupe toute la communauté de complexité, et malgré des progrès importants et le développement de techniques extrêmement sophistiquées on est encore loin de comprendre la structure exacte de la hiérarchie des classes de complexité.

Depuis les années 1960, les seules séparations de classes de complexité que l'on connaisse sont prouvées essentiellement par des techniques de diagonalisation. Par exemple, on sait que si pour deux fonctions "constructibles" $t1(n), t_2(n)$, si $\lim_{n\to\infty} \frac{t_1(n) \log t_1(n)}{t_2(n)} = 0$, alors il existe une fonction qui soit calculable en temps $t_2(n)$ mais pas en temps de $t_1(n)$, ce qui donne par exemple $P \neq EXP$. On connaît aussi des résultats semblables pour l'espace [HIS65] et pour le calcul nondéterministe [Iba72, Sze88, Coo73]. Quant aux résultats montrant que deux classes bien connues sont égales, on en connaît tout aussi peu. On peut citer $NL = coNL$ [Imm88, Sze88] et plus

récemment $SL = L$ [Tri05, Rei05], pour les classes de complexité en temps logarithmique, et $IP = PSPACE$, qui dit que tout problème calculable en espace polynomial possède une preuve interactive et vice versa, ainsi que $MIP = NEXP$, un résultat semblable pour les preuves interactives à plusieurs prouveurs.

En dehors de la question centrale $P \stackrel{?}{=} NP$, plusieurs questions fondamentales demeurent irrésolues. Une des plus importantes est celle de trouver une fonction explicite qui nécessite des circuits booléens de taille exponentielle. On sait par comptage que de telles fonctions existent, mais à ce jour, on ne sait même pas en trouver pour lesquelles on puisse prouver qu'une taille de plus que $5n$ soit nécessaire [LR01, IM02]. Cette question est liée de près à la question de savoir si $BPP = P$, c'est-à-dire si tout algorithme probabiliste à erreur bornée peut se dispenser d'aléa. Ce domaine est un des plus actifs en complexité et on peut espérer qu'une percée importante soit faite dans la prochaine décennie.

1.2 Survol des principales notions

Plusieurs notions récurrentes apparaîssent dans mes travaux. La première est la complexité de Kolmogorov, qui est l'outil principal que j'ai utilisé pour démontrer des bornes inférieures dans plusieurs modèles de calcul. La seconde est le calcul quantique, modèle de calcul auquel je me suis particulièrement intéressée dans les dernières années. Finalement, la complexité de la communication apparaît dans plusieurs travaux, car pour plusieurs modèles de calcul, il suffit de prouver une borne inférieure en complexité de la communication pour obtenir la borne souhaitée dans le modèle d'origine.

1.2.1 Complexité de Kolmogorov

La complexité de Kolmogorov permet de mesurer la quantité d'"aléa" contenue de façon inhérente dans une chaîne de symboles, sans que celle-ci soit associée à une distribution de probabilité ou à une source aléatoire. On mesure la complexité de Kolmogorov par la longueur du plus court programme qui imprime cette chaîne. Ainsi, une longue chaîne de zéros peut être imprimée avec un programme très court par rapport à la longueur de cette chaîne : il suffit de spécifier le nombre de zéros. Une telle chaîne contient peu d'"aléa". Par contre, si on considère une longue chaîne dont chaque symbole est tiré au hasard, (par exemple, 1001100100010010010110100101010...) on peut s'attendre à ce qu'un programme qui l'imprime doive être au moins aussi long que cette chaîne, qu'il doive en quelque sorte contenir explicitement la séquence de symboles qui la compose. La théorie de la probabilité nous dit que ces deux chaînes sont équiprobables si elles sont issues d'une distribution uniforme. Pourtant notre intuition nous dit que la chaîne de zéros a l'air moins aléatoire que l'autre. La complexité de Kolmogorov permet de concrétiser cette intuition avec une définition formelle de ce qu'est l'aléa d'une chaîne fixe quelconque. Cette notion peut également être vue comme un raffinement de la théorie de l'information ; en effet, l'entropie est égale, à une constante près, à la moyenne pondérée des complexités de Kolmogorov, à condition que la distribution sous-jacente soit calculable.

Plus précisément, la complexité de Kolmogorov se définit comme suit.

Définition 1. *Soit M une machine de Turing, et x, y des chaînes finies.*

1. *La* complexité de Kolmogorov de x étant donné y par rapport à M, *notée* $\mathsf{C}_M(x|y)$ *est définie ainsi :*
$$\mathsf{C}_M(x|y) = \min(|P| \text{ tel que } M(P,y) = x).$$
2. *Un ensemble de chaînes est appelé* sans préfixe *si aucune chaîne dans l'ensemble n'est préfixe d'une autre.*
3. *La* complexité de Kolmogorov préfixe de x étant donné y par rapport à M, *notée* $\mathsf{K}_M(x|y)$ *est définie comme suit :*
$$\mathsf{K}_M(x|y) = \min(|p| \text{ tel que } M(p,y) = x),$$
où p est pris dans un ensemble sans préfixe.

Dans la suite, M représentera une machine de Turing universelle fixe. On écrira C et K plutôt que C_M et K_M. Quand y est la chaîne vide, on écrit $\mathsf{K}(x)$ plutôt que $\mathsf{K}(x|y)$.

Pour plus de détails sur la complexité de Kolmogorov, le lecteur est invité à consulter [LV97].

1.2.2 Calcul quantique

Dans le calcul quantique, on travaille sur des "bit quantiques" plutôt que des bit binaires classiques. Ces bit peuvent être en superposition, c'est-à-dire qu'ils peuvent être "simultanément" 0 et 1, dans une proportion déterminée par des coefficients complexes. Les chaînes de n qubit sont représentées par des vecteurs unitaires dans un espace de Hilbert de dimension 2^n. Chacune des dimensions de l'espace est associée à vecteur de base noté $|i\rangle$ qui correspond à la $i$ème chaîne classique de longueur n. Un état quantique sur n qubit s'écrit $|\psi\rangle = \Sigma\alpha_i|i\rangle$ où les coefficients sont complexes et le vecteur $|\psi\rangle$ est de norme 1 (pour la norme L_2). La probabilité associée à une chaîne classique donnée s'obtient en prenant le carré du coefficient complexe correspondant. Les opérations que l'on admet sur les chaînes de qubit sont les opérateurs unitaires, tel que le dictent les lois de la physique quantique.

L'application d'opérateurs unitaires à ces états quantiques donne lieu à des phénomènes proprement quantiques comme l'interférence, qui provient du fait que les coefficients des états quantiques peuvent être négatifs, et donc il se peut que deux coefficients s'annullent. Cela permet dans certains de faire ressortir des solutions plus efficacement que classiquement. Tout l'art de l'algorithmique quantique consiste à savoir exploiter ces phénomènes. Pour prouver des bornes inférieures, en revanche, il n'est pas nécessaire de bien comprendre ces phénomènes, car on ne s'intéresse pas un algorithme particulier qui résoud le problème. Souvent il suffit de savoir que l'algorithme est composé d'opérations untaires.

Toute la problématique de la complexité classique peut se traduire dans le cadre des modèles de calcul quantique. Tout calcul quantique peut être simulé par un calcul classique, mais la simulation entraîne en général une explosion de la complexité. Pour certains modèles de complexité et certains problèmes, on sait qu'il existe une marge exponentielle entre la complexité classique et quantique. Cependant, dans plusieurs cas, cette marge est plutôt d'ordre quadratique. Encore très peu de techniques sont connues pour établir des bornes inférieures quantiques.

Pour une introduction plus complète au calcul quantique, voir [NC00].

1.2.3 Complexité de la communication

En complexité de la communication, deux joueurs ont chacun une partie de l'entrée d'une fonction qu'ils souhaitent calculer. Soit $f : \{0,1\}^N \times \{0,1\}^N \to \{0,1\}$ une telle fonction booléenne. Dans ce modèle, on compte uniquement le nombre de bit échangés entre Alice et Bob, tel que dicté par un protocole, en pire cas sur toutes les entrées de f, pour le meilleur protocole.

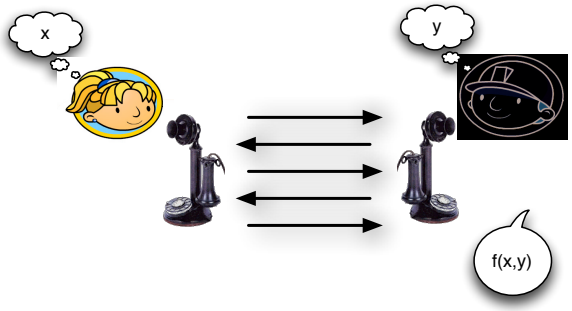

FIG. 1.1 – Complexité de la communication. Alice et Bob ont chacun un argument d'une fonction à deux arguments. Ils souhaitent calculer la fonction en minimisant la communication.

La *communication* $\mathsf{D}(P; x, y)$ dans un protocole P sur entrée x, y est le nombre de bit envoyés par Alice et par Bob. La complexité de la communication de f, notée $\mathsf{D}(f)$, est le minimum, sur tous les protocoles P, du maximum, sur toutes les entrées x, y, de $\mathsf{D}(P; x, y)$.

La complexité de la communication est utilisée pour prouver des bornes inférieures dans plusieurs modèles de calcul, notamment pour les formules booléennes (chapitre 3.2). et les OBDD (chapitre 5), On utilise aussi le cadre de la complexité de la communication pour mieux comprendre les corrélations quantiques (chapitre 6.2).

1.3 Résumé des travaux réalisés

Au cours de ma thèse j'ai travaillé principalement sur les applications de la complexité de Kolmogorov aux bornes inférieures classiques, et en particulier à la séparation de classes de complexité par ces techniques.

Depuis la fin de ma thèse, j'ai travaillé sur la complexité de Kolmogorov à ressources bornées, sur la complexité de Kolmogorov quantique, sur les bornes inférieures pour le test de propriété (*property testing*) dans le modèle des OBDD, et plus récemment, sur les techniques dites d'adversaire quantique, et la simulation des corrélations quantiques.

Mes travaux ont suivi un axe principal, d'où partent quelques branches connexes. Dans l'axe principal, j'ai cherché à développer la méthode de l'incompressibilité et l'appliquer à plusieurs modèles de calcul.

Ce parcours a commencé avec un article avec Fortnow où on a redémontré à l'aide de la complexité de Kolmogorov, la borne inférieure exponentielle de Håstad pour les circuits à profondeur

constante calculant la fonction parité (chapitre 2.1). La preuve originale de Håstad [Hås89] était très compliquée, et la nouvelle preuve qu'on a présentée, basée sur une preuve combinatoire de Razborov [Raz95], rend accessible la preuve à un public non-spécialiste.

Dans deux articles subséquents [FFLN98, BL99], on a obtenu une séparation entre les autoréductions aléatoires adaptatives et les autoréductions nonadaptatives (chapitre 2.2). Une fonction est autoréductible si on peut la calculer étant donné un accès à un oracle pour cette même fonction, avec quelques restrictions sur l'accès à l'oracle pour que le problème ne devienne pas trivial. La restriction la plus élémentaire donne lieu à la notion de cohérence : une fonction est cohérente si elle est autoréductible avec la restriction qu'on ne puisse pas obtenir de l'oracle la valeur de la fonction sur l'entrée que l'on souhaite calculer. L'autoréduction peut être probabiliste. Une fonction est autoréductible aléatoirement (*random-self-reducible*) si la distribution des questions posées à l'oracle ne dépend pas de l'entrée. On sait que toute fonction autoréductible aléatoirement est aussi cohérente moyennant un conseil (*advice*) polynomial. On a montré qu'il existe des fonctions autoréductibles aléatoirement si on permet aux questions d'être posées à l'oracle de façon adaptative, mais elle n'est même pas cohérente avec conseil polynomial si on retire la possibilité d'adaptativité.

Au centre de ces deux articles est l'idée de comptabiliser la quantité d'information contenue dans une requête à l'oracle. L'argument se présente comme un couteau à deux tranchants. Soit l'algorithme formule des requêtes "pertinentes", c'est-à-dire porteuses d'information sur l'oracle, soit les requêtes ne sont le sont pas. Dans le premier cas, le programme qui énonce la requête contient de l'information sur l'oracle. Mais comme l'algorithme est de taille constante, et on arrive à une contradiction dès que cette quantité d'information est trop grande, donc peu de requêtes peuvent être pertinentes. Dans le deuxième cas, on a peu de chances d'obtenir de l'oracle une information utile. En combinant ces deux arguments, on montre que le nombre de requêtes devra être grand pour résoudre le problème.

C'est cette technique qu'on a par la suite raffinée et généralisée, afin de l'appliquer à la complexité en requêtes quantique et probabiliste [LM04].

Espérant appliquer la méthode de l'incompressibilité à la complexité quantique, on a d'abord posé les bases de la complexité de Kolmogorov quantique, analogue de la complexité de Kolmogorov permettant de mesurer la quantité d'information quantique contenue dans une chaîne quantique (chapitre 6.1). Jusqu'à maintenant, la complexité de Kolmogorov standard a suffi pour les bornes inférieures dans les modèles de calcul qu'on a étudiés, mais on s'attend à ce que la formulation proprement quantique soit nécessaire pour d'autres modèles de calcul. Pour des raisons de continuité dans la présentation, cette thématique est présentée dans le chapitre sur le calcul quantique (chapitre 6).

En utilisant l'idée du "couteau à double tranchant", on a obtenu une technique extrêmement générale pour prouver des bornes inférieures en complexité des requêtes quantiques (chapitre 3.1). La méthode générale s'exprime de façon assez simple, et son expression fait intervenir la complexité de Kolmogorov des requêtes pertinentes, étant donné un exemplaire du problème. À notre surprise, cette technique s'est avérée généraliser toute une famille de méthodes en complexité quantique, dites techniques de l'adversaire [Amb02, Amb03, Aar04, BSS03]. De plus, alors qu'on croyait jusque là que la méthode de l'adversaire était une méthode uniquement quantique, on a pu isoler la partie combinatoire de la partie proprement quantique de la méthode, et l'appliquer ainsi au modèle des arbres de décision probabilistes et déterministes.

Par la suite, la même technique a donné lieu à une méthode générale de bornes inférieures en complexité des formules booléennes [LLS05] (chapitre 3.2). Comme dans le cas de la méthode de l'adversaire, la méthode qu'on a proposée pour les formules booléennes généralise un grand nombre de méthodes générales pour la complexité de formules booléennes.

Mes projets immédiats sont de continuer à travailler à l'extension de ces méthodes à d'autres modèles de calcul, et à l'application de ces méthodes à des fonctions particulières pour obtenir de nouvelles bornes inférieures.

En dehors de cet axe, mes travaux ont porté sur quelques autres problèmes connexes.

Complexité de Kolmogorov à ressources bornées Dans le problème de la compression des langages (chapitre 4), on cherche à donner un programme court qui fonctionne en temps polynomial pour reconnaître chacune des chaînes d'un langage. Sipser a introduit la complexité de reconnaissance (complexité CD) en temps polynomial afin de donner la première preuve que BPP est dans la hiérarchie polynomiale. On a montré

- Des programmes de longueur $2\log|A\cap\Sigma^n|$ suffisent [BFL02] ;
- Des programmes de longueur $(1+\varepsilon)\log|A\cap\Sigma^n|$ suffisent pour décrire un fraction $(1-\varepsilon)$ des chaînes de A [BFL02] ;
- Des programmes de longueur $\log|A\cap\Sigma^n|$ suffisent si on donne accès aux programmes à un oracle pour Σ_2^p [BLM99].
- Des programmes de longueur $2\log|A\cap\Sigma^n|$ sont nécessaires en général pour décrire toutes les chaînes [BLM99].

Bornes inférieures pour les OBDD Dans un article avec Lassaigne, Magniez, Peyronnet et de Rougement [LLM$^+$02], on a proposé une technique de vérification de modèle (*model checking*) basée sur le test de propriété (*property testing*) (chapitre 5). La vérification de modèle est une technique de vérification de programmes. Dans un grand nombre de cas, la représentation des données occupe un espace exponentiel, et la méthode devient inapplicable. Notre contribution a été d'introduire une technique probabiliste qui réduit la taille des données, au prix d'une perte de précision dans le résultat. Ma contribution a été la preuve de borne inférieure pour le test de la bipartition d'un graphe. Cette borne inférieure utilise la complexité de la communication. En utilisant la nouvelle technique probabiliste, l'espace devient constant, mais demeure exponentiel avec les méthodes traditionnelles, même si on relâche la condition d'exactitude.

Simulation de corrélations quantiques

Le célèbre paradoxe Einstein-Podolsky-Rosen (EPR) en physique quantique concerne l'apparente communication instantanée à distance entre deux particules intriquées. Lorsque deux particules maximalement intriquées sont partagées entre deux participants, et qu'ils mesurent indépendamment leurs particules, le résultat de leurs mesures sont corrélées alors qu'aucune communication n'a pu avoir lieu. Pour tenter d'expliquer le phénomène, ils ont postulé que les particules étaient accompagnées par des variables aléatoires cachées, qui permettaient aux participants de reproduire ces corrélations. Cependant, Bell en 1964 démontra que les corrélations ne pouvaient pas être simulées uniquement par un modèle à variable cachées et qu'il se produisait un phénomène proprement quantique qui ne pouvait pas s'expliquer avec un modèle probabiliste. Pour mieux comprendre la nature de la non-localité des expériences de type EPR, le cadre de la communication de la communication est utilisé pour quantifier la communication nécessaire pour obtenir les mêmes corrélations, avec un aléa partagé.

Plusieurs protocoles très différents les uns des autres ont été proposés pour simuler les corrélations quantiques avec communication, post-sélection, et avec des "boîtes non-locales". Dans notre travail, on a réduit le problème de simulation des corrélations à un problème d'échantillonnage distribué. En recadrant les protocoles existants dans cette perspective, on a pu redériver tous les protocoles existants dans un cadre unifié, donnant une explication beaucoup plus intuitive de pourquoi ces protocoles fonctionnaient, et on a facilement pu obtenir un nouveau protocole pour les mesures de type POVM avec communication et boîtes non-locales (chapitre 6.2).

