Complexity measures through the lens of two-player games and signatures of the hypercube

Anupa Sunny

Université Paris Cité

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

UNIVERSITÉ PARIS CITÉ

ÉCOLE DOCTORALE SCIENCES MATHÉMATIQUES DE PARIS CENTRE (ED 386)

INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE

THÈSE DE DOCTORAT EN INFORMATIQUE

Complexity measures through the lens of two-player games and signatures of the hypercube

---

Par:
Anupa Sunny

Dirigée par:
Dr. Sophie Laplante

Présentée et soutenue publiquement le 15 février 2023, devant un jury composé de :

Sophie Laplante	PR	Université Paris Cité, IRIF	*Directrice de thèse*
Rahul Jain	PR	National University of Singapore	*Rapporteur*
Troy Lee	PR	University of Technology Sydney	*Rapporteur*
Stacey Jeffery	DR	CWI, Amsterdam	*Examinatrice*
Robin Kothari	CR	Google Quantum AI	*Examinateur*
Sylvain Schmitz	PR	Université Paris Cité, IRIF	*Président du jury*

Abstract

Complexity measures of Boolean functions capture various aspects of the hardness of computing a function and their study is about finding connections between different complexity measures.

In the first part of this thesis, we introduce and study Certificate Game complexity, a measure of complexity based on the probability of winning a game in which two players are given inputs with different function values and are asked to output some index i where their inputs differ, in a zero-communication setting. We give upper and lower bounds for private coin, public coin, shared entanglement and non-signaling strategies, and give some separations. We show that complexity in the public coin model is bounded above by Randomized query and Certificate complexities. On the other hand, it is bounded below by fractional certificate complexity, making it a good candidate to prove strong lower bounds on randomized query complexity. Complexity in the private coin model is bounded below by zero-error randomized query complexity. The quantum measure highlights an interesting and surprising difference between classical and quantum query models. While public coin certificate game complexity is bounded above by randomized query complexity, quantum certificate game complexity can be quadratically larger than quantum query complexity. We use non-signaling, a notion from quantum information, to give a lower bound of n on the quantum certificate game complexity of the OR function, whose quantum query complexity is $\Theta(\sqrt{n})$ and then go on to show that this "non-signaling bottleneck" applies to all functions with high sensitivity, block sensitivity or fractional block sensitivity. We also consider the single-bit version of certificate games, where the inputs of the two players are restricted to having Hamming distance 1. We prove that the single-bit version of certificate game complexity with shared randomness is equal to sensitivity up to constant factors, thus giving a new characterization of sensitivity. On the other hand, the single-bit version of certificate game complexity with private randomness is equal to λ^2, where λ is the spectral sensitivity.

In the second part of this thesis, we revisit the celebrated proof of the sensitivity conjecture by Hao Huang. Using spectral techniques, Huang proved that every subgraph of the hypercube H_n of dimension n induced on more than half the vertices has maximum degree at least $\sqrt{n}$. Combined with earlier work, this completed a proof of the sensitivity conjecture. We show an alternate proof of Huang's result using only linear dependency of vectors associated with the vertices of the hypercube. Our approach helps gain insight on more structural properties of the induced subgraph in addition to the largest degree. In particular, we prove that in any induced subgraph of H_n with more than half the number of vertices, there are two vertices, one of odd parity and the other of even parity, each with at least n vertices at distance at most 2. As an application, we show that for any Boolean function f, the polynomial degree is bounded above by the product of 0-sensitivity and 1-sensitivity, $\mathsf{s}_0(f)\mathsf{s}_1(f)$, a strictly stronger statement which implies Huang's theorem. We also obtain structural relations for induced subgraphs at distance 3.

A key implement in Huang's proof was signed hypercubes with the property that every cycle of length 4 is assigned a negative sign. We take a detailed look at this signature and give a nearly optimal signature that uses the minimum number of negative edges while ensuring that every 4-cycle is negative. This problem turns out to be related to one of Erdős' problems on the largest 4-cycle free subgraph of the hypercube.

Keywords: Complexity measures, Boolean functions, sensitivity, certificate complexity, signed hypercubes, zero-communication two-player games

Résumé

Les mesures de complexité des fonctions booléennes capturent divers aspects de la difficulté du calcul d'une fonction et leur étude consiste à trouver des connexions entre différentes mesures de complexité.

Dans la première partie de cette thèse, nous introduisons et étudions la complexité de jeux de certificats, une mesure de complexité basée sur la probabilité de gagner un jeu dans lequel deux joueurs reçoivent des entrées avec des valeurs de fonctions différentes et doivent produire un indice i pour lequel leurs entrées diffèrent, sans communiquer. Nous donnons des bornes supérieures et inférieures pour les stratégies à base de pièces privées, de pièces publiques, d'intrication partagée et de non-signalisation, et nous prouvons quelques résultats de séparations. D'une part, nous montrons que la complexité dans le cas des pièces publiques est majorée par les complexités de requête aléatoire et de certificat. D'autre part, nous montrons qu'elle est minorée par la complexité fractionnelle de certificat, ce qui en fait un bon candidat pour trouver des bornes inférieures fortes sur la complexité de requête aléatoire. La complexité dans le cas des pièces privées est minorée par la complexité de requête aléatoire à erreur nulle. La mesure quantique met en évidence une différence intéressante et surprenante entre les modèles de requête classiques et quantiques. Alors que la complexité de jeux de certificats dans le cas des pièces publiques est majorée par la complexité de requête aléatoire, la complexité de jeux de certificats quantiques peut être quadratiquement plus grande que la complexité de requête quantique. Nous utilisons la non-signalisation, une notion d'information quantique, pour minorer par n la complexité de jeux de certificats quantiques de la fonction OR, dont la complexité de requête quantique est de $\Theta(\sqrt{n})$, puis nous montrons que ce "goulot d'étranglement de non-signalisation" s'applique à toutes les fonctions à sensibilité, à sensibilité de bloc ou à sensibilité de bloc fractionnaire élevée. Nous considérons également la version mono-bit des jeux de certificats, où les entrées des deux joueurs sont restreints à une distance de Hamming de 1. Nous prouvons que la version mono-bit de la complexité de jeux de certificats avec aléa partagé est égale à la sensibilité à un facteur constant près, ce qui donne une nouvelle caractérisation de la sensibilité. D'autre part, la version mono-bit de la complexité de jeux de certificats avec aléa privé est égale à λ^2, où λ est la sensibilité spectrale.

Dans la deuxième partie de cette thèse, nous revisitons la célèbre preuve de la conjecture de la sensibilité par Hao Huang. En utilisant des techniques spectrales, Huang a prouvé que tout sous-graphe de l'hypercube H_n de dimension n induit sur plus de la moitié des sommets a un degré maximal d'au moins $\sqrt{n}$. Combiné avec des travaux antérieurs, ce résultat a complété une preuve de la conjecture de la sensibilité. Nous en donnons une preuve alternative en utilisant seulement la dépendance linéaire des vecteurs associés aux sommets de l'hypercube. Notre approche permet de mieux comprendre les propriétés structurelles du sous-graphe induit, en plus du plus grand degré. En particulier, nous prouvons que dans tout sous-graphe induit de H_n avec plus de la moitié du nombre de sommets, il existe deux sommets, l'un de parité impaire et l'autre de parité paire, chacun ayant au moins n sommets à une distance au plus égale à 2. Comme application, nous montrons que pour toute fonction booléenne f, le degré polynomial est majoré par le produit de la sensibilité 0 et de la sensibilité 1, $s_0(f)s_1(f)$, une affirmation strictement plus forte qui implique le théorème de Huang. Nous obtenons également des relations structurelles pour les sous-graphes induits à distance 3.

Un ingrédient clé de la preuve de Huang était des hypercubes signés avec la propriété que chaque cycle de longueur 4 est affecté d'un signe négatif. Nous examinons en détail cette signature et donnons une signature quasi-optimale qui utilise le nombre minimum de bords négatifs tout en garantissant que chaque cycle de longueur 4 est négatif. Ce problème s'avère être lié à l'un des problèmes d'Erdös sur le plus grand sous-graphe de l'hypercube exempt de 4-cycles.

Mots clés: mesures complexes, complexité des certificats, signatures de l'hypercube, jeux à deux joueurs

Acknowledgements

I thank Sophie Laplante for being a very relaxed and understanding advisor who let me work at my own pace, and in particular, for being incredibly generous with her encouragement and support. Her enthusiasm and ability to simplify obscure concepts are traits that I hope to emulate.

I thank each of my other collaborators for having inspired me in multiple ways: Sourav Chakraborty with his ability to see the big picture from the very beginning, Anna Gál with her mathematical rigour, Rajat Mittal with his ability to formalise ideas very quickly and his helpfulness, and Reza Naserasr with his persistence when wading through computations.

I also thank everyone else who has worked with me for making research truly enjoyable. In particular, Alexandre Nolin who set a great example with his readiness to explain concepts and his passion for science, and Zhouningxin Wang for her diligence.

I would like to thank Sourav Chakraborty for inviting me to the Complexity Workshop at ISI Kolkata which marked the start of a great collaboration.

I express my gratitude to Rahul Jain and Troy Lee for agreeing to review my manuscript and for their helpful comments. I would also like to thank Stacey Jeffery, Robin Kothari and Sylvain Schmitz for being a part of my jury. It has truly been an honour.

I would like to thank MathInParis for funding me and Ariela Briani for being the best coordinator ever who made administrative tasks an absolute breeze. I would also like to extend my gratitude to Christoph Dürr and Sylvie Corteel for being a part of my PhD committee and who, despite their hectic schedules, have always found time for meetings at very short notices. A special thanks to Amina Hariti at the École Doctorale for her promptness and help with all kinds of administrative tasks.

IRIF has been a great place with its stellar administrative staff and wonderful permanent and non-permanent members. I would like to thank my officemates and others (Abhishek, Alexandre, Alessandro, Avinandan, Daniel(s), Enrique, Hening, Isaac, Klara, Pierre, Robin, Sander, Simon, Simona, Wael, Yassine, Zeinab, Zhouningxin and many more) for great conversations and for helping me out with tasks (especially those that required French). The lunch congregation at the Cantine followed by “coffee" (or water in my case) by the common area at IRIF has been an huge source of fun with it introducing me to a variety of board games and its light-hearted banter. The IRIF Cake is also very close to my heart having presented the opportunity to try my amateur baking skills.

I would also like to thank Partha Mukhopadhyay who was my Master's thesis advisor at CMI for introducing me to complexity theory and for his support.

Most importantly, I thank my family for being with me on the highs and the lows. I cannot begin to thank Renjan enough for his steadfast emotional support and for understanding me better than I do. He has been the superior half of me, and I am truly grateful to have him as my companion and best friend for life. I thank him

for the evening walks, drives, road trips, weekend trips, the Himalayan trek and for whisking me off to dreamy getaways. I thank him for most carefully going through this manuscript (and all of my writing) and for all his valuable comments that helped better the flow and cohesion. I thank our parents (Appa, Mumma, Mummy and Papa) for their understanding and for their great company. Mumma in particular deserves a giant shout-out for her exceptional appams and fish moilee that has been a delight to savour on a weekly basis. Annakuttan has been an equally strong presence with her humour and empathy that have helped me through many a difficult time.

Introduction en français

L'objectif fondamental de l'informatique est de développer des méthodes ou des algorithmes qui nous permettent d'effectuer des tâches informatiques de manière efficace. Presque toutes les activités que nous effectuons, telles que la planification des courses, la planification de l'itinéraire d'un point A à un point B, ou même la disposition des cartes dans notre main dans un jeu, peuvent être considérées comme des tâches informatiques. Le domaine de la complexité informatique tente de déterminer à quelle vitesse nous pouvons espérer accomplir ces tâches et pas mieux. Nous pouvons reformuler ces tâches comme des problèmes dont la réponse est "oui" ou "non". Par exemple, nous pouvons demander si toutes les courses de la journée peuvent être faites en 2 heures ou s'il existe un chemin d'une longueur maximale de 5 kilomètres entre un point A et un point B, ou si l'on peut ranger 10 cartes dans l'ordre croissant en effectuant au maximum 20 mouvements. Ces questions peuvent être modélisées à l'aide de fonctions booléennes qui produisent soit 0 soit 1 en sortie, où une réponse "oui" correspond à un 1 et une réponse "non" correspond à un 0. Notre objectif est maintenant de déterminer la difficulté de calculer la sortie de ces fonctions booléennes.

Les mesures de complexité tentent de saisir la difficulté de calculer une fonction en quantifiant ce qui rend le calcul difficile. Idéalement, nous aimerions trouver la complexité temporelle d'une fonction, c'est-à-dire le temps qu'il faut pour calculer la sortie d'une fonction. Cependant, il est très difficile de fournir des limites inférieures à la complexité temporelle d'une fonction. Par conséquent, nous nous tournons vers des mesures de complexité plus faibles. Par exemple, nous pouvons penser à certains modèles de calcul où l'accès à certaines ressources utilisées dans le calcul est restreint. Le nombre d'accès à ces ressources qui sont nécessaires pour le calcul peut être considéré comme une mesure de la dureté. Par exemple, imaginons un modèle de calcul où l'entrée est cachée et où vous avez accès à un dispositif qui révèle l'entrée à une position que vous demandez/interrogez. Le nombre de positions qui doivent être révélées pour déterminer la sortie de la fonction peut être considéré comme une mesure de dureté. Cette mesure de complexité est appelée la complexité d'interrogation d'une fonction.

On peut également imaginer de choisir de révéler des positions de l'entrée au hasard et de deviner la sortie. Si la réponse devinée est correcte la plupart du temps, nous avons réussi et le nombre de requêtes qui ont dû être faites s'appelle la complexité de requête aléatoire. De même, on peut penser à une variante quantique de la complexité des requêtes où les requêtes à l'entrée sont faites en superposition. Bien qu'il puisse sembler que les mesures de la complexité des requêtes soient trop simplistes et trop faibles, c'est le cadre qui a le mieux réussi à montrer un avantage définitif des algorithmes quantiques sur les algorithmes classiques. Par exemple, il existe des problèmes dont la résolution classique (par un algorithme aléatoire) nécessite un nombre exponentiel de requêtes, mais dont la résolution quantique ne nécessite qu'un nombre constant de requêtes [91, 39]. La question des limites inférieures de la complexité des requêtes, en particulier de la complexité des requêtes quantiques, est intéressante.

Une longue histoire de travaux dans ce domaine de recherche a été initiée par une mesure de complexité appelée la limite de l'adversaire. De nombreuses formulations de cette mesure ont ensuite été montrées comme étant toutes équivalentes les unes aux autres, et elles ont abouti à une mesure qui caractérise exactement la complexité des requêtes quantiques.

Le domaine des mesures de complexité pour les fonctions booléennes est un vieux champ de recherche et, au fil du temps, plusieurs mesures de complexité ont été étudiées. Par exemple, une fonction booléenne peut être représentée par un polynôme et le degré du polynôme peut être considéré comme une mesure de sa dureté. Ces mesures sont intéressantes en soi, mais il est beaucoup plus intéressant de les comparer. Par exemple, une grande valeur sur une mesure implique-t-elle une grande valeur sur l'autre ? Il est également intéressant de voir si certaines mesures sont asymptotiquement égales les unes aux autres.

Dans le domaine de l'analyse des fonctions booléennes, la conjecture de la sensibilité était l'une des plus insaisissables car elle est restée non résolue pendant environ trois décennies depuis son apparition dans la littérature de la théorie de la complexité et de la théorie des graphes à la fin des années 1980. Cette conjecture visait à caractériser la différence entre les deux mesures de complexité que sont la sensibilité et la sensibilité de bloc. La sensibilité d'une fonction mesure la sensibilité de la sortie d'une fonction au basculement d'un bit dans son entrée. En revanche, dans le cas de la sensibilité par bloc, nous sommes autorisés à basculer plus d'un bit à la fois pour modifier la valeur de la fonction.

La notion de sensibilité a été introduite pour la première fois par Cook et Dwork et, de manière indépendante, par Reischuk [41, 87] pour limiter en dessous le temps nécessaire au calcul d'une fonction dans un modèle CREW (Concurrent Read Exclusive Write) PRAM (Parallel Random Access Machines). Il a été démontré par la suite que la complexité temporelle de ce modèle était exactement caractérisée par le logarithme de la sensibilité des blocs dans un article de Noam Nisan [80]. D'après leurs définitions, on peut voir que la sensibilité est plus petite que la sensibilité de bloc.

La question qui restait était de savoir à quel point la sensibilité pouvait être petite par rapport à la sensibilité de bloc pour une fonction. Une percée majeure dans cette direction a été faite par Nisan et Szegedy qui ont montré une limite supérieure de la sensibilité de bloc d'une fonction en termes de son degré polynomial [81]. Après la démonstration de la borne supérieure de la sensibilité aux blocs en termes de degré en 1992, l'attention s'est portée sur la recherche d'une borne supérieure du degré (ou de toute autre mesure de complexité ayant une borne supérieure polynomiale de la sensibilité aux blocs) en termes d'un certain polynôme de la sensibilité.

Sous une autre forme, en théorie des graphes, Chung, Füredi, Graham et Seymour [37] ont cherché une borne inférieure sur le plus grand degré d'un sous-graphe induit suffisamment grand d'un hypercube. Ces deux problèmes apparemment sans rapport ont été démontrés comme étant équivalents par Gotsman et Linial [50] en 1992. La meilleure borne inférieure connue de la sensibilité par rapport à la sensibilité des blocs a été logarithmique pendant des décennies jusqu'à ce que Huang, en 2019, résolve cette conjecture en montrant que la sensibilité des blocs peut être au maximum quartiquement plus grande que la sensibilité [56].

On ne sait pas si c'est la meilleure relation entre les deux mesures, car toutes les fonctions qui ont été étudiées jusqu'à présent ont au plus une séparation quadratique entre elles. Cette séparation a été obtenue par la fonction de Rubinstein qui est une fonction sur n bits avec une sensibilité de $\sqrt{n}$ et une sensibilité de bloc de $n/2$ [89].

Cette thèse

Dans la première partie de cette thèse, nous introduisons une nouvelle mesure de complexité appelée "Complexité des jeux de certificats" et nous la comparons à d'autres mesures de complexité bien étudiées des fonctions booléennes. Pour décrire cette mesure, considérons un jeu dans lequel deux joueurs, disons Alice et Bob, reçoivent chacun des entrées telles que les valeurs de leurs fonctions sont différentes de celles de l'autre. Leur but est de trouver une position où leurs entrées diffèrent sans communiquer entre eux. On dit qu'ils gagnent ce jeu sur une paire d'entrées si les indices que les deux joueurs produisent correspondent et si c'est un endroit où leurs entrées diffèrent vraiment. Nous considérons ce jeu dans différents contextes, en donnant aux joueurs l'accès à l'aléa privé, à l'aléa public, à l'intrication partagée ou en leur permettant de jouer toute stratégie non signalante. Dans le chapitre 2, nous donnons une description formelle des jeux de certificats comme mesure de complexité dans ces modèles.

Nous étudions en détail les jeux de certificats avec un caractère aléatoire public dans le chapitre 3 où nous donnons des limites supérieures et inférieures à cette mesure en termes de mesures de complexité bien connues. Dans le chapitre 4, nous examinons d'autres variantes des jeux de certificats, comme ceux avec un caractère aléatoire privé, un enchevêtrement partagé ou des stratégies de non-signalisation, et nous donnons des limites supérieures et inférieures à ces mesures. Dans le chapitre 5, nous étudions les implications de nos résultats et montrons les séparations entre les jeux de certificats et d'autres mesures. Ces chapitres sont basés sur le manuscrit suivant :

> [35] Sourav Chakraborty, Anna Gál, Sophie Laplante, Rajat Mittal, and Anupa Sunny. "Certificate Games". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Vol. 251. Leibniz International Proceedings in Informatics (LIPIcs). 2023, 32:1–32:24. ISBN: 978-3-95977-263-1. DOI: 10.4230/LIPIcs.ITCS.2023.32. URL: https://drops.dagstuhl.de/opus/volltexte/2023/17535

Nous commençons la deuxième partie de cette thèse en examinant les résultats de la célèbre conjecture de sensibilité et en donnant une preuve alternative dans le chapitre 6. Dans sa preuve, Huang a utilisé de manière cruciale des graphes signés qui sont des graphes avec des signes positifs ou négatifs assignés aux bords. En utilisant ces graphes, il a pu montrer une borne inférieure sur le plus grand degré d'un sommet dans un graphe particulier.

Nous identifions ce qui a permis à la preuve de fonctionner et dans le Chapitre 7, nous donnons plus d'informations structurelles sur ce graphe autres que son plus grand degré. En particulier, nous analysons la relation structurelle entre les sommets à la distance 2 pour obtenir une limite supérieure sur le degré polynomial d'une fonction en termes de sa sensibilité à 0 et de sa sensibilité à 1. Ces résultats apparaissent dans l'article suivant :

[69] Sophie Laplante, Reza Naserasr, and Anupa Sunny. "Sensitivity Lower Bounds from Linear Dependencies". In: *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). 2020, 62:1–62:14. DOI: 10.4230/LIPIcs.MFCS.2020.62. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12732

La preuve de Huang de la conjecture de sensibilité donne une borne inférieure sur le degré du graphe lorsque le nombre de sommets est suffisamment grand. Nous pouvons renforcer ce résultat de deux manières. Tout d'abord, nous pouvons affaiblir l'hypothèse à tout graphe présentant une dépendance linéaire, quel que soit le nombre de sommets dans la dépendance linéaire. Deuxièmement, nous pouvons exploiter davantage la dépendance linéaire pour extraire des informations structurelles sur le graphe autres que son plus grand degré. Nous donnons les relations structurelles entre les sommets du sous-graphe induit à la distance 3 dans le chapitre 7.

Un ingrédient principal de la preuve de Huang était une affectation particulière des signes +/- aux arêtes d'un hypercube. En particulier, la signature utilisée par Huang a la propriété que tous les quatre-cycles sont négatifs (c'est-à-dire que le produit des signes des arêtes est négatif le long de tout cycle de longueur 4) et toute signature ayant cette propriété fonctionne pour la preuve de Huang. Dans le chapitre 8, nous examinons de plus près le graphe signé et étudions diverses autres signatures qui auraient pu conduire aux mêmes résultats tout en minimisant le nombre d'arêtes négatives utilisées. La question d'une signature qui minimise le nombre d'arêtes négatives utilisées s'avère être liée à l'un des problèmes d'Erdős sur la recherche du plus grand sous-graphe sans C_4 d'un hypercube. Ces résultats font partie du manuscrit suivant qui est en préparation :

[69] Sophie Laplante, Reza Naserasr, Anupa Sunny, and Zhouningxin Wang. "Sensitivity Conjecture and Signed Hypercubes". In preparation

Contents

List of Figures

List of Tables

For Renjan

Introduction

The fundamental goal of computer science is to develop methods or algorithms that perform computational tasks efficiently. Almost any activity that we do such as scheduling errands, planning how to get from point A to B or even arranging the cards in our hand while playing a game can be thought of as a computational task. The field of computational complexity tries to determine how fast can one hope to perform these tasks and not any better. We can reformulate these tasks into problems that have a "yes" or "no" answer, for instance, we can ask if all the errands for the day can be done in 2 hours. These can be modelled using Boolean functions that produce either 0 or 1 as the output. Our goal now is to find out how difficult it is to compute the output of a Boolean function.

Complexity measures try to capture the hardness of computing a function by quantifying what makes the computation hard. Ideally, we would like to find the time complexity of a function, i.e. the amount of time it takes for the computation. However, it is very hard to provide lower bounds on the time complexity of a function. As a result we turn our attention to weaker complexity measures. For instance, we can think of certain computational models where you restrict accesses to certain resources used in the computation. The number of accesses to these resources that are needed for the computation can be thought of as a measure of the hardness. As an example of this, let us think of a computational model where the input is hidden and you have access to a device that reveals the input at a position that you request/query. The number of positions that should be revealed to figure out the function value can be thought of as a measure of hardness. This complexity measure is called the query complexity of a function. One could also think of choosing to reveal positions of the input at random and guessing the output. If the guessed answer is correct most of the time, we are successful and the number of queries that had to be made is called the randomised query complexity. Similarly, one can also think of a quantum variant of query complexity where the queries to the input are made in superposition. Although it might seem that query complexity measures are overly simplistic and too weak, this is the setting that has been most successful in showing a definitive advantage for a quantum algorithm over classical ones. For instance, there exist problems that require exponentially many queries to solve classically (by a randomised algorithm), but only a constant number of queries quantumly [91, 39].

The field of complexity measures for Boolean functions is an old field of research and over the course of time several complexity measures have been studied. For instance, a Boolean function can be represented using a polynomial and the degree of the polynomial can be seen as a measure of its hardness. These measures are interesting on their own right, but it is much more interesting to compare them. For instance, does a large value on one measure imply a large value on the other? It is also interesting to see if certain measures are asymptotically equal to each other.

One of the most prominent open problems in this field had been the sensitivity conjecture. This conjecture hoped to characterise how different the two complexity measures sensitivity and block sensitivity can be. Sensitivity of a function measures how susceptible the output of a function is to the flip of a bit in its input. On the

other hand, for block sensitivity we are allowed to flip more than one bit at a time to change the function value. From their definitions, one can see that sensitivity is smaller than block sensitivity. The question that remained was how small can sensitivity be compared to block sensitivity for a function. The best known lower bound on sensitivity with respect to block sensitivity was logarithmic for decades until Huang in 2019 resolved this conjecture by showing that block sensitivity can at most be quartically larger than sensitivity [56]. It is not known if this is the best relation between the two measures as all the functions that have been studied so far have at most a quadratic separation between them.

This Thesis

In the first part of this thesis, we introduce a new measure of complexity called the "Certificate Game Complexity" and compare it with other well studied complexity measures of Boolean functions. To describe this measure, let us consider a game in which two players, say Alice and Bob, are each given inputs such that their function values are different from that of the other. Their goal is to figure out a position where their inputs differ without communicating with each other. They are said to win this game on a pair of inputs if the indices that both players output match and if it is a place where their inputs truly differ. We consider this game in various settings, by giving the players access to private randomness, public randomness, shared entanglement or by allowing them to play any non-signalling strategy. In Chapter 2, we give a formal description of Certificate Games as a complexity measure in these models. We study Certificate Games with public randomness in detail in Chapter 3 where we give upper and lower bounds on this measure in terms of well known complexity measures. In Chapter 4, we look at other variants of certificate games such as those with private randomness, shared entanglement or non-signalling strategies and give upper and lower bounds on these measures. In Chapter 5, we study the implications of our results and show separations between certificate games and other measures. These chapters are based on the following manuscript:

> [35] Sourav Chakraborty, Anna Gál, Sophie Laplante, Rajat Mittal, and Anupa Sunny. "Certificate Games". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Vol. 251. Leibniz International Proceedings in Informatics (LIPIcs). 2023, 32:1–32:24. ISBN: 978-3-95977-263-1. DOI: 10.4230/LIPIcs.ITCS.2023.32. URL: https://drops.dagstuhl.de/opus/volltexte/2023/17535

We start the second part of this thesis by looking into the celebrated sensitivity conjecture results and giving an alternate proof in Chapter 6. Huang in his proof crucially used signed graphs which are graphs with positive or negative signs assigned to the edges. Using these graphs, he was able to show a lower bound on the largest degree of a vertex in a particular graph. We identify what made the proof work and in Chapter 7, give more structural information about this graph other than its largest graph degree. In particular, we analyse the structural relation between vertices at distance 2 to obtain an upper bound on the polynomial degree of a function in terms of its 0-sensitivity and 1-sensitivity. These results appear in the following paper:

> [69] Sophie Laplante, Reza Naserasr, and Anupa Sunny. "Sensitivity Lower Bounds from Linear Dependencies". In: *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). 2020, 62:1–62:14. DOI: 10.