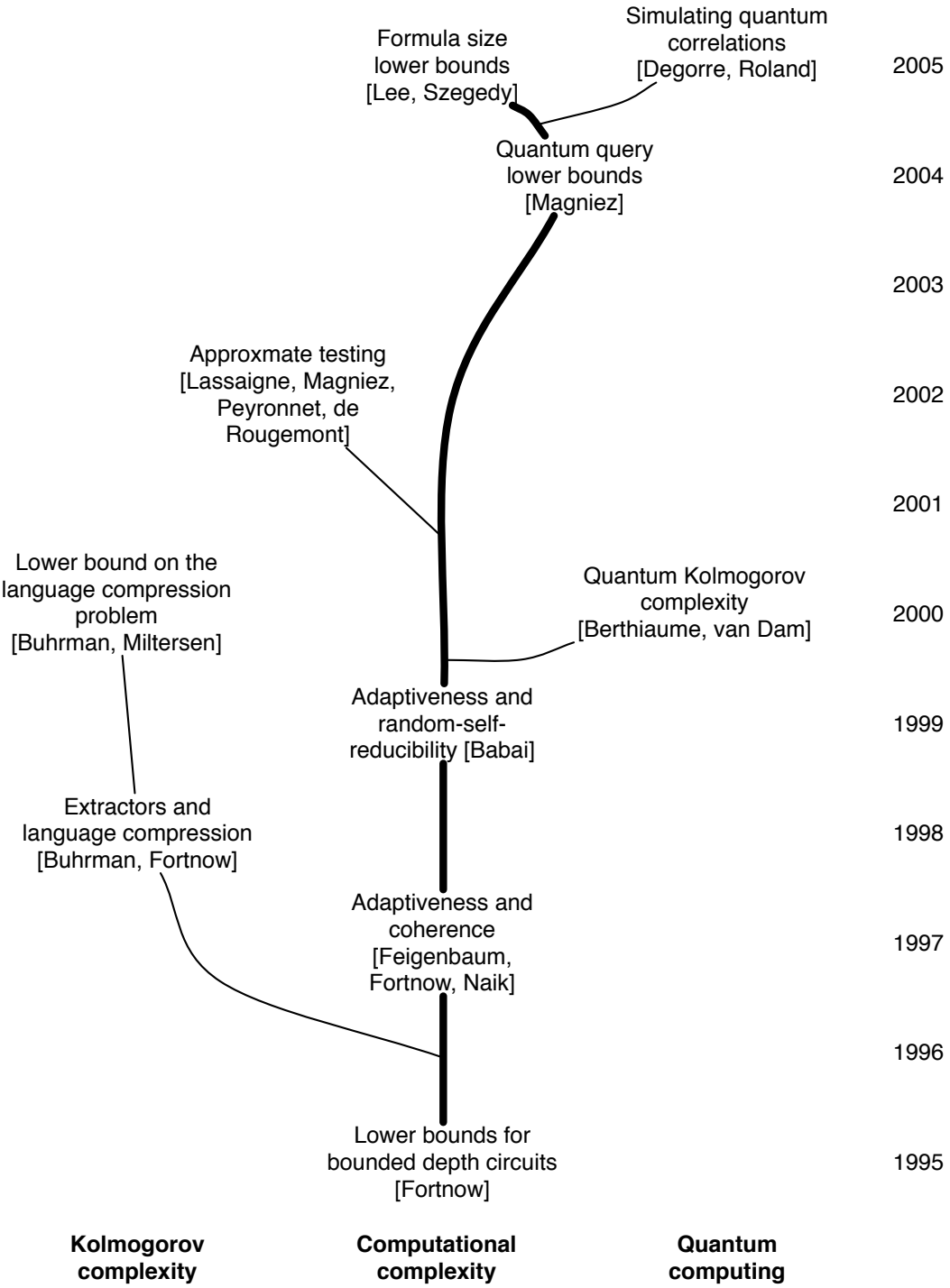

FIG. 1.2 – Résumé des travaux réalisés.

Chapitre 2

Bornes inférieures par la méthode de l'incompressibilité

2.1 Circuits à profondeur bornée

Dès 1949, Shannon introduit la notion de circuits booléens et propose la taille du plus petit circuit calculant une fonction comme mesure de complexité d'une fonction [Sha49].

Un circuit booléen est un graphe orienté sans cycle, dont les sources (sommets de degré entrant 0) sont étiquetées par des variables $x_1, \cdots, x_n$, appelées les entrées du circuit, et un puits unique (sommet de degré sortant 0) qui représente la sortie. Les autres sommets sont des portes logiques, étiquetées par $\wedge, \vee, \neg$. La profondeur du circuit est la longueur du chemin le plus long d'une entrée vers la sortie. La taille du circuit est le nombre de sommets du graphe. L'entrance du circuit (*fan-in*) est le plus grand degré entrant d'une porte $\vee, \wedge$. Le circuit peut être évalué récursivement depuis les variables en remontant jusqu'à la sortie. On dit que le circuit calcule la fonction booléenne f si la valeur du circuit coïncide avec la valeur de la fonction sur toute les entrées.

Shannon observe que le nombre de fonctions booléennes sur n variables est 2^{2^n}, mais que le nombre de circuits de taille s est au plus $(2(s+2n+2)^2)^s$. Il existe donc, par simple comptage, des fonctions dont les circuits doivent être de taille exponentielle. Cependant, la question de trouver une fonction explicite dont le plus petit circuit est de taille exponentielle demeure ouvert depuis lors, et le meilleur résultat connu est de Lachish et Raz, et Iwama et Morizumi qui ont prouvé une borne inférieure de $5n$ pour une fonction explicite en 2001 [LR01, IM02], ce qui demeure très loin de la borne exponentielle souhaitée.

En plus de son importance fondamentale en complexité, trouver des bornes inférieures pour les circuits revêt une importance encore plus importante aujourd'hui depuis que le lien a été établi entre bornes inférieures sur les circuits et dérandomisation [IW97]. En effet, des nouvelles bornes inférieures en complexité des circuits peuvent améliorer les constructions de générateurs pseudo-aléatoires, et ainsi de réduire la quantité d'aléa nécessaire aux algorithmes probabilistes, ce qui pose une pierre de plus vers une preuve que $BPP = P$.

On sait qu'il est déjà très difficile de démontrer des bornes inférieures pour des modèles de calcul plus réalistes comme les machines de Turing ou les RAM, donc on pourrait espérer que ce soit plus facile pour ce modèle de calcul en apparence très simple. Cependant, mis à part des

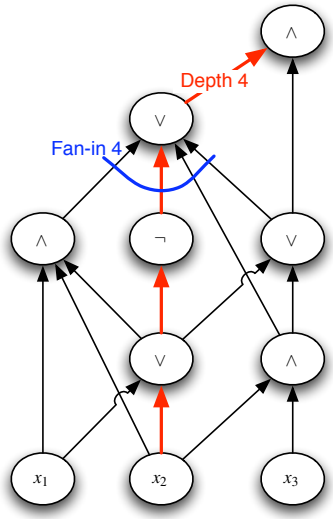

FIG. 2.1 – Circuit booléen de profondeur 4 et d'entrance 4

avancées importantes dans les années 1980 pour des familles restreintes de circuits (notamment les circuits à profondeur bornée) peu de progrès a été réalisé dans ces trente dernières années malgré le développement de techniques de plus en plus sophistiquées.

Une de ces percées fut le résultat de Håstad démontrant que tout circuit d'entrance arbitraire mais de profondeur constante pour la fonction parité a une taille exponentielle. La preuve est basée sur un lemme appelé lemme de commutation (*switching lemma*), qui dit que si on applique une restriction aléatoire aux variables d'un circuit, avec probabilité $1-p$ de laisser la variable libre, et $p/2$ de fixer la variable à 0, et $p/2$ de fixer la variable à 1, indépendamment pour chaque variable, alors on peut le réduire le circuit, avec probabilité dépendant de p, à un circuit composé d'une disjonction d'entrance arbitraire de conjonctions dont l'entrance est petite. En appliquant cette réduction de façon itérative avec les bons paramètres, on arrive à la borne inférieure exponentielle.

Le cœur de la preuve réside dans ce lemme de commutation dont la preuve utilise des arguments fins pour analyser les probabilités conditionnelles, et elle était extrêmement difficile à expliquer. Nous avons redémontré une variante du lemme de commutation dans le cadre de la complexité de Kolmogorov, éliminant ainsi les arguments probabilistes difficiles. Notre preuve s'inspire de la preuve combinatoire de Razborov [Raz95].

Plutôt que d'appliquer une restriction aléatoire avec paramètre p, on considère les restrictions qui laissent exactement l (de l'ordre de $n(1-p)$) variables libres, et dont la complexité de Kolmogorov est maximale.

Définition 2. *1. Une* restriction *est une chaîne de longueur n sur l'alphabet $\{0,1,\star\}$. Le symbole $\star$ représente le fait que la variable correspondante reste libre. Une l-restriction est une restriction qui contient exactement l occurrences du symbole $\star$. On appelle $\mathcal{R}^l$ l'ensemble des l-restrictions.*

2. Un circuit booléen est appelé t-fermé *si le circuit est composé d'une conjonction d'entrance*

arbitraire de disjonctions dont l'entrance est au plus t.

3. *Un circuit booléen est appelé* s-ouvert *si le circuit est composé d'une disjonction d'entrance arbitraire de conjonctions dont l'entrance est au plus s.*
4. *Une fonction booléenne est dite* t-fermée *(respectivement* s-ouverte*) si elle possède un circuit t-fermé (respectivement, s-ouvert).*
5. *La* restriction de f à ρ*, noté $f|_\rho$ consiste à fixer les variables à* 0 *et* 1 *respectivement lorsque le symbole de la restriction correspondant à la variable est* 0 *ou* 1*.*

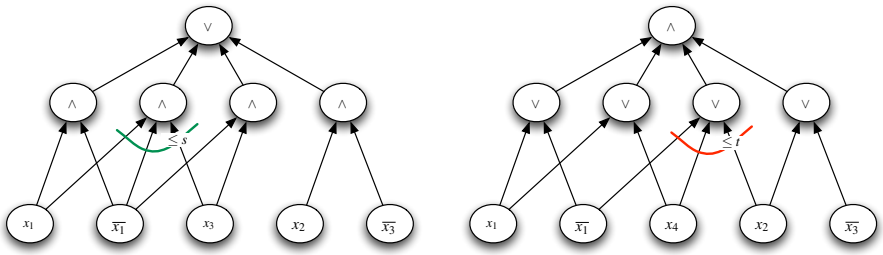

FIG. 2.2 – À gauche, un circuit s-ouvert (OU de ET), et à droite, un circuit t-fermé (ET de OU).

Dans le nouveau lemme, on montre que si une restriction a une complexité de Kolmogorov suffisante, alors le circuit se réduit à un circuit composé d'une disjonction d'entrance arbitraire de conjonctions dont l'entrance est petite.

Lemme 1 (Lemme de commutation)**.** *Soit f une fonction booléenne t-fermée sur n variables , s, l deux entiers $s < l < n, n \geq 2l - s$, et une restriction $\rho \in \mathcal{R}^l$. Si*

$$C(\rho|f, n, l, s, t) \geq \log + n - l + s\log(8t) + O(1)$$

alors $f|_\rho$ est s-ouverte.

La preuve, quoiqu'un peu compliquée, est essentiellement algorithmique et les principaux outils mathématiques utilisés sont des techniques de dénombrement élémentaires. On peut ensuite soit redémontrer le switching lemma original à partir de celui-ci, soit l'utiliser de façon inductive pour obtenir la borne inférieure.

Théorème 1. *Si un circuit de profondeur k a au plus $2^{(1/7)^{k/k-1}n^{1/k-1}}$ portes, alors il ne peut pas calculer la fonction parité correctement.*

2.2 Séparation de classes pour les autoréductions

Avec Feigenbaum, Fortnow, Naik, puis avec Babai, on a étudié les classes de fonctions autoréductibles, et en particulier les fonctions aléatoirement autoréductibles (*random-self-reducible*) en k rondes [FFLN98, BL99]. L'autoréductibilité aléatoire est une propriété fondamentale à la base

d'un grand nombre de protocoles cryptographiques, les PCP (*probabilistically checkable proofs*), les auto-testeurs et auto-correcteurs.

Une fonction f est autoréductible aléatoirement si pour tout x, il est possible de calculer $f(x)$ en temps polynomial à partir d'un nombre polynomial de valeurs $f(y_i)$ choisies aléatoirement. La distribution marginale de y_i, pris individuellement, doit être indépendante de x (mais peut dépendre de la longueur de x). Par contre les y_i peuvent (et souvent, doivent) être corrélés. Les valeurs de $f(y_i)$ sont obtenues en posant les questions $y_1, y_2, \ldots$ à un oracle pour f. Si les questions peuvent être posées en plusieurs rondes, alors on dit que l'autoréduction est adaptative, et si elles doivent être posées toutes en même temps, alors l'autoréduction est non-adaptative.

Tous les problèmes complets pour les classes $\#P$ et $PSPACE$ sont autoréductible aléatoirement, mais pour les problèmes complets au $i$ème niveau de la hiérarchie polynomiale, on ne connaît pas d'autoréduction aléatoire. Cependant, on sait que s'ils étaient autoréductibles aléatoirement nonadaptativement, alors la hiérarchie s'écroulerait deux niveaux plus haut, rendant très invraisemblable cette hypothèse [FF93]. Par exemple, pour le problème SAT, complet au premier niveau de la hiérarchie (NP), on ne sait toujours pas s'il existe une autoréduction, mais si cette autoréduction était nonadaptative, alors la hiérarchie s'écroulerait au troisième niveau, c'est-à-dire $\Sigma_3^p = \Sigma_4^p = \cdots$.

Une notion plus faible d'autoréductibilité probabiliste a été proposée par Yao [Yao90] : la cohérence. Une fonction f est cohérente si on peut la calculer sur toute entrée x avec un algorithme probabiliste qui a accès à un oracle pour f, sans jamais demander à l'oracle la valeur de la fonction sur x (c'est-à-dire qu'on interdit $y_i = x$, sans quoi toute fonction serait trivialement cohérente).

On sait que toute fonction aléatoirement autoréductible possède un auto-testeur et auto-correcteur. On ne sait pas si l'inverse est vrai pour les fonctions aléatoirement autoréductibles, mais les fonctions testables sont cohérentes.

On se penche donc sur la question de savoir s'il existe une autoréduction aléatoire, et donc un auto-testeur, pour SAT. Comme on sait qu'une autoréduction aléatoire non-adaptative ferait écrouler la hiérarchie polynomiale au troisième niveau, reste à savoir si une autoréduction nonadaptative est encore vraisemblable. Si on montrait que toute autoréduction aléatoire peut être convertie en autoréduction aléatoire nonadaptative, on saurait que l'existence d'une autoréduction, quelle qu'elle soit, entraînerait l'écroulement de la hiérarchie polynomiale. Cette question est la motivation pour étudier la question de l'adaptativité dans les autoréductions aléatoires.

Les résultats qui comparent les autoréductions adaptatives aux autoréductions nonadaptatives sont résumés dans le tableau 2.2.

Beigel et Fortnow [BF92] ont montré que les fonctions aléatoirement autoréductibles sont nonuniformément cohérentes, c'est-à-dire que pour tout x, il est possible de calculer $f(x)$ en temps polynomial probabiliste, avec un conseil de longueur polynomiale (comme pour $BPP/poly$), à partir d'un nombre polynomial de valeurs $f(y_i)$, où la seule contrainte est que $y_i \neq x$. Avec Babai, Fortnow, Feigenbaum et Naik [FFLN98, BL99, Lap97] , on a démontré que quel que soit k, la classe des fonctions aléatoirement autoréductibles en k rondes contient strictement la classe des fonctions nonuniformément cohérentes avec un nombre de rondes inférieur à k. La preuve réunit des outils de complexité de Kolmogorov et d'algèbre linéaire.

Théorème 2. *Il existe une fonction f qui a une autoréduction aléatoire adaptative mais n'a pas d'autoréduction aléatoire nonadaptative.*

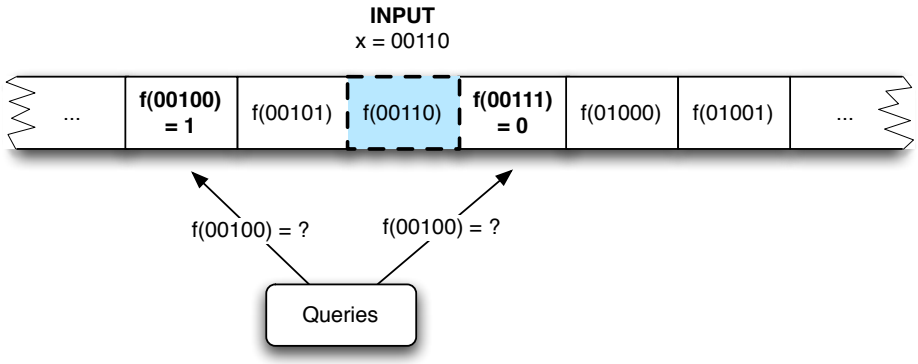

FIG. 2.3 – Une fonction f est cohérente s'il existe un algorithme probabiliste qui peut consulter un oracle pour f sans toutefois demander $f(x)$ sur entrée x. Elle est autoréductible aléatoirement (RSR) si les requêtes y_i ne révèlent pas la valeur de x. Comme l'algorithme est probabiliste, la $i$ème requête donne lieu à une distribution D_i qui doit être indépendante de l'entrée x qu'on est en train de calculer. Dans la figure, l'oracle est représenté par un ruban qui contient toutes les valeurs de la fonction, qui est consulté par l'algorithme par le biais de requêtes.

La fonction qui sépare les fonctions cohérentes des fonctions nonadaptativement cohérentes avec conseil de longueur polynomiale est représentée à la figure 2.2. Pour la preuve pour les autoréductions aléatoires, on prend l'extension multilinéaire d'une fonction de référencement de pointeurs (pointer chasing function) sur un corps fini de taille appropriée. La preuve que la fonction n'admet pas d'autoréduction aléatoire est techniquement beaucoup plus compliquée mais certaines idées de la preuve sont semblables.

On définit f sur des entrées de longueur n où n croît rapidement, de sorte que sur une entrée de longueur n, les seules questions posées à l'oracle soient de longueur n, ou beaucoup plus courtes, mais pas de longueur polynomialement plus grande. Cela simplifie les détails techniques de la preuve.

Les 2^n entrées de longueur n sont regroupées en $2^{n/2}$ blocs de $2^{n/2}$ chaînes consécutives. Sur chaque bloc, la fonction est strictement croissante, et elle passe de 0 à 1 selon un indice $0 \leq s_i < 2^{n/2}$, sauf la dernière valeur du bloc où la valeur de la fonction est la parité de s_i. On appelle cette valeur la parité du bloc. Les indices s_i sont choisis de sorte que la séquence $s_1, \cdots s_{2^{n/2}}$ soit incompressible.

Pour montrer que la fonction est cohérente, deux cas sont à prévoir. Pour calculer la valeur d'un point à l'intérieur d'un bloc, il suffit de demander la valeur des deux points voisins. S'ils sont égaux, la fonction vaut la même chose que les voisins. sinon, on consulte la valeur de la parité du bloc pour départager entre les deux valeurs possibles. Pour calculer la parité du bloc, on peut faire une fouille binaire pour trouver la valeur de s_i, ce qui peut se faire avec $\log(2^{n/2})$ requêtes. Ces requêtes sont fondamentalement adaptatives, et on voit dans la seconde partie de la preuve que cette adaptativité est nécessaire, ci-dessous.

Pour montrer que la fonction n'est pas cohérente si les requêtes sont nonadaptatives, même avec

Hypothèse	Complexité	Séparation	Référence
$NEEE \not\subseteq BPEEE$	$NP \setminus P$	RSR $\not\subseteq$ RSR nonadaptative	[FFLS94]
$NP \not\subseteq BPE$	$NP \setminus BPP$	RSR $\not\subseteq$ SR nonadaptative	[HNOS96]
–	–	cohérent $\not\subseteq$ cohérent nonadaptatif/poly	[FFLN98]
–	–	RSR $\not\subseteq$ cohérent nonadaptatif/poly	[BL99]

TAB. 2.1 – Séparations de classes d'autoréductions. La première colonne donne l'hypothèse de complexité utilisée pour obtenir la séparation, s'il y a lieu. La seconde colonne indique la complexité de la fonction témoin de la séparation. La troisième est la séparation obtenue. Le résultat de la dernière ligne implique une séparation inconditionnelle de RSR et RSR nonadaptative.

un conseil polynomial, on utilise un argument "à deux tranchants", par contradiction. Supposons que A soit un algorithme nonadaptatif qui calcule f avec un nombre polynomial de requêtes, avec un conseil de longueur polynomiale. Regardons le comportement de A sur la parité du bloc i. Supposons que A formule précisément la requête s_i. Alors, s_i peut être décrit succinctement, en spécifiant le numéro de la requête faite par A, ce qui demande un nombre de bit logarithmique dans le nombre de requêtes qu'on a supposé polynomial. Mais on a choisi les s_i incompressibles de complexité logarithmique en la taille des blocs. Ce n'est donc pas possible.

Ainsi, on peut supposer que A ne pose pas souvent la question s_i quand il calcule la parité du $i$ème bloc. S'il ne le fait pas, alors on peut simuler A pour trouver les parités de tous les blocs, sans avoir à connaître le dernier bit des s_i ! Mais c'est impossible car les s_i sont incompressibles. Quoique fasse l'algorithme, et même si on lui donne une quantité polynomiale d'information sous forme de conseil, on arrive à une contradiction, et on conclut qu'un algorithme efficace ne peut pas exister.

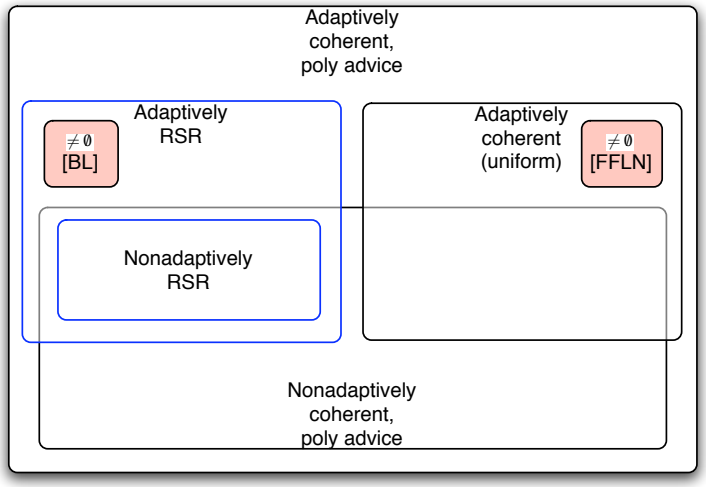

FIG. 2.4 – Séparations de classes d'autoréductions

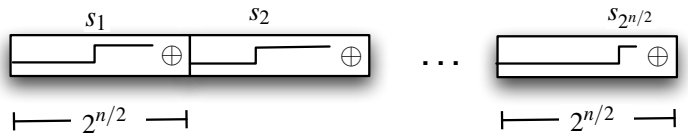

FIG. 2.5 – Fonction utilisée pour séparer les fonctions cohérentes des fonctions nonadaptativement cohérentes avec conseil polynomial. La fonction est définie sur des intervalles appelés blocs, et dans chaque intervalle, la fonction est une fonction en escalier : 0 jusqu'à un certain seuil s_i, puis 1. Sur la dernière valeur de l'intervalle (représenté par $\oplus$) la fonction a pour valeur la parité du seuil.

Chapitre 3

Bornes inférieures et la méthode de l'adversaire

Peu de méthodes permettent de prouver des bornes inférieures sur le temps nécessaire pour résoudre un problème de calcul avec un ordinateur quantique. Le seul modèle de calcul pour lequel il existe des techniques générales est l'analogue quantique des arbres de décision, appelé modèle de requêtes quantiques. Deux principales méthodes ont été développées. La première est la méthode polynomiale, qui a été inspirée de la méthode polynomiale en calcul classique. La seconde méthode, appelée méthode de l'adversaire quantique, n'avait pas de pendant classique au départ (malgré son nom), et semblait uniquement s'appliquer au modèle quantique.

Dans nos travaux, on a extrait de la méthode de l'adversaire quantique une méthode générale exprimée en termes de complexité de Kolmogorov, et qui s'applique au calcul quantique mais aussi au calcul probabiliste et déterministe. Cette formulation en termes de complexité de Kolmogorov est une extension et une adaptation des travaux sur les autoréductions aléatoires (chapitre 2.2). Cette méthode s'est avérée la plus générale des méthodes d'adversaire connues, et a permis à Špalek et Szegedy de montrer que toutes les extensions de la méthode de l'adversaire quantique sont équivalentes. Subséquemment, on a montré que la méthode se transportait à un autre modèle de calcul tout à fait classique : les formules booléennes (chapitre 3.2). Comme dans le cas de la méthode de l'adversaire, la méthode pour les formules booléennes généralise les principales méthodes générales connues pour la complexité des formules.

3.1 Bornes inférieures quantiques

3.1.1 Modèle de requêtes quantiques

Le modèle de requêtes quantiques a été introduit implicitement par Deutsch, Jozsa, Simon and Grover [Deu85, DJ92, Sim97, Gro96], et explicitement par Beals, Buhrman, Cleve, Mosca and de Wolf [BBC$^+$01]. Dans ce modèle, on comptabilise le nombre de fois que l'algorithme accède à l'entrée, modélisée sous forme d'oracle. Pour refléter la réalité du calcul quantique, l'algorithme accède à plusieurs bit de l'entrée en superposition. Pour une entrée x l'oracle est modélisé par

l'opérateur O_x, défini sur les vecteurs de base comme suit :

$$O_x(|i, z, w\rangle) = |i, x_i \oplus z, w\rangle,$$

où i représente l'indice qu'on veut consulter, z est un registre pour la réponse, et w est le contenu des autres registres du calcul.

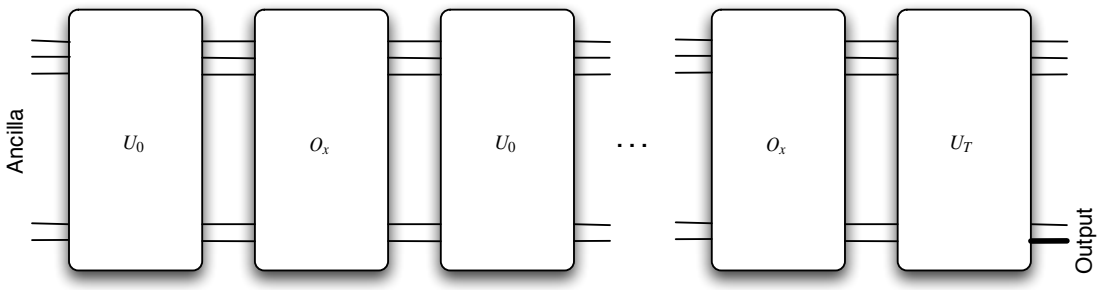

FIG. 3.1 – Modèle de requêtes quantiques

Le calcul est modélisé comme une application successive d'opérateurs unitaires arbitraires $U_0, \cdots U_T$ et de l'opérateur O_x sur un registre initialisé à l'état $|0, 0, 0\rangle$. On applique la séquence de transformations $U_0, O_x, U_1, O_x, \ldots, U_{T-1}, O_x, U_T$. Dans ce cas, on dit que l'algorithme fait T requêtes.

Un algorithme quantique *ε-calcule* une fonction f (qu'on supposera booléenne pour simplifier l'exposition) si l'observation du dernier qubit du registre de travail coïncide avec $f(x)$ avec probabilité au moins $1 - \varepsilon$ pour tout x. On note $\mathsf{QQC}(f)$ le nombre de requêtes fait par le meilleur algorithme quantique qui ε-calcule f, pour un ε fixé, par exemple $\varepsilon = 1/3$.

On peut définir un modèle de requêtes probabilistes de la même façon, en utilisant des transformations stochastiques plutôt qu'unitaires.

3.1.2 Méthode par complexité de Kolmogorov

La complexité de Kolmogorov, et plus particulièrement la méthode de l'incompressibilité, est très utile pour démontrer des bornes inférieures sur la complexité des problèmes, que ce soit en temps, en quantité de communication, d'aléa, etc. [LV97]. Cependant, la méthode demeure assez *ad hoc*, c'est-à-dire que son application dépend de la structure particulière du problème étudié. De plus, la méthode est bien connue pour les modèles classiques, mais n'avait pas été utilisée pour la complexité quantique.

Avec Magniez [LM04], on a mis de l'avant une technique de preuve très générale qui donne simultanément une borne inférieure classique (probabiliste) et une borne inférieure quantique (avec une marge d'ordre typiquement quadratique entre les deux). Il s'agit d'une généralisation de la méthode introduite par Ambainis [Amb00], l'une des méthodes les plus générales connues pour les bornes inférieures sur le nombre de requêtes en calcul quantique. Ce travail poursuit l'approche développée dans des travaux antérieurs [FFLN98, BL99]. Nous avons donné une méthode de borne

inférieure très générale, qui englobe toutes les méthodes de type "adversaire" connues pour les bornes inférieures en requêtes quantiques.

Théorème 3. *Soit f une fonction booléenne. Si A ε-calcule f avec T requêtes, alors $\forall x, y \in \Sigma^n$ tel que $f(x) \neq f(y)$:*

1. *Si A est un algorithme quantique,*

$$T \geq C \times \frac{1-2\sqrt{\varepsilon(1-\varepsilon)}}{\sum_{i:x_i \neq y_i} \sqrt{2^{-\mathsf{K}(i|x)-\mathsf{K}(i|y)}}};$$

2. *Si A est un algorithme probabiliste,*

$$T \geq C \times \frac{1-2\varepsilon}{\sum_{i:x_i \neq y_i} \min\left(2^{-\mathsf{K}(i|x)}, 2^{-\mathsf{K}(i|y)}\right)}.$$

L'idée de la preuve de ce théorème suit l'idée du couteau à deux tranchants développée au chapitre 2.2. Supposons qu'un algorithme calcule f, et qu'on l'exécute sur deux entrées x, y qui donnent lieu à deux réponses différentes. Une requête est pertinente si elle donne un poids important à des indices i où x et y diffèrent. Soit l'algorithme pose peu des requêtes pertinentes, ce qui veut dire que l'algorithme se trompera sur x ou sur y, soit il pose des questions pertinentes, dans lequel cas il suffit de donner un numéro de requête pour trouver un indice où x et y diffèrent, et la taille de la description est logarithmique dans le nombre de requêtes. On peut alors borner la complexité en requêtes en fonction de la complexité des indices pertinents, sachant x ou sachant y.

Le même raisonnement peut être appliqué au calcul classique comme au calcul quantique. Ce qui diffère est l'impact d'une requête sur le progrès réalisé par l'algorithme.

La preuve se fait en deux étapes. Dans un premier temps on évalue, pour chaque modèle de calcul, l'impact d'une requête sur l'avancement de l'algorithme.

Lemme 2 (Lemme de divergence)**.** *Notons $\overline{p}^x(i)$ la probabilité moyenne, sur toutes les requêtes faites par l'algorithme, de poser la question i sur entrée x (la probabilité dépend du carré des amplitudes de i dans le cas d'une reqête quantique). Pour tout $x, y \in \Sigma^n$ tel que $f(x) \neq f(y)$,*

1. *Pour les algorithmes quantiques :*

$$2T \sum_{i:x_i \neq y_i} \sqrt{\overline{p}^x(i)\overline{p}^y(i)} \geq 1 - 2\sqrt{\varepsilon(1-\varepsilon)}.$$

2. *Pour les algorithmes probabilistes :*

$$2T \sum_{i:x_i \neq y_i} \min\left(\overline{p}^x(i), \overline{p}^y(i)\right) \geq 1 - 2\varepsilon.$$

Dans un deuxième temps, on montre que la complexité de Kolmogorov des indices où x et y diffèrent peuvent être évaluée en termes des probabilités de poser les requêtes pertinentes.

Lemme 3 (Information des requêtes)**.** *Il existe une constante $c \geq 0$ tel que pour tout $x \in \Sigma^n$ et toute position $i \in \{1, \ldots n\}$,*

$$\mathsf{K}(i|x) \leq \log(\tfrac{1}{\overline{p}^x(i)}) + c.$$

Dans la méthode de l'adversaire originale de Ambainis, on obtient la borne inférieure en choisissant une relation R entre des paires x, y avec $f(x) \neq f(y)$, puis en examinant le degré du graphe qui représente la relation. Avec notre analyse, on obtient une borne semblable pour le calcul probabiliste. La borne inférieure se présente comme le quotient d'un terme qui dépend de la relation en entier, par un terme qui dépend des sous-relations déterminées par les paires pour laquelle une requête est pertinente.

Cette structure est récurrente dans les autres variantes de la méthode de l'adversaire. On peut l'interpréter de la façon suivante. À la fin de l'algorithmes, toutes les paires de la relation doivent être distinguées par les requêtes, car les paires en relation donnent lieu à une valeur de f différente. On veut donc évaluer le nombre de requêtes nécessaires pour distinguer toutes les paires. Chaque sous-relation dit combien de paires une requête permet de différencier. Pour connaître le nombre de requêtes nécessaire, on prend le quotient des deux quantités. Dans toutes les méthodes de l'adversaire, on donne une notion différente de "taille" de la relation, et des sous-relations associées à une requête.

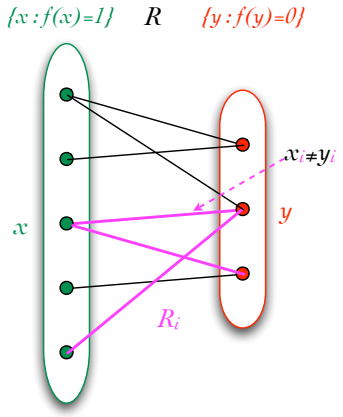

FIG. 3.2 – Dans la méthode de l'adversaire, on définit une relation R entre les entrées qui évaluent à 0 et les entrées qui évaluent à 1. La sous-relation R_i est composée des arêtes (x, y) de R telles que $x_i \neq y_i$.

Théorème 4 (Méthode d'Ambainis)**.** *Soit $R \subseteq S \times S$, une relation telle que $(x, y) \in R \Longrightarrow f(x) \neq f(y)$, et notons R_i la sous-relation de R des paires x, y telles que $x_i \neq y_i$. En considérant R comme un graphe biparti avec deux parties $X = f^{-1}(0)$ et $Y = f^{-1}(1)$, on définit l, l', m, m' comme suit.*

- *m est le plus petit degré des $x \in X$,*
- *m' est le plus petit degré des $y \in Y$,*
- *l est le plus grand degré dans R_i, sur tous les $x \in X$ et $i, 1 \leq i \leq n$.*
- *l' est le plus grand degré dans R_i, sur tous les $y \in Y$ et $i, 1 \leq i \leq n$.*

Alors

$$\mathsf{QQC}(f) = \Omega\left(\sqrt{\frac{mm'}{ll'}}\right)$$

et

$$\mathsf{RQC}(f) = \Omega\left(\max\{\sqrt{\tfrac{m}{l}}\sqrt{\tfrac{m'}{l'}}\}\right).$$

Ce théorème est corollaire du théorème principal, et de la proposition suivante qui est standard en complexité de Kolmogorov :

$$\mathsf{K}(i|x) \geq \mathsf{K}(x,y) - \mathsf{K}(x) - \mathsf{K}(y|i,x) + \mathsf{K}(i|x,y,\mathsf{K}(x,y)) - O(1).$$

Il suffit de montrer les bornes nécessaires sur les autres termes pour borner le terme $\mathsf{K}(i|x)+\mathsf{K}(i|y)$.

1. On observe que $|R| \geq \max\{m|X|, m'|Y|\}$, alors $\exists x,y\ \mathsf{K}(x,y) \geq \max\left(\log(m|X|), \log(m'|Y|)\right)$.
2. $\forall x \in X, \mathsf{K}(x) \leq \log(|X|)$ et $\mathsf{K}(y) \leq \log(|Y|)$, pour tout $y \in Y$.
3. $\forall x,y,i$ with $(x,y) \in R_i, \mathsf{K}(y|i,x) \leq \log(l)$ et de même, $\mathsf{K}(x|i,y) \leq \log(l')$.

Pour tout i tel que $x_i \neq y_i$, on obtient que

$$\mathsf{K}(i|x) = \log(\tfrac{m}{l}) + \mathsf{K}(i|x,y,\mathsf{K}(x,y)).$$

De même, on montre que $\mathsf{K}(i|y) \geq \log(\frac{m'}{l'}) + \mathsf{K}(i|x,y,\mathsf{K}(x,y))$. En appliquant le Théorème 3 et l'inégalité de Kraft sur les codes préfixe, on obtient

$$\mathsf{QQC}(f) = \Omega\left(\sqrt{\tfrac{mm'}{ll'}}\right).$$

et

$$\mathsf{RQC}(f) = \Omega\left(\max\{\sqrt{\tfrac{m}{l}}\sqrt{\tfrac{m'}{l'}}\}\right).$$

Ambainis obtient des bornes inférieures plus puissantes en introduisant des poids sur les arêtes du graphe R. Avec notre analyse, on obtient une borne semblable pour le calcul probabiliste. Dans la méthode pondérée, la borne inférieure est le quotient d'un terme qui regroupe tous les poids, par un terme qui ne dépend que des poids qui correspondent aux paires d'entrées pour lesquelles une requête est pertinente, reprenant la structure de la méthode d'Ambainis sans les poids.

Théorème 5 (Méthode d'Ambainis pondérée)**.** *Soit f une fonction booléenne, et soit une famille de poids sur les arêtes d'une relation R :*

- *Les arêtes de R ont des poids $w(x,y)$,*
- *Les arêtes de R_i ont des poids $w'(x,y,i)$.*

Pour tout x,i, posons des termes de normalisation

$$\begin{aligned} wt(x) &= \sum_y w(x,y), \quad \textit{et} \\ v(x,i) &= \sum_y w(x,y,i). \end{aligned}$$

Si $w'(x,y,i)w'(y,x,i) \geq w^2(x,y)$ pour tout x,y,i tels que $x_i \neq y_i$, alors

$$\mathsf{QQC}(f) = \Omega\left(\min_{\substack{x,y,i \\ w(x,y)\neq 0, x_i \neq y_i}} \left(\sqrt{\tfrac{wt(x)wt(y)}{v(x,i)v(y,i)}}\right)\right).$$

De plus si $w'(x,y,i), w'(y,x,i) \geq w(x,y)$ *pour tout* x,y,i *tels que* $x_i \neq y_i$*, alors*

$$\mathsf{RQC}(f) = \Omega\left(\min_{\substack{x,y,i \\ w(x,y)\neq 0, x_i \neq y_i}} \left(\max\left(\frac{wt(x)}{v(x,i)}, \frac{wt(y)}{v(y,i)}\right)\right)\right).$$

On utilise de façon essentielle deux résultats en complexité de Kolmogorov pour obtenir la méthode d'Ambainis comme corollaire du théorème principal

Proposition 1 (Théorème du codage de Shannon)**.** *Soit une source* $\mathcal{S}$ *de chaînes classiques, où* x *est émis avec probabilité* $p(x)$*. Alors pour tout encodage de* $\mathcal{S}$*, la longueur moyenne de l'encodage est au moins l'entropie de la source, c'est-à-dire, si* x *est encodé par* $c(x)$*,*

$$H(\mathcal{S}) = \sum_{x:p(x)\neq 0} p(x)\log(\tfrac{1}{p(x)}) \leq \sum_{x:p(x)\neq 0} p(x)|c(x)|.$$

On applique cette proposition à un codage optimal dans le sens de Kolmogorov, pour prouver l'existence de chaînes x dont on borne la complexité de Kolmogorov inférieurement en fonction de sa probabilité.