4230/LIPIcs.MFCS.2020.62. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12732

In addition to this, we give structural relations between vertices in the induced subgraph at distance 3 in Chapter 7. In Chapter 8, we take a closer look at the signed graph and study various other signatures that could have led to the same results while minimising the number of negative edges used. The question of a signature which minimises the number of negative edges used turns out be related to one of Erdős' problems on finding the largest C_4-free subgraph of a hypercube. These results are part of the following manuscript which is under preparation:

[69] Sophie Laplante, Reza Naserasr, Anupa Sunny, and Zhouningxin Wang. "Sensitivity Conjecture and Signed Hypercubes". In preparation

Chapter 1

Preliminaries

The main object of our study is a Boolean function which takes as inputs n-bit strings and produces an output of either 0 or 1, i.e. $f : \mathcal{D} \to \{0,1\}$ where the domain $\mathcal{D} \subseteq \{0,1\}^n$. When the function is defined for all n-bit strings, i.e. $\mathcal{D} = \{0,1\}^n$, it is said to be *total*. It is said to be *partial* when the function is defined over a subset of $\{0,1\}^n$, i.e. $\mathcal{D} \subset \{0,1\}^n$. For partial functions, we use f^{-1} to denote the domain of the function f, i.e. $f^{-1} = f^{-1}(0) \cup f^{-1}(1)$. In the rest of this thesis, we denote both partial and total Boolean functions by $f : \{0,1\}^n \to \{0,1\}$, and make the distinction only where it is necessary.

The indices of an n-bit string come from the set $[n] = \{1, 2, \cdots, n\}$ and let x_i denote the ith bit of a string x. For an input $x \in \{0,1\}^n$ and $S \subseteq [n]$, we write x^S to denote the string x with all the bits in positions corresponding to elements in S flipped. By an abuse of notation, we denote by x^i the input x with the ith bit flipped for an $i \in [n]$.

Hamming weight and distance: We denote the Hamming weight of a string x, which is the number of 1s in x, by $|x|$. The Hamming distance between two strings $x,\ y \in \{0,1\}^n$ is the number of indices where the two string differ and is denoted as $|x - y|$.

Boolean hypercube: The n-bit inputs to a Boolean function can be thought of as vertices of the Boolean hypercube which is denoted H_n. Two vertices in H_n are adjacent if their Hamming distance is 1, i.e. $u \sim v$ if $u = v^i$ for some $i \in [n]$ and this edge corresponds to the index i. A hypercube is a bipartite graph with a natural bipartition based on the parity of the Hamming weight of its vertices:

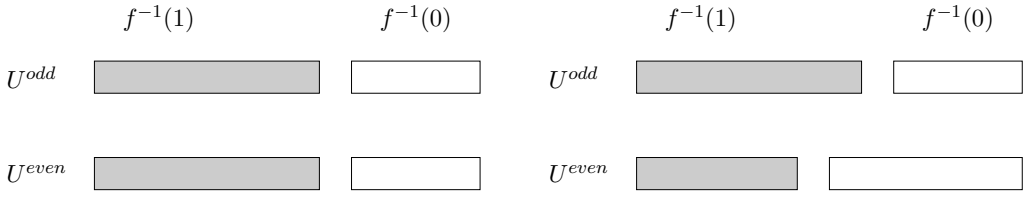
- vertices with an odd number of 1s, called *odd vertices* and denoted U_n^{odd}, form one part
- vertices with an even number of 1s, called *even vertices* and denoted U_n^{even}, form the other part.

The following definition will be useful in our discussion.

Definition 1.0.1. *A Boolean function is said to be* parity-balanced *if the number of even vertices that evaluate to 1 is the same as the number of odd vertices that evaluate to 1 (see Figure 1.1).*

Composition of functions: For any two (possibly partial) Boolean functions $f : \{0,1\}^n \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}$, we define the composed function $f \circ g$ on an input $x \in \{0,1\}^{nm}$ as follows:

$$f \circ g(x) = f(g(x_1, \cdots, x_m), \cdots, g(x_{(n-1)m+1}, \cdots, x_{nm}))$$



(A) Parity-balanced function (B) Not parity-balanced function

FIGURE 1.1: Parity-balance in a Boolean function can be thought of as the equality of the shaded regions in the figure above.

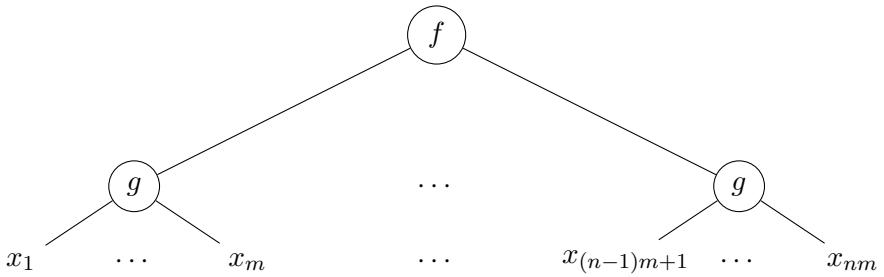


FIGURE 1.2: Composed function $f \circ g$ on an input $x \in \{0,1\}^{nm}$.

where the inputs to the intermediate functions are valid inputs, i.e. inputs to g are from g^{-1} and the input to the function f is from f^{-1}.

1.1 Complexity Measures

We will now define some well known complexity measures of Boolean functions. An important question regarding complexity measures is how these measures are related to each other and how it behaves under function composition. Most often, complexity measures are submultiplicative under composition, i.e. $\mathsf{M}(f \circ g) = O(\mathsf{M}(f) \cdot \mathsf{M}(g))$ for a complexity measure M. On the other hand, it is often not known if $\mathsf{M}(f \circ g) = \Omega(\mathsf{M}(f) \cdot \mathsf{M}(g))$ and these are given by composition theorems. A complexity measure M is said to *compose* perfectly if $\mathsf{M}(f \circ g) = \Theta(\mathsf{M}(f) \cdot \mathsf{M}(g))$.

1.1.1 Query complexity and its variants

In query complexity, we study a simple model of computation in which the input is unknown and we are given an oracle access to it. Query complexity looks at the number of queries made to the oracle to compute the function value at an input. In the following we discuss three variants of query complexity measures.

Deterministic Query complexity

In the classical setting, a query $i \in [n]$ to an oracle on an input x returns the value x_i. A deterministic query algorithm makes a series of queries to the oracle about the unknown input string x. Based on the oracle answers, it makes subsequent queries until it settles on the function value at x. Such an algorithm can be represented by a decision tree for which nodes are labelled by query indices $i \in [n]$ and edges are labelled by query answers which are either 0 or 1. Leaf nodes are also labelled by 0 or 1. On an input x, the query answers dictate the path from the root to a leaf. The decision tree is said to compute a function f if all the inputs that end up on a leaf have the function value labelled by it.

The *deterministic query complexity* of a Boolean function f is given by the minimal depth of the decision tree computing it. In other words, it is the minimum number of queries made to the oracle to evaluate the function value on the worst case input.

Randomized Query Complexity

In the randomized setting, the query algorithm uses randomness to decide on the queries to make. This can be represented by a probability distribution over deterministic decision trees. We say that a randomized decision tree computes a function f on an input x with probability at least $1-\epsilon$, if the decision tree sampled according to the distribution outputs $f(x)$ with probability at least $1-\epsilon$. We consider two measures on the depth of a randomised decision tree: worst-case depth and average depth. The worst-case depth of a randomised decision tree is the maximum depth of a deterministic decision tree in its support. The expected depth of a randomised decision tree is the expected number of queries made to compute f on an input for the worst case input. The *bounded error randomized query complexity* $\mathsf{R}(f)$ is the minimal worst-case depth of a randomised decision tree that computes the function f with probability at least $2/3$. The *zero-error randomised query complexity* $\mathsf{R}_0(f)$ is the minimal expected depth of a randomised decision tree that computes f correctly on all inputs, i.e. it makes zero error.

Quantum Query Complexity

In the quantum setting, the oracle $\mathcal{O}_x$ acts on a quantum state as follows:

$$\mathcal{O}_x|i\rangle = (-1)^{x_i}|i\rangle \ ,$$

where $i \in [n]$. A t-query quantum algorithm can be described by a series of operations $U_0\mathcal{O}_x \cdots \mathcal{O}_x U_t$ that act on an initial state $|\Psi_0\rangle$ where U_i are unitary transformations that are independent of the input x. The algorithm computes the function with probability at least $1-\epsilon$ if the final state when measured on an input x gives an output $f(x)$ with probability at least $1-\epsilon$ on all inputs $x \in f^{-1}$. The *bounded error quantum query complexity* of a function is the minimum number of quantum queries made by an algorithm that computes f with probability at least $2/3$.

It is worth noting that deterministic query complexity D and quantum query complexity Q were shown to compose perfectly [79, 95, 71, 86] even for partial functions. Although it was shown that randomised query complexity R does not compose for partial functions [21], it is not known if R composes for total functions.

1.1.2 Sensitivity and its variants

Sensitivity is one of the oldest complexity measures that has been studied on Boolean functions [41]. Given a Boolean function f, an input z is *sensitive at index i* if flipping the bit at index i (which we denote by z^i) changes the value of the function to $1-f(z)$.

Definition 1.1.1 (Sensitivity)**.** *For a (possibly partial) Boolean function f and $z \in f^{-1}$, $\mathsf{s}(f;z)$ is the number of sensitive indices of z. The sensitivity of f, denoted $\mathsf{s}(f)$, is the maximum $\mathsf{s}(f;z)$ over all $z \in f^{-1}$. The 0-sensitivity of f, denoted $\mathsf{s}_0(f)$, is the maximum sensitivity over inputs that evaluate to 0 on f. The 1-sensitivity of f, denoted $\mathsf{s}_1(f)$, is defined similarly.*

If B is a subset of indices, an input z is *sensitive to block B* if simultaneously flipping all the bits in B (which we denote by z^B) changes the value of the function to $1-f(z)$.

Definition 1.1.2 (Block sensitivity)**.** *For a (possibly partial) Boolean function f and $z \in f^{-1}$, $\mathsf{bs}(f;z)$ is the maximum number of disjoint sensitive blocks of z. The block sensitivity of f, $\mathsf{bs}(f) = \max_z \mathsf{bs}(f;z)$.*

If a block B is a minimal sensitive block for an input x, the size of the block is bounded above by sensitivity, i.e. $|B| \leq \mathsf{s}(f)$. This is true as the input x^B is sensitive to all the indices in B.

The *fractional block sensitivity* can be expressed as the following linear program whose integer solution corresponds to block sensitivity.

Definition 1.1.3 (Fractional block sensitivity)**.** *For any (possibly partial) Boolean function f and input $z \in f^{-1}$, let $\mathcal{B}_z$ denote the set of sensitive blocks of z, i.e. $\mathcal{B} = \{B \mid \exists z'\ f(z') = 1 - f(z) \text{ and } z^B = z'\}$. The fractional block sensitivity of f, $\mathsf{fbs}(f) = \max_{z \in f^{-1}} \mathsf{fbs}(f,z)$ where*

$$\mathsf{fbs}(f,z) = \max_{w} \sum_{B \in \mathcal{B}} w_{z,B}$$

$$\text{subject to } \sum_{\substack{B \in \mathcal{B} \\ i \in B}} w_{z,B} \leq 1 \quad \text{for all } i \in [n]$$

and w is a collection of variables such that $w_{z,B} \geq 0$.

Aaronson et al. [4] recently revived interest in a measure λ which was termed spectral sensitivity. It was first introduced by Koutsoupias [64] and can be viewed as a spectral relaxation of sensitivity.

Definition 1.1.4 (Spectral sensitivity)**.** *For a (possibly partial) Boolean function f, let F be the $|f^{-1}| \times |f^{-1}|$ matrix, with rows and columns indexed by elements of f^{-1}, defined by $F(x,y) = 1$ when $f(x) = 1 - f(y)$ and x, y differ in 1 bit, and $F(x,y) = 0$ otherwise. The spectral sensitivity $\lambda(f) = \|F\|$, where $\|\cdot\|$ is the spectral norm. In other words, $\lambda(f)$ is the largest eigenvalue of the matrix F.*

Note that F can also be taken to be a $|f^{-1}(0)| \times |f^{-1}(1)|$ matrix with rows indexed by elements of $f^{-1}(0)$ and columns by elements of $f^{-1}(1)$. It is easy to show that the two ways of defining F give the same spectral norm.

1.1.3 Certificate complexity and its variants

Certificate complexity is a measure which was first introduced by Vishkin and Wigderson to get a lower bound on the CRCW (Concurrent Read Concurrent Write) PRAM (Parallel Random Access Machines) model [97].

For a total Boolean function f, a *certificate* is a partial assignment of the bits of an input that forces the value of the function to be constant, regardless of the value of the other bits. A *certificate for input x* is a partial assignment consistent with x which is also a certificate for f.

Definition 1.1.5 (Certificate complexity)**.** *For any total Boolean function f and input x, $\mathsf{C}(f;x)$ is the size of the smallest certificate for x. The certificate complexity of the function is $\mathsf{C}(f) = \max_{0,1}\{\mathsf{C}^0(f), \mathsf{C}^1(f)\}$, where $\mathsf{C}^b(f) = \max_{x \in f^{-1}(b)}\{\mathsf{C}(f;x)\}$.*

Note that certificate complexity in the above definition has only been given for total functions. Definitions for certificate complexity with respect to partial functions and

additional details regarding the choice of definitions of sensitivity and block sensitivity for partial functions are included in Section 1.3.

Randomized certificate complexity RC was introduced by Aaronson as a randomized version of certificate complexity as follows [2].

Definition 1.1.6 (Randomized certificate complexity)**.** *For any (possibly partial) Boolean function f and $z \in f^{-1}$, $\mathsf{RC}^Z(f)$ is the minimum expected number of queries used by a randomized query algorithm that on an input $z' \in f^{-1}$ always accepts if $z' = z$ and rejects with probability at least $1/2$ if $f(z') = 1 - f(z)$. The randomised certificate complexity $\mathsf{RC}(f)$ is the maximum $\mathsf{RC}^Z(f)$ over all $z \in f^{-1}$.*

It was also shown that the version of randomised certificate complexity when the randomised query algorithm is allowed to make only non-adaptive queries (denoted RC_{na}) is equivalent to RC. The measure RC was subsequently shown to be equivalent (up to constant factors) to fractional block sensitivity and fractional certificate complexity [95, 47].

Definition 1.1.7 (Fractional certificate complexity)**.** *For any (possibly partial) Boolean function f, $\mathsf{FC}(f) = \max_{z \in f^{-1}} \mathsf{FC}(f, z)$ where*

$$\mathsf{FC}(f, z) = \min_{v} \sum_{i} v_{z,i}$$
$$\text{subject to} \sum_{i : z_i \neq z'_i} v_{z,i} \geq 1 \quad \text{for all } z' \text{ such that } f(z) = 1 - f(z')$$

and v is a collection of variables such that $v_{z,i} \geq 0$.

By rescaling the variables, we get an equivalent formulation,

$$\mathsf{FC}(f) = \min_{w} \max_{\substack{z, z' \in f^{-1} \\ f(z) = 1 - f(z')}} \frac{\sum_i w_{z,i}}{\sum_{i : z_i \neq z'_i} w_{z,i}},$$

where w is a collection of non-negative variables $w_{z,i}$.

The expectational certificate complexity was introduced as a quadratically tight lower bound on R_0, the zero-error randomised query complexity [58].

Definition 1.1.8 (Expectational certificate complexity)**.** *For any (possibly partial) Boolean function f,*

$$\mathsf{EC}(f) = \min_{w} \max_{z \in f^{-1}} \sum_{i \in [n]} w_{z,i}$$

where w is a collection of variables such that $0 \leq w_{z,i} \leq 1$ and $\sum_{i : z_i \neq y_i} w_{z,i} w_{z',i} \geq 1$ for all z and z' such that $f(z) = 1 - f(z')$.

Since the weights are between 0 and 1, we can associate to each i a Bernoulli variable. The players can sample from each of these variables independently and output the set of indices where the outcome was 1. The constraint says that the expected number of indices i in both sets that satisfy $z_i \neq z'_i$ should be bounded below by 1. The complexity measure is the expected size of the sets which can be thought of as behaving like a certificate.

The following relations are known to hold for any total Boolean function f.

Proposition 1.1.9 ([58])**.** $\mathsf{FC} \leq \mathsf{EC} \leq \mathsf{C} \leq O(\mathsf{R}_0) \leq O(\mathsf{EC}^2)$.

1.1.4 Adversary Bounds

Quantum adversary bounds were introduced to give lower bounds on quantum query complexity Q. The original version which is also called the unweighted version was introduced by Ambainis [9]. This was generalised in several works [15, 8, 102, 68] and were shown to be all equivalent by Spalek and Szegedy [93]. These are referred to as the *positive weight adversary bound*. We use the minimax formulation MM here.

Definition 1.1.10 (Positive weight adversary method, minimax formulation)**.** *For any (possibly partial) Boolean function f,*

$$\mathsf{MM}(f) = \min_{p} \max_{x \in f^{-1}(0), y \in f^{-1}(1)} \frac{1}{\sum_{i:x_i \neq y_i} \sqrt{p_{x,i} p_{y,i}}}$$

where p is taken over all families of nonnegative $p_{x,i} \in \mathbb{R}$ such that for all $x \in f^{-1}$, $\sum_{i \in [n]} p_{x,i} = 1$.

These were further generalised to negative weight adversary bounds [55] and it was shown that they were equivalent to the quantum query complexity Q [86, 71].

A classical version of the adversary bound was introduced as a lower bound for randomised query complexity R [1, 68]. Several formulations of the classical adversary bound were shown to be equivalent to fractional block sensitivity fbs for total functions [11]. In the case of partial functions, there is an unbounded separation between fbs and the classical adversary bound formulations. We use the minimax formulation of the classical adversary bound which is the largest of all the formulations.

Definition 1.1.11 (Classical Adversary Bound)**.** *For any (possibly partial) Boolean function f, the minimax formulation of the Classical Adversary Bound is as follows:*

$$\mathsf{CMM}(f) = \min_{p} \max_{\substack{x,y \in S \\ f(x) \neq f(y)}} \frac{1}{\sum_{i:x_i \neq y_i} \min\{p_x(i), p_y(i)\}}$$

where $\{p_x\}_{x \in S}$ is a probability distribution over $[n]$.

Proposition 1.1.12 ([4, 95, 47, 67])**.** *For any (possibly partial) Boolean function f,*

$$\lambda(f) \leq \mathsf{s}(f) \leq \mathsf{bs}(f) \leq \mathsf{FC}(f) \text{ and } \lambda(f) \leq \mathsf{MM}(f)$$

1.1.5 Polynomial degree

For a $v \in \{0,1\}^n$, we define an indicator polynomial $P_v : \mathbb{R}^n \to \mathbb{R}$ as follows:

$$P_v = \prod_{\substack{i \in [n] \\ v_i = 1}} x_i \prod_{\substack{i \in [n] \\ v_i = 0}} (1 - x_j).$$

This polynomial has the property that when restricted to the elements of $\{0,1\}^n$, it takes the value 1 on v and 0 everywhere else. Note that this polynomial is multilinear i.e., every variable appears with a degree at most 1. The degree of this indicator polynomial is n and the coefficient of the highest degree term $x_1 x_2 \dots x_n$ is either $+1$ or -1 depending on the parity of v.

A polynomial $p : \mathbb{R}^n \mapsto \mathbb{R}$ *represents* a Boolean function f if for all $x \in \{0,1\}^n$, $p(x) = f(x)$. Every Boolean function can be represented by a multilinear polynomial P_f which is the sum of polynomials P_v such that $f(v) = 1$ i.e.,

$$P_f = \sum_{v;f(v)=1} P_v \ .$$

We now define the *degree* of a Boolean function f, denoted $\deg(f)$, as the minimum degree of a polynomial that represents f. It can be seen that there is a unique multilinear polynomial that represents p which is given by P_f (a proof of which is given in [31]). Observe that a Boolean function f has full degree (i.e. degree n) if and only if it is not parity-balanced (see Definition 1.0.1). This can be found in [31] and was attributed to Yao and Shi.

The *approximate degree* of a Boolean function $f : \{0,1\}^n \to \{0,1\}$, denoted $\widetilde{\mathsf{deg}}(f)$, is defined as the minimum degree of a polynomial $p : \mathbb{R}^n \mapsto \mathbb{R}$ that satisfies $|f(x) - p(x)| \leq 1/3$ for all $x \in \{0,1\}^n$. The polynomial method introduced by Beals et al. [17] uses approximate degree to give a lower bound on the quantum query complexity i.e, $\mathsf{Q} = \Omega(\widetilde{\mathsf{deg}})$.

1.1.6 Complexity measures inspired by R

A composition theorem for randomised query complexity R of total Boolean functions is not known despite multiple attempts [49, 22, 13, 46, 16, 21]. These attempts have however produced interesting complexity measures that achieve partial progress towards a composition theorem. Some of these measures are described below.

Sabotage Complexity

The *sabotage complexity* of a function f, denoted $\mathsf{RS}(f)$, is defined using a concept of *sabotaged* inputs $P_f \subseteq \{0,1,*\}^n$ which is the set of all partial assignments of a function f consistent with a $0-$input and a $1-$input. Let $P_f^\dagger$ be defined similarly with the symbol $*$ being replaced by $\dagger$.

Definition 1.1.13 (Sabotage Complexity [22])**.** *Given a (possibly partial) function f, we define a partial function $f_{sab} : P_f \cup P_f^\dagger \to \{0,1\}$ where $f_{sab}(x) = 1$ if $x \in P_f$ and $f_{sab}(x) = 0$ if $x \in P_f^\dagger$. The sabotage complexity is defined as the bounded-error randomized query complexity of f_{sab} i.e. $\mathsf{RS}(f) = R(f_{sab})$.*

It was shown that sabotage complexity composes and that it can be used to give a composition theorem for R with a loss in terms of the inner function, i.e. for any (possibly partial) Boolean functions f and g, $\mathsf{R}(f \circ g) = \Omega(\mathsf{R}(f)\ \mathsf{RS}(g))$ [22].

noisyR Complexity

The noisyR measure was instrumental in showing that R does not compose for partial functions [21]. This measure is based on a *noisy oracle* model. A *noisy oracle*, on a query to a bit b made with a bias γ, returns b with probability $\frac{1+\gamma}{2}$, and $1-b$ with probability $\frac{1-\gamma}{2}$. *Noisy randomised query algorithms* have access to a noisy oracle, and a query made with a bias γ costs γ^2. It is said to compute a function f with probability at least $1-\epsilon$ if its output on an input x equals $f(x)$ with probability at least $1-\epsilon$ for all inputs $x \in f^{-1}$.

Definition 1.1.14 (noisyR [21])**.** *For any (possibly partial) Boolean function f, the cost of the noisy randomised algorithm for an input x is the expected cost of all the queries made to compute $f(x)$. The cost of the algorithm computing f is the maximum cost needed to compute the function value for any input. $\mathsf{noisyR}(f)$ is defined as the minimum cost of an algorithm computing f with probability at least $2/3$.*

A composition theorem for R with a loss in terms of the outer function was shown using noisyR, i.e. for any (possibly partial) Boolean functions f and g, $\mathsf{R}(f \circ g) = \Omega(\mathsf{noisyR}(f)\ \mathsf{R}(g))$.

The relations between various measures is illustrated in Figure 1.3.

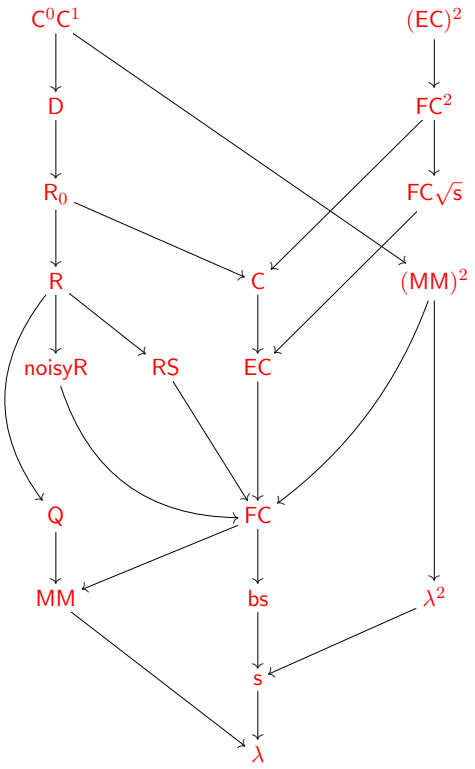


FIGURE 1.3: Some relations between complexity measures for total functions. An arrow from A to B indicates that for every total Boolean function f, $B(f) = O(A(f))$. References are omitted from the diagram for space considerations. Most references can be found in the tables in [99, 4] and we cite others elsewhere in this thesis. Known relations about EC are given in [58], and $\mathsf{FC} = O((\mathsf{MM})^2)$ is implicit in [12]. Fractional certificate complexity FC is equal to fractional block sensitivity and to randomized certificate complexity RC (up to multiplicative constants). MM is the minimax formulation of the positive adversary method. $\mathsf{MM} = O(\mathsf{FC})$ is proven in [65] and $\mathsf{MM}^2 \leq O(\mathsf{C_0C_1})$ was shown in [93].

1.2 Some useful functions

Interesting examples of total and partial Boolean functions are very important to understand the relations between various complexity measures. In fact, constructing interesting functions is a commonly used technique to prove separation between measures. A number of interesting functions have been constructed for this purpose in previous works [47, 3, 14, 33, 89]. We will be using some of them in this thesis and they are defined in this section.

Symmetric functions are those whose output depends only on the Hamming weight of the input. OR and AND are symmetric functions that are probably the first ones to be studied for any complexity measure.

Definition 1.2.1 (OR and AND)**.** *The* OR *function takes as input an n-bit string and evaluates it to 0 if it is all zeroes and to 1 otherwise. Similarly the* AND *evaluates a string to 1 only if it is all ones.*

$$\mathsf{OR}_n(x) = \bigvee_{i=1}^{n} x_i$$
$$\mathsf{AND}_n(x) = \bigwedge_{i=1}^{n} x_i \ .$$

The bounds on OR_n follow from $\lambda(\mathsf{OR}) = \Theta(\sqrt{n})$ [4], $\mathsf{Q}(\mathsf{OR}_n) = O(\sqrt{n})$ [51] and the observation that $\mathsf{s}(\mathsf{OR}_n) = \Theta(n)$.

Parity is another symmetric function that is widely studied.