Lemme 4. *Soit* $\mathcal{S}$ *une source où* x *est émis avec probabilité* $p(x)$*. Alors il existe* x *pour lequel* $p(x) \neq 0$ *et* $\mathsf{K}(x|\sigma) \geq \log(\frac{1}{p(x)})$*.*

En contrepartie, le code de Shannon-Fano encode chaque mot avec $\lceil \log(\frac{1}{p(x)}) \rceil$ bit.

Proposition 2 (Code de Shannon-Fano)**.** *Pour chaque source* $\mathcal{S}$*, pour tout* x *avec* $p(x) \neq 0$*,* $\mathsf{K}(x|\mathcal{S}) \leq \log(\frac{1}{p(x)}) + O(1)$*.*

De même, on peut obtenir la méthode spectrale de Barnum, Saks ans Szegedy [BSS03] comme un corollaire du Théorème 3. Pour toute matrice carrée Γ, $\lambda(\Gamma)$ dénote la plus grande valeur propre de Γ. La borne de la méthode spectrale est le quotient de la norme spectrale de la matrice de tous les poids, par la norme spectrale de la sous-matrice des poids qui correspondent à des paires pour lesquelles la requête est pertinente (les autres poids sont mis à zéro).

Théorème 6 (Méthode spectrale de Barnum-Saks-Szegedy)**.** *Soit* $f : S \to S'$*. Soit* Γ *une matrice réelle positive* $S \times S$ *satisfaisant* $\Gamma(x,y) = 0$ *lorsque* $f(x) = f(y)$*. Pour tout* $i = 1, \ldots, n$ *soit* Γ_i *la matrice*

$$\Gamma_i(x,y) = \begin{cases} 0 & \text{si } x_i = y_i\text{ ;} \\ \Gamma(x,y) & \text{sinon.} \end{cases}$$

Alors

$$\mathsf{QQC}(f) = \Omega\left(\frac{\lambda(\Gamma)}{\max_i \lambda(\Gamma_i)}\right).$$

Se basant sur l'article avec Magniez, Špalek et Szegedy [ŠS05] ont démontré qu'en fait toutes les méthodes de type adversaire connues étaient équivalentes. Notre méthode a aussi été utilisée pour démontrer une borne inférieure sur la recherche de minima locaux [SS04].

3.2 Complexité des formules

Avec Szegedy et Lee, on a poursuivi l'étude des méthodes de l'adversaire et mis de l'avant une nouvelle technique, qui étonnamment s'applique aux formules booléennes classiques. Cette méthode généralise la plupart des méthodes générales en complexité des formules, notamment la méthode de Khrapchenko et ses généralisations, mais aussi une méthode de Håstad, basée sur les restrictions aléatoires.

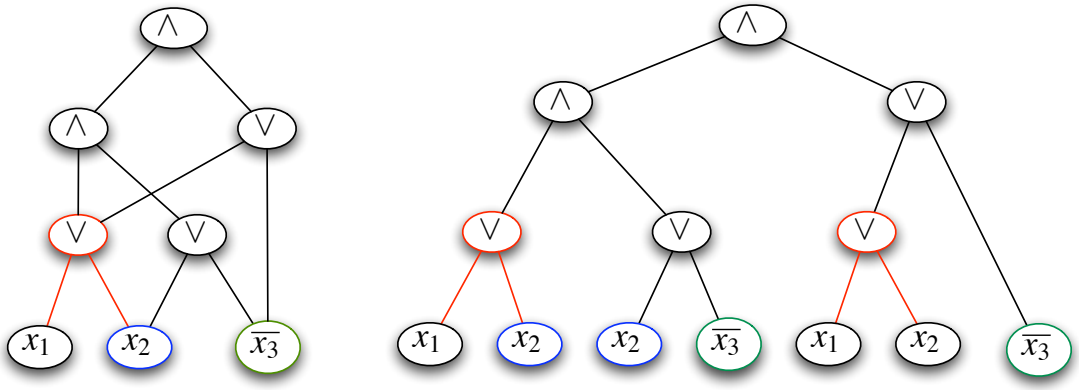

FIG. 3.3 – Circuit booléen, et formule booléenne calculant la même fonction.

Les formules booléennes sont semblables aux circuits booléens, sauf que dans une formule, le graphe sous-jacent est un arbre, et non un graphe orienté acyclique comme dans le cas des circuits. Dans les cas des circuits booléens, on ne sait pas démontrer de bornes inférieures plus que linéaires (voir chapitre 2.1). Dans le cas des formules booléennes, on peut faire un peu mieux. En effet, Håstad donne une borne inférieure de n^3 pour une fonction explicite [Hås98].

On mesure deux quantités en complexité des formules booléennes, la profondeur de la formule, notée $\mathsf{d}(f)$, et le nombre de feuilles, noté $\mathsf{L}(f)$.

Une notion importante qui intervient dans la complexité des formules booléennes est la complexité de la communication (voir chapitre 1.2.3).

La matrice de communication M_f d'une problème de communication f est la matrice $(M_f)_{x,y} = f(x,y)$. La notion de rectangles monochromatiques est importante pour les bornes inférieures sur la complexité de la communication. Un rectangle $R = X \times Y$, où X est un sous-ensemble des entrées de Alice, et Y est un sous-ensemble des entrées de Bob, est monochromatique si $\forall x, y \in R, f(x,y)$ a la même valeur.

Pour une fonction booléenne f, on définit la relation associée $R_f = \{x, y, i : f(x) \neq f(y) \text{ et } x_i \neq y_i\}$. Karchmer et Wigderson ont montré que $\mathsf{d}(f)$ est égale à la complexité de la communication $\mathsf{D}(R_f)$ [KW88]. Ils ont aussi montré que $L(f) \geq \mathsf{C}^{\mathsf{P}}(R_f)$, où C^{P} représente le nombre de rectangles dans la plus petite partition de la matrice de communication en rectangles monochromatiques. C'est un résultat classique de la complexité de la communication que $\mathsf{D}(f) \geq \log(\mathsf{C}^{\mathsf{P}}(f))$.

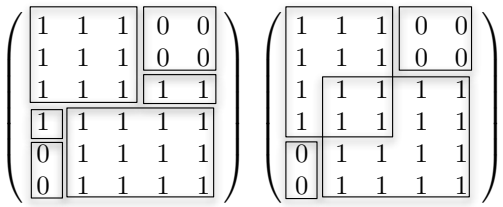

FIG. 3.4 – À gauche, une *partition* de la matrice de communication en rectangles monochromatiques. À droite, un *recouvrement* en rectangles monochromatiques.

3.2.1 Les quantités KI, sumPI et maxPI

Le fait que la méthode de l'adversaire quantique ait plusieurs formulations équivalentes indique qu'il s'agit d'une propriété combinatoire naturelle des fonctions booléennes, tout comme le degré, qu'on utilise pour prouver des bornes inférieures dans plusieurs modèles de calcul.

On choisit une des définitions formulées dans [LM04] comme définition de la quantité qu'on appelle sumPI (somme des probabilités des indices). Soit $S \subseteq \{0,1\}^n$ et $f : S \to \{0,1\}$, une fonction booléenne. On définit

$$\mathsf{sumPI}(f) = \min_{p} \max_{\substack{x,y \\ f(x)\neq f(y)}} \frac{1}{\sum_{\substack{i \\ x_i \neq y_i}} \sqrt{p_x(i)p_y(i)}}, \tag{3.1}$$

où $p = \{p_x : x \in S\}$ est une famille de distributions de probabilité indicées par les entiers dans $[n]$. On sait (chapitre 3.1) que si $\mathsf{QQC}(f)$ dénote la complexité en requêtes quantiques de f, alors $\mathsf{QQC}(f) = \Omega(\mathsf{sumPI}(f))$. On montre que de plus, $\mathsf{sumPI}^2(f)$ est une borne inférieure sur la taille de la plus petite formule booléenne pour f. Qui plus est, $\mathsf{sumPI}^2(f)$ permet de généraliser plusieurs bornes inférieures connues, notamment, la méthode de Khrapchenko et ses généralisations [Khr71, Kou93], ainsi qu'un lemme clef de Håstad [Hås98] qu'il utilise pour montrer la meilleure borne inférieure connue sur la taille de formule d'une fonction explicite.

On introduit deux autres quantités, notamment

$$\mathsf{KI}(f) = \min_{\alpha\in\Sigma^*} \max_{\substack{x,y \\ f(x)\neq f(y)}} \min_{i:x_i\neq y_i} K(i|x,\alpha) + K(i|y,\alpha).$$

Cette formulation provient des bornes inférieures probabilistes dans [LM04]. Cette formulation donne une intuition très simple pour les bornes inférieures sur la complexité des formules. On peut montrer très facilement que la profondeur d'un circuit est bornée par $\mathsf{d}(f) \geq \mathsf{KI}(f)$, en utilisant la caractérisation de la profondeur des circuits de Karchmer et Wigderson [KW88].

On introduit également une quantité proche de 2^{KI}, qu'on appelle maxPI (maximum des probabilités des indices).

$$\mathsf{maxPI}(f) = \min_{p} \max_{\substack{x,y \\ f(x)\neq f(y)}} \frac{1}{\max_{i:x_i\neq y_i} \sqrt{p_x(i)p_y(i)}}. \tag{3.2}$$

Cette expression ressemble à sumPI sauf que la somme est remplacée par le maximum. Ainsi, par définition, maxPI est plus grand que sumPI, mais le carré de maxPI reste une borne inférieure sur la

taille des formules. Cependant, maxPI n'est plus une borne inférieure sur la complexité en requêtes quantiques en général. En effet, pour la fonction f qui trouve une collision, $\mathsf{maxPI}(f) = \sqrt{n/2} \gg \mathsf{QQC}(f) = \Theta(n^{1/3})$ [AS04, BHT97], alors que $\mathsf{sumPI}(f) = 2$.

Pour voir comment ces quantités sont liées à la complexité de Kolmogorov, montrons d'abord que KI est une borne inférieure sur la profondeur d'une formule booléenne.

Théorème 7. *Pour toute fonction booléenne f, $\mathsf{KI}(f) \leq \mathsf{d}(f)$.*

Démonstration. Soit P un protocole pour R_f. Soient x, y avec $f(x) \neq f(y)$, et T_A la transcription des messages envoyés de Alice vers Bob sur entrée x, y. De même, soit T_B la transcription des messages envoyés de Bob vers Alice. Soit i la sortie du protocole, avec $x_i \neq y_i$. Pour imprimer i étant donné x, il suffit de simuler P en utilisant x pour la partie de Alice, et T_B à la place de Bob. Pour imprimer i étant donné y, on peut simuler P en utilisant y pour la partie de Bob et T_A à la place de Alice. Ainsi, $\forall x, y : f(x) \neq f(y), \exists i : x_i \neq y_i, K(i|x, \alpha) + K(i|y, \alpha) \leq |T_A| + |T_B| \leq \mathsf{D}(R_f)$, où α est une description du protocole. □

Notre principal résultat est le suivant.

Théorème 8.

$$\mathsf{sumPI}^2(f) \leq \mathsf{maxPI}^2(f) \leq \mathsf{C}^{\mathsf{P}}(R_f) \leq \mathsf{L}(f)$$

On sait déjà que $\mathsf{sumPI}^2(f) \leq \mathsf{maxPI}^2(f)$ (par définition) et que $\mathsf{C}^{\mathsf{P}}(R_f) \leq \mathsf{L}(f)$, par la caractérisation de Karchmer et Wigderson. Afin de démontrer $\mathsf{maxPI}^2(f) \leq \mathsf{C}^{\mathsf{P}}(R_f)$, on passe par une formulation différente de maxPI, qu'on obtient en dualisant son expression, vue comme un programme semidéfini. Pour introduire l'expression duale, on doit d'abord introduire une nouvelle notation. Les fonctions de sélection d'indices correspondent à une partition de $X \times Y$, qui maintient la propriété que la paire x, y peut être dans la $i$ème part seulement si $x_i \neq y_i$. On représente les parts sous forme matricielle, à raison d'une matrice par part, pour simplifier la notation ci-après.

Définition 3 (Fonctions de sélection d'indices)**.** *Soit $X{=}f^{-1}(0)$, and $Y{=}f^{-1}(1)$. Pour $i \in [n]$ on note D_i la matrice de dimension $|X|\times|Y|$ indicatrice des paires x, y avec $x_i \neq y_i$ (définie par $D_i[x, y] = 1 - \delta_{x_i,y_i}$), et E la matrice dont toutes les entrées sont 1. Une famille de matrices booléennes $\{P_i\}_{i\in n}$ sont appelées* fonctions de sélection d'indices *si*

1. *$\sum_i P_i = E$ (les P_i représentent une partition de $X \times Y$.)*
2. *$P_i \leq D_i$, terme à terme (pour tout $x \in X$, $y \in Y$ on ne sélectionne que des indices i tels que $x_i \neq y_i$, c'est-à-dire que la partition des P_i raffine le recouvrement par les D_i).*

Théorème 9 (Version spectrale de maxPI)**.** *Soit A une matrice de poids réels positifs de dimension $|X|\times|Y|$ avec $A[x, y] = 0$ quand $f(x) = f(y)$. Alors*

$$\mathsf{maxPI}(f) = \min_{\{P_i\}_i} \max_A \frac{\|A\|_2}{\max_i \|A \circ P_i\|_2},$$

où $\{P_i\}_i$ est pris parmi toutes les fonctions de sélection d'indices possibles. La notation $U{\circ}V$ représente le produit matriciel terme à terme, et $\|U\|_2$ est la norme spectrale, c'est-à-dire $\|U\|_2 = \max_{u,v:|u|=|v|=1} |u^T U v|$

Pour arriver au résultat principal, on prouve un résultat d'une forme beaucoup plus générale, où on généralise la norme spectrale par des mesures de rectangles.

Définition 4. *Une fonction* $\mu : 2^{X\times Y} \to \mathbb{R}^+$ *est appelée* mesure de rectangles *si les propriétés suivantes sont vérifiées.*

1. *(Subadditivité) Pour toute partition de rectangles,* $\mathbf{R}$ *de* $X\times Y$, $\mu(X\times Y) \leq \sum_{R\in\mathbf{R}} \mu(R)$.
2. *(Monotonicité) Pour tout rectangle* $R \subseteq X\times Y$, *et sous-ensemble* $S \subseteq X\times Y$, *si* $R \subseteq S$ *alors* $\mu(R) \leq \mu(S)$.

Cette mesure de rectangles reprend l'idée de mesurer la taille de la relation, et la taille de la sous-relation correspondant à une requête pertinente qu'on a vu au chapitre 3.1. Cependant, ici les rectangles reprennent le rôle tenu auparavant par les sous-relations.

Théorème 10. *Soit* μ *une mesure de rectangles sur* $X\times Y$, $\mathcal{S}$ *un recouvrement de* $X\times Y$ *et* $\mathbf{R}$ *une partition en rectangles de* $X\times Y$ *telle que* $\mathbf{R} \prec \mathcal{S}$ *(*$\mathbf{R}$ *raffine* $\mathcal{S}$*). Alors* $|\mathbf{R}| \geq \frac{\mu(X\times Y)}{\max_{S\in\mathcal{S}} \mu(S)}$.

Pour conclure, il ne reste plus qu'à montrer que la norme spectrale est une mesure de rectangles. Le lemme suivant, qui établit la subadditivité de la norme spectrale, constitue la clef de cette preuve.

Lemme 5. *Soit* A *une matrice de dimension* $|X|\times|Y|$ *et* $\mathbf{R}$ *une partition de* $X\times Y$. *Alors* $\|A\|_2^2 \leq \sum_{R\in\mathbf{R}} \|A_R\|_2^2$.

3.2.2 Méthode de Khrapchenko

Une des méthodes les plus générales et les plus connues pour prouver des bornes inférieures en complexité de formules est la méthode introduite par Khrapchenko en 1971 [Khr71], pour montrer une borne inférieure de $\Omega(n^2)$ pour la fonction parité. Comme la méthode d'Ambainis, la méthode est basée sur un graphe biparti dont une partie est constituée de $f^{-1}(0)$ et l'autre de $f^{-1}(1)$. La borne inférieure est le produit des degrés moyens des deux parties.

Théorème 11 (Khrapchenko)**.** *Soit* $S \subseteq \{0,1\}^n$ *and* $f : S \to \{0,1\}$. *Soit* $A \subseteq f^{-1}(0)$ *et* $B \subseteq f^{-1}(1)$. *soit* C *l'ensemble des paires* $(x,y) \in A\times B$ *avec distance de Hamming 1, c'est-à-dire,* $C = \{(x,y) \in A\times B : d_H(x,y) = 1\}$. *Alors* $\mathsf{L}(f) \geq \mathsf{sumPI}(f)^2 \geq \frac{|C|^2}{|A||B|}$.

Koutsoupias [Kou93] donne une extension spectrale de la méthode de Khrapchenko. Les poids sont 1 pour les entrées de valeur différente et 0 ailleurs.

Théorème 12 (Koutsoupias)**.** *Soient* $f : \{0,1\}^n \to \{0,1\}$, $A \subseteq f^{-1}(0)$, *et* $B \subseteq f^{-1}(1)$. *Soit* $C = \{(x,y) \in A\times B : d_H(x,y) = 1\}$. *Soit* Q *une matrice* $|B|\times|A|$, $Q[x,y] = C(x,y)$, *où* C *dénote la fonction indicatrice de l'ensemble* C. *Alors* $\mathsf{L}(f) \geq \mathsf{sumPI}(f)^2 \geq \|Q\|_2^2$.

Dans les deux cas, on peut montrer assez facilement que la borne découle de la méthode de l'adversaire. Dans le cas de la borne de Khrapchenko, il suffit de mettre des poids uniformes et utiliser la formulation d'Ambainis pondérée. Dans le cas de la borne de Koutsoupias, on utilise la formulation spectrale. Donc dans les deux cas, la méthode de l'adversaire (au carré) généralise les deux méthodes.

3.2.3 La méthode de Håstad

L'exposant de rétrécissement (*shrinkage exponent*) d'une formule booléenne est la plus petite borne supérieure γ telle que si on fixe les variables d'une formule avec probabilité $1-p$, la formule booléenne passe de taille L à la taille $p^\gamma L$. Andreev [And87] a défini une fonction f pour laquelle $\mathsf{L}(f) = n^{1+\gamma}$, donc pour obtenir de meilleures bornes inférieures sur la taille des formules, il suffit de montrer que cette quantité est grande. Håstad [Hås98] a montré que l'exposant de rétrécissement des formules est 2, donc il obtient une formule de taille au moins n^3. C'est à ce jour la meilleure borne inférieure sur la taille d'une formule explicite. À la base de cette preuve est un lemme qui donne une borne inférieure sur la taille des formules en fonction de restrictions aléatoires appliquées aux variables. Håstad montre que ce lemme généralise la méthode de Khrapchenko ; on montre qu'à son tour, le lemme de Håstad est un cas particulier de la méthode sumPI. Ce résultat est très surprenant a priori car la technique est restrictions aléatoires semble à première vue complètement différente de la méthode de l'adversaire.

Rappelons la définition des restrictions aléatoires des variables d'une fonction.

Définition 5. *Pour toute fonction $f : \{0,1\}^n \to \{0,1\}$,*

1. *Une* restriction *est une chaîne sur l'alphabet $\{0,1,\star\}^n$ où $\star$ signifie que la variable reste libre, et 0 ou 1 signifie que la variable correspondante est fixée à 0 ou à 1, respectivement.*
2. *la fonction restreinte $f|_\rho$ est la fonction f sur les variables laissées libres dans ρ, et où les autres variables sont fixées selon ρ.*
3. *R_p est la distribution sur les restrictions, où on laisse chaque variable libre avec probabilité p et on la fixe à 0 ou 1 avec probabilité $\frac{(1-p)}{2}$.*
4. *Un* filtre Δ *est un ensemble de restrictions tel que si $\rho \in \Delta$, alors ρ' obtenu en fixant une des variables restant libre dans ρ est aussi dans Δ.*
5. *Lorsque p est fixé, pour tout événement E, et tout filtre Δ, on écrira $\Pr[E|\Delta]$ plutôt que $\Pr_{\rho\in R_p}[E|\rho \in \Delta]$.*

Théorème 13 (Håstad, Lemme 4.1)**.** *Soit $f : \{0,1\}^n \to \{0,1\}$. Soit A l'événement qu'une restriction obtenue selon R_p réduit f à la constante 0, B l'événement qu'une restriction obtenue selon R_p réduit f à la constante 1, et C l'événement qu'une restriction obtenue selon R_p réduit f à un seul littéral. Alors*

$$\mathsf{L}(f) \geq \frac{Pr[C|\Delta]^2}{Pr[A|\Delta]Pr[B|\Delta]} \left(\frac{1-p}{2p}\right)^2$$

Implicite dans la preuve de Håstad est la relation suivante entre A and B. Pour tout $\rho \in C$, $f|_\rho$ se réduit à un seul littéral, c'est-à-dire que $f|_\rho = x_i$ (or $\neg x_i$ si la variable est négative). On écrit ρ^b pour dénoter ρ avec x_i fixée à b, $b \in \{0,1\}$ (ou x_i fixée à $1-b$ si la variable est négative). Alors ρ^0, ρ^1 sont les paires dans la relation utilisée dans la méthode de l'adversaire. On peut ensuite mettre des poids appropriés sur les arêtes de la relation et appliquer la formulation d'Ambainis pondérée pour conclure.

3.2.4 Méthode de Razborov

Razborov [Raz90] propose une technique de borne inférieure sur les formules booléennes qui utilise le rang.

Pour un sous-ensemble $S \subseteq X \times Y$ on définit :

$$\hat{A}_S[x,y] = A[x,y], \text{ if } (x,y) \in S \text{ and 0 otherwise.} \tag{3.3}$$

et $\alpha(\mathcal{S}) = \min\{|\mathbf{R}| : \mathbf{R} \text{ est une partition en rectangles qui raffine } \mathcal{S}\}$.

Théorème 14 (Razborov)**.** *Soit $\mathcal{S}$ un recouvrement de $X \times Y$, A une matrice non-nulle $|X| \times |Y|$ et $\mathbf{R}$ une partition en rectangles de $X \times Y$ avec $\mathbf{R} \prec \mathcal{S}$ ($\mathbf{R}$ raffine $\mathcal{S}$). Alors*

$$\max_A \frac{\mathrm{rk}(A)}{\max_{S \in \mathcal{S}} \mathrm{rk}(\hat{A}_S)} \leq \alpha(\mathcal{S}).$$

Pour obtenir ce théorème comme corollaire du théorème 10, il suffit de montrer que le rang est une mesure de rectangles.

3.2.5 Limites des méthodes d'adversaire

Un des avantages du fait que la méthode de l'adversaire ait plusieurs formulations équivalentes est le fait qu'on puisse en utiliser certaines pour bien voir les limites de la méthode. Les méthodes dans lesquelles la relation a degré 1 sont bornées par la susceptibilité $s_b(f)$ (*sensitivity*) qui est le nombre de façons qu'on peut changer une variable d'une entrée, pour passer d'une valeur de fonction b à $1-b$, au maximum sur toutes les entrées de valeur b. On note $s(f)$ le maximum entre $s_0(f)$ et $s_1(f)$.

Lemme 6. *Les bornes obtenues par la méthode de Khrapchenko, (Théorème 11), Koutsoupias (Théorème 12), et Håstad (Théorème 13) pour une fonction f est au plus $s_0(f)s_1(f) \leq s^2(f)$.*

Les limites de la méthode de l'adversaire quantique étaient déjà connues [Amb02, LM04, Sze03, Zha04, ŠS05]. Špalek and Szegedy ont donné une preuve très simple de cette limitation. Cette preuve s'applique aussi à maxPI. La limitation s'exprime en fonction de la complexité des certificats de f. Un 0-certificat est un ensemble minimal de variables qui forcent la valeur de la fonction à 0. La complexité $C_0(f)$ est la taille du plus grand 0-certificat. On définit $C_1(f)$ de façon analogue.

Lemme 7. *Soit $f : \{0,1\}^n \to \{0,1\}$ une fonction booléenne totale ou partielle. Si f est totale (respectivement, partielle) alors* $\mathsf{maxPI}(f) \leq \sqrt{C_0(f)C_1(f)}$ *(respectivement,* $\min\{\sqrt{nC_0(f)}, \sqrt{nC_1(f)}\}$*).*

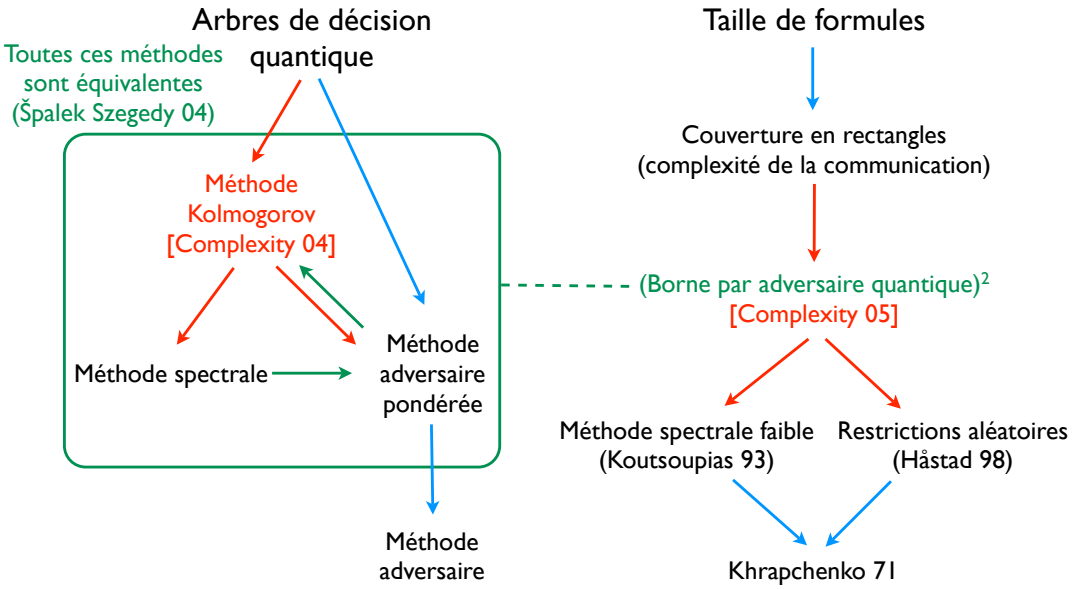

FIG. 3.5 – Méthode de l'adversaire et formules booléennes.

Chapitre 4

Complexité de Kolmogorov à ressources bornées

En complexité de Kolmogorov classique, on décrit des chaînes de caractères à l'aide de programmes, sans contrainte sur le temps de calcul. Il est utile pour plusieurs applications en complexité de se limiter aux descriptions calculables en temps polynomial.

Pour un polynôme p donné, la complexité C^p d'une chaîne $x \in \Sigma^n$ est la longueur du plus court programme p qui produit x en temps $p(|x|)$ sur l'entrée vide. La complexité CD^p de x est la longueur du plus court programme qui sur l'entrée $z \in \Sigma^n$, s'exécute en temps $p(|z|)$, et accepte uniquement x (rejette toutes les autres chaînes $z \neq x$). On dit alors que p reconnaît x. Dans les deux cas, on peut également permettre au programme p d'accéder à un oracle A.

Dans problème de la compression de langage, on fixe un langage A et on cherche à reconnaître toutes les chaînes appartenant à A avec des programmes les plus courts possibles. Les programmes peuvent accéder à l'oracle A. Par le principe des tiroirs, si $A \cap \Sigma^n$ est de taille m, alors il faudra au moins un programme de longueur $\log m$ pour reconnaître toutes les chaînes appartenant à $A \cap \Sigma^n$, car un programme ne peut servir à reconnaître qu'une seule chaîne. Si on n'impose pas de limite sur le temps d'exécution sur le programme, $\log m + O(1)$ suffit : pour accepter la $i$ème chaîne de $A \cap \Sigma^n$, il suffit de préciser i, et le programme peut parcourir l'oracle jusqu'à la $i$ème chaîne, et accepter uniquement celle-là. Il suffit de $\log m$ bit pour coder i, et le programme qui parcourt l'oracle et accepte la $i$ème est de taille constante. Cependant, cette borne supérieure n'est plus valide quand le programme doit s'exécuter en temps polynomial, car il faut en général un temps exponentiel pour parcourir l'oracle.

Sipser [Sip83] a montré qu'il était quand même possible de reconnaître A avec des programmes de taille $\log m$, si ces programmes ont accès à une chaîne auxiliaire commune de taille polynomiale.

Théorème 15. *[Sip83] Il existe un polynôme p tel que pour tout langage A il existe une chaîne r, $|r| \leq p(n)$ $\forall x \in A^{\leq n} : \mathsf{CD}^{p,A\cap\Sigma^n}(x \mid r) \leq \log(\|A \cap \Sigma^n\|) + O(\log(n))$.*

Buhrman et Fortnow ont montré [BFL02] que des programmes de taille $2\log m$ suffisent, sans cette chaîne auxiliaire.

Théorème 16. *[BFL02] Il existe un polynôme p tel que pour tout langage A et pour tout $x \in A \cap \Sigma^n$,*

$$\mathsf{CD}^{p,A\cap\Sigma^n}(x) \leq 2\log(\|A \cap \Sigma^n\|) + O(\log(n)).$$

Fraction des chaînes	Puissance de calcul	Type de borne	Borne	Référence
Toutes	$P/poly$	Supérieure	$\log \lvert A \cap \Sigma^n \rvert$	[Sip83]
Toutes	P	Supérieure	$2 \log \lvert A \cap \Sigma^n \rvert$	[BFL02]
$1 - \varepsilon$	P	Supérieure	$(1 + \varepsilon) \log \lvert A \cap \Sigma^n \rvert$	[BFL02]
Toutes	$P^{\Sigma_2^p}$	Supérieure	$\log \lvert A \cap \Sigma^n \rvert$	[BLM99]
Toutes	P	Inférieure	$2 \log \lvert A \cap \Sigma^n \rvert$	[BLM99]
Toutes	NP	Supérieure	$\log \lvert A \cap \Sigma^n \rvert$	[BLvM04]

TAB. 4.1 – Bornes pour la compression des langages.

En utilisant ces programmes de longueur $2 \log m$ pour reconnaître un ensemble, on peut obtenir une méthode pour approximer la taille de cet ensemble, méthode qu'on peut appliquer par exemple au nombre de chemins acceptants d'un calcul probabiliste à erreur bornée. En utilisant les programmes de longueur $2 \log m$ pour reconnaître les chemins acceptants, on peut redémontrer plus simplement le résultat de Gács et Sipser que BPP est dans le deuxième niveau de la hiérarchie polynomiale [Sip83].

Une autre application de la compression des langages est qu'il existe une réduction aléatoire qui transforme les formules booléennes en formules booléennes ayant au plus une assignation satisfaisante. La preuve est plus intuitive que la preuve originale de Valiant et Vazirani [VV86].

Plusieurs autres applications sont détaillées dans [BFL02].

On a longtemps cru qu'il serait possible d'améliorer la borne de $2 \log m$ en général. Pour essayer d'améliorer cette borne, on a montré qu'on peut décrire tous sauf une fraction ϵ des éléments de $A \cap \Sigma^n$ avec $\log \lvert A \cap \Sigma^n \rvert$ bit plus un terme polylogarithmique qui dépend de ϵ.

Théorème 17. *[BFL02] Il existe un polynôme p tel que pour tout langage A, tout $\epsilon < 0$ et pour tout $x \in A \cap \Sigma^n$, pour une fraction au moins $1 - \epsilon$ des x dans $A \cap \Sigma^n$,*

$$\mathsf{CD}^{p, A \cap \Sigma^n}(x) \leq \log(\lVert A \cap \Sigma^n \rVert) + (\log(\frac{n}{\epsilon}))^{O(1)}.$$

Ce résultat est une application étonnante des extracteurs, un objet combinatoire avec des propriétés d'expansion, très étudié [NTS99], utilisé dans le cadre de la génération pseudo-aléatoire et la dérandomisation. Ces résultats ont été généralisés et les rapports avec la génération pseudo-aléatoire ont été étudiés par la suite dans [BLvM04].

Cependant, on a montré qu'en général, on ne peut pas faire mieux que $2 \log m$ si on veut reconnaître toutes les chaînes [BLM99].

Théorème 18. *[BLM99] Pour tout n il existe un langage $A \subseteq \{0,1\}^n$, $\lVert A \rVert > 2^{\Omega(n)}$, et une chaîne $x_0 \in A$ telle que $\mathsf{CD}^{poly, A}(x_0) \geq 2 \log(\lVert A \rVert)$.*

La preuve emploie un lemme combinatoire qui donne une borne sur la taille de familles sans k-recouvrement (*k-cover free*) [DR82].

Soit $\mathcal{F}$ une famille d'ensembles sur un univers fini. On note $m(\mathcal{F})$ la taille de l'univers duquel les éléments sont pris, c'est-à-dire $m(\mathcal{F}) = \lVert \bigcup_{F \in \mathcal{F}} F \rVert$. Un recouvrement d'un ensemble est une famille d'ensembles dont l'union contient tous les éléments de l'ensemble recouvert.

Définition 6. *Une famille $\mathcal{F}$ est sans k-recouvrement si quels que soient les ensembles $F_0, ..., F_k \in \mathcal{F}, F_0 \not\subseteq \bigcup_{i=1}^{k} F_i$.*

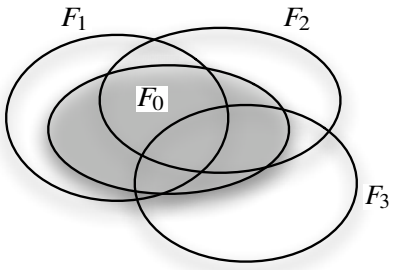

FIG. 4.1 – La famille $\mathcal{F} = \{F_0, F_1, F_2, F_3\}$ n'est pas sans 3-recouvrement, car F_0 est inclus dans $F_1 \cup F_2 \cup F_3$.

Lemme 8. *[DR82] Si $\mathcal{F}$ contient N ensembles, est sans k-recouvrement, et $N > k^3$, alors $m(\mathcal{F}) \geq \frac{k^2 \log N}{2 \log k + c}$, où c est une constante.*

Pour obtenir la borne inférieure, on montre l'existence d'un ensemble B dans lequel les chaînes sont incompressibles, qui contient forcément un sous-ensemble pour lequel des programmes de reconnaissance de longueur $2 \log m$ ne suffisent pas. Pour chaque $x \in B$, on pose

$$F_x = \{i : p_i^{\{x\}}(x) \text{ accepte}\},$$

où p_i a une taille bornée par $2 \log m$. On pose $\mathcal{F} = \bigcup_x F_x$, et on démontre que cet ensemble est sans k-recouvrement, où k est un entier plus petit que $\|B\|$, choisi judicieusement. En effet, soient $F_{x_0}, ..., F_{x_k}$ et $A = \{x_0, ..., x_k\}$, et soit p le programme $\mathsf{CD}^{t,A}$ pour x_0. On peut montrer, utilisant les propriété de B, que $i \in F_{x_0}$. Par contre, i ne peut pas être dans $\bigcup_{i=1}^{k} F_i$, car p n'accepte que x_0. Le lemme sur la taille de l'univers des familles sans k recouvrement donne la borne inférieure sur le nombre de programmes nécessaires pour les éléments du langage.

Par ailleurs, on montre que pour les ensembles "aléatoires" $R \subseteq \{0,1\}^n$ le facteur 2 n'est pas nécessaire : quel que soit $x \in R : \mathsf{CD}^{p,R} \leq \log(\|R\|) + O(\log(n))$. Notre notion de "aléatoire" est correspond à prendre une longue chaîne y de complexité de Kolmogorov maximale, et à la couper en d morceaux de longueur n qui forment l'ensemble R. Ceci semble paradoxal à prime abord car l'ensemble B duquel on a extrait un sous-ensemble A pour la borne inférieure était justement un tel ensemble.

Comme en temps polynomial, le facteur 2 est nécessaire, mais en temps exponentiel il ne l'est plus, on s'est demandé à partir de quelle puissance de calcul le facteur 2 n'était plus nécessaire. Nous avons montré que si on a une puissance de calcul correspondant à $\mathbf{NP} \cap \mathbf{coNP}$ le facteur 2 est nécessaire en général, mais qu'à partir de Σ_2^p il peut être éliminé.

Subséquemment, Buhrman, Lee et van Melkebeek ont amélioré ce résultat et ont montré, en utilisant des techniques provenant de la génération pseudo-aléatoire, que le facteur 2 n'est pas nécessaire si le programme de reconnaissance est nondéterministe [BLvM04].