Definition 1.2.2 (Parity)**.** *The parity function on an n-bit input string outputs the parity of ones in the string, i.e.* $\mathsf{Parity} : \{0,1\}^n \to \{0,1\}$ *is defined for an input $z \in \{0,1\}^n$ as,*

$$\mathsf{Parity}(z) = \bigoplus_{i \in [n]} z_i \ .$$

One of the most useful properties of the Parity function is that λ, which is the smallest measure we have considered in this thesis, is the largest it can be i.e., $\lambda(\mathsf{Parity}) = \Theta(n)$ [77].

$\mathsf{Tribes}_{m,n} = \mathsf{OR}_m \circ \mathsf{AND}_n$ is a non-symmetric function, made by composing the symmetric functions OR and AND.

Definition 1.2.3 (Tribes)**.** *The $\mathsf{Tribes}_{m,n}$ function is a composition of two functions, $\mathsf{Tribes}_{m,n} = \mathsf{OR}_m \circ \mathsf{AND}_n$ i.e., $\mathsf{Tribes}_{m,n} : \{0,1\}^{mn} \to \{0,1\}$ is defined as*

$$\mathsf{Tribes}_{m,n}(x) = \bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} x_{i,j}.$$

It can be verified that $\mathsf{C}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(\sqrt{n})$, and $\lambda(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \mathsf{Q}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(\sqrt{n})$ follows from composition [4, 71]. We also have $\mathsf{R}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$ from Jain and Klauck [57]. The bounds for other measures follow from these.

The Rubinstein function was introduced to show that the block sensitivity of a function can be quadratically larger than its sensitivity [89].

Definition 1.2.4 (Rubinstein [89])**.** *The Rubinstein function $f_R : \{0,1\}^n \to \{0,1\}$ is defined by dividing the input into $\sqrt{n}$ blocks of size $\sqrt{n}$ each. The function evaluates to 1 if there exists a block with exactly 2 consecutive ones and zeroes everywhere else in the block. On all other inputs x, $f_R(x) = 0$.*

The Rubinstein function has sensitivity $\Theta(\sqrt{n})$ and block sensitivity $\Theta(n)$ [89].

The function GSS_1 was introduced to give a quadratic separation between FC and C.

Definition 1.2.5 (Gilmer-Saks-Srinivasan [47])**.** *The function GSS_1 is defined on $\{0,1\}^{n^2}$ bits as $\mathsf{GSS}_1 := \mathsf{OR} \circ g$ where the function g is constructed probabilistically on n bits by picking $2^{n/50}$ inputs randomly (with replacement) and setting their outputs to 1. All the other inputs are set to 0.*

It was argued that with high probability a random function g constructed as above leads to a GSS_1 function with the following properties: $\mathsf{C}(\mathsf{GSS}_1) = \Theta(n^2)$, $\mathsf{EC}(\mathsf{GSS}_1) = \Theta(n)$ and $\mathsf{FC}(\mathsf{GSS}_1) = \Theta(n)$ [47, 58].

Ambainis constructed a function to show that the positive adversary bound can give better lower bounds on quantum query complexity than the polynomial method.

Definition 1.2.6 (Ambainis [8]). *The Ambainis function f_A is defined on inputs with 4-bits. The function evaluates the inputs* 0000, 0001, 0011, 0111, 1111, 1110, 1100, 1000 *to 1 and the rest to 0. In other words, $f_A(x) = 1$ if $x_1 \leq x_2 \leq x_3 \leq x_3$ or $x_1 \geq x_2 \geq x_3 \geq x_4$. The function is composed with itself for larger inputs, i.e. $f_A^d : \{0,1\}^{4^d} \to \{0,1\}$*

$$f_A^d = f_A \circ f_A^{d-1},$$

where $f_A^1 = f_A$.

Some of the important properties of the Ambainis function are: $\mathsf{s}(f_A^d) = 2^d$, $\widetilde{\mathsf{deg}}(f_A^d) = 2^d$ while the positive adversary bound $\mathsf{MM}(f_A^d) = 2.5^d$ and quantum query complexity $\mathsf{Q}(f_A^d) = 2.5135^d$ [8, 67, 55].

The function BKK was constructed by Aaronson et al. to show a quadratic separation (up to log factors) between certificate complexity and quantum query complexity.

Definition 1.2.7 ([3]). *The function $\mathsf{BKK}_{n^2,k} : \{0,1\}^{n^2} \to \{0,1\}$ is defined using two functions* Block k-sum *and* k-sum *both of which act on n bit strings. The* Block k-sum *function on n bits is defined by splitting the input on n bits to blocks of size $10k \log n$ each. A block is said to be balanced if there are an equal number of zeroes and ones in it, and each of the balanced blocks represents an element in an alphabet M of size n^k. The function* Block k-sum *evaluates an input to 1 if and only if there exists k balanced blocks whose corresponding elements in M sum to 0 and all the other blocks have at least as many ones as zeroes. The function k-sum $: [M]^n \to \{0,1\}$ evaluates to 1 on an input x if there exist k elements in the input string $x_1, x_2, \cdots, x_n \in [M]$ that sum to 0. The function* BKK *is defined as* Block k-sum *composed with a Boolean version of the k-sum function.*

When $k = \log n$, $\mathsf{C}(\mathsf{BKK}) = O(n \log^3 n)$ and $\mathsf{Q}(\mathsf{BKK}) = \Omega\left(\frac{n^2}{\log^5 n}\right)$ which gives a quadratic separation between certificate complexity and quantum query complexity up to log factors.

In certain cases, some complexity measures are not known to be separated by total functions but only by partial functions. This is the case for the classical adversary bound and fractional certificate complexity which are separated by a partial function called "Greater than Half".

Definition 1.2.8 (Greater than Half [11]). *The "Greater than Half" function is defined only on n bit strings that have Hamming weight 1. The function evaluates to 1 on an input x if the position i where the input bit is 1 is in the second half of the string and is zero if i is in the first half of the string, i.e. $\mathsf{GTH} : \{0,1\}^n \to \{0,1\}$ is defined as $\mathsf{GTH}(x) = 1$ if $x_i = 1$ at $i > n/2$ and $\mathsf{GTH}(x) = 0$ if $x_i = 1$ at $i \leq n/2$.*

For the GTH function, FC is constant and $\mathsf{CMM}(\mathsf{GTH}) = \Theta(n)$ which provides an arbitrary separation between these measures [11].

Some complexity measures for the functions we consider is compiled in Table 1.1.

1.3 Additional definitions for partial functions

Extending the definition of certificates to partial functions is slightly complicated. For instance, it is not clear if a 1-certificate for a 1-input should distinguish it from all 0-inputs, or if it should exclude all inputs whose outputs are not defined as well. Taking

Function	λ	s	bs	FC	MM	Q	R	EC	C
OR_n	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Parity_n	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
$\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
f_R	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n)$			$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
GSS_1		$\Theta(n)$	$\Theta(n)$	$\Theta(n)$				$\Theta(n)$	$\Theta(n^2)$
f_A		$\Theta(2^d)$			$\Theta(2.5^d)$	$\Theta(2.5135^d)$			
$\mathsf{BKK}_{n^2,\log n}$						$\Omega\left(\frac{n^2}{\log^5 n}\right)$			$O(n\log^3 n)$

TABLE 1.1: Some of the commonly referred total functions and their complexity measures.

this into account, for a partial Boolean function $f:\{0,1\}^n\to\{0,1,*\}$ it is natural to define the measures $\mathsf{C}^0(f)$, $\mathsf{C}^1(f)$, as well as $\mathsf{C}^{\{0,*\}}(f)$ and $\mathsf{C}^{\{1,*\}}(f)$ as follows:

Definition 1.3.1. *For $f:\{0,1\}^n\to\{0,1,*\}$ and $b\in\{0,1\}$, a partial assignment α is a b-certificate for $x\in f^{-1}(b)$ if α is consistent with x and $f(x')=b$ for any x' consistent with α.*

For $f:\{0,1\}^n\to\{0,1,*\}$ and $b\in\{0,1\}$, a partial assignment α is a $\{b,*\}$-certificate for $x\in f^{-1}(b)$ if α is consistent with x and $f(x')\in\{b,*\}$ for any x' consistent with α.

For $b\in\{0,1\}$ and $x\in f^{-1}(b)$, $\mathsf{C}^b(f;x)$ is the size of the smallest b-certificate for x and $\mathsf{C}^b(f)=\max_{x\in f^{-1}(b)}\{\mathsf{C}^b(f;x)\}$.

For $b\in\{0,1\}$ and $x\in f^{-1}(b)$, $\mathsf{C}^{\{b,*\}}(f;x)$ is the size of the smallest $\{b,*\}$-certificate for x and $\mathsf{C}^{\{b,*\}}(f)=\max_{x\in f^{-1}(b)}\{\mathsf{C}^{\{b,*\}}(f;x)\}$.

Note that while one can think of 0-certificates for x certifying that $f(x)=0$, a $\{0,*\}$-certificate for x certifies that $f(x)\neq 1$. We also note that in the definition of $\mathsf{C}^{\{b,*\}}(f)$ we take the maximum over $x\in f^{-1}(b)$ and we do not include inputs x where the function is not defined i.e., where $f(x)=*$.

The above definitions are fairly straightforward and natural, but it is not immediately clear how to define $\mathsf{C}(f)$ for partial functions. We use the following definition.

Definition 1.3.2. *For a partial Boolean function $f:\{0,1\}^n\to\{0,1,*\}$ we define*

$$\mathsf{C}(f)=\max\{\mathsf{C}^{\{0,*\}}(f),\mathsf{C}^{\{1,*\}}(f)\}$$

and $\mathsf{C}'(f)=\max\{\mathsf{C}^0(f),\mathsf{C}^1(f)\}$.

Notice that $\mathsf{C}(f)\leq\mathsf{C}'(f)$ for any f and for total functions $\mathsf{C}(f)=\mathsf{C}'(f)$. However, for partial functions $\mathsf{C}(f)$ can be much smaller than $\mathsf{C}'(f)$. The "Greater than Half" function (Definition 1.2.8) is an example of a partial function on n bits with $\mathsf{C}(f)=O(1)$ while $\mathsf{C}'(f)=\Theta(n)$. For any $x\in f^{-1}(b)$, the place where its input is 1 certifies that its output is either b or $*$, but the certificate has to be significantly bigger if it has to exclude inputs with undefined outputs.

It turns out that some results known for total functions remain valid for partial functions with respect to $\mathsf{C}(f)$ but not $\mathsf{C}'(f)$ and vice versa. As a result, it is important to distinguish between the two versions. We prefer to use this definition for $\mathsf{C}(f)$ since with this definition, $\mathsf{C}(f)$ remains a lower bound on deterministic query complexity (and on R_0 as well) for partial functions. On the other hand, it is easy to construct partial functions with deterministic query complexity $O(1)$ but $\mathsf{C}'(f)=\Omega(n)$. Some of our results for total functions involving $\mathsf{C}(f)$ no longer hold for partial functions, even though they remain valid with respect to $\mathsf{C}'(f)$.

A property of certificates often exploited in proofs is that every 0-certificate must intersect (and contradict) every 1-certificate and this remains the case for partial functions. However, this property no longer holds for $\{0,*\}$ versus $\{1,*\}$-certificates. Proofs based on this property remain valid for partial functions with respect to $\mathsf{C}'(f)$, but may no longer hold for partial functions with respect to $\mathsf{C}(f)$. An important example where this happens is the result that $\mathsf{EC}(f) \leq \mathsf{C}(f)$ [58]. This result does not hold for partial functions, as shown by the "Greater than Half" function (Definition 1.2.8) which has $\mathsf{C}(f) = O(1)$ and $\mathsf{EC}(f) = \Theta(n)$, but remains valid with respect to $\mathsf{C}'(f)$.

For sensitivity (block sensitivity) of partial functions, we consider an input x in the domain $f^{-1}(0) \cup f^{-1}(1)$ to be sensitive to an index (or to a block) if flipping it gives an input where f is defined and takes the complementary value $1 - f(x)$. We do not consider an input to be sensitive to an index (or block) if flipping it gives an input where f is undefined. Notice that with our definition, sensitivity can be 0 even for non-constant partial functions. Our definition preserves equality between fractional block sensitivity and fractional certificate complexity (as defined above).

Chapter 2

An Introduction to Certificate Games

To better understand the relation between various complexity measures, we introduce a new complexity measure called the certificate game complexity based on the Karchmer-Wigderson relation of a Boolean function. This relation, introduced by Karchmer and Wigderson [59], has been extensively studied in communication complexity.

Definition 2.0.1 (Karchmer-Wigderson relation [59]). *Let $f : \{0,1\}^n \to \{0,1\}$ be a (possibly partial) Boolean function. The Karchmer-Wigderson relation $R_f \subseteq f^{-1}(0) \times f^{-1}(1) \times [n]$ is defined as $R_f \coloneqq \{(x, y, i) : x_i \neq y_i\}$.*

It was shown by Karchmer and Wigderson [59] that the communication complexity of R_f is equal to the circuit depth of f. As a matter of convention, for a Boolean function f we will use x to denote an input in $f^{-1}(0)$ and y to denote an input in $f^{-1}(1)$ (unless otherwise stated). We study the following 2-player *certificate game*, where the goal of the players is to solve the Karchmer-Wigderson relation in a zero-communication setting.

Definition 2.0.2 (Certificate game). *Let $f : \{0,1\}^n \to \{0,1\}$ be a (possibly partial) Boolean function. One player is given $x \in f^{-1}(0)$ and the other player is given $y \in f^{-1}(1)$. Their goal is to produce a common index i such that $x_i \neq y_i$, without communicating with each other.*

We look at how well the players can solve this task in several zero-communication settings. We consider four models: (i) when they only have private coins, (ii) when they share a public random source, (iii) when they share an entangled quantum state (also called the quantum model) that does not depend on their inputs and (iv) when we allow all non-signalling strategies which we describe in Section 2.2.4. In all these models, we consider the probability of success that they can achieve, for the best strategy and the worst case input pair. The multiplicative inverse of the winning probability is called the certificate game complexity of the function (CG for the private coin model, $\mathsf{CG}^{\mathrm{pub}}$ for the public coin model, CG^* for the shared entanglement model and $\mathsf{CG}^{\mathsf{ns}}$ for the non-signalling model).

To illustrate how to achieve such a task without communication, we consider the following simple strategy. Let f be a total function whose 0-certificate complexity is c_0 and whose 1-certificate complexity is c_1. On an input $x \in f^{-1}(0)$, Alice can output a random i in a minimal 0-certificate for x (and similarly for Bob from a minimal 1-certificate for y). Since the certificates intersect, the probability that they output the same index is at least $\frac{1}{c_0 \cdot c_1}$. This shows that $\mathsf{CG}(f) \leq \mathsf{C}^0(f) \cdot \mathsf{C}^1(f)$. This simple upper bound is tight for many functions including OR and Parity, but there are other examples where it can be much smaller, and it is interesting to see what other upper

and lower bounds can apply. We will also see that access to shared randomness can significantly reduce the complexity.
In this chapter we give formal definitions of certificate game complexity when the players are allowed to have private coin, public coin, shared entanglement or non-signalling strategies. In the following chapters, we will show that the certificate game complexity measures in the four different models hold a pivotal position with respect to other measures, making them good candidates for proving strong lower and upper bounds on various measures. Their operational interpretation as winning probabilities of certificate games makes them convenient for proving upper bounds. Furthermore, the public coin and non-signalling versions are linear programs and hence their dual formulation is convenient for proving lower bounds.

2.1 Motivation for certificate games

The two main ingredients in our certificate games are two-player zero-communication games and the Karchmer-Wigderson relation. Two-player zero-communication games have been studied in many different contexts. They are called two-prover games in the context of parallel repetition theorems that are central to the study of Probabilistically Checkable Proofs (PCPs) and the Unique Games Conjecture [61, 85, 19]. They also appear under the name of zero-communication protocols in the context of communication and information complexity. Finally, they are known as local or quantum games in the study of quantum nonlocality, an extensive field motivated by the study of quantum entanglement and the relative power of quantum over classical behaviours. Quantum behaviours are modeled by two parties making measurements on a shared bipartite quantum state, whereas in the classical setup, the two parties can share "hidden variables" (shared randomness). There has been immense work, for instance, on simulating quantum behaviours with various resources such as communication, post-selection, noise and more [26, 7, 32, 5, 74, 48, 94, 96, 90]. There are also strong connections between finding separations between quantum and classical communication complexity, and between quantum and classical zero-communication games [66, 30]. A survey on quantum non-locality can be found in references [28, 82], and on the interactions between communication complexity and nonlocality in reference [29].

The Karchmer-Wigderson relation R_f appears in many contexts in the study of complexity measures, including the Adversary bound on quantum query complexity, and its variants (see Section 1.1.4) [9, 93]. It is key to understanding how hard a function is and captures the intuition that if one is to distinguish the 0-instances from the 1-instances of a function, then some i in the relation has to play a key role in computing the function. Another measure where the Karchmer-Wigderson relation appears implicitly is randomized certificate complexity (RC) defined by Aaronson [1]. The non-adaptive version of randomized certificate complexity (RC_{na}) can be viewed as a one-player game where the player with an input x should output an index i. The player wins against an adversary with an input y (with $f(y) \neq f(x)$) if $x_i \neq y_i$ i.e, $(x, y, i) \in R_f$. $\mathsf{RC}_{na}(f, x)$ is the multiplicative inverse of the probability of winning the nonadaptive game for x, against the worst y. $\mathsf{RC}_{na}(f)$ is the maximum over all inputs x of $\mathsf{RC}_{na}(f, x)$ and it can be expressed as a linear program. This was shown to be equal (asymptotically) to fractional certificate complexity whose dual formulation gives the fractional block sensitivity [95, 47]. These are also equal to the classical adversary bound for total functions.

2.2 Formal Definitions of Certificate Games

In this section, we give the formal definitions of the Certificate Game complexity measures that we introduce.
A two-player game G is given by a relation $R(x, y, a, b) \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{A} \times \mathcal{B}$, where $x \in \mathcal{X}$ is the first player's input and $y \in \mathcal{Y}$ is the second player's input. The players output a pair of values $a \in \mathcal{A}$ and $b \in \mathcal{B}$ respectively, and they win if $R(x, y, a, b)$ holds.

A *deterministic strategy* is a pair of functions $A : f^{-1}(0) \to \mathcal{A}$ and $B : f^{-1}(1) \to \mathcal{B}$ for Alice and Bob that depend solely on their respective inputs. A *randomized strategy with private randomness* is the product of two mixed individual strategies. A *randomized strategy with shared randomness* is a mixture of pairs of deterministic strategies. A *quantum* or *shared entanglement strategy* is given by a shared bipartite state that does not depend on the input, and a measurement for Alice indexed by her input and one for Bob indexed by his input.

For any strategy, we write $p(a, b|x, y)$ to denote the probability that the players output a and b when their inputs are x and y respectively. The marginal distribution of Alice's output is $p(a|x, y) = \sum_b p(a, b|x, y)$ and $p(b|x, y) = \sum_a p(a, b|x, y)$ is Bob's marginal distribution.

Non-signalling is a notion that comes from quantum games which says that if players are spatially separated, they cannot convey information to each other instantaneously. All the types of strategies described above satisfy the non-signalling condition.

Definition 2.2.1 (Non-signalling strategy)**.** *Let $p(a, b|x, y)$ be the probability that the players on input x and y output a and b respectively. We say that p is non-signalling if $p(a|x, y) = p(a|x, y')$ and $p(b|x, y) = p(b|x', y)$ for all inputs x, x', y, y' and all outcomes a, b.*

Since nonsignalling implies that Alice's output does not depend on Bob's input, Alice's marginal distribution can be denoted by $p(a|x)$ and $p(b|y)$ for Bob.

Surprisingly, non-signalling strategies are characterized by *affine combinations* of local deterministic strategies that lie in the positive orthant as stated below (Propositio 2.2.2). This has been known since the 1980s [45, 84, 62, 98] and a more recent proof is given in [83].

Proposition 2.2.2 (Characterization of non-signalling strategies)**.** *A strategy p is non-signalling if and only if it is given by a family of coefficients $\lambda = \{\lambda_{AB}\}$ (not necessarily nonnegative) where AB ranges over pairs (A, B) of deterministic strategies such that $p(a, b|x, y) = \sum_{AB:A(x)=a,B(y)=b} \lambda_{AB}$, and λ satisfies the constraints $\sum_{AB} \lambda_{AB} = 1$, and $\sum_{AB:A(x)=a,B(y)=b} \lambda_{AB} \geq 0$ for all a, b, x, y.*

We will now formally define some variants of certificate games.

2.2.1 Certificate games with private coins

In case of private coins, a randomized strategy for each player amounts to assigning for every input $x \in \{0, 1\}^n$, a probability $p_{x,i}$ of producing i as its outcome for each $i \in [n]$.

Definition 2.2.3 (Private coin certificate game complexity)**.** *For any (possibly partial) Boolean function f,*

$$\mathsf{CG}(f) = \min_{p} \max_{x,y \in f^{-1}(0) \times f^{-1}(1)} \frac{1}{\omega(p; x, y)},$$

where p is a collection of nonnegative variables $\{p_{x,i}\}$, with $x \in f^{-1}(0) \cup f^{-1}(1)$ and $i \in [n]$ that satisfy the constraint:

$$\sum_{i \in [n]} p_{x,i} = 1 \qquad \forall x \in f^{-1}(0) \cup f^{-1}(1).$$

The winning probability $\omega(p; x, y)$ is the probability that both players output a common index i that satisfies the Karchmer-Wigderson relation R_f, i.e.

$$\omega(p; x, y) = \sum_{i : x_i \neq y_i} p_{x,i} p_{y,i}.$$

We will see in later chapters that it is possible for CG of a function to be quadratically larger than the number of bits the function is defined on. In particular, we will see that $\mathsf{CG}(\mathsf{Parity}_n) = \Omega(n^2)$.

2.2.2 Certificate games with public coins

When the players share randomness, a *public-coin randomized strategy* is a distribution over pairs (A, B) of deterministic strategies. We assign a nonnegative variable p_{AB} to each pair of strategies (A, B) and require that they sum to 1.

Definition 2.2.4 (Public coin certificate game complexity)**.** *For any (possibly partial) Boolean function f,*

$$\mathsf{CG}^{pub}(f) = \min_{p} \max_{x,y \in f^{-1}(0) \times f^{-1}(1)} \frac{1}{\omega^{pub}(p; x, y)},$$

where p is a collection of nonnegative variables $\{p_{AB}\}$ for each pair of strategies (A, B) that satisfy $\sum_{(A,B)} p_{AB} = 1$ and

$$\omega^{pub}(p; x, y) = \sum_{(A,B) \text{ correct on } x,y} p_{AB}.$$

We say that a pair of strategies (A, B) is correct on x, y *if $A(x) = B(y) = i$ and $x_i \neq y_i$.*

We note that maximizing the winning probability in the worst case can be written as a linear program in the public coin variant. This gives us a dual formulation and this form will be more convenient when proving lower bounds (since it becomes a minimization problem, and we are considering its multiplicative inverse). The dual variables $\mu_{x,y}$ can be thought of as a hard distribution on pairs of inputs, and the objective function is the μ-size of the largest set of input pairs where any deterministic strategy is correct.

Proposition 2.2.5 (Dual formulation of CG^{pub})**.** *For a two-player certificate game G_f corresponding to a Boolean function f,*

$$\mathsf{CG}^{pub}(f) = 1/\omega^{pub}(G_f),$$

where the winning probability $\omega^{pub}(G_f)$ *is given by the following linear program.*

$$\begin{aligned}\omega^{pub}(G_f) = \quad & \min_{\delta,\mu} \quad \delta \\ \text{such that} \quad & \sum_{x,y:\ A,B \text{ correct on } x,y} \mu_{x,y} \leq \delta \quad \text{for every deterministic strategy } A,B \\ & \sum_{x,y} \mu_{xy} = 1, \quad \mu_{x,y} \geq 0,\end{aligned}$$

where $\mu = \{\mu_{x,y}\}_{x\in f^{-1}(0),\, y\in f^{-1}(1)}$.

From the definitions, it is easy to show that randomised certificate complexity RC is a lower bound on $\mathsf{CG}^{\mathrm{pub}}$. A successful RC strategy can be obtained by one of the players playing according to a $\mathsf{CG}^{\mathrm{pub}}$ strategy and ignoring the second player. A more detailed proof using the FC definition will be presented in Proposition 3.5.1.

It is also easy to see that unlike the private coin variant CG, $\mathsf{CG}^{\mathrm{pub}}$ of a function on n bits is always bounded above by n. This holds as the players can output the same index $i \in [n]$ picked at random using shared randomness. Since this random index will have a probability at least $1/n$ of being a position where their inputs differ, we get this naive upper bound. In fact, we will see in Chapter 3 that $\mathsf{CG}^{\mathrm{pub}}$ is bounded above by R, C and EC. These upper bounds will also hold for all the variants of certificate games we define below.

2.2.3 Certificate games with quantum strategies

We extend the definition of certificate games to quantum strategies where the players can use shared entanglement.

The definition of certificate games with quantum strategies is similar to that of non-local games (as presented in reference [40]). A quantum strategy for a certificate game consists of a shared state $|\Psi_{AB}\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$ between the two players, and two families of projective measurements $M_A = \{M_A(x)\}_{x\in f^{-1}(0)}$ and $M_B = \{M_B(x)\}_{x\in f^{-1}(1)}$ made on their respective part of the shared state $|\Psi_{AB}\rangle$. Here $\mathcal{H}_A$ and $\mathcal{H}_B$ denote the Hilbert spaces of the players Alice and Bob respectively, and the definition of projective measurements is as below.

Definition 2.2.6 (Projective measurement)**.** *Let* $\mathcal{H}$ *be a Hilbert space. A projective measurement* M *is a collection of orthogonal projectors* $\{P_i\}_{i\in[n]}$ *for each of the possible outcomes* i *such that* $\{P_i\}$ *are positive semidefinite matrices that satisfy*

$$\begin{aligned}\sum_i P_i &= I \\ P_i P_j &= \delta_{i,j} P_i\end{aligned}$$

where $\delta_{i,j}$ *is the Kronecker delta, i.e.* $\delta_{i,j} = 1$ *if* $i = j$ *and is 0 otherwise.*

The quantum certificate game complexity of a Boolean function is defined as follows.

Definition 2.2.7 (Shared entanglement certificate game complexity)**.** *For any (possibly partial) Boolean function* f*,*

$$\mathsf{CG}^*(f) = \min_{|\Psi_{AB}\rangle, M_A, M_B} \quad \max_{x,y\in f^{-1}(0)\times f^{-1}(1)} \quad \frac{1}{\omega^\star((|\Psi_{AB}\rangle, M_A, M_B); x, y)},$$

where $\omega^\star((|\Psi_{AB}\rangle, M_A, M_B); x, y)$ *is the winning probability of strategy* $(|\Psi_{AB}\rangle, M_A, M_B)$ *on* x, y

$$\omega^\star((|\Psi_{AB}\rangle, M_A, M_B); x, y) = \sum_{i: x_i \neq y_i} \langle \Psi_{AB} | P_{A;x,i} \otimes P_{B;y,i} | \Psi_{AB} \rangle.$$

We note that $\langle \Psi_{AB} | P_{A;x,i} \otimes P_{B;y,j} | \Psi_{AB} \rangle$ is the probability that Alice and Bob output i and j on inputs x and y.