Chapitre 5

Bornes inférieures pour les OBDD

La vérification de programmes est une tâche fondamentale en informatique, où la logique, la complexité et la combinatoire ont contribué de nouvelles idées qui se sont avéré utiles dans des applications pratiques. Dans un article avec Lassaigne, Magniez, Peyronnet et de Rougement [LLM$^+$02], on a proposé une technique de vérification de modèle (*model checking*) basée sur le test de propriété (*property testing*). La vérification de modèle est une technique de vérification de programmes, qui passe par une représentation compacte du système de transition du programme et d'une formule temporelle qui décrit le comportement attendu du programme. Malheureusement dans un grand nombre de cas, le système de transition a une taille exponentielle, et la méthode devient inapplicable. Il est parfois possible de définir des "abstractions" pour réduire la taille du système de transition. Notre contribution a été d'introduire des abstractions probabilistes, en appliquant les techniques du test de propriété. Nous avons montré que dans le cas de programmes pour tester la bipartition d'un graphe, les techniques standard donnent lieu à un système de transition de taille exponentielle, même en relâchant l'exactitude du test. Cette borne inférieure utilise la complexité de la communication. Cependant, on a montré qu'en appliquant une abstraction probabiliste, le système de transition passe à une taille constante.

5.1 La vérification de modèle

La vérification de modèle est une méthode algorithmique qui permet de décider si un programme, modélisé comme un système de transition, vérifie une spécification, exprimée comme une formule de la logique temporelle telle que CTL ou CTL* [CGP99]. Cette vérification peut être effectuée en temps linéaire en la taille de la spécification et du système de transition. La difficulté à appliquer cette méthode en pratique vient de l'explosion exponentielle de la taille du système de transition, en fonction de la taille des entrées admises par le programme.

L'approche appelée vérification de modèle symbolique vise à réduire la taille du système de transition en en donnant une représentation plus compacte [McM93, CGP99]. Les diagrammes de décision binaires ordonnés (OBDD) sont semblables aux arbres de décision mais le graphe sous-jacent est une graphe orienté acyclique plutôt qu'un arbre. Le diagramme est présenté par niveaux déterminés par la distance par rapport à la racine. Chaque niveau est étiqueté par une variable et chaque variable apparaît à un seul niveau. Chaque nœud interne a deux successeurs, un étiqueté 0

et l'autre 1. On évalue une entrée en parcourant le graphe de la racine vers un puits en suivant à chaque nœud le successeur dont l'étiquette correspond à la valeur de la variable du niveau auquel on se trouve.

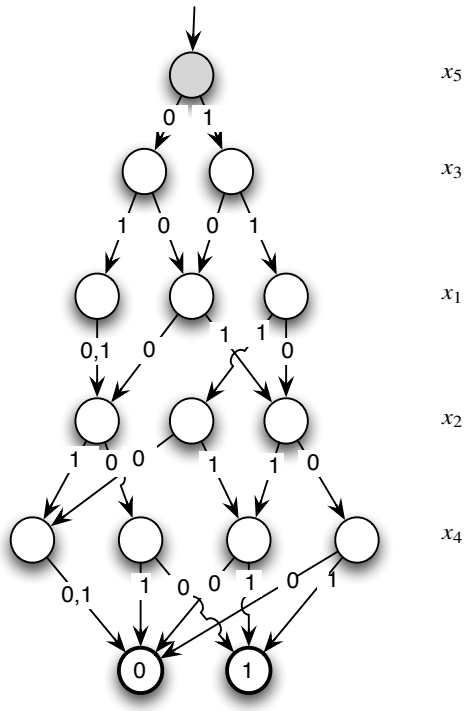

FIG. 5.1 – Un BDD, binary decision diagram, est semblable à un arbre de décision, mais le graphe sous-jacent est un graphe orienté acyclique plutôt qu'un arbre. Un BDD est ordonné (OBDD) si tous les nœuds d'un même niveau sont étiquetés par la même variable, et chaque variable n'apparaît qu'à un seul niveau.

Dans plusieurs cas pratiques, l'OBDD donne une représentation compacte du système de transition. Cependant, ce n'est pas toujours le cas, par exemple, dans le cas de la multiplication d'entiers, ou pour la bipartition d'un graphe, où la taille de l'OBDD demeure exponentielle.

La méthode de l'abstraction [CGL94] propose une solution dans certains cas lorsque les OBDD sont trop grands. Une abstraction permet de regrouper des états dans des classes d'équivalence, et ainsi un système de transition très grand est remplacé par un système de transition beaucoup plus petit qui mime le comportement du système de transition original. Dans le cas de la multiplication, l'abstraction est basée sur le théorème du reste chinois, et les entiers sont regroupés en classes d'équivalence modulo des nombres premiers. On peut ainsi vérifier la spécification sur le petit système de transition de façon plus efficace.

Pour arriver à réduire la taille des systèmes de transition pour les problèmes sur les graphes comme la bipartition, on s'inspire des techniques du test de propriété, où on peut évaluer de façon approchée si un graphe est biparti en n'examinant qu'un tout petit nombre de sommets du graphe.

5.2 Le test de propriété

Le test de propriété est une technique algorithmique qui consiste à tester si une instance d'un problème a une propriété, par exemple, si un graphe donné est biparti. Le testeur est satisfait d'une réponse imparfaite, c'est-à-dire qu'il doit accepter toute instance qui a la propriété, et qu'il doit rejeter les instances qui sont "loin" d'avoir la propriété, où "loin" est déterminé en fonction d'une mesure, par exemple, la plus petite distance de Hamming à l'instance la plus proche qui a la propriété. Le testeur peut donner une réponse arbitraire dans les autres cas. Le testeur se contente de regarder seulement une partie de l'instance, par exemple, pour déterminer la bipartition il ne regardera qu'un sous-graphe induit par un sous-ensemble de sommets du graphe. La complexité des testeurs est souvent constante.

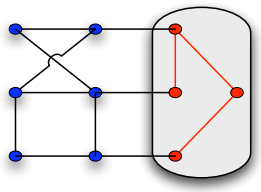

FIG. 5.2 – Le test de propriété permet de déterminer si un objet a, ou est loin d'avoir, une propriété, en faisant uniquement quelques tests locaux. Par exemple, pour tester si un graphe est biparti, il suffit de regarder un petit sous-graphe et voir s'il est biparti. Le test peut parfois se tromper, mais on est correct avec grande probabilité si le graphe est effectivement biparti, ou s'il est loin de l'être, dans le sens où il faudrait retirer un grand nombre d'arêtes pour qu'il le devienne.

Le test de propriété prend ses origines à la fin des années 1980 avec la théorie du *program checking* et du *self-testing/correcting* introduite en 1988 par Blum et Kannan [BK95], ainsi que Blum, Luby et Rubinfeld [BLR93]. Dans cette approche, on teste le bon comportement d'un programme en vérifiant une propriété de cohérence entre la sortie du programme sur des entrées choisies au hasard de façon corrélée, tout comme dans les autoréductions aléatoires (voir chapitre 2.2).

Rubinfeld and Sudan [RS96] ont formulé la notion de test de propriété qui émerge des testeurs connus. On souhaite tester si un objet possède une propriété φ en effectuant des vérifications locales sur l'objet. Le but est de distinguer entre deux cas : celui des objets qui ont la propriété, et ceux qui sont loin de l'avoir, où "loin" est paramétré par une constante ε. On souhaite que le résultat soit correct avec grande probabilité, où l'erreur est paramétrée par la constante δ. Dans le cas de plusieurs de propriétés, notamment les propriétés de graphe, il suffit de faire un nombre constant de vérifications locales lorsque ε est fixé.

Goldreich, Goldwasser, and Ron [GGR98, GR02, GR99] ont étudié les propriétés de graphe telles que la k-colorabilité. Alon, Fischer, Krivelevich, and Szegedy [AFKS00] ont montré un résultat très général pour toutes les propriétés de graphe qui s'expriment comme une formule du premier ordre de la forme $\exists\forall$.

5.3 Application du test de propriété à la vérification de modèle

L'application du test de propriété permet, dans le cadre de la vérification de modèle, de réduire considérablement la taille des systèmes de transition des programmes, et donc de réduire la complexité de la vérification.

On donne un programme simple pour tester la bipartition d'un graphe, ainsi qu'une spécification vérifiant que le programme calcule correctement la propriété de bipartition. On démontre que l'abstraction probabiliste obtenue en réduisant un graphe à un sous-graphe induit choisi au hasard conserve le comportement de la spécification vis-à-vis du programme (le système de transition réduit vérifie la spécification si le système de transition original le vérifiait.) Cela permet de confirmer l'applicabilité des abstractions probabilistes dans le cas de la bipartition car la complexité de la vérification de modèle passe d'exponentielle à constante, moyennant le relâchement de la propriété vérifiée.

5.4 Borne inférieure pour la bipartition relâchée

Hajnal, Maass et Turán ont démontré que tout OBDD pour la bipartition nécessite une taille exponentielle [HMT88], mais qu'en est-il de la bipartition relâchée (qui départage entre le cas des graphes bipartis, et ceux loin de tout graphe biparti) ? Il se pourrait qu'un OBDD pour cette propriété soit déjà beaucoup plus petit, et que la vérification de modèle symbolique soit suffisamment performante, ce qui remettrait en question l'utilité des abstractions probabilistes.

Ce n'est pas le cas ; en effet, la taille de l'OBDD demeure exponentielle. La borne inférieure est prouvée par le biais de la complexité de la communication.

Le problème de la ε-bipartition est la fonction partielle suivante.

Définition 7 (ε-bipartition)**.** *Soit $\varepsilon > 0$. La* ε-bipartition sur V *est une fonction partielle f sur l'ensemble des graphes G sur l'ensemble de sommets V :*

$$f(G) = \begin{cases} 1 & \text{si } G \text{ est biparti,} \\ 0 & \text{s'il faut retirer plus que } \varepsilon|V|^2 \text{ arêtes de } G \text{ pour qu'il devienne biparti,} \\ \perp \text{ (non-défini)} & \text{sinon.} \end{cases}$$

Un OBDD calcule une fonction partielle si sa sortie coïncide avec la valeur de la fonction lorsque celle-ci est définie.

Théorème 19. *Pour tout $\varepsilon > 0$ suffisamment petit, tout OBDD qui calcule la ε-bipartition sur des graphes à n sommets a une taille $2^{\Omega(n)}$.*

Pour prouver cette borne inférieure, on passe par la complexité de la communication. Deux joueurs ont chacun une partie de l'entrée d'une fonction qu'ils souhaitent calculer. Soit $f : \{0,1\}^N \to \{0,1\}$ la fonction booléenne qu'ils veulent calculer. Traditionnellement, on étudie des fonctions totales, mais on a été amené à étudier la complexité des fonctions partielles. Soit $A \dot\cup B$ une partition des N variables d'entrée en deux parts de même taille. L'entrée x d'Alice est formée à partir des bit A de l'entrée, et l'entrée y de Bob à partir de B. On écrit alors f comme une fonction à deux arguments $f^{A:B}(x, y)$.

En complexité de la communication, on compte le nombre de bit échangés par Alice et Bob, tel que dicté par un protocole, en pire cas sur toutes les entrées de f. Dans le modèle général, les échanges peuvent se faire en plusieurs rondes. Cependant, pour obtenir une borne sur les OBDD, on ne considère que les protocoles consistant d'un seul message allant de Alice vers Bob.

La *communication à sens unique* $\mathsf{D}_{\rightarrow}^{A:B}(P;x,y)$ dans un protocole protocole P sur entrée x,y selon la partition A,B est le nombre de bit envoyés par Alice. La *complexité de la communication à sens unique* de f suivant A,B est notée $\mathsf{D}_{\rightarrow}^{A:B}(f)$, et est égale au minimum sur tous les protocoles à sens unique P pour f, de $\max\{\mathsf{D}_{\rightarrow}^{A:B}(P;x,y)\}$, où le maximum est pris sur tous les x,y tels que $f^{A:B}(x,y)\neq\perp$.

Pour obtenir une borne pour les OBDD on doit considérer la complexité en meilleur cas sur les partitions A,B. La *complexité de la communication à sens unique pour la meilleure partition* est $\mathsf{D}_{\rightarrow}^{best}(f)=\min_{A\dot{\cup}B}\{\mathsf{D}_{\rightarrow}^{A:B}(f)\}$, où $A:B$ est pris sur l'ensemble des partitions avec $|A|=|B|\pm 1$.

La matrice de communication M_f d'une problème de communication $f^{A:B}$ est la matrice $(M_f^{A:B})_{x,y}=f^{A:B}(x,y)$.

On peut obtenir une borne inférieure sur les OBDD en prouvant une borne inférieure sur $\mathsf{D}_{\rightarrow}^{best}(f)$. La borne inférieure s'obtient grâce à une caractérisation combinatoire de la complexité de la communication à sens unique. Elle est égale au logarithme du nombre de lignes distinctes dans la matrice de communication. Comme la fonction considérée peut être partielle, if faut préciser ce qu'on entend par "distinctes". On dit que deux lignes sont distinctes sans ambiguïté s'il existe une colonne où la valeur sur une ligne est zéro et un sur l'autre ligne (autrement dit, on exclut le cas des points où la fonction n'est pas définie).

Proposition 3 ([KN97, Page 144])**.**

1. *Si un OBDD de largeur w calcule f, alors $\mathsf{D}_{\rightarrow}^{best}(f)\leq\log w$.*
2. *Soit M_f la matrice de communication de $f^{A:B}$. Alors $\mathsf{D}_{\rightarrow}^{A:B}(f)\geq\log(l)$, où l est le nombre de lignes de M_f distinctes sans ambiguïté deux à deux.*

On peut voir que la première partie est vraie en observant que si f possède un OBDD, alors on peut donner à Alice les $N/2$ premières variables dans l'ordre donné par l'OBDD, et les autres à Bob. Le protocole consiste à ce que Alice calcule l'état de l'OBDD après l'exécution de l'OBDD jusqu'à la moitié et envoie cet état à Bob en $\log(w)$ bit. Bob peut ensuite compléter l'exécution de l'OBDD et déterminer la valeur de la fonction.

Pour la deuxième partie, on peut observer que si Alice envoie le même message sur deux entrées différentes x,x' Bob donnera la même valeur à la fonction sur $f(x,y)$ ainsi que $f(x',y)$, pour tout y. Les lignes x et x' ne sont donc pas distinctes sans ambiguïté. Il doit donc y avoir autant de messages différents que de lignes de M_f deux à deux distinctes sans ambiguïté.

On introduit un problème intermédiaire pour prouver la borne inférieure sur la ε-bipartition, et on montre qu'il se réduit au problème de la ε-bipartition.

Définition 8 (Problème des demi-ensembles disjoints)**.** *Soit S un ensemble fini. Le problème des demi-ensembles disjoints est une fonction partielle g sur les sous-ensembles $T_1,T_2\subset S$ de taille $\lfloor|S|/4\rfloor$:*

$$g(T_1,T_2)=\begin{cases}1 & \text{si } T_1\cap T_2=\emptyset,\\ 0 & \text{si } |T_1\cap T_2|\geq|T_1|/2,\\ \perp & \text{sinon.}\end{cases}$$

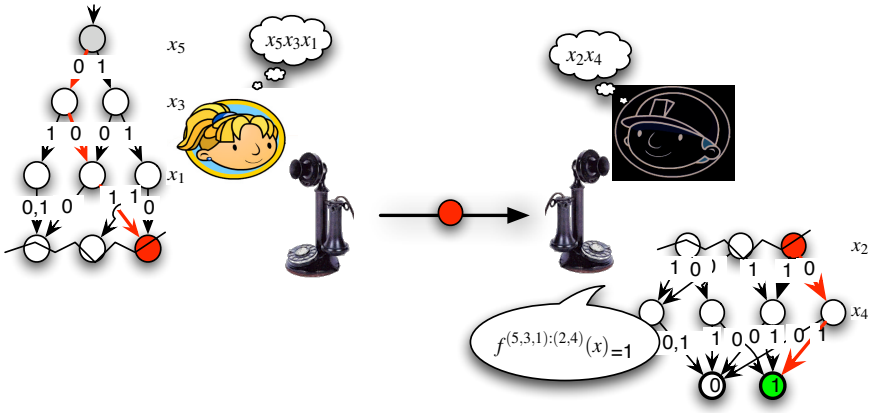

FIG. 5.3 – Alice et Bob simulent l'exécution d'un OBDD avec un protocole à sens unique avec communication bornée par le logarithme de la largeur de l'OBDD. Par conséquent, pour prouver une borne inférieure sur la taille d'un OBDD, il suffit de montrer une borne inférieure sur la complexité de la communication à sens unique.

Chaque joueur a un ensemble T_1, T_2, et leur but est de déterminer si ces ensembles sont disjoints ou s'ils ont une grande intersection. On montre que le nombre de lignes de M_g distinctes sans ambiguïté deux à deux est exponentiel. Le lemme combinatoire suivant est utilisé de façon fondamentale dans la preuve, pour identifier un grand ensemble de lignes deux à deux distinctes sans ambiguïté.

Proposition 4. *Il existe au moins* $\lfloor 2^{|S|/64} \rfloor$ *sous-ensembles* $T_i \subset S$ *de taille* $\lfloor |S|/4 \rfloor$ *tels que pour chaque paire* $T_i \neq T_j$, $|T_i \cap T_j| \leq |T_i|/2$.

Ce résultat peut se prouver par la méthode probabiliste, ou par un argument de complexité de Kolmogorov.

Lemme 9. *Soit* S *un ensemble. La matrice de communication du problème des demi-ensembles disjoints a au moins* $\lfloor 2^{|S|/64} \rfloor$ *lignes deux à deux distinctes sans ambiguïté.*

Pour compléter la preuve, il faut montrer que le problème des demi-ensembles distincts se réduit à la ε-bipartition. Pour cela, on adapte la preuve de Hajnal, Maass and Turán [HMT88]. Cette preuve utilise le lemme de régularité de Szemeredi de façon essentielle, afin d'avoir une borne inférieure qui tienne quelle que soit la répartition des variables entre Alice et Bob.

Chapitre 6

Calcul quantique

6.1 Complexité de Kolmogorov quantique

La théorie de l'information quantique, une extension de la théorie de l'information de Shannon, est bien établie depuis les années 1960. Vu les liens étroits entre complexité de Kolmogorov et théorie de l'information, le problème se pose naturellement de donner une extension de complexité de Kolmogorov pour le calcul quantique. Trois articles ont proposé des définitions de complexité de Kolmogorov quantique. Paul Vitányi [Vit00] donne une définition qui décrit la quantité d'information classique dans une chaîne quantique. La définition qu'on a proposée avec Berthiaume et van Dam [BvDL01] décrit le contenu d'information quantique dans une chaîne quantique. Gács [Gác01] approche le problème du point de vue purement abstrait des matrices de semi-densité, et cette approche lui permet d'unifier les deux approches précédentes.

La notion de complexité de Kolmogorov quantique qu'on a proposée possède, comme son pendant classique, des liens étroits avec la théorie de l'information quantique. Elle permet de donner un résultat qui implique le célèbre théorème de non-clonage quantique. Notre mesure de complexité de Kolmogorov donne un résultat quantitatif, c'est-à-dire qu'à chaque chaîne quantique est associée une valeur : plus cette valeur est grande, plus cette chaîne est difficile à cloner.