2.2.4 Certificate games with non-signalling strategies

Non-signalling strategies (Definition 2.2.1) are a generalization of quantum strategies and are useful to give lower bounds on quantum games.

Definition 2.2.8 (Non-signalling certificate game complexity)**.** *For any (possibly partial) Boolean function* f*,*

$$\mathsf{CG}^{\mathsf{ns}}(f) = \min_{\lambda} \quad \max_{x,y \in f^{-1}(0) \times f^{-1}(1)} \quad \frac{1}{\omega^{\mathsf{ns}}(\lambda; x, y)},$$

where λ *is a collection of (possibly negative) variables* $\{\lambda_{AB}\}$ *and* AB *ranges over all pairs of deterministic strategies* A, B*. They must satisfy* $\sum_{(A,B)} \lambda_{AB} = 1$ *and the winning probability is given by,*

$$\omega^{\mathsf{ns}}(\lambda; x, y) = \sum_{\substack{A,B: A(x)=B(y)=i \\ \text{and } x_i \neq y_i}} \lambda_{AB}.$$

To prove lower bounds on CG^*, we cannot proceed in the same way as in the case of $\mathsf{CG}^{\mathrm{pub}}$ since the value of CG^* cannot be written as a linear program. However, a key observation is that in many cases (and in all the cases we have considered in this thesis), the fundamental bottleneck for proving lower bounds on quantum strategies is the non-signalling property, which says that in two-player games with shared entanglement, the outcome of one of the players' measurements cannot reveal the other player's input. This was the original motivation for defining $\mathsf{CG}^{\mathsf{ns}}$: if we only require the non-signalling property of quantum strategies, it suffices to prove a lower bound on $\mathsf{CG}^{\mathsf{ns}}$, which is a lower bound on CG^*. Using the characterization of non-signalling strategies in terms of an affine polytope (see Proposition 2.2.2), we obtain a convenient linear programming formulation for $\mathsf{CG}^{\mathsf{ns}}$.

Proposition 2.2.9 (Dual formulation of $\mathsf{CG}^{\mathsf{ns}}$)**.** *For a certificate game* G *corresponding to a (possibly partial) Boolean function* f*,* $\mathsf{CG}^{\mathsf{ns}}(f) = 1/\omega^{ns}(G_f)$*, where winning probability* $\omega^{ns}(G_f)$ *can be written as the following linear program.*

$$\begin{aligned}
\omega^{\mathsf{ns}}(G_f) = \quad & \min_{\mu,\gamma,\delta} \quad \delta \\
\text{such that} \quad & \sum_{x,y:\ A,B \text{ correct on } x,y} \mu_{x,y} + \sum_{x,y} \gamma_{A(x),B(y),x,y} = \delta \quad \text{for every deterministic strategy } A, B \\
& \sum_{x,y} \mu_{xy} = 1, \quad \mu_{x,y} \geq 0, \quad \gamma_{a,b,x,y} \geq 0,
\end{aligned}$$

where $\mu = \{\mu_{x,y}\}_{x \in f^{-1}(0),\, y \in f^{-1}(1)}$ *and* $\gamma = \{\gamma_{i,j,x,y}\}_{i,j \in [n], x \in f^{-1}(0),\, y \in f^{-1}(1)}$ *.*

We will see in Section 4.2 that the certificate game complexity variant with non-signalling strategies $\mathsf{CG}^{\mathsf{ns}}$ of a function is bounded below by fractional certificate complexity FC and the classical adversary bound CMM.

Since we have considered progressively stronger models, the following holds trivially.

Proposition 2.2.10. *For any Boolean function f,*

$$\mathsf{CG}^{\mathsf{ns}}(f) \leq \mathsf{CG}^{*}(f) \leq \mathsf{CG}^{pub}(f) \leq \mathsf{CG}(f).$$

A natural question that arises is how separated these measures are. For instance, what advantage does shared randomness give over private randomness and similarly shared entanglement over shared randomness? In the following chapters, we will see more lower and upper bounds on the different variants of certificate game complexity and see how they fit in the landscape of previously studied complexity measures of Boolean functions.

Chapter 3

Certificate Games with public randomness

The main challenge in constructing a certificate game strategy in any model is in getting the two players to coordinate their strategies so that the index they output is the *same* without any communication. In the public coin setting, we can take advantage of having access to shared randomness to achieve this task. In this chapter, we prove strong (and arguably surprising) upper bounds on CG^{pub} by constructing certificate game strategies using shared randomness. In particular, we show that $\mathsf{R}, \mathsf{RS}, \mathsf{C}$, and even EC (Section 3.3, Section 3.4) are upper bounds on CG^{pub}. The main tools in constructing these strategies are hash functions and permutations. The ideas behind our public coin strategies can be expressed in a general framework based on hash functions which is shown in Section 3.2. We will also show a lower bound of FC on CG^{pub} in Section 3.5.

We illustrate the idea of using hash functions and permutations by constructing a CG^{pub} strategy for the Tribes function.

3.1 Public coin certificate game for the Tribes function

The Tribes function on n bits where $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}} = \mathsf{OR}_{\sqrt{n}} \circ \mathsf{AND}_{\sqrt{n}}$ (Definition 1.2.3) is a very well studied problem in complexity theory. It has full randomized query complexity, i.e. $\mathsf{R}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$. Since the constituent functions $\mathsf{AND}_{\sqrt{n}}$ and $\mathsf{OR}_{\sqrt{n}}$ have full sensitivity $\sqrt{n}$ by Proposition 3.5.1, the CG^{pub} of $\mathsf{AND}_{\sqrt{n}}$ and $\mathsf{OR}_{\sqrt{n}}$ is $\Theta(\sqrt{n})$. In this section, we prove that the CG^{pub} of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ is $O(\sqrt{n})$ (Theorem 3.1.1). This implies that the function $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ demonstrates a quadratic separation between R and CG^{pub}. It also implies that CG^{pub} is not preserved under composition, i.e. CG^{pub} value is not the product of the CG^{pub} values of the individual functions.

For the $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ function, we want to construct a strategy that wins the certificate game with probability $\Omega(1/\sqrt{n})$ (instead of the obvious $\Omega(1/n)$). The input of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ consists of $\sqrt{n}$ blocks, each of $\sqrt{n}$ bits. We will reduce the general problem to the case where all blocks of Alice's input have a single 0, and Bob has exactly one block with all 1's. Alice and Bob win when they both output the unique index i where Alice's bit is 0 and Bob's bit is 1. We will now look at the description of the strategy for this special case before we discuss the more general strategy.

Let us view Alice's input as an array A of $\sqrt{n}$ values where the entries specify the position of the 0 in each block and each entry is in $[\sqrt{n}]$. On the other hand, Bob's input can be thought of as an index, say $j \in [\sqrt{n}]$, which identifies his all-1 block.

Alice wants to find j and Bob wants to find $A[j]$ which would enable them to both output a position where their inputs differ.

Let us begin with the simple case where each entry of Alice's array is distinct. Bob picks a random number r and outputs the r-th index of the j-th block. Alice can use the same r (due to shared randomness), and find the unique j such that $A[j] = r$. Whenever Bob picks r such that $A[j] = r$, they win the game. The probability that a random r matches $A[j]$ is $1/\sqrt{n}$.

For the harder case when some of the entries of A coincide, we use the shared randomness to permute entries of each block. This ensures that, with constant probability, we have a unique j such that $A[j] = r$. This gives the required success probability $\Omega(1/\sqrt{n})$.

We now give a formal CG^{pub} strategy for the $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ function.

Theorem 3.1.1. *The public coin certificate game complexity of the* Tribes *function is,*

$$\mathsf{CG}^{pub}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = O(\sqrt{n}).$$

Proof. Let x and y be the two strings given to Alice and Bob respectively, i.e. $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}(x) = 0$ and $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}(y) = 1$.

Since $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}(x) = 0$, in every block $i \in \left[\sqrt{n}\right]$ there exists at least one position a_i such that $x_{i,a_i} = 0$. For each block $i \in \left[\sqrt{n}\right]$, Alice arbitrarily picks an a_i such that $x_{i,a_i} = 0$. A new string x' is constructed in which

$$x'_{(i,j)} = \begin{cases} 0 & \text{if } j = a_i \\ 1 & \text{otherwise} \end{cases}$$

for all $i, j \in \left[\sqrt{n}\right]$. Note that the a_i's are not necessarily unique.

Similarly, $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}(y) = 1$ implies the existence of a block b such that every entry of that block is 1, i.e. $y_{b,j} = 1$ for all $j \in \left[\sqrt{n}\right]$. Once again, note that there may be multiple blocks b but Bob picks one such b and constructs an input y' as follows,

$$y'_{(i,j)} = \begin{cases} 1 & \text{if } i = b \\ 0 & \text{otherwise} \end{cases}$$

for all $i, j \in \left[\sqrt{n}\right]$.

We now have that (b, a_b) is the unique index (i, j) such that $x'(i, j) = 0$ and $y'(i, j) = 1$. We will now present a protocol for Alice and Bob that correctly guesses (b, a_b) with probability at least $1/\sqrt{n}$, which would imply the theorem as (b, a_b) is an index where the original inputs x and y differ.

- Alice and Bob use shared randomness to select the same list of $\sqrt{n}$ permutations $\sigma_1, \ldots, \sigma_{\sqrt{n}} : \left[\sqrt{n}\right] \to \left[\sqrt{n}\right]$, where the permutations are drawn (with replacement) independently and uniformly at random from the set of all possible permutations from $\left[\sqrt{n}\right]$ to $\left[\sqrt{n}\right]$.
- Both Alice and Bob pick the same index t between 1 and $\sqrt{n}$ using shared randomness.
- Bob outputs $(b, \sigma_b^{-1}(t))$.

- Alice picks a number $i \in [\sqrt{n}]$ such that $\sigma_i(a_i) = t$ and outputs (i, a_i). In case no such i exists, Alice outputs a random index.

The probability of success of the protocol crucially depends on the fact that Alice and Bob can use shared randomness to pick the same set of permutations $\sigma_1, \ldots, \sigma_{\sqrt{n}}$ while maintaining that the permutations are picked uniformly at random.

We will show that with constant probability there exists a unique i which satisfies $\sigma_i(a_i) = t$. Under the condition that this holds, we will show that the probability of success of the above protocol is at least $1/\sqrt{n}$ which would prove the theorem.

We state the following claim that will be proven later.

Claim 3.1.2. *If $\sigma_1, \ldots, \sigma_{\sqrt{n}} : [\sqrt{n}] \to [\sqrt{n}]$ are permutations selected uniformly at random from $\mathcal{S}_{\sqrt{n}}$ and $a_1, \ldots, a_{\sqrt{n}} \in [\sqrt{n}]$, there exists a unique i such that $\sigma_i(a_i) = t$ with probability at least $(1 - 1/\sqrt{n})^{\sqrt{n}-1} \approx e^{-1}$ for any fixed number $t \in [\sqrt{n}]$ where $\mathcal{S}_{\sqrt{n}}$ is the permutation group which is the set of all $\sqrt{n}!$ permutations.*

Note that the permutation σ_b is picked uniformly at random from $\mathcal{S}_{\sqrt{n}}$, i.e. σ_b is a random bijection from $[\sqrt{n}]$ to $[\sqrt{n}]$. For a $b \in [\sqrt{n}]$, $\sigma_b(a_b) = t$ with probability $1/\sqrt{n}$. If we assume that $\sigma_b(a_b) = t$ and that there exists a unique i such that $\sigma_i(a_i) = t$, both Alice and Bob output (b, a_b) in the protocol described above. Hence we have that the probability of success of the protocol is $\Omega(1/\sqrt{n})$. □

Proof of Claim 3.1.2. Consider $\mathcal{E}_k$ to be the event that there is a unique k such that $\sigma_k(a_k) = t$, i.e.

$$\mathcal{E}_k := \sigma_k(a_k) = t \text{ and for all } i \neq k,\ \sigma_i(a_i) \neq t.$$

The probability that the event $\mathcal{E}_k$ occurs is $\frac{1}{\sqrt{n}} \cdot (1 - \frac{1}{\sqrt{n}})^{\sqrt{n}-1}$. The event that there exists a unique i such that $\sigma_i(a_i) = t$ is $\cup_{k=1}^{\sqrt{n}} \mathcal{E}_k$. Since the events $\mathcal{E}_k$ are disjoint, we have proven the claim. □

We will now give a generic framework for public coin strategies based on random hash functions to isolate a common index where the inputs x and y differ.

3.2 A framework for upper bounds based on hashing

The idea of using hash functions and permutations can be extended for more general $\mathsf{CG}^{\mathrm{pub}}$ strategies, where Alice and Bob have their respective set of possible answers (indices) and want to find a common index. The hashing framework given below captures the intuition used in the $\mathsf{CG}^{\mathrm{pub}}$ strategy for Tribes function. The framework can be described concisely as follows: The players share a common hash function which maps the set of indices to a set S. Using shared randomness, a random element r in S is picked. Both the players answer from the intersection of the preimage of r and their set of possible answers. This framework is formally described below.

Let $f : \{0,1\}^n \to \{0,1\}$ be a (possibly partial) Boolean function. Alice is given $x \in f^{-1}(0)$ and Bob is given $y \in f^{-1}(1)$. Their goal is to produce a common index $i \in [n]$ such that $x_i \neq y_i$.

Let $T \subseteq [n]$ be a set of potential outputs which is known to both players, and let S be a finite set. The sets T and S are fixed in advance as a part of the strategy, i.e. they do not depend on the input and depend only on the function f. Let A_x and B_y denote the set of potential outputs of Alice and Bob on inputs x and y respectively where $A_x, B_y \subseteq T$. The players proceed as follows.

1. Using shared randomness, they select a random hash function $h : T \to S$.
2. Using shared randomness, they select a random element $z \in S$.
3. Alice outputs a (possibly random) element of $h^{-1}(z) \cap A_x$. If this set is empty, she outputs an arbitrary element. Similarly, Bob outputs a (possibly random) element of $h^{-1}(z) \cap B_y$ or an arbitrary element if this set is empty.

This general strategy will be correct with good enough probability, if the following two conditions are ensured:

(i) $h^{-1}(z) \cap W$ is not empty, where $W \subseteq A_x \cap B_y$ denotes the set of correct outputs from $A_x \cap B_y$, i.e. $x_i \neq y_i$ for any $i \in W$.

(ii) $h^{-1}(z) \cap A_x$ and $h^{-1}(z) \cap B_y$ are "small enough".

Note that the Condition (i) implies that the sets, $h^{-1}(z) \cap A_x$ and $h^{-1}(z) \cap B_y$, are not empty.

We will apply this general framework in the following chapters in several different ways. We use it for proving that $\mathsf{CG}^{\mathrm{pub}}$ is bounded above by C and EC. We also use it to get a strong upper bound for the Approximate Index function ApInd (Definition 5.2.1). Finally, we use the hashing framework to prove that the single-bit version of $\mathsf{CG}^{\mathrm{pub}}$ characterizes sensitivity up to constant factors. While each of these proofs fits into the framework we described above, their analyses are technically different.

3.3 Upper bounds on $\mathsf{CG}^{\mathrm{pub}}$ by C and EC

We will take advantage of having access to shared randomness by using the hashing based approach outlined in Section 3.2. We start with a simple argument to show that $\mathsf{CG}^{\mathrm{pub}}$ is always bounded above by certificate complexity C. A slightly more involved argument will show a stronger upper bound by the expectational certificate complexity EC.

Theorem 3.3.1. *For a total Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{CG}^{pub}(f) \leq O(\mathsf{C}(f)).$$

Proof. Let S be a finite set of cardinality $\mathsf{C}(f)$. An element $z \in S$ is fixed as a part of the strategy, i.e. z does not depend on the input.

Using shared randomness, the players select a function $h : [n] \to S$ as follows. Let $h : [n] \to S$ be a random hash function such that for each $i \in [n]$, $h(i)$ is selected independently and uniformly at random from S.

For an input $x \in f^{-1}(0)$, we fix an optimal 0-certificate C_x, and $A_x \subseteq [n]$ denotes the set of indices fixed by C_x. Similarly, $B_y \subseteq [n]$ is the set of indices fixed by an optimal 1-certificate C_y for an input $y \in f^{-1}(1)$.

After selecting h using shared randomness, the players proceed as follows. On an input x, Alice outputs an index $i \in A_x$ such that $h(i) = z$. Similarly, Bob on input y outputs an index $j \in B_y$ such that $h(j) = z$. If they have several valid choices, they select their outputs randomly. If they have no valid choice, they output arbitrary indices.

Let $i^* \in A_x \cap B_y$ such that $x_{i^*} \neq y_{i^*}$. By the definition of certificates, such an element i^* exists for any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$ as 0-certificates and 1-certificates

intersect. Alice and Bob win on input (x, y) if both players output i^*. We now estimate the probability that both players output i^*.

Recall that by the definition of the hash function h, the probability that $h(i^*) = z$ is $\frac{1}{|S|} = \frac{1}{\mathsf{C}(f)}$. We also note that for any $i \in A_x \cup B_y$, the number of elements that are different from i in $A_x \cup B_y$ is $\ell = |A_x \cup B_y| - 1 \leq |A_x| + |B_y| - 2$. For any $z \in S$ and any $i \in A_x \cup B_y$, the probability over the choice of h that no element other than i in $A_x \cup B_y$ is mapped to z by h is

$$\left(1 - \frac{1}{|S|}\right)^{\ell} \geq \frac{1}{e^2},$$

since $\max\{|A_x|, |B_y|\} \leq \mathsf{C}(f) = |S|$ and hence $\ell \leq 2(|S| - 1)$. Thus the players output a correct answer with probability at least $\frac{1}{e^2}\frac{1}{\mathsf{C}(f)}$. □

We will now obtain a stronger upper bound of EC (see Definition 1.1.8) on CG^{pub}.

Theorem 3.3.2. *For a (possibly partial) Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{CG}^{pub}(f) \leq O(\mathsf{EC}(f)).$$

Proof. The proof will be similar to that of the upper bound by C but will be slightly more involved. We will rely on the "weights" $w_{x,i}$ from the definition of $\mathsf{EC}(f)$ to construct a CG^{pub} strategy.

Let S be a finite set of cardinality $\lceil \mathsf{EC}(f) \rceil$. Using shared randomness, the players select a hash function $h : [n] \to S$ and an element $z \in S$ as follows. Let $h : [n] \to S$ be a random hash function such that $h(i)$ is selected independently and uniformly at random from S for each $i \in [n]$. In addition, z is selected uniformly at random from S and independently from the choices for the hash function h.

For all inputs $x \in \{0,1\}^n$, let $w_{x,i}$ be the optimal weights achieving $\mathsf{EC}(f)$. Let EC_x denote the sum $\sum_{i \in [n]} w_{x,i}$. Recall that for each $x \in \{0,1\}^n$, $\mathsf{EC}_x \leq \mathsf{EC}(f)$ by the definition of EC.

For a given $z \in S$, consider the preimage $h^{-1}(z)$. We use the following notation,

$$W_x(z) = \sum_{i \in h^{-1}(z)} w_{x,i} \ .$$

Notice that for any $z \in S$, the expected value of $W_x(z)$

$$\mathbb{E}[W_x(z)] = \sum_{i \in [n]} \frac{w_{x,i}}{|S|} = \frac{EC_x}{|S|} \ ,$$

where the expectation is over the choice of hash functions h.

After selecting h and z using shared randomness, the players proceed as follows. On an input $x \in f^{-1}(0)$, Alice selects an index i from $h^{-1}(z)$ such that each i is chosen with a probability $\frac{w_{x,i}}{W_x(z)}$. Similarly, Bob on an input $y \in f^{-1}(1)$ selects an index i from $h^{-1}(z)$ such that each i is chosen with a probability $\frac{w_{y,i}}{W_y(z)}$. Note that these choices are made using Alice's and Bob's private randomness, and hence Alice's choices are independent of Bob's choices for a fixed z and h. However, both of their choices depend on z and h. In what follows, we denote the probabilities that are only over the choice of z and h by $\Pr_z$ and $\Pr_h$ respectively.

Recall that $W_x(z)$ and $W_y(z)$ are measures of the preimage of z with respect to the weights for x and y respectively. Since $\frac{\mathsf{EC}_x}{|S|} \leq 1$ for any $x \in \{0,1\}^n$, the preimage

of most elements in S will have a small measure. We now estimate the probability that a given element i is mapped to a value $h(i)$ whose preimage has small measures $W_x(h(i))$ and $W_y(h(i))$. Note that this only depends on the choice of h.

For a given i, consider the selection of the values $h(j)$ for all $j \neq i$ from $[n]$. Let us now look at the measure of the preimages of elements in S at this point (without taking into account what happens to i). Since $\frac{\mathsf{EC}_x - w_{x,i}}{|S|} \leq 1$ for any $x \in \{0,1\}^n$, at most $\frac{1}{t-1}$ of the elements in S can have a measure more than $t-1$ at this point. Since $w_{x,i} \leq 1$, we get that for any $x \in \{0,1\}^n$ and $i \in S$,

$$\Pr_h[W_x(h(i)) > t] \leq \frac{1}{t-1}$$

by Markov's inequality. For an $i \in [n]$, let $\mathbf{Small}_i$ denote the event that both $W_x(h(i))$ and $W_y(h(i))$ are at most t. The probability of the event $\mathbf{Small}_i$ is,

$$\Pr_h[\mathbf{Small}_i] \geq 1 - \frac{2}{t-1}.$$

For a given $i \in [n]$, let $\mathbf{Both}_i$ denote the event that both players select i. Let $I_{xy} = \{i | x_i \neq y_i\}$. Since $f(x) \neq f(y)$, $I_{xy} \neq \emptyset$.

Recall that the goal of the players is that they both output the same i from I_{xy}. Let us denote by $P(x,y)$ the corresponding winning probability. Note that $P(x,y)$ is at least as large as the probability that they both output the same i from I_{xy} and that both $W_x(h(i))$ and $W_y(h(i))$ are at most t.

Using the fact that the events $\mathbf{Both}_i$ are pairwise disjoint, we have

$$\begin{aligned} P(x,y) &\geq \sum_{i \in I_{xy}} \Pr\left[\mathbf{Both}_i \cap (z{=}h(i)) \cap \mathbf{Small}_i\right] \\ &= \sum_{i \in I_{xy}} \Pr[\mathbf{Both}_i | (z{=}h(i)) \cap \mathbf{Small}_i] \Pr[(z{=}h(i)) \cap \mathbf{Small}_i] \,. \end{aligned}$$

Note that the events $z = h(i)$ and $\mathbf{Small}_i$ are independent, since the choice of z is independent of the choice of h. For any $i^* \in I_{xy}$ and $h : [n] \to S$,

$$\Pr_z[z = h(i^*)] = \frac{1}{|S|}.$$

We get,

$$\begin{aligned} \Pr[z = h(i) \cap \mathbf{Small}_i] &= \Pr_z[z = h(i)] \Pr_h[\mathbf{Small}_i] \\ &= \frac{1}{|S|} \Pr_h[\mathbf{Small}_i] \geq \frac{1}{|S|}\left(1 - \frac{2}{t-1}\right) \end{aligned}$$

For any $i \in [n]$, we have

$$\Pr[\mathbf{Both}_i \mid z = h(i)] = \frac{w_{x,i}}{W_x(z)} \frac{w_{y,i}}{W_y(z)} \text{ and } \Pr[\mathbf{Both}_i \mid z = h(i) \cap \mathbf{Small}_i] \geq \frac{w_{x,i}}{t} \frac{w_{y,i}}{t} \,.$$

On putting these together, we get

$$P(x,y) \geq \frac{1}{t^2} \frac{1}{|S|} \left(1 - \frac{2}{t-1}\right) \sum_{i \in I_{xy}} w_{x,i} w_{y,i} \geq \frac{1}{t^2} \frac{1}{|S|} \left(1 - \frac{2}{t-1}\right)$$

where the last inequality follows from the definition of $\mathsf{EC}(f)$.

Setting $t = 5$, we get that the players output the same element from I_{xy} with probability at least $\frac{1}{50}\frac{1}{\lceil \mathsf{EC}(f) \rceil} = \Omega(\frac{1}{\mathsf{EC}(f)})$. □

3.4 Upper bound on $\mathsf{CG}^{\mathrm{pub}}$ by R and RS

In this section we show that $\mathsf{CG}^{\mathrm{pub}}$ is bounded above by R.

Theorem 3.4.1. *For any Boolean (possibly partial) function f, $\mathsf{CG}^{pub}(f) \leq O(\mathsf{R}(f))$.*

Proof. From the definition of $\mathsf{R}(f)$ there is a randomized decision tree $\mathcal{R}$ that on any input x outputs $f(x)$ correctly with probability at least $2/3$, and $\mathcal{R}$ only reads at most $\mathsf{R}(f)$ number of bits of x. To prove $\mathsf{CG}^{\mathrm{pub}}(f) \leq \mathsf{R}(f)$, let us consider the following strategies used by the two players:

Both the players run the algorithm $\mathcal{R}$ on their respective inputs using the same random coins (using the shared randomness). Both the player also use shared randomness to pick a number t uniformly at random between 1 and $\mathsf{R}(f)$. Both the players output the t-th index that is queried by $\mathcal{R}$.

Let $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$ be the inputs to the players respectively. Since $f(x) \neq f(y)$, the algorithm $\mathcal{R}$ will output different answers when the players run the algorithm on their respective inputs with probability at least $4/9$. Since the algorithm $\mathcal{R}$ is run using the same internal coins, the initial sequence of indices queried by both the runs of the algorithm is the same until the algorithm queries an index k such that $x_k \neq y_k$. Note that with probability at least $1/\mathsf{R}(f)$, the random number t picked by the players is the same as k. Hence, the players correctly output the same index k such that $x_k \neq y_k$ with probability $\frac{4}{9} \cdot \frac{1}{\mathsf{R}(f)}$. This simple strategy shows that $\mathsf{CG}^{\mathrm{pub}}(f) \leq O(\mathsf{R}(f))$. □

Using the same idea we can show that $\mathsf{CG}^{\mathrm{pub}}$ is bounded above by randomized sabotage complexity RS (Definition 1.1.13), a measure of complexity introduced to study the behaviour of randomized query complexity R under composition [22]. It was shown that RS is a lower bound on R and that it behaves perfectly under composition. We show that $\mathsf{CG}^{\mathrm{pub}}$ is a lower bound on RS.

Proposition 3.4.2. *The CG^{pub} of a (possibly partial) Boolean function f is at most its sabotage complexity,*

$$\mathsf{CG}^{pub}(f) \leq \frac{3}{2}\mathsf{RS}(f).$$

Proof. We show this by using the sabotage complexity protocol to build a $\mathsf{CG}^{\mathrm{pub}}$ protocol. Assuming that Alice has input $x \in f^{-1}(0)$ and Bob an input $y \in f^{-1}(1)$, we construct a sabotaged input $z_{x,y}$ that is consistent with x and y as follows:

$$z_{x,y}(i) = \begin{cases} x(i) & \text{if } x(i) = y(i) \\ * & \text{otherwise.} \end{cases}$$

The $\mathsf{CG}^{\mathrm{pub}}$ protocol is as follows: using public randomness, Alice and Bob sample a decision tree from the RS protocol and follow the path on the decision tree according to their respective inputs for at most $\mathsf{RS}(f)$ steps. With probability at least $2/3$, the randomly chosen tree finds a $*$ on input $z_{x,y}$ in $\mathsf{RS}(f)$ steps. Since the sabotaged input $z_{x,y}$ is consistent with both Alice's and Bob's inputs, the path on x and y on

the decision tree is the same as that on $z_{x,y}$ until they reach a place where they differ (or encounter a $*$ in $z_{x,y}$). Alice and Bob pick a random position t such that $1 \leq t \leq \mathsf{RS}(f)$ and output the *t-th* query made in their corresponding paths on the tree. With probability $\frac{1}{\mathsf{RS}(f)}$, it is a place corresponding to a $* \in z_{x,y}$ and they succeed in finding a place where the inputs differ. This gives a success probability $\geq \frac{2}{3} \cdot \frac{1}{\mathsf{RS}(f)}$ as the random decision tree finds a $*$ on the sabotaged input $z_{x,y}$ with probability $\geq \frac{2}{3}$. □

We will now switch gears and look at the following lower bound on $\mathsf{CG}^{\mathrm{pub}}$ in terms of fractional certificate complexity FC (see Definition 1.1.7).

3.5 A lower bound on $\mathsf{CG}^{\mathrm{pub}}$

Proposition 3.5.1. *For any Boolean (possibly partial) function f,*

$$\mathsf{FC}(f) \leq \mathsf{CG}^{pub}(f)$$

Proof. By the definition of $\mathsf{CG}^{\mathrm{pub}}(f)$ (Definition 2.2.4), there is a collection of non-negative quantities $\{p_{AB}\}$ for pairs of strategies (A, B) that satisfy $\sum_{(A,B)} p_{AB} = 1$.

We also have that for any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$

$$\sum_{\substack{(A,B) \,:\, A(x)=B(y)=i, \\ x_i \neq y_i}} p_{AB} \geq \frac{1}{\mathsf{CG}^{\mathrm{pub}}(f)}.$$

Let $\mathsf{CG}^{\mathrm{pub}}(f) = \delta^\star$, and $v_{x,i} = \delta^\star \left(\sum_{(A,B) \,:\, A(x)=i} p_{A,B} \right)$ be the weights for $\mathsf{FC}(f)$. For any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, we get

$$\sum_{i \,:\, x_i \neq y_i} v_{x,i} \geq \delta^\star \left(\sum_{\substack{(A,B) \,:\, A(x)=B(y)=i, \\ x_i \neq y_i}} p_{A,B} \right) \geq 1$$

Thus if we use these weights $v_{x,i}$, we have $\mathsf{FC}(f) \leq \max_x \sum_{i \in [n]} v_{x,i} = \mathsf{CG}^{\mathrm{pub}}(f)$. □

This lower bound result will be improved in the next chapter as we look at the non-signalling variant $\mathsf{CG}^{\mathsf{ns}}$.

Chapter 4

Other variants of certificate games

In this chapter, we turn our attention to other variants of certificate games such as those with private randomness, quantum and non-signalling strategies. We will see the limitations of private randomness in certificate games and show that for certain functions CG can be as large as quadratic in the number of bits. We will obtain lower bounds on CG in terms of the positive Adversary bound MM (Theorem 4.1.2) and zero-error randomized query complexity R_0 (Theorem 4.1.3). We will also have upper bounds in terms of CG^{pub} (Theorem 4.1.5) and certificate complexity (Theorem 4.1.8).

For certificate games with quantum strategies, we will see that the non-signalling model gives very simple and useful lower bounds in terms of fractional certificate complexity FC (Theorem 4.2.2) and the classical Adversary bound CMM (Theorem 4.2.3). This serves as a lower bound for all the versions of certificate games discussed thus far. Our final set of results is in the context of single-bit versions of certificate games. We show that the single-bit version of CG is equal to λ^2 (Theorem 4.3.11) and that the single-bit version of CG^{pub} is asymptotically equal to $\mathsf{s}(f)$ (Theorem 4.3.6). This gives a new interpretation of sensitivity which is one of the central complexity measures in this area.

4.1 Certificate Games with private randomness

We begin our discussion on CG with the following formulation in terms of weights, the essential idea of which is rescaling.

Proposition 4.1.1 (Equivalent formulation for CG)**.**

$$\begin{aligned} \mathsf{CG}(f) = \min_{\{w_{x,i}\}} \max_{x} & \left\{ \sum_i w_{x,i} \right\}^2 \\ \textit{such that} \quad \sum_{i:x_i \neq y_i} & w_{x,i} w_{y,i} \geq 1 \quad \forall x \in f^{-1}(0), y \in f^{-1}(1) \\ & w_{x,i} \geq 0 \quad \forall x, i \end{aligned}$$

Proof. We will first show that the value of the objective function in the formulation in terms of weights is at most CG. Let p be an optimal probability distribution that achieves $\mathsf{CG}(f)$ and let

$$\Delta = \min_{x,y:f(x)\neq f(y)} \sum_{i:x_i \neq y_i} p_{x,i} p_{y,i} = \frac{1}{\mathsf{CG}(f)}.$$

We construct the following weight scheme using p,

$$w_{x,i} = \frac{p_{x,i}}{\sqrt{\Delta}}.$$

This is a feasible solution as,

$$\sum_{\substack{\forall x,y \\ f(x)\neq f(y)}} \sum_{i:x_i\neq y_i} w_{x,i}w_{y,i} = \frac{1}{\Delta}\sum_{i:x_i\neq y_i} p_{x,i}p_{y,i} \geq \frac{\Delta}{\Delta} = 1.$$

We now have,

$$\min_{\{w'_{x,i}\}} \max_x \left\{\sum_{i\in[n]} w'_{x,i}\right\}^2 \leq \max_x \left\{\sum_{i\in[n]} w_{x,i}\right\}^2 = \max_x \left\{\sum_{i\in[n]} \frac{p_{x,i}}{\sqrt{\Delta}}\right\}^2 = \frac{1}{\Delta} = \mathsf{CG}.$$

For the other direction, let w be an optimal weight scheme that minimises $\max_x \sum_i w_{x,i}$. We construct the following family of probability distributions

$$p_{x,i} = \frac{w_{x,i}}{\sum_j w_{x,j}}.$$

This gives,

$$\begin{aligned}\mathsf{CG}(f) &\leq \max_{\substack{x,y \\ f(x)\neq f(y)}} \frac{1}{\sum_{i:x_i\neq y_i} p_{x,i}p_{y,i}} = \max_{\substack{x,y \\ f(x)\neq f(y)}} \frac{\sum_j w_{x,j}\sum_j w_{y,j}}{\sum_{i:x_i\neq y_i} w_{x,i}w_{y,i}} \\ &\leq \max_{\substack{x,y \\ f(x)\neq f(y)}} \sum_j w_{x,j}\sum_j w_{y,j}\end{aligned}$$

Thus we have $\mathsf{CG}(f) \leq \max_x \left\{\sum_j w_{x,j}\right\}^2$. □

We will now prove some upper and lower bounds for CG and see how it compares with other well studied complexity measures for Boolean functions.

4.1.1 Upper and lower bounds for CG

We begin with a lower bound on CG in terms of the positive Adversary bound MM (see Definition 1.1.10). The positive adversary method was introduced by Ambainis as a lower bound for quantum query complexity and we use the minimax formulation here.

Theorem 4.1.2. *For any (possibly partial) Boolean function f, $\mathsf{MM}(f)^2 \leq \mathsf{CG}(f)$*

Proof. Let $p = \{p_{x,i}\}$ be an optimal solution for $\mathsf{CG}(f)$ and we have the winning probability of the game $\omega(p;x,y) \geq \frac{1}{\mathsf{CG}(f)}$ for all $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$. Using the same assignment of the variables $p_{x,i}$ for MM (Definition 1.1.10), we get

$$\begin{aligned}\frac{1}{\mathsf{MM}(f)^2} &\geq \min_{\substack{x\in f^{-1}(0) \\ y\in f^{-1}(1)}} \left(\sum_{i:x_i\neq y_i} \sqrt{p_{x,i}p_{y,i}}\right)^2 \\ &\geq \min_{\substack{x\in f^{-1}(0) \\ y\in f^{-1}(1)}} \sum_{i:x_i\neq y_i} p_{x,i}p_{y,i} \geq \frac{1}{\mathsf{CG}}\end{aligned}$$

and hence $\mathsf{MM}(f)^2 \leq \mathsf{CG}(f)$. □

The following upper and lower bounds on CG come from its formulation in terms of weights and they follow from a direct application of the technique used in reference[58].

Theorem 4.1.3 (implied by [58], Theorem 1)**.** *For any total Boolean function f,*

$$\mathsf{R}_0(f) \leq \mathsf{CG}(f) \leq O(\mathsf{EC}(f)^2)$$

Proof. The formulation of $\sqrt{CG}$ in terms of weights in Proposition 4.1.1 is a relaxation of the definition of EC. This is true as EC has the additional constraint that the weights $w_{x,i}$ are bounded above by 1, and we have $\sqrt{CG}(f) \leq \mathsf{EC}(f)$.

To get the first inequality, the proof of $\mathsf{R}_0 \leq O(\mathsf{EC}^2)$ in reference [58] is directly applied since their proof does not make use of the constraint that the weights $w_{x,i}$ are bounded above by 1. The proof is adapted for CG and reproduced here for completeness. The proof proceeds by constructing randomised query algorithms with one-sided error ϵ from an optimal CG strategy.

Claim 4.1.4 (implied by [58], Proposition 1)**.** *For a total Boolean function f, $b \in \{0,1\}$, and $0 < \epsilon \leq 1$, $\mathsf{R}^b_\epsilon \leq \lceil \frac{\mathsf{CG}}{\epsilon} \rceil$.*

If this claim holds, $\mathsf{R}_0 \leq O(\mathsf{CG}^2)$ since a zero error randomised algorithm can be constructed as in the proof of $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$. To prove this claim, a one-sided error algorithm R^0_ϵ is constructed from the optimal weights for CG following Proposition 4.1.1. The algorithm for R^0_ϵ is constructed analogously. Let μ_y be the probability distribution given by, $\mu_y(i) = \frac{w_{y,i}}{\sum_{j\in[n]} w_{y,j}}$ for every input $y \in \{0,1\}^n$.

Algorithm 1 One-sided Randomized query algorithm $\mathcal{A}$ for R^0_ϵ from CG

Input: $x \in \{0,1\}^n$

1. Repeat the following $\lceil \frac{\mathsf{CG}}{\epsilon} \rceil$ times:
   - Pick the lexicographically first 1-input y that is consistent with the queries made so far.
   - Sample an index i from μ_y and query x_i.
   - If the indices queried so far form a c-certificate, return c.
2. Return 1.

Note that if the algorithm $\mathcal{A}$ terminates before reaching step (2), it would have found a certificate for the input and does not make any error. We also see that the algorithm does not make any error on any 1-input. The proof of the claim lies in showing that the algorithm terminates on any 0-input within $\lceil \frac{\mathsf{CG}}{\epsilon} \rceil$ iterations of step (1) with probability at least $1 - \epsilon$. To prove this, a random variable T_k is defined as follows:

$$T_k := \begin{cases} \frac{1}{\sqrt{CG(f)}} & \text{if } \mathcal{A} \text{ terminates before the } k\text{-th iteration of step (1)} \\ w_{x,i} & \text{if at the } k\text{-th iteration } \mathcal{A} \text{ has queried } i \text{ for the first time} \\ 0 & \text{otherwise.} \end{cases}$$

Let a variable T be defined as

$$T = \sum_{k=1}^{\lceil \mathsf{CG}/\epsilon \rceil} T_k.$$

If the algorithm $\mathcal{A}$ has not terminated in $\left\lceil \frac{\mathsf{CG}}{\epsilon} \right\rceil$ iterations of step (1), the maximum value T can take is $\sum_{i\in[n]} w_{x,i} \leq \sqrt{\mathsf{CG}(f)}$. Thus if $T > \sqrt{\mathsf{CG}(f)}$, the algorithm must have terminated in step (1) in which case it makes no error. Hence if it can be shown that $T > \sqrt{\mathsf{CG}(f)}$ with probability $\geq 1-\epsilon$, we have Claim 4.1.4. Let p denote this probability, i.e. $p = \Pr[T > \sqrt{\mathsf{CG}(f)}]$. To prove that $p \geq 1-\epsilon$, we find upper and lower bounds on the expected value of T.

- **Upper bound on $\mathbb{E}(\mathbf{T})$:** The maximum value T can take is,

$$T \leq \left\lceil \frac{\mathsf{CG}}{\epsilon} \right\rceil \cdot \frac{1}{\sqrt{\mathsf{CG}(f)}} + \sum_{i\in[n]} w_{x,i} \leq \sqrt{\mathsf{CG}(f)} \left(1 + \frac{1}{\epsilon}\right).$$

  We get the following upper bound on $\mathbb{E}(T)$,

$$\mathbb{E}(T) \leq p\left(1+\frac{1}{\epsilon}\right)\sqrt{\mathsf{CG}(f)} + (1-p)\sqrt{\mathsf{CG}(f)} \leq \left(1+\frac{p}{\epsilon}\right)\sqrt{\mathsf{CG}(f)}\,.$$

- **Lower bound on $\mathbb{E}(\mathbf{T})$:** Let $\mathcal{E}_k$ denote the event that the algorithm $\mathcal{A}$ terminates before the k-th iteration of step 1. If $\mathcal{A}$ performs the k-th iteration, let y denote the lexicographically first consistent 1-input chosen and let i_k denote the query made by $\mathcal{A}$ in this run. We will get a lower bound on $\mathbb{E}(T_k)$ and use it to get $\mathbb{E}(T)$ by using the linearity of expectation.

$$\begin{aligned}
\mathbb{E}(T_k) &= \frac{1}{\sqrt{\mathsf{CG}(f)}} \Pr[\mathcal{E}_k] + \Pr[\overline{\mathcal{E}_k}] \cdot \mathbb{E}[w_x(i_k) \mid \overline{\mathcal{E}_k}] \\
&\geq \frac{1}{\sqrt{\mathsf{CG}(f)}} \Pr[\mathcal{E}_k] + \Pr[\overline{\mathcal{E}_k}] \cdot \sum_{i:x_i\neq y_i} w_{x,i} \frac{w_{y,i}}{\sum_{j\in[n]} w_{y,j}} \\
&\geq \frac{1}{\sqrt{\mathsf{CG}(f)}} \Pr[\mathcal{E}_k] + \Pr[\overline{\mathcal{E}_k}] \cdot \frac{1}{\sqrt{\mathsf{CG}(f)}} \sum_{i:x_i\neq y_i} w_{x,i} w_{y,i} \\
&\geq \frac{1}{\sqrt{\mathsf{CG}(f)}}\,.
\end{aligned}$$

  Note that a query i_k contributes non-zero weight to T_k only if it has not been queried so far. Since x and y are consistent with the queries made so far, the set of indices where x and y differ is a subset of the set of indices that has not been queried yet. The first inequality is due to this fact. The second and third inequalities follow from the weight formulation of CG (Proposition 4.1.1), i.e. $\sum_{j\in[n]} w_{y,j} \leq \sqrt{\mathsf{CG}(f)}$ and $\sum_{i:x_i\neq y_i} w_{x,i} w_{y,i} \geq 1$. By linearity of expectation, we get

$$\mathbb{E}(T) = \sum_{k=1}^{\lceil \mathsf{CG}/\epsilon \rceil} \mathbb{E}(T_k) \geq \frac{\sqrt{\mathsf{CG}(f)}}{\epsilon}$$

From the upper and lower bounds, we have $\frac{\sqrt{\mathsf{CG}(f)}}{\epsilon} \leq \mathbb{E}(T) \leq \left(1+\frac{p}{\epsilon}\right)\sqrt{\mathsf{CG}(f)}$. This gives that $\mathcal{A}$ terminates in step 1 with probability at least $1-\epsilon$ which proves the

claim. □

The following upper bound on CG for total Boolean functions in terms of CG^{pub} and sensitivity s follows from an upper bound on EC in terms of FC and s.

Theorem 4.1.5. *For any total Boolean function f,*

$$\mathsf{CG}(f) \leq O(\mathsf{CG}^{pub}(f)^2 \mathsf{s}(f)).$$

Proof. Jain et al. [58] showed that $\mathsf{EC}(f)^2 \leq O(\mathsf{FC}(f)^2 \mathsf{s}(f))$, the proof will be reproduced below (Theorem 4.1.6) for completeness. From Theorem 4.1.3, we have $\mathsf{CG}(f) \leq O(\mathsf{EC}(f)^2)$, and we have $\mathsf{FC}(f) \leq \mathsf{CG}^{\text{pub}}(f)$ from Proposition 3.5.1. We get the desired result by combining the three inequalities. □

Theorem 4.1.6 ([58], Lemma 3)**.** *For any total Boolean function f,*

$$\mathsf{EC}(f) \leq O(\mathsf{FC}(f)\sqrt{\mathsf{s}(f)}).$$

Before we look into the proof of this theorem, we state and prove the following lemma given by Kulkarni et al. [65] which will come in handy for the proof.

Lemma 4.1.7 ([65], Lemma 6.2)**.** *For a total Boolean function $f : \{0,1\}^n \to \{0,1\}$, let $\{v_x\}_{x\in\{0,1\}^n}$ be a feasible solution of the linear program for fractional certificate complexity FC. For any $x, y \in \{0,1\}^n$ such that $f(x) \neq f(y)$, we have,*

$$\sum_{i:x_i\neq y_i} \min\{v_{x,i}, v_{y,i}\} \geq 1 \,.$$

Proof. Consider an input $z \in \{0,1\}^n$, that lies in a shortest path between x and y in the hypercube, which is defined as follows:

$$z_i = \begin{cases} y_i & \text{if } v_{x,i} < v_{y,i} \\ x_i & \text{otherwise.} \end{cases}$$

From the above definition, we have

$$\sum_{i:x_i\neq z_i} v_{x,i} = \sum_{i:v_{x,i}<v_{y,i} \text{ and } x_i\neq y_i} v_{x,i} \leq \sum_{i:x_i\neq y_i} \min\{v_{x,i}, v_{y,i}\}.$$

Similarly, we also have $\sum_{i:y_i\neq z_i} v_{y,i} \leq \sum_{i:x_i\neq y_i} \min\{v_{x,i}, v_{y,i}\}$. Since $f(x) \neq f(y)$, the function value at z must differ from that at either x or y. Without loss of generality, let us assume that $f(x) \neq f(z)$. Since $\{v_x\}_{x\in\{0,1\}^n}$ forms a feasible solution for FC, we have $\sum_{i:x_i\neq z_i} v_{x,i} \geq 1$ which implies that $\sum_{i:x_i\neq y_i} \min\{v_{x,i}, v_{y,i}\} \geq 1$. □

Proof of Theorem 4.1.6. The main idea of this proof is to modify the weights $\{v_x\}_{x\in\{0,1\}^n}$, given by an optimal solution for FC, by pruning out small weights and boosting the larger weights by a constant factor. These modified weights are used to construct a feasible solution for $\mathsf{EC}(f)$. The weights are modified as follows:

$$v'_{x,i} = \begin{cases} \min\{\frac{3}{2} v_{x,i}, 1\} & \text{if } v_{x,i} \geq \frac{1}{3\mathsf{s}(f)} \\ 0 & \text{otherwise.} \end{cases}$$

We now check that these modified weights are a feasible solution for $\mathsf{FC}(f)$. For any input $x \in \{0,1\}^n$, let $\mathcal{B}$ be the minimal sensitive block for x. Since $\{v_x\}_{x\in\{0,1\}^n}$ are a

feasible solution for FC, we have

$$1 \leq \sum_{i\in\mathcal{B}} v_{x,i} = \sum_{\substack{i\in\mathcal{B}\\ v_{x,i}\geq 1/3\mathsf{s}(f)}} v_{x,i} + \sum_{\substack{i\in\mathcal{B}\\ v_{x,i}<1/3\mathsf{s}(f)}} v_{x,i}$$
$$< \sum_{\substack{i\in\mathcal{B}\\ v_{x,i}\geq 1/3\mathsf{s}(f)}} v_{x,i} + \frac{1}{3\mathsf{s}(f)}\cdot\mathsf{s}(f) = \sum_{\substack{i\in\mathcal{B}\\ v_{x,i}\geq 1/3\mathsf{s}(f)}} v_{x,i} + \frac{1}{3},$$

where the inequality comes from the fact that the size of a minimally sensitive block is at most $\mathsf{s}(f)$. This shows that

$$\sum_{\substack{i\in\mathcal{B}\\ v_{x,i}\geq 1/3\mathsf{s}(f)}} v_{x,i} > \frac{2}{3}$$

and thus $\sum_{i\in\mathcal{B}} v'_{x,i} \geq 1$. A feasible solution for EC is constructed from these modified weights as,

$$w_{x,i} = \sqrt{v'_{x,i}}.$$

These weights lie between 0 and 1 since $0 \leq v'_{x,i} \leq 1$. For all x, y such that $f(x) \neq f(y)$,

$$\sum_{i:x_i\neq y_i} w_{x,i}w_{y,i} = \sum_{i:x_i\neq y_i} \sqrt{v'_{x,i}v'_y(i)} \geq \sum_{i:x_i\neq y_i} \min\{v'_{x,i}, v'_y(i)\} \geq 1.$$

The first inequality holds as $\sqrt{ab} \geq \min\{a,b\}$ for any $a, b \geq 0$ and the second inequality follows from Lemma 4.1.7 and the fact that $\{v'_x\}_{x\in\{0,1\}^n}$ forms a feasible solution for $\mathsf{FC}(f)$. This shows that $\{w_x\}_{x\in\{0,1\}^n}$ is a feasible solution for EC and for any $x \in \{0,1\}^n$,

$$\sum_{i\in[n]} w_{x,i} = \sum_{i\in[n]} \sqrt{v'_{x,i}} = \sum_{\substack{i\in[n]\\ v'_{x,i}>0}} \frac{v'_{x,i}}{\sqrt{v'_{x,i}}} \leq \sqrt{3\mathsf{s}(f)} \sum_{i\in[n]} v'_{x,i} \leq \sqrt{3\mathsf{s}(f)}\frac{3}{2}\mathsf{FC}(f).$$

The first and second inequalities are due to the fact that $\frac{3}{2}v_x(i) \geq v'_{x,i} \geq \frac{1}{3\mathsf{s}(f)}$. We note that the above proof does not hold for partial functions as the existence of an input corresponding to a minimally sensitive block cannot be guaranteed in such a function. □

Lastly we prove the following upper bound on CG in terms of 0-certificate and 1-certificate complexity.

Theorem 4.1.8. *For any total Boolean function f,*

$$\mathsf{CG}(f) \leq \mathsf{C}^0(f)\mathsf{C}^1(f).$$

Proof. It is easy to see that $\mathsf{CG}(f) \leq \mathsf{C}^0(f)\cdot\mathsf{C}^1(f)$: on input x, each player outputs uniformly at random some index i in a minimal certificate for their input. The certificates must intersect in at least one index, otherwise we could simultaneously fix the value of f to 0 and to 1 by fixing both certificates. The strategy therefore succeeds when both players output the same index in the intersection, which occurs with probability at least $\frac{1}{\mathsf{C}^0(f)}\frac{1}{C^1(f)}$. □

4.2 Certificate games with quantum and non-signalling strategies

One surprising result of the work on certificate games concerns the shared entanglement model. In order to prove lower bounds for this model, we introduce non-signalling certificate games. Non-signalling states that when making a quantum measurement the outcome on one side should not leak any information about the measurement made on the other side. This "non-signalling bottleneck" is shared by all of our certificate game complexity measures. Identifying it turned out to be the key insight which led to a very strong lower bound on all these measures, including the quantum model, with a single, simple proof, not involving any of the technical overhead inherent to the quantum setting. The simplicity of the proof comes from the fact that the non-signalling model has several equivalent formulations as linear programs, and the strength of the bounds comes from the fact that it captures precisely a fundamental computational bottleneck. It also neatly highlights one of the key differences between quantum and classical query models, since the quantum query model somehow averts this bottleneck. In this section, we give a very short and simple proof that fractional certificate complexity is a lower bound on all of our certificate game models.

To illustrate the idea behind the proof and the technique we use, we start with a lower bound on the OR function for the quantum model. Consider a hypothetical strategy with shared entanglement that would allow two players to win the certificate game with probability more than $1/n$. The players could then use this strategy for the certificate game as a black box to convey information (without using communication) in the following way. Assume Alice wants to send an integer $i \in \{1, \ldots, n\}$ to Bob. Bob uses the input $x = 0^n$ and Alice the input $y = x^i$, i.e. all zero input with the i-th bit being 1. By running this game several times, Bob could learn i by taking the majority output of several runs of this game which would violate the non-signalling principle of quantum information.

In order to give a formal proof, we show that the non-signalling certificate game complexity of the OR function (Definition 1.2.1) is at least n.

Proposition 4.2.1. $\mathsf{CG}^{\mathsf{ns}}(\mathsf{OR}_n) \geq n$.

Since for every (possibly partial) Boolean function f, $\mathsf{CG}^{\mathsf{ns}}(f) \leq \mathsf{CG}^{*}(f)$, this proposition implies $\mathsf{CG}^{*}(\mathsf{OR}_n) \geq n$.

Proof of Proposition 4.2.1. We give a feasible solution to the dual formulation which consists of a hard distribution μ and an assignment to the variables $\gamma_{i,j,x,y}$ that satisfy the constraints given in Proposition 2.2.9. Let $x = 0^n$ and $\delta = \frac{1}{n}$. For the hard distribution, we only pick input pairs consisting of the all-zero input and inputs with Hamming weight 1, i.e. $\mu_{xy} = \frac{1}{n}$ when $y = x^i$, and is 0 otherwise. To satisfy the correctness constraint, we use γ to pick up a weight $1/n$ whenever a strategy AB fails on some pair (x, x^i). To this end, we define γ as follows: $\gamma_{i,j,x,x^i} = \frac{1}{n}$ for all $j \neq i$ and 0 everywhere else. To see that this satisfies the constraints, consider any strategy AB and let i be the output of A on x.