Une première tentative de définition de la complexité de Kolmogorov quantique d'un état $|\psi\rangle$ serait la longueur du plus court programme quantique qui produit $|\psi\rangle$ en sortie. Cependant, plusieurs questions fondamentales sont soulevées avant d'arriver à une véritable définition.

Quel modèle de calcul ? Malgré que le circuit quantique soit le modèle de calcul le plus utilisé pour sa simplicité, la taille d'un circuit quantique n'est pas une bonne mesure de la complexité d'un état quantique. En effet, il se peut qu'un très grand circuit quantique possède une description classique très petite. Le modèle de machine de Turing, moins élégant pour le calcul quantique, est plus approprié.

Bit ou qubit ? On peut souhaiter mesurer la quantité d'information contenue dans $|\psi\rangle$ en bit ou en qubit. Comme l'ensemble des états quantiques n'est pas dénombrable, un nombre dénombrable de programmes classiques ne peuvent pas espérer décrire exactement l'ensemble des états quantiques. Vitányi [Vit00] confronte cette apparente contradiction en admettant que certains états quantiques soient décrits avec une erreur importante. Nous prenons plutôt le parti des descriptions quantiques des états quantiques.

Description exacte ou inexacte ? On souhaite idéalement une description exacte des états quantiques. Cependant, une des propriétés de la complexité de Kolmogorov que l'on souhaite maintenir est l'invariance, qui dit qu'on peut passer d'une description par une machine universelle à une autre sans allonger la complexité par plus qu'une constante qui dépend uniquement des deux machines universelles. Or dans le cadre quantique, les machines universelles peuvent introduire une petite distorsion dans la simulation d'un calcul quantique. On doit donc se contenter de descriptions inexactes.

Si on admet des descriptions de fidélité bornée, on entre dans un scénario où pour certains états, toute la complexité réside dans la partie qui n'est pas décrite exactement, par exemple un état dont la différence avec l'état $|0\rangle$ est très petite en termes de fidélité, mais néanmoins très complexe. On choisira donc un modèle où on demande que la fidélité puisse être rendue arbitrairement grande, empruntant à la définition de schémas d'approximation provenant des algorithmes d'approximation.

Qu'est-ce qu'un programme quantique ? On dira qu'un programme quantique est n'importe quelle entrée à une machine de Turing quantique universelle. Comme on souhaite compter la quantité d'information en qubit, ce programme doit pouvoir être un état quantique. Ces programmes peuvent être vus comme des programmes binaires avec des états auxiliaires quantiques codés en dur dans le programme.

Description unique ou répétable ? Dans le cadre classique, le programme qui imprime x peut être exécuté à plusieurs reprises pour obtenir plusieurs copies de x. Dans le cadre quantique, on sait que le théorème de non-clonage interdit de dupliquer des états quantiques arbitraires exactement. En général, on ne pourra pas réexecuter une copie d'un programme quantique sans perdre la sortie d'une exécution préalable. On opte donc pour un modèle où le programme ne peut pas être réutilisé.

On mesure la fidélité entre l'état $|\varphi\rangle$ qu'on veut décrire et l'état produit $|\tilde{\varphi}\rangle$ par la fonction $\text{Fidelity}(|\varphi\rangle, |\tilde{\varphi}\rangle) = |\langle\varphi|\tilde{\varphi}\rangle|$.

On propose donc la définition suivante pour la complexité de Kolmogorov quantique, où la fidélité est paramétrée par une fonction f.

Définition 9. *(Complexité de Kolmogorov avec fidélité f) Pour toute machine de Turing quantique M et état quantique $|\varphi\rangle$ la f-complexité de Kolmogorov quantique, notée $\mathsf{QC}^f_M(|\varphi\rangle)$, est la longueur de la plus courte chaîne de qubit P tel que pour tout paramètre de fidélité k,* $\text{Fidelity}(|\varphi\rangle, M(P, 1^k)) \geq f(k)$.

On peut prouver un théorème d'invariance pour une fonction de fidélité qui tend vers 1 lorsque k tend vers l'infini.

Définition 10. *(Complexité de Kolmogorov quantique avec fidélité convergeant vers 1.) La complexité $\mathsf{QC}^{\uparrow 1}_M(|\varphi\rangle)$ est définie comme étant $\mathsf{QC}^f_M(|\varphi\rangle)$, avec $f(k) = 1 - \frac{1}{k}$.*

C'est cette version que nous adoptons comme définition de la complexité de Kolmogorov quantique. On prouve donc un théorème d'invariance pour la complexité de Kolmogorov quantique, ce qui essentiel pour établir que c'est une notion robuste.

Théorème 20. *Il existe une machine de Turing quantique universelle U telle que pour toute machine de Turing quantique M et tout état quantique $|\varphi\rangle$,*

$$\mathsf{QC}^{\uparrow 1}_U(|\varphi\rangle) \quad \leq \quad \mathsf{QC}^{\uparrow 1}_M(|\varphi\rangle) + c_M,$$

où c_M est une constante qui dépend uniquement de M.

La preuve repose sur l'existence d'une machine de Turing quantique universelle, montrée par Bernstein and Vazirani [BV97].

On peut démontrer un théorème d'incompressibilité pour les états quantiques, en utilisant des outils de la théorie de l'information quantique. On cite la version générale pour les états mixtes, qui sont des mélanges statistiques des états quantiques habituels.

On note $p|\varphi\rangle\langle\varphi| + (1-p)|\psi\rangle\langle\psi|$ le mélange de l'état $|\varphi\rangle$ avec probabilité p et $|\psi\rangle$ avec probabilité $1-p$, où $\langle\varphi|$ représente le vecteur φ conjugué et transposé. La matrice résultante s'appelle matrice de densité. L'entropie de Von Neumann est une généralisation de l'entropie de Shannon définie sur les mélanges statistiques d'états quantiques purs.

Définition 11 (Entropie de Von Neumann)**.** *L'entropie de Von Neumann d'un état mixte ρ égale $S(\rho) = \mathrm{tr}(-\rho \log \rho)$. En particulier, si on décompose ρ dans sa base de vecteurs propres φ_i avec coefficients p_i,*

$$S(\rho) \;=\; S\left(\sum_i p_i |\varphi_i\rangle\langle\varphi_i|\right) \;=\; H(p),$$

où $H(p)$ est l'entropie de Shannon pour une source $p_1, p_2, \ldots$

Théorème 21. *[Incompressibilité d'états quantiques] Pour tout ensemble d'états quantiques $\rho_1, \ldots, \rho_M$ (en général des états mixtes représentés par leur matrice de densité) tel que $\forall i, QC(\rho_i) \leq l$, l est borné inférieurement par*

$$l \;\geq\; S(\rho) - \frac{1}{M}\sum_i S(\rho_i),$$

où ρ est la matrice de densité "moyenne", $\rho = \frac{1}{M}\sum_i |\varphi_i\rangle\langle\varphi_i|$.

Pour les états quantiques purs, cela veut dire qu'il existe toujours parmi M états distincts, un dont la complexité est au moins $\log M$, tout comme en complexité de Kolmogorov classique.

Quelques différences sont à noter avec la complexité de Kolmogorov classique, par exemple, pour la sous-additivité. On sait que classiquement, $\mathsf{C}(x,y) \leq \mathsf{C}(x) + \mathsf{C}(y|x) + O(1)$. Implicite dans ce résultat est le fait qu'on puisse réutiliser x. En quantique on doit produire deux copies de x, une pour la sortie, et une comme chaîne auxiliaire pour produire y.

Proposition 5. *Pour tout $|\varphi\rangle, |\psi\rangle$, $\mathsf{QC}(|\varphi\rangle, |\psi\rangle) \leq \mathsf{QC}(|\varphi\rangle, |\varphi\rangle) + \mathsf{QC}(|\psi\rangle||\varphi\rangle) + O(1)$.*

On peut aussi étudier de plus près le phénomène de non-clonage quantique. Malgré le phénomène de non-clonage, les copies d'états quantiques peuvent être compressés. Ce résultat est prouvé à l'aide des sous-espaces symétriques de l'espace de Hilbert.

Théorème 22. *Pour tout état quantique $|\varphi\rangle$, et entier m*

$$\mathsf{QC}(|\varphi\rangle^{\otimes m}|m) \;\leq\; \log \binom{m + 2^{\mathsf{QC}(|\varphi\rangle)} - 1}{2^{\mathsf{QC}(|\varphi\rangle)} - 1} + O(1), \tag{6.1}$$

par conséquent, $\mathsf{QC}(|\varphi\rangle^{\otimes m}) \leq \log \binom{m+2^{\mathsf{QC}(|\varphi\rangle)}-1}{2^{\mathsf{QC}(|\varphi\rangle)}-1} + O(\log m)$.

La complexité de Kolmogorov quantique permet ainsi de "mesurer" la clonabilité des états, avec l'expression $\mathsf{QC}(|\varphi\rangle^{\otimes m}|m)$. On montre, par exemple, qu'il existe des états maximalement non-clonables.

Théorème 23 (Incompressibilité de copies d'états quantiques)**.** *Pour tout m et n, il exitste un état à n qubit $|\varphi\rangle$ tel que*

$$\mathsf{QC}(|\varphi\rangle^{\otimes m}) \geq \log \binom{m + 2^n - 1}{2^n - 1}.$$

La plupart des résultats sont cités ici pour les états purs, mais nos résultats tiennent également pour les états mixtes, c'est-à-dire des mélanges statistiques d'états purs.

6.2 Simulation de corrélations quantiques

Au tout début de la mécanique quantique, plusieurs physiciens se sont interrogés sur les effets étranges prédits par cette théorie. Einstein, Podolsky and Rosen ont montré que lorsque deux partis, appelés Alice et Bob, partagent un état enchevêtré, si l'un et l'autre mesurent indépendamment leur portion de cet état, non seulement le résultat est probabiliste, mais ils sont aussi corrélés, même s'ils sont très éloignés lorsqu'ils font leur mesure. Cela suggère que le résultat des mesures serait "communiqué" au moins partiellement à une vitesse supérieure à la vitesse de la lumière. Devant cet apparent paradoxe, ils se sont donc demandés si la mécanique quantique était une description complète et cohérente de la réalité physique [EPR35].

Pour résoudre le paradoxe, appelé aujourd'hui paradoxe EPR, il a été proposé que l'aléa apparent dans les expériences quantiques pouvaient s'expliquer par la présence d'aléa caché, qui aurait été créé localement avec l'état (prétendument) quantique, ce qui expliquerait l'aléa apparent du modèle quantique. Cependant Bell montra en 1964 que les corrélations quantiques apparentes dans les expériences EPR ne pouvaient pas s'expliquer par un modèle de variables aléatoires cachées [Bel64].

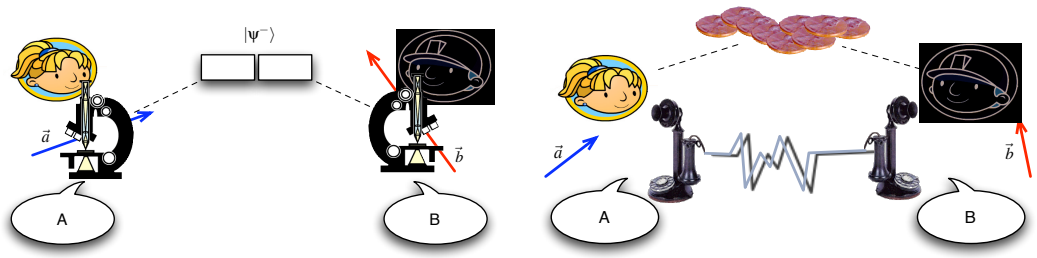

FIG. 6.1 – À gauche, l'expérience EPR. À droite, simulation des corrélations EPR avec des variables aléatoires partagées et de la communication classique.

Avec l'aide de la complexité de la communication, on a poursuivi les travaux qui consistent à évaluer quelles et combien ressources non-quantiques seraient nécessaires, en plus de l'aléa partagé, suffisent à reproduire les corrélations quantiques. Cette quantité permet de quantifier la non-localité quantique. Les ressources considérées sont les suivantes :

La communication classique. En plus de partager un nombre infini de variables aléatoires partagées, Alice et Bob disposent d'un canal de communication (classique) qui leur permet de s'échanger des messages. On compte le nombre de bit échangés entre les deux joueurs, comme dans le modèle de complexité de la communication.

La postsélection. Alice et Bob peuvent occasionnellement refuser de compléter le protocole. Dans ce cas on ne tient compte que des résultats où les deux joueurs complètent le protocole. Cela correspond à la situation en laboratoire où l'un ou l'autre des détecteurs ne donne aucun résultat. Dans ce cadre, on compte l'efficacité du protocole comme la probabilité que chaque joueur complète le protocole.

La boîte non-locale. Il s'agit d'une boîte noire dans laquelle Alice peut déposer une valeur $x \in \{0,1\}$ et Bob peut déposer une valeur $y \in \{0,1\}$. A la sortie, Alice récupère une valeur $\alpha \in \{0,1\}$ et Bob récupère $\beta \in \{0,1\}$, tels que α, β prises individuellement sont uniformes, mais ils sont corrélés selon $x \wedge y = \alpha \oplus \beta$. On sait qu'un bit de communication suffit pour simuler une utilisation d'une boîte non-locale, mais le contraire n'est pas vrai. (On peut montrer qu'une boîte non-locale ne permet pas de communication.) Ainsi, un protocole faisant k utilisations d'une boîte non-locale est plus économe qu'un protocole avec k bit de communication échangés.

Les résultats connus sont résumés dans le tableau qui suit.

Ressource	Quantité	Moyenne/Pire cas	Plan/sphère	Référence
Communication	1.17	Moyenne	Plan	[Mau92]
Communication	8	Pire cas	Sphère	[BCT99]
Postsélection	1/3	Pire cas	Sphère	[GG99]
Communication	1.48	Moyenne	Plan	[Ste00]
Communication	1.19	Moyenne	Sphère	[CGM00]
Communication	1	Pire cas	Sphère	[TB03]
Boîte non locale	1	Pire cas	Sphère	[CGMP04]

TAB. 6.1 – Simulation des corrélations quantiques.

Dans notre travail, on a montré que le problème de simulation des corrélations quantiques se réduit à un problème d'échantillonnage distribué.

Dans le modèle à variables cachées, Alice possède une entrée $\vec{a}$ dans la sphère unitaire, et produit une sortie A booléenne, et Bob a pour entrée $\vec{b}$ et sortie B. Pour simuler les corrélations quantiques, la sortie doit avoir les probabilités conjointes suivantes :

$$p(A,B) = \frac{1 - AB\ \vec{a} \cdot \vec{b}}{4}.$$

Si Alice et Bob, à partir d'une source aléatoire partagée, pouvaient partager un échantillon d'un point sur la sphère de rayon unité selon la distribution

$$\rho_{\vec{a}}(\lambda) = \frac{|\lambda \cdot \vec{a}|}{2\pi},$$

où $\vec{a}$ est un vecteur unitaire décrivant sa mesure, alors Alice et Bob pourraient reproduire les corrélations quantiques selon le protocole suivant.

Simulation de corrélations quantiques

1. Alice et Bob échantillonnent une variable $\vec{\lambda} \sim \rho_{\vec{a}}$,
2. Alice produit $A = \text{sgn}(\vec{a} \cdot \lambda)$
3. Bob produit $B = \text{sgn}(\vec{a} \cdot \lambda)$

On a donné plusieurs méthodes qui consistent à échantillonner selon $\rho_{\vec{a}}$ de sorte qu'à la fin, Alice et Bob partagent l'échantillon, mais en minimisant la quantité de ressources en plus de l'aléa partagé. La méthode la plus simple utilise la postsélection. Une autre permet d'obtenir un échantillon avec un seul bit de communication. Finalement, on identifie un problème d'échantillonnage un peu plus simple qui suffit pour simuler les corrélations. Alice obtiendra un échantillon de $\rho_{\vec{a}}$ mais Bob n'obtient que $\text{sgn}(\vec{a} \cdot \lambda)$, ce qui suffit pour obtenir les corrélations quantiques. Ce problème d'échantillonnage simplifié peut être réalisé avec uniquement une utilisation d'une boîte non-locale.

Ces méthodes d'échantillonnage permettent de retrouver, et surtout, d'expliquer, les méthodes données à la Figure 6.2. Nos techniques donnent également lieu à un nouveau protocole pour une famille généralisée de mesures, les POVM, qui utilise 6 bit de communication en moyenne.

Chapitre 7

Projet de recherche

Je souhaite orienter ma recherche autour de deux principaux projets à court terme : le premier consiste à poursuivre mes travaux sur des nouvelles techniques de preuve pour les bornes inférieures classiques et quantiques. Les techniques de bornes inférieures développées pour les modèles quantiques ont déjà permis, dans plusieurs cas, d'améliorer les bornes classiques [SV01, KW03, Aar04]. Le but de mes travaux est de poursuivre dans ce sens, en apportant au classique des techniques quantiques et vice versa, pour arriver à des techniques unifiées qui s'appliqueraient à plusieurs modèles de calcul (déterministe, probabiliste, quantique, complexité de la communication...).

Le deuxième vise à obtenir des bornes concrètes en complexité des circuits, où un des problèmes ouverts les plus importants est de prouver une borne inférieure surlinéaire sur la taille de circuit pour une fonction explicite. Pour cela, il faudra développer des méthodes ad hoc. Dans les deux projets, je voudrais appliquer à ce problème des méthodes issues de la théorie de l'information.

Trois projets à moyen et long terme viennent compléter mon programme de recherche. Un est de développer une nouvelle notion de sécurité des instances individuelles des protocoles cryptographiques. Le second est l'extension de travaux précédents sur la complexité de Kolmogorov quantique. Le dernier concerne le lien entre complexité de Kolmogorov à ressources bornées et l'analyse des classes probabilistes et déterministes (séparation ou dérandomisation).

Tout comme mes travaux antérieurs, ces projets s'inscrivent dans un but commun. La complexité d'un problème peut provenir de plusieurs sources. Les preuves par diagonalisation, par exemple, mettent en évidence la faiblesse d'un modèle de calcul face à un trop grand nombre de problèmes à résoudre. Un autre obstacle au calcul efficace est l'information que le calcul doit extraire pour résoudre un exemplaire d'un problème. C'est cet aspect qui m'intéresse, et mes projets de recherche ont pour but de faire ressortir le rôle de la composante informationnelle dans les techniques de preuve existantes, de mettre de l'avant de nouvelles techniques à la fois plus simples et plus générales, pour ainsi obtenir des séparations de classes de complexité ou des bornes inférieures sur la complexité de problèmes concrets. La complexité de Kolmogorov se prête particulièrement bien à cette étude, et l'intuition qu'on en retire donne souvent lieu à des résultats inattendus.

7.1 Bornes inférieures classiques et quantiques

Suite aux travaux réalisés avec Magniez, Lee et Szegedy [LM04, LLS05], je prévois concentrer mon travail sur la question de l'application de techniques issues de la complexité quantique et la complexité de Kolmogorov aux bornes inférieures, avec une méthode unifiée pour plusieurs modèles de calcul.