Case 1: When $B(x^i) = i$, AB is correct on the input pair (x, x^i) but it cannot be correct on any other input pair with non-zero weight under μ. Therefore,

$$\sum_{x',y' : A(x') = B(y') = i \text{ and } x'_i \neq y'_i} \mu_{x',y'} = \frac{1}{n} \quad \text{and} \quad \sum_{x',y'} \gamma_{A(x'),B(y'),x',y'} = 0.$$

Case 2: When $B(x^i) = j$ where $j \neq i$, AB is incorrect on all the input pairs with non-zero weight under μ and we have

$$\sum_{x',y':A(x')=B(y')=i \text{ and } x'_i \neq y'_i} \mu_{x',y'} = 0 \quad \text{and} \quad \sum_{x',y'} \gamma_{A(x'),B(y'),x',y'} = \frac{1}{n}.$$

Since $\delta = \frac{1}{n}$, this is a satisfying assignment which shows that

$$\mathsf{CG}^{\mathsf{ns}}(\mathsf{OR}) = \omega^{\mathsf{ns}}(G_{\mathsf{OR}})^{-1} \geq n.$$

□

Note that the quantum query complexity $\mathsf{Q}(\mathsf{OR})$ is $\Theta(\sqrt{n})$ [51, 17]. Thus OR shows that there exists a function for which $\mathsf{CG}^*(f) = \omega(\mathsf{Q}(f))$ (as opposed to the randomized model where $\mathsf{CG}^{\mathrm{pub}}(f) \leq O(\mathsf{R}(f))$. On the other hand, note that the function BKK (see Definition 1.2.7) constructed by [3] demonstrates that there exists a total Boolean function f with $\mathsf{C}(f) = O(\sqrt{\mathsf{Q}(f)})$; this f also shows that $\mathsf{CG}^{\mathrm{pub}}(f) \leq O(\sqrt{\mathsf{Q}(f)})$.

This lower bound on the OR function can be generalized, with a slightly more complicated weight assignment, to show that block sensitivity is a lower bound on the non-signalling value of the certificate games. However, using a different technique, we can prove an even stronger result. We do this by going back to the original definition of $\mathsf{CG}^{\mathsf{ns}}$ (Definition 2.2.8) and giving a very simple proof that $\mathsf{CG}^{\mathsf{ns}}$ is an upper bound on FC.

Theorem 4.2.2 (Lower bound on $\mathsf{CG}^{\mathsf{ns}}$)**.** *For any (possibly partial) Boolean function f, $\mathsf{bs}(f) \leq \mathsf{FC}(f) \leq \mathsf{CG}^{\mathsf{ns}}(f)$.*

Proof. Let $p(i,j|x,y)$ be the distribution over outcomes in an optimal nonsignalling strategy for $\mathsf{CG}^{\mathsf{ns}}(f)$. This distribution p verifies the nonsignalling condition, i.e. $\sum_j p(i,j|x,y) = \sum_j p(i,j|x,y')$ for all x,y,y',i and hence we can write the marginal distribution for x as $p(i|x) = \sum_j p(i,j|x,y)$ since it does not depend on y. Notice that $p(i|x) = \sum_j p(i,j|x,y) \geq p(i,i|x,y)$ for all x,y,i.

With $\delta = \frac{1}{\mathsf{CG}^{\mathsf{ns}}(f)}$, we have that $\sum_{i:x_i \neq y_i} p(i,i|x,y) \geq \delta$ for all x,y such that $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$. Let $v_{x,i} = p(i|x)/\delta$ for all $x \in \{0,1\}^n$ and $i \in [n]$. Using these weights, we get $\sum_i v_{x,i} = \frac{1}{\delta}$ for all x (since p is a distribution). For any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, we have

$$\sum_{i:x_i \neq y_i} v_{x,i} = \sum_{i:x_i \neq y_i} p(i|x)/\delta \geq \sum_{i:x_i \neq y_i} p(i,i,x,y)/\delta \geq 1.$$

Since this is a feasible solution to FC, we have $\mathsf{FC}(f) \leq \mathsf{CG}^{\mathsf{ns}}(f)$. □

The lower bound can be improved by slightly modifying the proof to hold for the Classical Adversary bound CMM (Definition 1.1.11). This measure was introduced in [1, 68] as a lower bound for randomized query complexity R and was shown to equal fractional certificate complexity FC for total functions (but can be larger for partial functions) [11].

Theorem 4.2.3 (Lower bound on $\mathsf{CG}^{\mathsf{ns}}$ by CMM)**.** *For any (possibly partial) Boolean function f, $\mathsf{CMM}(f) \leq \mathsf{CG}^{\mathsf{ns}}(f)$.*

Proof. We build the probability distributions $p(i|x)$ and $p(i|y)$ from the marginal distribution for x and y as in the proof above, i.e. $p(i|x) = \sum_j p(i,j|x,y)$ and $p(j|y) = \sum_i p(i,j|x,y)$. Since $p(i|x) = \sum_j p(i,j|x,y) \geq p(i,i|x,y)$ and similarly $p(i|y) \geq p(i,i|x,y)$ for all x,y,i by definition, we have $\min\{p(i|x), p(i|y)\} \geq p(i,i|x,y)$.

$$\mathsf{CMM}(f) = \min_p \max_{\substack{x\in f^{-1}(0)\\ y\in f^{-1}(1)}} \frac{1}{\sum_{i:x_i\neq y_i} \min\{p(i|x), p(i|y)\}} \leq \min_p \max_{\substack{x\in f^{-1}(0)\\ y\in f^{-1}(1)}} \frac{1}{\sum_{i:x_i\neq y_i} p(i,i|x,y)}$$

Since $\sum_{i:x_i\neq y_i} p(i,i|x,y) \geq \frac{1}{\mathsf{CG}^{\mathsf{ns}}(f)}$ for all $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, we get $\mathsf{CMM}(f) \leq \mathsf{CG}^{\mathsf{ns}}(f)$. □

To summarize the key idea of this section, introducing the non-signalling model of certificate games provides a very clean and simple way to give lower bounds on all of our previous models, including the shared entanglement model. It has several linear formulations, making it very easy to give upper and lower bounds. Finally, it captures an essential feature of zero-communication games, which we think of as the "non-signalling bottleneck". As a bonus, it allows us to give proofs on the shared entanglement model without having to get into the technicalities of what characterizes quantum games.

4.3 Single bit versions of certificate games

Aaronson et al. [4] defined single-bit versions of several formulations of the adversary method and showed that they are all equal to the spectral sensitivity λ. Informally, single-bit versions of these measures are obtained by considering the constraints only with respect to pairs (x,y) such that $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, and x and y differ only in a single bit.

Let $|x-y|$ denote the Hamming distance between x and y, and x^i the string obtained from x by flipping the value of the i-th bit. The single-bit version of $\mathsf{MM}(f)$ was defined in [4] as follows.

Definition 4.3.1 (Single bit minimax adversary [4])**.** *For any (possibly partial) Boolean function $f : \{0,1\}^n \to \{0,1\}$*

$$\mathsf{MM}_{[1]}(f) = \min_w \max_x \sum_i w_{x,i}$$
$$\text{such that } w_{x,i}\, w_{x^i,i} \geq 1 \quad \forall x \in f^{-1}(0),\ x^i \in f^{-1}(1)$$

where $x \in \{0,1\}^n$ and $i \in [n]$.

Using a proof similar to that of Proposition 4.1.1, it can be shown that this definition is equivalent to the following formulation in terms of weights which we include for comparison with other definitions.

Proposition 4.3.2 (Equivalent formulation for $\mathsf{MM}_{[1]}$)**.** *For any (possibly partial) Boolean function $f : \{0,1\}^n \to \{0,1\}$*

$$\mathsf{MM}_{[1]}(f) := \min_p \max_{\substack{x,y\in f^{-1}(0)\times f^{-1}(1)\\ |x-y|=1}} \frac{1}{\sum_{i:x_i\neq y_i} \sqrt{p_{x,i}\, p_{y,i}}} = \min_p \max_{\substack{x\in f^{-1}(0)\\ x^i\in f^{-1}(1)}} \frac{1}{\sqrt{p_{x,i}\, p_{x^i,i}}} \quad (4.1)$$

where p ranges over all families of nonnegative $p_{x,i} \in \mathbb{R}$ such that $\sum_{i\in[n]} p_{x,i} = 1$ for all x.

Aaronson et al. [4] showed that λ is equivalent to a host of single-bit complexity measures including $\mathsf{MM}_{[1]}$.

Theorem 4.3.3 ([4], Theorem 28)**.** *For any (possibly partial) Boolean function* $f : \{0,1\}^n \to \{0,1\}$*,*

$$\lambda(f) = \mathsf{MM}_{[1]}(f) \ .$$

In this section, we consider single-bit versions of $\mathsf{CG}^{\mathrm{pub}}$ and CG and show that they characterise sensitivity and λ^2 respectively up to constant factors.

Definition 4.3.4 (Single-bit private coin certificate game complexity)**.** *For any (possibly partial) Boolean function* $f : \{0,1\}^n \to \{0,1\}$*,*

$$\mathsf{CG}_{[1]}(f) := \min_{p} \max_{\substack{x,y \in f^{-1}(0)\times f^{-1}(1) \\ |x-y|=1}} \frac{1}{\omega(p;x,y)} = \min_{p} \max_{\substack{x\in f^{-1}(0) \\ x^i \in f^{-1}(1)}} \frac{1}{p_{x,i}\, p_{x^i,i}},$$

where p *is a collection of nonnegative variables* $\{p_{x,i}\}_{x,i}$ *that satisfy* $\sum_{i\in[n]} p_{x,i} = 1$ *for all* $x \in \{0,1\}^n$*, and* $\omega(p;x,x^i)$ *is the probability that both players output the unique index* i *where* x *and* x^i *differ.*

Note that the winning probability for the single bit version of CG is given by

$$\omega(p;x,x^i) = p_{x,i}\, p_{x^i,i}.$$

Recall that when the players share randomness, a public-coin randomized strategy is a distribution over pairs (A,B) of deterministic strategies. We assign a nonnegative variable p_{AB} to each strategy and require that they sum to 1. We say that a pair of strategies (A,B) is correct on x,y if $A(x) = B(y) = i$ and $x_i \neq y_i$.

Definition 4.3.5 (Single-bit public coin certificate game complexity)**.** *For any (possibly partial) Boolean function* $f : \{0,1\}^n \to \{0,1\}$*,*

$$\mathsf{CG}^{pub}_{[1]}(f) := \min_{p} \max_{\substack{x,y \in f^{-1}(0)\times f^{-1}(1) \\ |x-y|=1}} \frac{1}{\omega^{pub}(p;x,y)} = \min_{p} \max_{\substack{x\in f^{-1}(0) \\ x^i \in f^{-1}(1)}} \frac{1}{\omega^{pub}(p;x,x^i)},$$

where p *is a collection of nonnegative variables* $\{p_{AB}\}$ *for pairs of strategies* (A,B) *that satisfy* $\sum_{(A,B)} p_{AB} = 1$ *and the winning probability*

$$\omega^{pub}(p;x,y) = \sum_{(A,B) \ \text{correct on } x,y} p_{AB} \ .$$

We prove the following.

Theorem 4.3.6. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$

$$\mathsf{CG}^{pub}_{[1]}(f) = \Theta(\mathsf{s}(f)) \ .$$

Proof.

Upper bound by sensitivity:

We use a hashing based approach similar to the upper bounds on $\mathsf{CG}^{\mathrm{pub}}$ by C and EC (Section 3.3).

Let S be a finite set of cardinality $\mathsf{s}(f)$. An element $z \in S$ is fixed as part of the specification of the protocol, that is z does not depend on the input.

Using shared randomness, the players select a hash function $h : [n] \to S$ as follows. Let $h : [n] \to S$ be a random hash function such that for each $i \in [n]$, $h(i)$ is selected independently and uniformly at random from S.

Let A_x be the set of indices corresponding to the sensitive bits of x, that is $A_x = \{i \in [n] \mid f(x^i) = 1\}$ for an $x \in f^{-1}(0)$. Similarly, let $B_y = \{i \in [n] \mid f(y^i) = 0\}$ for a $y \in f^{-1}(1)$.

After selecting h using shared randomness, the players proceed as follows. Alice on input x outputs an index $i \in A_x$ such that $h(i) = z$, and Bob on input y outputs an index $j \in B_y$ such that $h(j) = z$. If they have several valid choices or if they have no valid choice, they output arbitrary indices. Since $|x - y| = 1$, let $i^* \in A_x \cap B_y$ be the unique index where their inputs differ, i.e. $x_{i^*} \neq y_{i^*}$.

We now estimate the probability that both players output i^*. Recall that by the definition of h,

$$\Pr_h[h(i^*) = z] = \frac{1}{|S|} = \frac{1}{\mathsf{s}(f)} .$$

Notice that for any $i \in A_x \cup B_y$, the number of elements different from i in $A_x \cup B_y$ is $\ell = |A_x \cup B_y| - 1 \leq 2(|S| - 1)$, since $\max\{|A_x|, |B_y|\} \leq |S| = \mathsf{s}(f)$. Thus for any $z \in S$ and any $i \in A_x \cup B_y$, the probability (over the choice of h) that no element other than i in $A_x \cup B_y$ is mapped to z by h is $(1 - \frac{1}{|S|})^\ell \geq \frac{1}{e^2}$.

Hence the players output a correct answer with probability at least $\frac{1}{e^2}\frac{1}{\mathsf{s}(f)}$.

Lower bound by sensitivity:

We will now consider the dual formulation of $\mathsf{CG}^{\mathrm{pub}}_{[1]}$ which is similar to that of $\mathsf{CG}^{\mathrm{pub}}$ given in Proposition 2.2.5. The only difference is that the distribution μ takes nonzero values only on pairs (x, x^i), i.e. on pairs with Hamming distance 1. Let x^* be an input such that $\mathsf{s}(f; x^*) = \mathsf{s}(f) =: s$, and assume without loss of generality that $f(x^*) = 0$. Consider the following distribution μ over input pairs at Hamming distance 1: $\mu_{x^*,y} = \frac{1}{s}$ for $y \in f^{-1}(1)$ such that $|x^* - y| = 1$ and $\mu_{x^*,y} = 0$ for every other y. Furthermore, we choose $\mu_{x,y} = 0$ for any y and $x \neq x^*$. Thus, we only have s input pairs with nonzero measure.

Let A, B be any pair of deterministic strategies for Alice and Bob. Since A is a deterministic strategy, if $A(x^*) = i$, Alice will output the same index i for every pair (x^*, y). This means that the probability over μ that the players win is at most $\frac{1}{s(f;x)} = \frac{1}{s} = \frac{1}{\mathsf{s}(f)}$ for any pair of deterministic strategies.

□

Note that one can similarly define single-bit versions of FC and EC, and it is easy to see that both are equal to sensitivity.

Definition 4.3.7. *For any (possibly partial) Boolean function $f : \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{FC}_{[1]}(f) = \max_{x \in \{0,1\}^n} \mathsf{FC}_{[1]}(f, x),$$

where

$$\mathsf{FC}_{[1]}(f, x) = \min_v \sum_i v_{x,i},$$

subject to $v_{x,i} \geq 1$ for all i such that $f(x) \neq f(x^i)$, where v is a collection of variables $v_{x,i} \geq 0$.

Definition 4.3.8. *For any (possibly partial) Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathsf{EC}_{[1]}(f) = \min_{w} \max_{x} \sum_{i \in [n]} w_{x,i},$$

where w *is a collection of variables* $0 \leq w_{x,i} \leq 1$ *that satisfy* $w_{x,i} w_{x^i,i} \geq 1$ *for all* x *and* i *such that* $f(x) \neq f(x^i)$.

Proposition 4.3.9. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathsf{s}(f) = \mathsf{FC}_{[1]}(f) = \mathsf{EC}_{[1]}(f)\,.$$

Proof. We can think of the values $v_{x,i}$ and $w_{x,i}$ as weights assigned to the edges of the Boolean hypercube. We say that an edge (x, x^i) is sensitive with respect to the function f iff $f(x) \neq f(x^i)$. Notice that both definitions have constraints that require a weight at least 1 on each sensitive edge, and thus both $\mathsf{FC}_{[1]}(f)$ and $\mathsf{EC}_{[1]}(f)$ are at least $s(f)$. On the other hand, placing a weight 1 on each sensitive edge and a weight 0 on every other edge satisfies the constraints in both definitions, making $\mathsf{FC}_{[1]}(f)$ and $\mathsf{EC}_{[1]}(f)$ at most $s(f)$. □

Combining the results above we get the following.

Corollary 4.3.10. *For any Boolean function* $f : \{0,1\}^n \to \{0,1\}$

$$\mathsf{s}(f) = \mathsf{FC}_{[1]}(f) = \mathsf{EC}_{[1]}(f) = \Theta(\mathsf{CG}^{pub}_{[1]}(f))\,.$$

We now look at the single-bit version of $\mathsf{CG}^{\mathrm{pub}}$ about which we have the following theorem.

Theorem 4.3.11. *For any (possibly partial) Boolean function* $f : \{0,1\}^n \to \{0,1\}$,

$$\mathsf{CG}_{[1]}(f) = \lambda^2\,.$$

Proof. On comparing the definitions of $\mathsf{MM}_{[1]}$ and $\mathsf{CG}_{[1]}$, i.e. the formulation of $\mathsf{MM}_{[1]}$ in Proposition 4.3.2 with Definition 4.3.4, we see that $\sqrt{\mathsf{CG}_{[1]}} = \mathsf{MM}_{[1]}$. One can also restate Definition 4.3.4 with weights as in Proposition 4.1.1 and compare that with the formulation of $\mathsf{MM}_{[1]}$ in Definition 4.3.1. The statement then follows from Theorem 4.3.3. □

In the next chapter, we use the results we have so far for certificate game complexity to see what they imply and how these measures fit in the complexity landscape.

Chapter 5

Relations and separations between measures

In this chapter, we put together our results about certificate games in the context of known results and try to understand how tight the relations are (Section 5.1). In addition, we use the hashing framework to show an exponential separation between R and $\mathsf{CG}^{\mathrm{pub}}$ for a partial function (Section 5.2), the techniques used for which may be of independent interest.

5.1 Relationship between various models of certificate games

Understanding the relationships between various models of certificate game complexity would help us understand the power of shared randomness over private randomness and that of quantum shared entanglement over shared randomness in the context of certificate games. The following results follow as corollaries to our other results presented in the previous chapters and from other previously known results in this area. We start by relating $\mathsf{CG}^{\mathrm{pub}}$ and $\mathsf{CG}^{\mathsf{ns}}$.

Corollary 5.1.1. *For any total Boolean function f, $\mathsf{CG}^{\mathsf{ns}}(f) \leq \mathsf{CG}^{pub}(f) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^{3/2})$.*

Proof. The first inequality follows from the definitions. The second inequality follows from the following string of inequalities,

$$\mathsf{CG}^{\mathrm{pub}}(f) \leq O(\mathsf{EC}(f)) \leq O(\mathsf{FC}(f) \cdot \sqrt{\mathsf{s}(f)}) \leq O(\mathsf{CG}^{\mathsf{ns}}(f) \cdot \sqrt{\mathsf{s}(f)}) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^{3/2}),$$

where the first inequality follows from Theorem 3.3.2 and the second inequality (Theorem 4.1.6) was proven by Jain et al. [58]. The third and fourth inequalities follow from Theorem 4.2.2. □

The question of whether $\mathsf{FC} = \Theta(\mathsf{EC})$ was posed as an open problem by Jain et al. [58] and if true, would imply $R_0 \leq O(\mathsf{FC}^2)$ which will answer a well-known open problem by Aaronson [2]. We do not yet know of a total Boolean function for which FC is significantly lower than $\mathsf{CG}^{\mathsf{ns}}$ or $\mathsf{CG}^{\mathrm{pub}}$ which would have implied a separation between FC and EC. We can however show a separation between FC and EC in the case of partial functions. Since $\mathsf{CG}^{\mathsf{ns}} \geq \mathsf{CMM}$ by Theorem 4.2.3, we have for the partial function "Greater than Half" (Definition 1.2.8) $\mathsf{CG}^{\mathsf{ns}}(\mathsf{GTH}) = \Theta(n)$ and $\mathsf{FC}(\mathsf{GTH}) = O(1)$ which provides an arbitrary separation between these measures. For total functions, we have the following set of open problems.

Open Problem 1 : Are any two complexity measures asymptotically separated by a total function in the following chain of inequalities?

$$\mathsf{FC}(f) \leq \mathsf{CG}^{\mathsf{ns}}(f) \leq \mathsf{CG}^{*}(f) \leq \mathsf{CG}^{\mathrm{pub}}(f) \leq O(\mathsf{EC}(f))$$

We now look at how big CG can be with respect to $\mathsf{CG}^{\mathsf{ns}}$ and the following is the best known relation between them.

Corollary 5.1.2. *For any total Boolean function f, $\mathsf{CG}^{\mathsf{ns}}(f) \leq \mathsf{CG}(f) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^3)$.*

Proof. The first inequality follows from the definitions and the second inequality is from

$$\mathsf{CG} \leq O(\mathsf{EC}(f)^2) \leq O(\mathsf{FC}^2(f) \cdot \mathsf{s}(f)) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^2 \cdot \mathsf{s}(f)) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^3),$$

where the first inequality follows from Theorem 4.1.3, the second from Theorem 4.1.6 proven in [58] and the last two inequalities follow from Theorem 4.2.2. □

We pose an open problem of whether there exists a better upper bound on CG in terms of $\mathsf{CG}^{\mathsf{ns}}$.

Open Problem 2 : Is there a $c < 3$ such that $\mathsf{CG}(f) \leq O(\mathsf{CG}^{\mathsf{ns}}(f)^c)$?

Even in the case of an upper bound on CG in terms of $\mathsf{CG}^{\mathrm{pub}}$, we do not have a tighter result. The best known separation between these two measures is quadratic and is given by Tribes and Parity functions.

Corollary 5.1.3. *While $\mathsf{CG}^{pub}(\mathsf{OR}_{\sqrt{n}}) = \mathsf{CG}^{\mathsf{ns}}(\mathsf{OR}_{\sqrt{n}}) = \Theta(\sqrt{n})$, the certificate game complexities for $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}} := \mathsf{OR}_{\sqrt{n}} \circ \mathsf{AND}_{\sqrt{n}}$ are as follows.*

- $\mathsf{CG}^{\mathsf{ns}}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \mathsf{CG}^{pub}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(\sqrt{n})$, *and*
- $\mathsf{CG}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$.

Proof. Firstly, since the functions OR and AND have full sensitivity, from Theorem 4.2.2 we have $\mathsf{CG}^{\mathrm{pub}}(\mathsf{OR}_{\sqrt{n}}) = \mathsf{CG}^{\mathsf{ns}}(\mathsf{OR}_{\sqrt{n}}) = \Theta(\sqrt{n})$.

The sensitivity of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ is $\Theta(\sqrt{n})$ and hence from Theorem 4.2.2 we have that the $\mathsf{CG}^{\mathrm{pub}}$ and $\mathsf{CG}^{\mathsf{ns}}$ of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ is $\Omega(\sqrt{n})$. The upper bound follows from Theorem 3.3.2 and the fact that the certificate complexity of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$ is at most $\sqrt{n}$. We have also provided a separate proof (Theorem 3.1.1) for the upper bound on $\mathsf{CG}^{\mathrm{pub}}$ of $\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$. Thus we have $\mathsf{CG}^{\mathsf{ns}}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \mathsf{CG}^{\mathrm{pub}}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(\sqrt{n})$.

For the certificate game complexity with private randomness, we know from Theorem 4.1.3 that CG is bounded below by R_0 and that $\mathsf{R}_0(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$. Theorem 4.1.3 also gives an upper bound of EC^2 on CG and since $\mathsf{EC}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) \leq \mathsf{C}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) \leq \sqrt{n}$, we have $\mathsf{CG}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$. □

In fact, any function with $\lambda(f) = \Theta(n)$, such as the Parity function, demonstrates a quadratic gap between CG and $\mathsf{CG}^{\mathrm{pub}}$. This is true as $\mathsf{CG}(f) = \Omega((\mathsf{MM}(f))^2)$ from Theorem 4.1.2 and $\mathsf{MM}(f) = \Omega(\lambda(f))$ [67]. This shows that for any such function CG is $\Theta(n^2)$ while $\mathsf{CG}^{\mathrm{pub}}$ is $\Theta(n)$. For any total function f, we have $\mathsf{CG}(f) \leq O(\mathsf{CG}^{\mathrm{pub}}(f)^2 \cdot \mathsf{s}(f))$. A better upper bound is not known.

Open Problem 3 : Is $\mathsf{CG}(f) \leq O(\mathsf{CG}^{\mathrm{pub}}(f)^2)$ for all functions f?

The bound $\mathsf{CG}(f) \leq O(\mathsf{EC}(f)^2)$ is indeed tight since CG of the Parity function is $\Theta(n^2)$, while its EC is $\Theta(n)$. On the other hand, for the OR function CG is much smaller than the upper bound EC^2. From Theorem 4.1.8, $\mathsf{CG} \leq \mathsf{C}^0 \cdot \mathsf{C}^1$ and we get

$\mathsf{CG}(\mathsf{OR}_n) = \Theta(n)$ since $\mathsf{C}^1(\mathsf{OR}_n) = 1$ and $\mathsf{C}^0(\mathsf{OR}_n) = n$. Since $\mathsf{FC}(\mathsf{OR}_n) = \Omega(n)$ we have $\mathsf{EC}(\mathsf{OR}_n) = \Theta(n)$.

Another question is: what is the biggest separation between $\mathsf{CG}(f)$ and $\mathsf{MM}(f)$? To the best of our knowledge, the best upper bound on CG for total functions in terms of MM is

$$\mathsf{CG} \leq O(\mathsf{FC}^2\mathsf{s}) \leq O(\mathsf{MM}^6),$$

where the final inequality follows from the fact that $\mathsf{FC} \leq \mathsf{MM}^2$[12] and $\mathsf{s} \leq \lambda^2 \leq \mathsf{MM}^2$. The biggest separation between CG and MM in this direction is cubic: Ambainis et al. constructed a "pointer function" g, for which $\mathsf{R}_0(g) = \Omega(\mathsf{Q}(g)^3)$ [10]. We observe that for this pointer function

$$\mathsf{CG}(g) \geq \Omega(\mathsf{R}_0(g)) \geq \Omega(\mathsf{Q}(g)^3) \geq \Omega(\mathsf{MM}(g)^3),$$

where the first inequality follows from Theorem 4.1.3 and the other inequalities follows from earlier known results. Another function that achieves the same separation is the cheat sheet version of $k-$Forrelation function that gives a cubic separation between Q and R [14, 3].

The lower bound on CG in terms of MM, i.e. $\mathsf{MM}^2 \leq O(\mathsf{CG})$ from Theorem 4.1.2, is tight for any function with full spectral sensitivity, such as Parity. In fact, the two quantities, CG and MM^2, are asymptotically identical for symmetric functions [77].

We also note that the lower bound on CG by R_0, i.e. $\mathsf{CG}(f) \geq \Omega(\mathsf{R}_0(f))$ from Theorem 4.1.3, is tight: for the function OR, $\mathsf{CG}(\mathsf{OR}) = \mathsf{R}_0(\mathsf{OR}) = \Theta(n)$. There exist functions like Parity for which this lower bound on CG is much smaller than CG since $\mathsf{CG}(\mathsf{Parity}) = \Theta(n^2)$ while $\mathsf{R}_0(\mathsf{Parity}) = \Theta(n)$.

Another upper bound on CG that we observe is $\mathsf{CG} \leq \mathsf{C}^0 \cdot \mathsf{C}^1$. While for some functions (such as Tribes) the two quantities CG and $\mathsf{C}^0 \cdot \mathsf{C}^1$ are asymptotically equal, we note that there are functions for which CG is significantly less than $\mathsf{C}^0 \cdot \mathsf{C}^1$.

Theorem 5.1.4 ([47, 58]). *The total function* $\mathsf{GSS}_1 : \{0,1\}^{n^2} \to \{0,1\}$ *(Definition 1.2.5) has the following properties:*

- $\mathsf{C}^0(\mathsf{GSS}_1) = \Theta(n^2)$ *and* $\mathsf{C}^1(\mathsf{GSS}_1) = \Theta(n)$,
- $\mathsf{EC}(\mathsf{GSS}_1) = \Theta(n)$,
- $\mathsf{CG}(\mathsf{GSS}_1) = \Theta(n^2)$.

Proof. Since $\mathsf{CG} \leq \mathsf{EC}^2$ by Theorem 4.1.3, $\mathsf{CG}(\mathsf{GSS}_1) = \Theta(n^2)$. The other properties stated above were shown in references [47, 58]. For this function, $C^0(\mathsf{GSS}_1) \cdot C^1(\mathsf{GSS}_1) = \Omega(\mathsf{CG}(\mathsf{GSS}_1)^{3/2})$. □

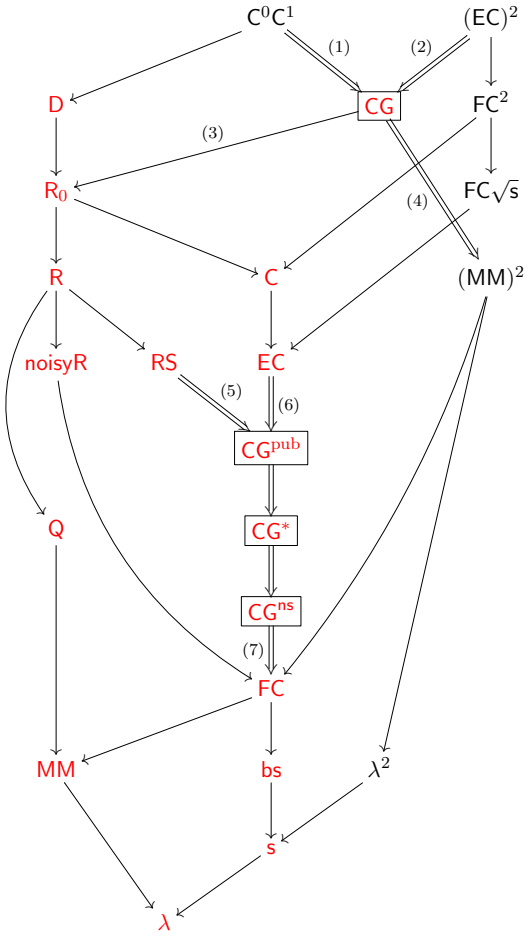
One of the most interesting open problems in this area of complexity theory is the quadratic sensitivity-block sensitivity conjecture. In Huang's seminal work [56] the degree of a Boolean function was bounded above by the square of sensitivity, and this is tight for Boolean functions. Since the degree of a Boolean function is quadratically related to the block sensitivity, we have $\mathsf{bs}(f) \leq O(\mathsf{s}(f)^4$. Unfortunately, this approach via degree will not be able to give any tighter bound on block sensitivity in terms of sensitivity.

Estimating certificate game complexity could be a possible way to prove a tighter bound on block sensitivity in terms of sensitivity. Given the result in Theorem 4.3.6, designing a strategy for $\mathsf{CG}^{\mathrm{pub}}$ using a strategy for $\mathsf{CG}^{\mathrm{pub}}_{[1]}$ may help us solve the sensitivity-block sensitivity conjecture.

Open Problem 4 : What is the smallest c such that, for any Boolean function f, $\mathsf{CG}^{\mathrm{pub}}(f) = O(\mathsf{CG}^{\mathrm{pub}}_{[1]}(f)^c)$?

Note that proving $\mathsf{CG}^{\mathrm{pub}}(f) = O(\mathsf{CG}^{\mathrm{pub}}_{[1]}(f)^2)$ would prove that $\mathsf{bs}(f) \leq O(\mathsf{s}(f)^2)$. It may seem too much to expect that the single-bit version of the game can help get upper bounds on the general public coin setting, but thanks to Huang's breakthrough result [56], we already know that $\mathsf{CG}^{\mathrm{pub}}(f) = O(\mathsf{CG}^{\mathrm{pub}}_{[1]}(f)^5)$.

Our main results for total functions in addition to known relations are illustrated in Figure 5.1. While most of our results also hold for partial functions, for simplicity we do not indicate them in the figure. Instead we specify in each theorem whether our result holds for partial functions or not.



1. Theorem 4.1.8. Separation: GSS_1 (follows from the fact that $\mathsf{C}^1(\mathsf{GSS}_1) = \Theta(n)$ and $\mathsf{C}^0(\mathsf{GSS}_1) = \Theta(n^2)$). Tightness: Parity.
2. Theorem 4.1.3, Separation: OR, Tightness: Parity.
3. Implicit in [58] (Theorem 4.1.3). Separation: Parity, Tightness: OR.
4. Theorem 4.1.2 Separation: Pointer function in [10] and the cheat sheet version of the $k-$Forrelation function [14, 3]. Tightness: OR.
5. Theorem 3.4.1 and Proposition 3.4.2. Separation: Tribes (Theorem 3.1.1 and $\mathsf{RS}(\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}) = \Theta(n)$ because RS composes [22]). Tightness: Parity.
6. Theorem 3.3.2. Separation: OPEN, Tightness: Parity.
7. Theorem 4.2.2. Separation: OPEN, Tightness: Parity.

FIGURE 5.1: Some relations among complexity measures including the certificate games complexity variants for total functions. An arrow from A to B indicates that for every total Boolean function f, $B(f) = O(A(f))$. Double arrows indicate results in this thesis, and boxes indicate new complexity measures. Single arrows indicate known results. For the examples on separation and tightness, see Table 5.1 and Table 5.2.

We compile various complexity measures of some of the total and partial functions mentioned in this thesis in Table 5.1 and Table 5.2 respectively. Blank spaces indicate that tight bounds are not known.

Function	λ	s	bs	FC	MM	Q	$\mathsf{CG}^{\mathrm{pub}}$	R	EC	C	CG
OR_n	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Parity_n	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n^2)$
$\mathsf{Tribes}_{\sqrt{n},\sqrt{n}}$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(n)$
GSS_1		$\Theta(n)$	$\Theta(n)$	$\Theta(n)$			$\Theta(n)$		$\Theta(n)$	$\Theta(n^2)$	$O(n^2)$

TABLE 5.1: Some of the commonly referred total functions and their complexity measures including certificate game complexity.

Function	λ	s	bs	FC	MM	Q	CMM	$\mathsf{CG}^{\mathrm{pub}}$	R	EC	C	CG
AplnD		0	$O(1)$	$O(1)$				$O(\log k)$	$\Theta(\sqrt{k \log k})$		$O(1)$	
GTH_n		0	$O(1)$	$O(1)$			$\Theta(n)$	$\Theta(n)$		$\Theta(n)$	$O(1)$	$\Theta(n)$

TABLE 5.2: The known complexity measures for AplnD and GTH_n.

5.2 Approximate Index: Exponential gap between R and $\mathsf{CG}^{\mathrm{pub}}$ for a *partial* Boolean function

We have seen that $\mathsf{CG}^{\mathrm{pub}}$ of a Boolean function lies between its randomized query complexity and randomized certificate complexity. The same is true for noisyR.

The measure noisyR (Definition 1.1.14) was introduced to study how randomised query complexity R behaves under composition and it was shown that $\mathsf{R}(f \circ g) = \Omega(\mathsf{noisyR}(f)\mathsf{R}(g))$ [21]. Since it was also shown that almost all lower bounds (except Q) on R are also lower bounds on noisyR, it is interesting to see whether $\mathsf{CG}^{\mathrm{pub}}$ is also a lower bound on noisyR.

Open Problem 5 : Is $\mathsf{CG}^{\mathrm{pub}}(f) \leq O(\mathsf{noisyR}(f))$ for all f?

Ben-David and Blais [21] constructed the approximate index function, which is the only function known where noisyR and R are different. The approximate index function that they construct is not a total Boolean function but a partial non-Boolean function which can be converted to a Boolean function by a suitable encoding. This would affect lower and upper bounds by at most a factor of two.

Let AplnD_k be the approximate index function where the input has an address part, say a, of k bits and a table with 2^k bits. The function is defined on inputs where all positions of the table labelled by strings within $\frac{k}{2} - \sqrt{k \log k}$ Hamming distance from a have the same value (either 0 or 1), and all positions that are farther away from a have 2 in them, i.e.

Definition 5.2.1. $\mathsf{AplnD}_k : \{0,1\}^k \times \{0,1,2\}^{2^k} \to \{0,1,*\}$ *is defined as*

$$\mathsf{AplnD}_k(a,x) = \begin{cases} x_a & \text{if } x_b = x_a \in \{0,1\} \text{ for all } b \text{ that satisfy } |b-a| \leq \frac{k}{2} - \sqrt{k \log k} \\ & \text{and } x_b = 2 \text{ for all other } b, \\ * & \text{otherwise.} \end{cases}$$

Ben-David and Blais showed that $\mathsf{noisyR}(\mathsf{AplnD}_k) = O(\log k)$, and $\mathsf{R}(\mathsf{AplnD}) = \Theta(\sqrt{k \log k})$. As an indication that $\mathsf{CG}^{\mathrm{pub}}$ could be a lower bound on noisyR, we show the following theorem.

Theorem 5.2.2. *The public coin certificate game complexity of* AplnD *on* $n = k + 2^k$ *bits is*

$$\mathsf{CG}^{pub}(\mathsf{AplnD}_k) = O(\log k).$$

We use the hashing framework to show an exponential separation between R and CG^{pub} of Approximate Index ApInd_k. The analysis of the strategy reduces to a very natural question: what is the intersection size of two Hamming balls of radius $\frac{k}{2} - \sqrt{k \log k}$ whose centers are at a distance $\frac{k}{\log k}$? We show that the intersection is at least an $\Omega(\frac{1}{\sqrt{\log k}})$ fraction of the total volume of the Hamming ball. This result and the techniques used could be of independent interest.

To bound the intersection size, we focus on the outermost $\sqrt{k}$ layers of the Hamming ball (since they contain a constant fraction of the total volume), and show that for each such layer the intersection contains an $\Omega(\frac{1}{\sqrt{\log k}})$ fraction of the elements in that layer.

For a single layer, the intersection can be expressed as the summation of the latter half of a hypergeometric distribution $P_{k,m,r}$ from $\frac{m}{2}$ to m ($m = \frac{k}{\log k}$ is the distance between the Hamming Balls and r is the radius of the layer). By using the "symmetric" nature of the hypergeometric distribution around $\frac{m}{2}$ for a sufficient range of values (Lemma 5.2.10), this reduces to showing a concentration result around the expectation with width $\sqrt{m}$ (as the expectation for our choice of parameters is $\frac{m}{2} - O(\sqrt{m})$).

We use the standard concentration bound on hypergeometric distribution with width $\sqrt{r}$ (Lemma 5.2.9) and reduce it to the required width $\sqrt{m}$ by noticing a monotonicity property of the hypergeometric distribution (Lemma 5.2.11).

We will now look at the proof in detail. A central ingredient of the proof of Theorem 5.2.2 is the following lemma that captures yet another application of the hashing based framework introduced in Section 3.2. We state it in a more general form than what we need.

Lemma 5.2.3. *Let L be an integer. Assume that for every $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$ there are sets A_x and B_y of size L that depend only on x and y respectively such that any element of $A_x \cap B_y$ is a correct output on the input pair (x, y), i.e. for any $i \in A_x \cap B_y$, we have $x_i \neq y_i$. If for any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$*

$$L = |A_x| = |B_y| \leq t|A_x \cap B_y| \,,$$

$\mathsf{CG}^{pub}(f) \leq O(t^2)$.

Proof. Let A_x and B_y be sets of size L guaranteed by the statement of the lemma. We can assume that t in the statement of the lemma is such that $20 \leq t \leq 0.1L$ holds, since $O(L^2)$ is a trivial upper bound on $\mathsf{CG}^{\text{pub}}(f)$. Let S be a finite set with $|S| = \lfloor \frac{L}{2t} \rfloor > 1$. Let z be a fixed element of S (for e.g., the first element of S) given as part of the specification of the protocol. Note that z could also be selected using shared randomness, but this is not necessary.

Let $T \subseteq [n]$ be a set of possible outputs that contains the sets A_x and B_y for every $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$. Let $h : T \to S$ be a random hash function such that for each $i \in T$, $h(i)$ is selected independently and uniformly at random from S. The players select a common hash function h using shared randomness. On input x, Alice outputs a uniformly random element from $h^{-1}(z) \cap A_x$, and she outputs an arbitrary element if this set is empty. On input y, Bob outputs a uniformly random element of $h^{-1}(z) \cap B_y$. If this set is empty, he outputs an arbitrary element.

Recall that in the hashing framework described in Section 3.2, two conditions regarding the sizes of the intersection of $h^{-1}(z)$ and the sets A_x and B_y were to be ensured. In this setting, we have the following two claims that help in this regard.

Claim 5.2.4. *For any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$,*

$$\Pr[h^{-1}(z) \cap A_x \cap B_y = \emptyset] \leq \frac{1}{e^2}$$

where the probability is over the choice of the hash function h.

Proof. Notice that our setting implies that for any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, $|A_x \cap B_y| \geq \frac{L}{t} \geq 2|S|$. Thus,

$$\Pr[h^{-1}(z) \cap A_x \cap B_y = \emptyset] = \left(1 - \frac{1}{|S|}\right)^{|A_x \cap B_y|} \leq \left(1 - \frac{1}{|S|}\right)^{2|S|} \leq \frac{1}{e^2} .$$

□

Claim 5.2.5. *For any $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$,*

$$\Pr[|h^{-1}(z) \cap A_x| > 3t] \leq \epsilon$$
$$\Pr[|h^{-1}(z) \cap B_y| > 3t] \leq \epsilon,$$

where $\epsilon = e^{-0.1t}$.

Proof. Notice that the expected size (over the choice of the hash function h) of the intersection of the pre-image of z with the set A_x is

$$\mathbb{E}[|h^{-1}(z) \cap A_x|] = \frac{|A_x|}{|S|} \leq 2.1t .$$

The claim follows by using the following form of the Chernoff bound [78]:

$$\Pr[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2 \mu}{2+\delta}}$$

where X is a sum of independent random variables with values from $\{0, 1\}$ and $\mu = \mathbb{E}[X]$. An analogous proof works for the claim with respect to B_y. □

Using the above two claims, we obtain that the following conditions hold with probability at least $1 - e^{-2} - 2e^{-0.1t} > \frac{1}{2}$:
(i) $h^{-1}(z) \cap A_x \cap B_y \neq \emptyset$ and
(ii) $h^{-1}(z) \cap A_x$ and $h^{-1}(z) \cap B_y$ are both nonempty and have size at most $3t$.
If $i^* \in h^{-1}(z) \cap A_x \cap B_y$, i^* is a correct output and the probability that both Alice and Bob select i^* as their output is at least $\frac{1}{9t^2}$. Thus on any input $x \in f^{-1}(0)$ and $y \in f^{-1}(1)$, the players output a correct answer with probability at least $\frac{1}{18t^2}$.

□

Before we see how the hashing lemma helps prove Theorem 5.2.2, we define the following notation. The Hamming Sphere of radius r centred at a k-bit string a, denoted as $\mathcal{S}_a(r)$, contains all strings $z \in \{0,1\}^k$ that are at distance exactly r from a. Similarly the Hamming Ball of radius r centred at a, denoted as $\mathcal{B}_a(r)$, contains all strings $z \in \{0,1\}^k$ such that $|a - z| \leq r$. For the APInd_k function, a valid input has the function value in all the positions in the table indexed by strings in $\mathcal{B}_a\left(\frac{k}{2} - \sqrt{k \log k}\right)$ where a is the address part.

Proof of Theorem 5.2.2. Let us suppose that Alice has an input $(a, x) \in f^{-1}(0)$ and Bob has $(b, y) \in f^{-1}(1)$. We consider two different strategies for different kinds of inputs: the first for when the Hamming distance between the address parts a and b of

the inputs is large, i.e. $|a-b| \geq k/\log k$ and the second when the distance is smaller. For the first case, Alice and Bob use public randomness to sample an index $i \in [k]$ and this bit differentiates a from b with probability $\geq 1/\log k$. In the other case, we first show that $\Omega(1/\sqrt{\log k})$ fraction of the Hamming Ball $\mathcal{B}_a\left(\frac{k}{2} - \sqrt{k \log k}\right)$ around a intersects that around b. We then use the hashing lemma (Lemma 5.2.3) for Alice and Bob to pick an index in the intersection with probability $\Omega(1/\log k)$. These strategies are described in detail below.

Public coin strategy for Apind: We will consider two separate strategies for Alice and Bob to win the public coin Certificate Game with probability $\Omega(\frac{1}{\log k})$. They choose to play either of the strategies with probability 1/2.

- **Strategy 1:**

  > Alice and Bob sample a random element $z \in [k]$ using public coins and output the element z.

  This strategy works for inputs that have large Hamming distance between the address parts a and b, i.e. $|a-b| \geq \frac{k}{\log k}$. The probability that this strategy succeeds, $\Pr[a_z \neq b_z] \geq \frac{1}{\log k}$.

- **Strategy 2:** We use the strategy described in Lemma 5.2.3 where $A_{(a,x)}$ and $B_{(b,y)}$ are Hamming Balls of radius $\frac{k}{2} - \sqrt{k \log k}$ centred at a and b respectively. Let S be a set of size $\left\lfloor \frac{|A_x|}{2\sqrt{\log k}} \right\rfloor$.

  > - Alice and Bob agree on a $z \in S$ in advance.
  > - They sample a random hash function $h : \{0,1\}^k \to S$ using public randomness.
  > - Alice outputs a uniformly random element from $h^{-1}(z) \cap A_{(a,x)}$ (if this set is empty, she outputs an arbitrary element). Similarly, Bob outputs a uniformly random element of $h^{-1}(z) \cap B_{(b,y)}$ and if empty, an arbitrary element.

  The proof that this strategy works for inputs where the Hamming distance between the address parts a and b is small, i.e. $|a-b| \leq \frac{k}{\log k}$ essentially relies on the following lemma.

Lemma 5.2.6. *(Intersection Lemma): For two $k-$bit strings a and b at Hamming distance $\frac{k}{\log k}$, a Hamming sphere of radius r centred at a has $\frac{c}{\sqrt{\log k}}$ fraction of it lying in the Hamming ball of the same radius centred at b*

$$\frac{|\mathcal{S}_a(r) \cap \mathcal{B}_b(r)|}{|\mathcal{S}_a(r)|} \geq \frac{c}{\sqrt{\log k}}$$

where $\frac{k}{2} - 100\sqrt{k \log k} \leq r \leq \frac{k}{2} - \sqrt{k \log k}$ and c is a constant.

The proof of Lemma 5.2.6 will be presented later in Section 5.2.1. The basic outline of the proof is as follows: the fraction $\frac{|\mathcal{S}_a(r) \cap \mathcal{B}_b(r)|}{|\mathcal{S}_a(r)|}$ is at least the sum of probabilities from a hypergeometric distribution $P_{k,m,r}$ from $\frac{m}{2}$ to m where $m = \frac{k}{\log k}$

is the distance between the Hamming Ball and the Sphere. We show in Lemma 5.2.10 that the hypergeometric distribution $P_{k,m,r}$ is symmetric about $\frac{m}{2}$ for a range up to $200\sqrt{m}$. The expected value E of $P_{k,m,r}$ for our choice of m and r lies between $\frac{m}{2}-100\sqrt{m}$ and $\frac{m}{2}-\sqrt{m}$. We have a concentration bound for $P_{k,m,r}$ by Hoeffding [54] stated in Lemma 5.2.9 that the sum of the probabilities around the expected value of width $\sqrt{r}$ is at least 0.7. Hypergeometric distributions have the property that they are monotone increasing up to the expected value E and monotone decreasing beyond it (Lemma 5.2.11). Using this we derive a concentration bound of width $\sqrt{m}$ around E that the probabilities in this range sum to at least $0.7 \times \frac{\sqrt{m}}{\sqrt{r}}$, which for our choice of m and r is at least $\frac{1}{\sqrt{\log k}}$. This gives $\frac{|\mathcal{S}_a(r)\cap\mathcal{B}_b(r)|}{|\mathcal{S}_a(r)|} \geq \frac{c'}{\sqrt{\log k}}$ for a constant c'.

Since we can show most of the weight of the Hamming ball is concentrated on outer layers (proof of which is given in the Lemma 5.2.12) and since the size of the intersection of the Hamming Balls increases as the distance between them decreases, we easily get the following corollary from Lemma 5.2.6.

Corollary 5.2.7. *For two $k-$bit strings a and b at Hamming distance at most $\frac{k}{\log k}$, the ratio of $k-$bit strings in the intersection between the Hamming balls of radius $\frac{k}{2}-\sqrt{k\log k}$ centred at a and b to the total size of each Hamming Ball is at least $\frac{c_1}{\sqrt{\log k}}$, i.e.*

$$\frac{|\mathcal{B}_a\left(\frac{k}{2}-\sqrt{k\log k}\right)\cap\mathcal{B}_b\left(\frac{k}{2}-\sqrt{k\log k}\right)|}{|\mathcal{B}_a\left(\frac{k}{2}-\sqrt{k\log k}\right)|} \geq \frac{c_1}{\sqrt{\log k}}$$

where c_1 is a constant.

Using the hashing-based framework described in Lemma 5.2.3 with $A_x = \mathcal{B}_a\left(\frac{k}{2}-\sqrt{k\log k}\right)$ and $B_y = \mathcal{B}_b\left(\frac{k}{2}-\sqrt{k\log k}\right)$, we get $\mathsf{CG}^{\text{pub}}(\mathsf{ApInd}) = O(\log k)$ as $t = \sqrt{\log k}/c$ where c is a constant. □

5.2.1 Proof of the Intersection Lemma (Lemma 5.2.6)

The Hamming sphere $\mathcal{S}_a(r)$ centred at the $k-$bit string a of radius r contains $\binom{k}{r}$ $k-$bit strings, i.e. $|\mathcal{S}_a(r)| = \binom{k}{r}$.

Suppose we denote the Hamming distance between a and b as m. For our purposes, we choose $m = \frac{k}{\log k}$. A $k-$bit string z at a distance r from a lies in $\mathcal{B}_b(r)$ if on the m indices that a differs from b, z is closer to b than a. The number of $k-$bit strings at a distance r from a that lie in $\mathcal{B}_b(r)$,

$$|\mathcal{S}_a(r)\cap\mathcal{B}_b(r)| = \left|\left\{z\in\{0,1\}^k \mid |a-z| = r \wedge |b-z| \leq r\right\}\right| \geq \sum_{j=m/2}^{m}\binom{m}{j}\binom{k-m}{r-j}.$$

The hypergeometric distribution on parameters k, m and r, for $0 \leq j \leq m$ is given by,

$$P_{k,m,r}(j) = \frac{\binom{m}{j}\binom{k-m}{r-j}}{\binom{k}{r}}.$$

Proposition 5.2.8. *The fraction of the size of the intersection to the size of the Hamming Ball can be expressed as a sum of probabilities from a hypergeometric distribution,*

$$\frac{|\mathcal{S}_a(r)\cap\mathcal{B}_b(r)|}{|\mathcal{S}_a(r)|} \geq \sum_{j=m/2}^{m} P_{k,m,r}(j)$$

The proof of the Intersection Lemma 5.2.6 relies on the following three lemmas about hypergeometric distributions.

Lemma 5.2.9. *(Concentration Lemma)[54]: For a hypergeometric distribution P with parameters k, m and r,*

$$\sum_{i=0}^{E-\sqrt{r}} P_{k,m,r}(i) \leq e^{-2}$$
$$\sum_{i=E+\sqrt{r}}^{r} P_{k,m,r}(i) \leq e^{-2}$$

where $E = \frac{mr}{k}$ is the expected value of the distribution P.

Lemma 5.2.10. *(Symmetric Property): For the hypergeometric distribution with parameters $m = \frac{k}{\log k}$ and $k/2 - c\sqrt{k \log k} \leq r \leq k/2 - \sqrt{k \log k}$*

$$\frac{P_{k,m,r}(m/2+j)}{P_{k,m,r}(m/2-j)} \geq c'$$

where $0 \leq j \leq 2c\sqrt{m}$ and c, c' are constants.

Proof. From the definition

$$\begin{aligned}\frac{P_{k,m,r}(m/2+j)}{P_{k,m,r}(m/2-j)} &= \frac{\binom{m}{m/2+j}\binom{k-m}{r-m/2-j}}{\binom{m}{m/2-j}\binom{k-m}{r-m/2+j}} \\ &= \frac{(r-m/2-j+1)\cdots(r-m/2+j)}{(k-m/2-r-j+1)\cdots(k-m/2-r+j)} \\ &\geq \left(\frac{r-m/2-j}{k-m/2-r+j}\right)^{2j} = \left(1-\frac{k-2r+2j}{k-m/2-r+j}\right)^{2j}\end{aligned}$$

where in the last line we have approximated all the terms in the numerator by a factor smaller than the smallest factor and in the denominator by the largest factor. On substituting the values for m and r, we have

$$\begin{aligned}\frac{P_{k,m,r}(m/2+j)}{P_{k,m,r}(m/2-j)} &\geq \left(1-\frac{1}{2j}\left(\frac{2j\left(2c\sqrt{k\log k}+2j\right)}{k/2-\frac{k}{2\log k}+\sqrt{k\log k}+j}\right)\right)^{2j} \\ &\approx e^{-\left(\frac{2j\left(2c\sqrt{k\log k}+2j\right)}{k/2-\frac{k}{2\log k}+\sqrt{k\log k}+j}\right)} \geq e^{-16c^2}.\end{aligned}$$

We get the last inequality after replacing j by the largest possible value that we consider, i.e. $2c\sqrt{m}$ and we get $c' \approx e^{-16c^2}$. □

Lemma 5.2.11. *(Monotonicity Property): For the hypergeometric distribution where k is large and $m = \frac{k}{\log k}$ and $k/2 - c\sqrt{k \log k} \leq r \leq k/2 - \sqrt{k \log k}$, $P_{k,m,r}(j+1) \geq P_{k,m,r}(j)$ for $j \leq E - 1/2$ and $P_{k,m,r}(j+1) \leq P_{k,m,r}(j)$ otherwise. Here, $E = \frac{mr}{k}$ is the expected value of the distribution P.*

Proof. From the definition of hypergeometric distribution, we have

$$\frac{P_{k,m,r}(j+1)}{P_{k,m,r}(j)} = \frac{\binom{m}{j+1}\binom{k-m}{r-j-1}}{\binom{m}{j}\binom{k-m}{r-j}} = \frac{(m-j)(r-j)}{(j+1)(k-m-r+j+1)}.$$

If $P_{k,m,r}(j+1) \geq P_{k,m,r}(j)$, we have $\frac{(m-j)(r-j)}{(j+1)(k-m-r+j+1)} \geq 1$. On simplifying this expression, we get $j \leq \frac{mr+m-k+r-1}{(k+2)}$. Similarly, we have $P_{k,m,r}(j+1) \leq P_{k,m,r}(j)$ when $j \geq \frac{mr+m-k+r-1}{(k+2)}$. When k is large, $k+2 \approx k$ and $\frac{mr+m-k+r-1}{(k+2)} \approx E - (1 - \frac{m+r}{k})$. On substituting for m and r, we get $\frac{m+r}{k} \approx 1/2 + \epsilon$ where $\epsilon \ll 0$. Thus we can conclude that when k is large enough, $P_{k,m,r}(j+1) \geq P_{k,m,r}(j)$ when $j \leq E - 1/2$ and $P_{k,m,r}(j+1) \leq P_{k,m,r}(j)$ otherwise. □

We can now prove the main result of this section.