Des travaux récents de Bar-Yossef, Jayram, Kumar and Sivakumar [BYJKS02b, BYJKS02a] en complexité classique montrent que des méthodes issues de la théorie de l'information sont adaptées aux bornes inférieures pour la complexité de la communication classique et pour les algorithmes sur les grands flux de données. D'autres travaux en complexité des requêtes quantiques ("query complexity") tendent dans le même sens [JRS02]. Les résultats de ce type peuvent habituellement être exprimés dans le langage de la complexité de Kolmogorov, ce qui donne lieu à une analyse plus simple basée sur la combinatoire, alors qu'en théorie de l'information, des résultats profonds en probabilité et en statistique sont souvent nécessaires. Il est donc utile de développer des outils qui permettent dans plusieurs cas de donner des preuves simples là où les techniques précédentes étaient beaucoup plus difficiles à appliquer.

Je souhaite développer ces méthodes afin d'obtenir des bornes classiques et quantiques avec une seule analyse, qui ne dépend pas des particularités du modèle. Le but serait de les appliquer à plusieurs modèles de calcul (complexité des requêtes, complexité de la communication, complexité des circuits classiques).

7.2 Bornes inférieures en complexité des circuits

Les techniques développées avec Lee et Szegedy [LLS05] ne sont pas susceptibles de donner lieu à des bornes inférieures surlinéaires pour la taille des circuits d'une fonction explicite. Seule une approche ad hoc pourrait permettre d'obtenir de tels résultats. Cependant, il est à espérer que ces idées, conjointement avec les travaux sur les bornes inférieures avec les méthodes de type théorie de l'information, iront dans ce sens.

La complexité de la communication est une technique bien connue pour prouver des bornes inférieures en complexité des circuits. Avec Troy Lee (CWI Amsterdam), on a des résultats préliminaires qui nous permettent de redémontrer des résultats connus en complexité de la communication probabiliste, dont la preuve antérieure utilise la théorie de l'information. Nos preuves utilisent la complexité de Kolmogorov, et ces preuves sont plus élémentaires que les preuves précédentes. En particulier, on n'utilise pas le principe "min-max" de Yao, jusqu'à maintenant incontournable dans les preuves de bornes inférieures sur la complexité de la communication probabiliste. Nos travaux se poursuivent avec Marc Kaplan, où on a développé une technique de preuve générale basée sur la complexité de Kolmogorov, qui généralise un grand nombre de techniques classiques en complexité de la communication. Lorsque les techniques seront mieux maîtrisées, nous souhaitons les appliquer à la taille des circuits.

7.3 Sécurité cryptographique des instances

En collaboration avec le laboratoire LIACC de l'Université de Porto (Luis Antunes, Armando Matos et une étudiante, Liliana Salvador), notre projet consiste à définir une notion de sécurité cryptographique basée sur la complexité de Kolmogorov. À l'heure actuelle, il existe deux grandes familles de niveaux de sécurité : la sécurité absolue, basée sur la théorie de l'information, et la sécurité conditionnelle, basée sur une hypothèse cryptographique, c'est-à-dire sur la complexité de problèmes tels que la factorisation, le logarithme discret sur les courbes elliptiques, etc.

L'inconvénient de ces approches est que la sécurité n'est jamais garantie pour une instance individuelle, seulement en moyenne sur l'ensemble des instances. Exprimer la sécurité d'un système cryptographique en termes de complexité de Kolmogorov serait une façon de quantifier la sécurité des instances individuelles.

La complexité de Kolmogorov admet aussi une version à ressources bornées, c'est-à-dire qu'on peut ne considérer que les programmes qui fonctionnement en temps polynomial, ou exponentiel, ou en espace linéaire, etc. Ces variantes permettraient de définir une notion de sécurité qui tiendrait à la fois de la théorie de l'information et de la complexité, faisant ainsi une transition lisse entre les deux approches existantes. Dans un premier temps on veut poser les bases dans le cadre classique, puis passer au modèle quantique.

Des travaux menés sur ce sujet ont donné lieu à un mémoire de DEA de Liliana Salvador, étudiante à l'université de Porto. Ses résultats, obtenus avec Luis Antunes (U. Porto) et moi-même, portent principalement sur le "one-time pad" et sur le partage de secrets (secret sharing schemes). Ces travaux sont encore au stade préliminaire et ne sont pas encore publiés. Ils ont fait l'objet d'un financement bilatéral Luso-français (EGIDE et Ambassade de France au Portugal).

7.4 Applications de la complexité de Kolmogorov quantique

Récemment, trois articles ont proposé des définitions pour adapter la complexité de Kolmogorov au cadre quantique [Vit00, Gác01, BvDL01]. Ces définitions ont pour but de quantifier l'aléa contenu dans une chaîne de qubit fixée. Ces notions devraient naturellement donner lieu à des applications, notamment des bornes inférieures, mais il reste à surmonter quelques obstacles techniques. J'aimerais d'abord étudier des problèmes simples en complexité de communication, par exemple dans le modèle de la complexité de la communication à sens unique (un seul message). Ensuite la technique pourrait être appliquée à des modèles plus généraux.

7.5 Simulation de corrélations quantiques

Je souhaite poursuivre l'étude de la simulation des corrélations quantiques avec l'approche de l'échantillonnage, afin de trouver des résultats pour des cadres plus généraux. Presque tous les résultats connus aujourd'hui sont pour les mesures dites de Von Neumann, òu les mesures sont faites selon des axes orthogonaux. Pour les POVM, des mesures plus générales, il n'existe aucun protocole avec un nombre borné de bit de communication en pire cas. On pourrait aussi considérer les états en dimension plus que 3. Finalement, on voudrait aussi simuler les corrélations obtenues pour des états non maximalement enchevêtrés.

7.6 Complexité du calcul probabiliste

Un projet de recherche à plus long terme porte sur la complexité du calcul probabiliste. Une des plus grandes questions en complexité du calcul à l'heure actuelle est de déterminer si la classe de calcul probabiliste BPP est égale à EXP, la classe de calcul déterministe en temps exponentiel, ou à P, la classe de calcul déterministe en temps polynomial. Beaucoup de travaux récents sur la dérandomisation laissent croire que ce sera l'objet des prochaines grandes percées de la complexité. Déjà, des résultats importants comme ceux de Impagliazzo et Wigderson [IW97] mettent en lumière le lien entre les bornes inférieures sur les modèles de calcul non-uniformes et la dérandomisation. (Ils démontrent que s'il existe des fonctions calculables en temps $2^{O(n)}$ qui sont difficiles pour les circuits de taille 2^n, alors il existe un générateur pseudo-aléatoire qui permet de dérandomiser BPP, c'est-à-dire BPP=P). Quand on regarde la preuve de près, celle-ci repose sur une hypothèse qui a trait à la taille de la description du germe d'un générateur pseudo-aléatoire en termes de la taille du circuit. J'aimerais obtenir un énoncé plus précis en termes de complexité de Kolmogorov du germe.

Nos travaux sur la compression des langages, et en particulier sur les extracteurs [BFL02], laissent entrevoir un autre lien entre dérandomisation et complexité de Kolmogorov. Il est intéressant de noter que Trevisan [Tre01] construit un extracteur en combinant un code correcteur d'erreurs et le générateur pseudo-aléatoire de Impagliazzo et Wigderson. J'aimerais explorer le lien précis entre compression de langages et génération pseudo-aléatoire, et en particulier, si on peut établir une équivalence entre les bornes sur la complexité de la compression des langages et la dérandomisation.

Bibliographie

[Aar04] S. Aaronson. Lower bounds for local search by quantum arguments lower bounds for local search by quantum arguments. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing of the thirty-fifth annual ACM symposium on Theory of computing*, pages 465–474, 2004.

[AFKS00] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy. Efficient testing of large graphs. *Combinatorica*, 20 :451–476, 2000.

[Amb00] A. Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the Thirty-Second ACM Symposium on Theory of Computing*, pages 636–643, 2000.

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64 :750–767, 2002.

[Amb03] A. Ambainis. Polynomial degree vs. quantum query complexity. In *Proceedings of 44th IEEE Symposium on Foundations of Computer Science*, pages 230–239, 2003.

[And87] A. E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of Π-schemes. *Moscow Univ. Math. Bull.*, 42(1) :63–66, 1987.

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4) :595 – 605, 2004.

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4) :778–797, 2001.

[BCT99] G. Brassard, R. Cleve, and A. Tapp. Cost of exactly simulating quantum entanglement with classical communication. *Phys. Rev. Lett.*, 83 :1874–1877, 1999.

[Bel64] J. S. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1 :195, 1964.

[BF92] R. Beigel and J. Feigenbaum. On being incoherent without being very hard. *Computational Complexity*, 2 :1–17, 1992.

[BFL02] Harry Buhrman, Lance Fortnow, and Sophie Laplante. Resource-bounded kolmogorov complexity revisited. *SIAM Journal on Computing*, 31(3) :887–905, 2002.

[BHT97] G. Brassard, P. Høyer, and A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology column)*, 28 :14–19, 1997.

[BK95] M. Blum and S. Kannan. Designing programs that check their work. *Journal of the ACM*, 42(1) :269–291, 1995.

[BL99] L. Babai and S. Laplante. Stronger separations for random-self-reducibility, rounds, and advice. In *Proceedings of the Fourteenth Annual Conference on Computational Complexity*, pages 98–104, 1999.

[BLM99] H. Buhrman, S. Laplante, and P. B. Miltersen. New bounds for the language compression problem. In *Proceedings of the Fifteenth Annual Conference on Computational Complexity*, pages 126–130, 1999.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3) :549–595, 1993.

[BLvM04] H. Buhrman, T. Lee, and D. van Melkebeek. Language compression and pseudorandom generators. In *19th Annual IEEE Conference on Computational Complexity*, pages 15–28, 2004.

[BSS03] H. Barnum, M. Saks, and M. Szegedy. Quantum decision trees and semidefinite programming. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 179–193, 2003.

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5) :1411–1473, 1997.

[BvDL01] A. Berthiaume, W. van Dam, and S. Laplante. Quantum kolmogorov complexity. *Journal of Computer System Sciences*, 63 :201–221, 2001.

[BYJKS02a] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. an information statistics approach to data stream and communication complexity. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 209–218, 2002.

[BYJKS02b] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. Information theory methods in communication complexity. In *Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, pages 93–102, 2002.

[CGL94] E. Clarke, O. Grumberg, and D. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5) :1512–1542, 1994.

[CGM00] Nicolas J. Cerf, Nicolas Gisin, and Serge Massar. Classical teleportation of a quantum bit. *Phys. Rev. Lett.*, 84 :2521–2524, 2000.

[CGMP04] N. J. Cerf, N. Gisin, S. Massar, and S. Popescu. Quantum entanglement can be simulated without communication. Technical Report 0410027, quant-ph ArXiv, 2004. quant-ph/0410027.

[CGP99] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[Chu41] A. Church. The calculi of lambda-conversion. *Annals of Mathematics Studies*, 6, 1941.

[Coo73] S. Cook. A hierarchy for nondeterministic time complexity. *Journal of Computer and System Sciences*, 7(4) :343–353, 1973.

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London A*, volume 400, pages 97–117, 1985.

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society A*, volume 439, 1992.

[DR82] A.G. Dyachkov and V.V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Informatsii*, 18(3) :7–13, 1982. In Russian.

[Edm65] J. Edmonds. Maximum matchings and a polyhedron with 0,1-vertices. *Journal of Research at the National Bureau of SStandards (Section B)*, 69B :125–130, 1965.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete ? *Phys. Rev.*, 47 :777–780, 1935.

[FF93] J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22 :994–1005, 1993.

[FFLN98] J. Feigenbaum, L. Fortnow, S. Laplante, and A. Naik. On coherence, random-self-reducibility and self-correction. *Computational Complexity*, 7 :174–191, 1998.

[FFLS94] J. Feigenbaum, L. Fortnow, C. Lund, and D. Spielman. The power of adaptiveness and additional queries in random-self-reductions. *Computational Complexity*, 4 :158–174, 1994.

[Gác01] P. Gács. Quantum algorithmic entropy. In *Proceedings of the Sixteenth Annual Conference on Computational Complexity*, pages 274–285, 2001.

[GG99] N. Gisin and B. Gisin. A local hidden variable model of quantum correlation exploiting the detection loophole. *Phys. Lett. A*, 260(5) :323–327, 1999.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4) :653–750, 1998.

[GR99] O. Goldreich and D. Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 19 :335–373, 1999.

[GR02] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2) :302–343, 2002.

[Gro96] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[Hås89] J. Håstad. *Randomness and Computation, Advances in Computing Research*, volume 5, chapter Almost Optimal Lower Bounds for Small Depth Circuits, pages 143–170. JAI Press, 1989.

[Hås98] J. Håstad. he shrinkage exponent of de morgan formulae is 2. *SIAM Journal on Computing*, 2719(1) :48–64, 1998.

[HIS65] J. Hartmanis, M. Lewis II, and R. Stearns. Hierarchies of memory limited computations. In *Proceedings of the Sixth Annual IEEE Symposium on Switching Circuit Theory and Logical Design*, pages 179–190, 1965.

[HMT88] A. Hajnal, W. Maass, and G. Turán. On the communication complexity of graph properties. In ACM, editor, *Proceedings of the twentieth annual ACM Symposium on Theory of Computing, Chicago, Illinois, May 2–4, 1988*, pages 186–191, 1988.

[HNOS96] E. Hemaspaandra, A. Naik, M Ogihara, and A. Selman. P-selective sets and reducing search to decision versus self-reducibility. *Journal of Computer and System Sciences*, 53(2) :194–209, October 1996.

[HS65] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117 :285–306, 1965.

[Iba72] O. Ibarra. A note concerning nondeterministic tape complexities. *Journal of the ACM*, 19(4) :608–612, 1972.

[IM02] K. Iwama and H. Morizumi. An explicit lower bound of $5n - o(n)$ for boolean circuits. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science*, pages 353–364, 2002.

[Imm88] N. Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on Computing*, 17(5) :935–938, 1988.

[IW97] R. Impagliazzo and A. Wigderson. P=bpp unless e has subexponential circuits : derandomizing the xor lemma. In *Proceedings of the 29th STOC*, pages 220–229, 1997.

[JRS02] R. Jain, J. Radhakrishnan, and P. Sen. Quantum lower bounds by quantum arguments. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 429–438, 2002.

[Khr71] V.M. Khrapchenko. Complexity of the realization of a linear function in the case of Π-circuits. *Math. Notes Acad. Sciences*, 9 :21–23, 1971.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity.* Cambridge University Press, 1997.

[Kou93] E. Koutsoupias. Improvements on Khrapchenko's theorem. *Theoretical Computer Science*, 116(2) :399–403, 1993.

[KW88] M. Karchmer and A. Wigderson. Monotone connectivity circuits require superlogarithmic depth. In *Proceedings of the 20th STOC*, pages 539–550, 1988.

[KW03] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing of the thirty-fifth annual ACM symposium on Theory of computing*, pages 106—115, 2003.

[Lap97] S. Laplante. *Kolmogorov Techniques in Complexity Theory.* PhD thesis, University of Chicago Department of Computer Science, 1997.

[LLM+02] S. Laplante, R. Lassaigne, F. Magniez, S. Peyronnet, and Michel de Rougemont. Probabilistic abstraction for model checking : An approach based on property testing. In *Logic in Computer Science*, pages 30–39, 2002.

[LLS05] S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and classical formula size lower bounds. In *Proceedings of the Twentieth Annual IEEE Conference on Computational Complexity*, pages 76–90, 2005.

[LM04] S. Laplante and F. Magniez. Lower bounds for randomized and quantum query complexity using kolmogorov arguments. In *Proceedings of the Nineteenth Annual IEEE Conference on Computational Complexity*, pages 294–304, 2004.

[LR01] O. Lachich and R. Raz. Explicit lower bound of 4.5n - o(n) for boolean circuits. In *Proceeding of the 33rd Symposium on Theory of Computing*, 2001.

[LV97] M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer Verlag, second edition, 1997.

[Mau92] T. Maudlin. Bell's inequality, information transmission, and prism models. In *Biennal Meeting of the Philosophy of Science Association*, pages 404–417, 1992.

[McM93] K. McMillan. *Symbolic Model Checking.* Kluwer Academic Publishers, 1993.

[NC00] M. A. Nielsen and I. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[NTS99] N. Nisan and A. Ta-Shma. Extracting randomness : A survey and new constructions. *Journal of Computer and System Sciences*, 58(1) :148–173, 1999.

[Raz90] A. Razborov. Applications of matrix methods to the theory of lower bounds in computational complexity. *Combinatorica*, 10(1) :81–93, 1990.

[Raz95] A. Razborov. *Feasible Mathematics II*, chapter Bounded Arithmetic and Lower Bounds in Boolean Complexity, pages 344–386. Birkhäuser Verlag, 1995.

[Rei05] O. Reingold. Undirected st-connectivity in log-space. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 376–385, 2005.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2) :23–32, 1996.

[Sha49] C. E. Shannon. Communication in the presence of noise. *IRE*, 37 :10–21, 1949.

[Sim97] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5) :1474–1483, 1997.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proc. 15th ACM Symposium on Theory of Computing*, pages 330–335, 1983.

[SS04] M. Santha and M. Szegedy. Quantum and classical query complexities of local search are polynomially related. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 494–501, 2004. Manuscript.

[ŠS05] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. In *Proceedings of 32nd International Colloquium on Automata, Languages and Programming*, pages 1299–1311, 2005. quant-ph/0409116.

[Ste00] M. Steiner. Towards quantifying non-local information transfer : finite-bit non-locality. *Phys. Lett. A*, 270 :239–244, 2000.

[SV01] P. Sen and S. Venkatesh. Lower bounds in the quantum cell probe model lower bounds in the quantum cell probe model lower bounds in the quantum cell probe model. In *Proceedings of Intenational Colloquium on Automata, Languages and Programming (ICALP)*, pages 358–369, 2001.

[Sze88] R. Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26 :279–284, 1988.

[Sze03] M. Szegedy. An $O(n^{1.3})$ quantum algorithm for the triangle finding problem. Technical Report 0310134, quant-ph, 2003. quant-ph/0310134.

[TB03] B. F. Toner and D. Bacon. Communication cost of simulating Bell correlations. *Phys. Rev. Lett.*, 91 :187904, 2003.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4) :860–879, 2001.

[Tri05] V. Trifonov. An o(log n log log n) space algorithm for undirected st-connectivity. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 626–633, 2005.

[Tur36] A. Turing. On computable numbers, with an application to the etscheidungs problem. *Proceedings of the London Mathematica Society*, 42 :230–265, 1936.

[Vit00] P. Vitanyi. Three approaches to the quantitative definition of information in an individual pure quantum state. In *Proceedings of the 15th Annual IEEE Conference on Computational Colmplexity*, pages 263–270, 2000.

[VV86] L. G. Valiant and V. V. Vazirani. Np is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1) :85–93, 1986.

[Yao90] A. C.-C. Yao. Coherent functions and program checkers. In *Proceedings of 22nd ACM Symposium on Theory of Computing*, pages 84–94, 1990.

[Zha04] S. Zhang. On the power of Ambainis's lower bounds. In *Proceedings of 31st International Colloquium on Automata, Languages and Programming*, 2004. To appear. Also in quant-ph/0311060.