Proof of Lemma 5.2.6. To prove this lemma, from Proposition 5.2.8 it is enough to show that

$$\sum_{j=m/2}^{m} P_{k,m,r}(j) \geq \frac{c'}{\sqrt{\log k}}$$

when $m = \frac{k}{\log k}$ and $k/2 - c\sqrt{k \log k} \leq r \leq k/2 - \sqrt{k \log k}$. From the monotonicity property in Lemma 5.2.11, we have

$$\sum_{j=E-\sqrt{m}}^{j=E+\sqrt{m}} P_{k,m,r}(j) \geq \frac{\sqrt{m}}{\sqrt{r}} \sum_{j=E-\sqrt{r}}^{j=E+\sqrt{r}} P_{k,m,r}(j) > \sqrt{\frac{2}{\log k}} \sum_{j=E-\sqrt{r}}^{j=E+\sqrt{r}} P_{k,m,r}(j).$$

From Lemma 5.2.9, we have

$$\sum_{j=E-\sqrt{r}}^{j=E+\sqrt{r}} P_{k,m,r}(j) \geq 0.72\,.$$

This gives,

$$\sum_{j=E-\sqrt{m}}^{j=E+\sqrt{m}} P_{k,m,r}(j) > \sqrt{\frac{2}{\log k}} \times 0.72 > \frac{1}{\sqrt{\log k}}.$$

For our choice of m and r, we have the expected value $m/2 - c\sqrt{m} \leq E \leq m/2 - \sqrt{m}$. Using Lemma 5.2.10, by the symmetric property of the hypergeometric distribution for our choice of m and r, on reflecting about $m/2$ we have

$$\sum_{j=m/2}^{m} P_{k,m,r}(j) \geq c' \sum_{j=E-\sqrt{m}}^{j=E+\sqrt{m}} P_{k,m,r}(j) \geq \frac{c'}{\sqrt{\log k}}.$$

where $c' \approx e^{-16c^2}$. □

We will now show that a constant fraction of the strings in the Hamming Ball lie on the outer surfaces.

5.2.2 Weight of a Hamming ball is concentrated on its outer surfaces

Lemma 5.2.12. *For a Hamming Ball of radius $r = k/2 - \sqrt{k \log k}$, the weight contributed by Hamming Spheres of radius $\leq k/2 - 100\sqrt{k \log k}$ is small.*

$$\frac{\sum_{i=0}^{\frac{k}{2}-100\sqrt{k\log k}} |\mathcal{S}_a(i)|}{|\mathcal{B}_a\left(\frac{k}{2} - \sqrt{k \log k}\right)|} \leq c_1$$

where c_1 is a constant.

Proof. We would like to show

$$\frac{\sum_{j=0}^{\frac{k}{2}-100\sqrt{k\log k}} \binom{k}{j}}{\sum_{j=0}^{\frac{k}{2}-\sqrt{k\log k}} \binom{k}{j}} \leq c_1.$$

We use the following form of Chernoff bound [78],

$$\Pr[X \leq (1-\delta)\mu] \leq e^{\frac{\delta^2 \mu}{2}}$$

for $0 \leq \delta \leq 1$ and apply it to the binomial distribution with $p = 1/2$ to get $\sum_{j=0}^{\frac{k}{2}-100\sqrt{k\log k}} \binom{k}{j} \leq 2^k k^{-10^4}$. We now use the following lower bound for the tail of the binomial distribution when $p = 1/2$ (which is stated in a slightly different form from the original form [73]).

$$\Pr[X \leq k/2 - \delta] \geq \frac{1}{15} e^{-16\delta^2/k}$$

for $\delta \geq 3k/8$. This gives $\sum_{j=0}^{\frac{k}{2}-\sqrt{k\log k}} \binom{k}{j} \geq 2^k \frac{1}{15} k^{-16}$. Thus we have

$$\frac{\sum_{j=0}^{\frac{k}{2}-100\sqrt{k\log k}} \binom{k}{j}}{\sum_{j=0}^{\frac{k}{2}-\sqrt{k\log k}} \binom{k}{j}} \leq \frac{15k^{-10^4}}{k^{-16}} \ll c_1 .$$

□

Although we have proven an upper bound on $\mathsf{CG}^{\mathrm{pub}}(\mathsf{ApInd})$, a lower bound has not been shown and we leave it as an open problem.

Open Problem 6 : Give a lower bound on $\mathsf{CG}^{\mathrm{pub}}(\mathsf{ApInd})$.

Chapter 6

Sensitivity Conjecture

In the field of Boolean function analysis, the sensitivity conjecture was one of the most elusive ones as it remained unsolved for about three decades since its appearance both in the complexity theory and graph theory literature in the late 1980s. In complexity theory, it was formally introduced by Nisan and Szegedy [81], where they sought to address if block sensitivity and sensitivity are polynomially related to each other. In an alternate form, in graph theory, Chung, Füredi, Graham and Seymour [37] sought for a lower bound on the largest degree of a large enough induced subgraph of a hypercube. These two seemingly unrelated problems were shown to be equivalent by Gotsman and Linial [50] in 1992. Partial progress was made in the next two decades [60, 80, 92, 23, 89, 95, 17, 75, 34], until the conjecture was finally resolved by Hao Huang in 2019 using a very short proof that crucially assigned $+/-$ signs to the edges of a hypercube [56]. In this chapter we revisit the journey of how the sensitivity conjecture came into being, its multiple formulations and their equivalence, and ultimately its proof. We also provide an alternate proof in Section 6.4, which naturally leads to many interesting questions and served as motivation for many of the questions addressed in this thesis.

6.1 First steps

The notion of sensitivity was first introduced by Cook and Dwork and independently by Reischuk [41, 87] to bound below the time taken to compute a function in a CREW (Concurrent Read Exclusive Write) PRAM (Parallel Random Access Machines) model. The time complexity in this model was later shown to be exactly characterised by the logarithm of block sensitivity in a paper by Noam Nisan [80]. By definition, we have that block sensitivity is larger than sensitivity,

$$\mathsf{bs}(f) \geq \mathsf{s}(f).$$

The largest separation found between these two measures was quadratic[1]. The natural question of whether block sensitivity is bounded above by some polynomial in sensitivity was then posed as the sensitivity conjecture. A major breakthrough in this direction was due to Nisan and Szegedy where they showed an upper bound on block sensitivity in terms of polynomial degree [81].

Theorem 6.1.1. *[81, 95] For a Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$, the block sensitivity of f is at most quadratic in its polynomial degree, i.e.*

$$\mathsf{bs}(f) \leq \deg(f)^2.$$

[1] This separation was achieved by the Rubinstein function (see Definition 1.2.4) which is a function on n bits with sensitivity $\sqrt{n}$ and block sensitivity $n/2$.

The proof can be outlined as follows: In a series of operations that could only potentially decrease the degree, the polynomial representing the Boolean function f is modified to a symmetric polynomial on which we can use lower bounds on degree from approximation theory. At first, the polynomial representing the Boolean function f is modified to be defined over $b = \mathsf{bs}(f)$ variables in a way that the degree of the resulting polynomial is at most the original degree d. This modified polynomial is such that it evaluates to 1 on input x if $|x| = 1$. This polynomial is now *"symmetrised"* such that the degree of the resulting symmetric polynomial is at most d and it evaluates to 0 on input 0 and 1 on input 1. This allows one to use the following lower bound on degree of the polynomial from approximation theory from [88, 43] which is a generalisation of a theorem by A. A. Markov [72, 36].

Theorem 6.1.2. *(Rivlin and Cheney; Ehlich and Zeller) For a univariate polynomial $p : \mathbb{R} \mapsto \mathbb{R}$ such that for any integer $0 \leq i \leq b$, $a_1 \leq p(i) \leq a_2$ and for some real $0 \leq x \leq b$, the derivative of p at x has absolute value $|p'(x)| \geq c$, then the degree of the polynomial p*

$$\deg(p) \geq \sqrt{\frac{cb}{c + a_2 - a_1}}.$$

Let us now come to the proof of the Nisan-Szegedy Theorem 6.1.1. The presentation of this theorem follows the survey by Buhrman and de Wolf [31].

Proof of Theorem 6.1.1. Let $p : \mathbb{R}^n \mapsto \mathbb{R}$ be the polynomial representing the Boolean function f with degree d. Let b be the block sensitivity $\mathsf{bs}(f)$ with z being the input at which maximum block sensitivity is achieved using blocks $B_1, B_2, \ldots B_b$. Without loss of generality, let us assume that $f(z) = 0$. From the polynomial p, a new polynomial $q : \mathbb{R}^b \mapsto \mathbb{R}$ is constructed such that $q(0^b) = p(z)$ and q evaluated on e_i corresponds to $p(z^{B_i})$, where $0 \leq i \leq b$. Here e_i is the input with zeroes everywhere except at the i^{th} position and z^{B_i} is the input z with i^{th} block flipped. This can be done by replacing every x_j in $p(x_1, \cdots, x_n)$ as follows:

- If $j \in B_i$ for some block and at the j^{th} position of the most block sensitive input $z_j = 0$, replace x_j by y_i.
- If $j \in B_i$ and $z_j = 1$, replace x_j by $1 - y_i$.
- If j does not belong to any block, then replace x_j by z_j.

Roughly, each of these variables $y_1, \cdots, y_b$ tries to encapsulate an entire block of z and flipping each of them flips the corresponding block of z. This process could only decrease the degree of the polynomial, i.e. $\deg(q) \leq \deg(p)$ as we are replacing the variables in the polynomial by new variables that are fewer in number. We also have the following two properties:

- At the all-zero input, q takes the same value as $p(z)$, i.e. $q(0^b) = p(z) = 0$.
- At the input e_i, q evaluates to 1 i.e. $q(e_i) = 1$ for all $0 \leq i \leq b$. This holds as for any $x \in \{0,1\}^b, q(x) = p(\tilde{z})$ where $\tilde{z}$ is obtained from z on flipping blocks corresponding to positions in x that have ones in them.

Now this polynomial q is "symmetrised" as described by Minski and Papert [76].

Definition 6.1.3. *For a polynomial $q : \mathbb{R}^b \mapsto \mathbb{R}$, its symmetric polynomial q_{sym} is defined as*

$$q_{sym}(x_1, \cdots, x_b) = \frac{\sum_{\pi \in \mathcal{S}_b} q(x_{\pi(1)} \cdots x_{\pi(b)})}{b!}$$

where $\mathcal{S}_b$ is the permutation group which is the set of all $b!$ permutations.

The following lemma by Minski and Papert shows the existence of a unique univariate polynomial that represents q_{sym} [76].

Lemma 6.1.4. *For a multivariate polynomial $q : \mathbb{R}^b \mapsto \mathbb{R}$, there exists a unique univariate polynomial $\tilde{q} : \mathbb{R} \mapsto \mathbb{R}$ such that for all $x \in \{0,1\}^b$, $q_{sym}(x) = \tilde{q}(|x|)$ and $\deg \tilde{q} \leq \deg(q)$.*

The proof of this lemma follows from the fact that q_{sym} is a symmetric polynomial and hence can be expressed as a linear combination of elementary symmetric polynomials (where the k^{th} elementary symmetric polynomial V_k is the sum of all $\binom{b}{k}$ products of k variables) by the fundamental theorem of symmetric polynomials. Using Lemma 6.1.4, we have that $\tilde{q}(0) = q_{\text{sym}}(0) = 0$ and $\tilde{q}(1) = q_{\text{sym}}(1) = 1$. We can now use Theorem 6.1.2 on $\tilde{q}$ as $0 \leq \tilde{q}(i) \leq 1$ for all integers $0 \leq i \leq b$ and since $\tilde{q}(0)$ and $\tilde{q}(1) = 1$, we have that for some $x \in [0,1]$, the derivative $\tilde{q}'(x) \geq 1$. This gives us the required lower bound of $\deg(\tilde{q}) \geq \sqrt{\frac{b}{2}}$. □

This bound in its current form is due to Tal who showed using a proof by contradiction that the constant factor 2 was not needed [95]. After the upper bound on block sensitivity in terms of degree was shown in 1992, the focus shifted to finding an upper bound on degree (or any other complexity measure that had a polynomial upper bound on block sensitivity) in terms of some polynomial in sensitivity.

6.2 An incomplete puzzle: its formulations

This famous problem was reduced by Gotsman and Linial in 1992 to the graph theoretic question posed by Chung et al. in 1988. Numerous other formulations of the sensitivity conjecture also appeared over the years and a detailed survey can be found in the work by Hatami et al. [53].

Chung, Füredi, Graham and Seymour [37] posed the following question: how small can the maximum degree of an induced subgraph of a hypercube H_n with strictly more than 2^{n-1} vertices be? They showed an upper bound of $\sqrt{n}$ on this quantity and a lower bound of $\Omega(\log n)$. They also noted that the lower bound (and its proof) is similar to the one on sensitivity of non-degenerate functions on n bits, where a non-degenerate function is one that depends on all n bits.

Gotsman and Linial showed that this question is equivalent to that of an upper bound on degree using some polynomial in sensitivity by the following theorem [50].

Theorem 6.2.1. (Gotsman-Linial) *The following are equivalent for any monotone function $h : \mathbb{N} \mapsto \mathbb{R}$:*

1. *For any induced subgraph G of H_n such that $|V(G)| \neq 2^{n-1}$, there exists a vertex with degree $\geq h(n)$ in either G or $H_n - G$.*

2. *For any Boolean function f, $\mathsf{s}(f) \geq h(deg(f))$.*

Proof. Firstly we show how it is sufficient to consider functions with full polynomial degree for 2. Consider a monomial of the largest degree d in the multilinear polynomial representing the Boolean function and discard all variables that do not appear in this monomial (i.e. set them to zero). This modified polynomial has degree d and the number of variables is also d. Let the function representing this modified polynomial be f'. The sensitivity of this function f' might be smaller than that of the original function i.e. $\mathsf{s}(f') \leq \mathsf{s}(f)$, but the polynomial degree of the function is preserved and f' now has full polynomial degree. If we can show that for all functions f' with

full polynomial degree, $\mathsf{s}(f') \geq h(\deg(f'))$, this implies that for the original function $\mathsf{s}(f') \geq h(\deg(f'))$. Henceforth in this proof, we will only consider functions with full polynomial degree.

1 $\implies$ 2: Consider a function f with full polynomial degree d. We use the fact that functions with full polynomial degree are not parity-balanced. Without loss of generality, let us assume that $|f^{-1}(1) \cap U^{\text{odd}}| > |f^{-1}(1) \cap U^{\text{even}}|$. This implies that there are more than 2^{d-1} inputs in $(f^{-1}(1) \cap U^{\text{odd}}) \cup (f^{-1}(0) \cap U^{\text{even}})$.

Notice that if we consider the graph induced by the vertices in $(f^{-1}(1) \cap U^{\text{odd}}) \cup (f^{-1}(0) \cap U^{\text{even}})$ (and similarly for $(f^{-1}(1) \cap U^{\text{even}}) \cup (f^{-1}(0) \cap U^{\text{odd}})$), the graph degree of a vertex x in this subgraph is the sensitivity of x with respect to f. Thus if we have that any induced subgraph G of the hypercube with $|V(G)| \geq 2^{d-1}$ has maximum degree $\geq h(d)$ in either G or $H_d - G$, then the input corresponding to the maximum degree vertex has sensitivity $\geq h(d)$. Thus we have $\mathsf{s}(f) \geq h(d)$.

2 $\implies$ 1: An induced subgraph G of H_n such that $|V(G)| \geq 2^{n-1}$ can be mapped to a function f with full degree such that inputs corresponding to odd vertices in G have function value 1 and even vertices have function value 0 (and vice versa for $H_n - G$). Since we have assumed that $\mathsf{s}(f) \geq h(n)$ for any function with full degree n, it implies the existence of a high degree vertex in G or $H_n - G$ as sensitivity corresponds to degree in G or $H_n - G$.

□

6.3 Final Piece

In 2019, Hao Huang showed a surprisingly simple proof of the sensitivity conjecture using signed hypercubes and Cauchy's Interlace theorem [56].

Theorem 6.3.1. *[56] Any induced subgraph of the n-dimensional hypercube with more than 2^{n-1} vertices has at least one vertex of degree larger than or equal to $\sqrt{n}$.*

We now outline the original proof by Huang. One of the main objects used in the proof is a *signed adjacency matrix* defined as follows:

Definition 6.3.2. *A signed adjacency matrix $A_{G,\sigma}$ of an undirected graph G is the adjacency matrix of the graph G with a signature σ of $+$ or $-$ signs associated to its edges where*

$$\sigma(x,y) = \begin{cases} +/- & \text{if } (u,v) \in E(G) \\ 0 & \text{otherwise}, \end{cases}$$

and

$$A_{G,\sigma}(x,y) = \begin{cases} +1 & \text{if } \sigma(x,y) = + \\ -1 & \text{if } \sigma(x,y) = - \\ 0 & \text{otherwise}. \end{cases}$$

An important observation made by Huang was that the maximum degree of a graph (denoted as Δ) is at least the largest eigenvalue (denoted as λ_1) of its signed adjacency matrix. This was already known for adjacency matrices without signatures.

Lemma 6.3.3. *[56] For any undirected graph G with an associated signature σ, the maximum degree of G is at least the largest eigenvalue of its signed adjacency matrix.*

$$\Delta(G) \geq \lambda_1(A_{G,\sigma}).$$

Proof. If u is an eigenvector corresponding to the eigenvalue $\lambda = \lambda_1(A_{G,\sigma})$, we have $A_{G,\sigma}u = \lambda u$. Let u_i be the entry in u with the maximum absolute value.

$$|\lambda u_i| = |(A_{G,\sigma}u)_i| = \left| \sum_{j \in V(G)} A_{G,\sigma}(i,j)u_j \right| \leq \sum_{j:(i,j)\in E(G)} |A_{G,\sigma}(i,j)|\,|u_i|$$

This gives us that the degree of vertex i in G is at least λ. □

The other important ingredient in this proof is the construction of a signature for the hypercube H_n that has sufficient number of large eigenvalues. In particular, Huang showed the following.

Lemma 6.3.4. *[56, 6] The signed adjacency matrix B_n defined iteratively as,*

$$B_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, B_n = \begin{bmatrix} B_{n-1} & I \\ I & -B_{n-1} \end{bmatrix}$$

has eigenvalues $\sqrt{n}$ and $-\sqrt{n}$ with multiplicity 2^{n-1}.

We also note that this lemma is implicit in the proof of Theorem 6.7 and its Corollary 6.9 in [6]. The proof of this lemma will be shown in terms of a more general signature in Lemma 6.4.2. Cauchy's interlace theorem (stated below) along with the above lemmas completes the proof of the sensitivity conjecture.

Lemma 6.3.5 (Cauchy's Interlace Theorem)**.** *Let A be a symmetric $n \times n$ matrix and B be an $m \times m$ principal submatrix of A with $m \leq n$. If the eigenvalues of A are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and those of B are $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$, then for all $1 \leq i \leq m$,*

$$\lambda_i \geq \mu_i \geq \lambda_{i+n-m}.$$

Putting these together, we have that the maximum eigenvalue of an induced subgraph of signed H_n with more than 2^{n-1} vertices is at least $\sqrt{n}$, hence proving that the maximum degree is at least $\sqrt{n}$.

In 2020, Aaronson, Ben-David, Kothari, Rao and Tal introduced the complexity measure called spectral sensitivity $\lambda(f)$ (Definition 1.1.4) for a Boolean function f and showed a short and compact proof of $\deg(f) \leq \lambda(f)^2$ combining ideas from Huang's proof of Theorem 6.3.1 and the Gotsman-Linial proof of Theorem 6.2.1 [4]. This result of $\deg(f) \leq \lambda(f)^2$ and $\lambda(f) \leq \mathsf{s}(f)$ were implicit in Huang's work, but this work also went on to show how $\lambda(f)$ was equivalent to many other measures such as the Koutsoupias complexity [64] and the single bit versions of the adversary method. Using the λ measure, they were able to show a tight upper bound on deterministic query complexity in terms of the quantum variant i.e. $\mathsf{D}(f) \leq O(\mathsf{Q}(f)^4)$ and on spectral sensitivity in terms of the approximate degree $\lambda(f) \leq O(\widetilde{\mathsf{deg}}(f))$.

6.4 Alternate Proof of Huang's result

We will now present our alternate proof of Huang's result using linear dependencies of vectors assigned to the vertices of the hypercube [69]. A similar idea of using linear dependencies to show an alternate proof of Huang's result was shown by Knuth [63] based on a comment attributed to Shalev Ben-David [20].

First we lay some groundwork such as the definition of the vectors assigned to every vertex in the hypercube.

Given a hypercube H_n, let σ be an assignment of $+$ or $-$ to the edges such that every 4-cycle has an odd number of negative edges. We will refer to a signed hypercube with this property as a signed hypercube with negative 4-cycles. In this alternate proof of Lemma 6.3.4 we shall see this is the only property of Huang's signature that is needed to derive the degree lower bound. Signed hypercubes with this property have been studied in the literature before and in Chapter 8 we will look at the construction of such a signature and how to find all such signatures.

A crucial step in our approach is the following definition. For any vertex x of H_n, we define a real valued vector x^+ of length 2^n whose entries are labeled by vertices y of H_n. We use $x^+(y)$ to denote the value at the index y of a vector x^+.

Definition 6.4.1. *For all vertices x of H_n, we define the vectors x^+ and x^- as follows, for all $y \in V(H^n)$,*

$$x^+(y) = \begin{cases} \sqrt{n} & \text{if } x = y \\ +1 & \text{if } \sigma(x,y) = + \\ -1 & \text{if } \sigma(x,y) = -\,, \end{cases} \qquad x^-(y) = \begin{cases} -\sqrt{n} & \text{if } x = y \\ +1 & \text{if } \sigma(x,y) = + \\ -1 & \text{if } \sigma(x,y) = -\,. \end{cases}$$

Note that each of the vectors x^+ or x^- is non-zero only at the coordinates corresponding to x and its neighbours in the hypercube. We use V^+ and V^- to denote the subspace generated by the vectors x^+ and x^-, i.e. $V^+ = \langle x_1^+, x_2^+, \ldots, x_{2^n}^+ \rangle$ and $V^- = \langle x_1^-, x_2^-, \ldots, x_{2^n}^- \rangle$.

We make the following observation that V^+ and V^- are the eigenspaces of the signed adjacency matrix A with negative 4-cycles.

Lemma 6.4.2. *The subspaces V^+ and V^- are the eigenspaces of the signed adjacency matrix A, with signature σ such that every 4-cycle is negative, with eigenvalues $+\sqrt{n}$ and $-\sqrt{n}$ respectively, i.e. for any $x \in V(H_n)$*

$$Ax^+ = \sqrt{n}x^+$$
$$Ax^- = -\sqrt{n}x^-.$$

Proof. Let us show this for the subspace V^+. Since the entries of the vector x^+ are non-zero only at the coordinates corresponding to x and its neighbours we have the following,

$$\left(Ax^+\right)(y) = \sum_{z \in \{0,1\}^n} A(y,z)x^+(z) = \sum_{i \in [n]} \sigma(y,x^i)\sigma(x,x^i) + \sigma(y,x)\sqrt{n}.$$

Depending on the distance between x and y, we consider the following 4 cases:

Case 1. If $y = x$, then the only non-zero contributions come from the neighbours of x since $\sigma(x,x) = 0$.

$$\left(Ax^+\right)(x) = \sum_{i \in [n]} \sigma(x,x^i)\sigma(x,x^i) + 0 = n = \sqrt{n}x^+(x).$$

Case 2. If y is a neighbour of x such that $y = x^k$, then the only non-zero contribution comes from x and y (since x and y do not share any common neighbours).

$$\left(Ax^+\right)(x^k) = \sum_{i \in [n]} \sigma(x^k,x^i)\sigma(x,x^i) + \sigma(x^k,x)\sqrt{n} = 0 + \sqrt{n}\sigma(x^k,x) = \sqrt{n}x^+(x^k).$$

Case 3. If y is at a distance 2 from x, i.e. $y = x^{\{k,j\}}$, observe that there are exactly two vertices (x^k and x^j) adjacent to both x and y. These form a 4-cycle and by the assumption on the signature σ, $\sigma(x^{\{k,j\}}, x^k)\sigma(x, x^k) = -\sigma(x^{\{k,j\}}, x^j)\sigma(x, x^j)$. Therefore,

$$\begin{aligned}\left(Ax^+\right)(x^{\{k,j\}}) &= \sum_{i\in[n]} \sigma(x^{\{k,j\}}, x^i)\sigma(x, x^i) + \sigma(x^{\{k,j\}}, x)\sqrt{n} \\ &= \sigma(x^{\{k,j\}}, x^k)\sigma(x, x^k) + \sigma(x^{\{k,j\}}, x^j)\sigma(x, x^j) + 0 \\ &= 0 = \sqrt{n}x^+(x^{\{k,j\}}).\end{aligned}$$

Case 4. If y is at a distance greater than 2 from x, $(Ax^+)(y) = \sqrt{n}x^+(y)$ is trivially satisfied as there are no common non-zero terms in the vectors.

The argument for V^- follows the same steps as above with a change of sign. □

We have the following corollary from the above lemma by taking into consideration that for every vertex x in the hypercube, the vector x^+ is the column corresponding to x in $A + \sqrt{n}I$.

Corollary 6.4.3. *The adjacency matrix A of a signed hypercube with negative 4-cycles satisfies $A^2 = nI$.*

Proof. From Lemma 6.4.2 and the fact that vectors x^+ are the columns corresponding to x in $A + \sqrt{n}I$, we have $A(A + \sqrt{n}I) = \sqrt{n}(A + \sqrt{n}I)$. This gives us the required property that $A^2 = nI$. □

It is easy to see that the following proposition also holds.

Proposition 6.4.4. *The subspaces V^+ and V^- are orthogonal to each other.*

Proof. From Corollary 6.4.3, we have that for the signed hypercube with negative 4-cycles, $A^2 = nI$. This gives us that $(A + \sqrt{n}I)(A - \sqrt{n}I) = 0$. Since the columns and rows of $A+\sqrt{n}I$ correspond to the spanning set of V^+ and those of $A-\sqrt{n}I$ correspond to the spanning set of V^-, the two subspaces are orthogonal to each other. □

We are interested in studying linear dependencies among the set of vectors assigned to the vertices. A set of vectors, $S = \{v_1, v_2, \ldots, v_k\}$, is said to have a *linear dependency* if we have $\sum_i a_i v_i = 0$ for some choice of real numbers a_i not all of which are zero, and where 0 is the all-zero vector.

Theorem 6.3.1 is an immediate corollary of the following facts:

Observation 6.4.5. *If the vectors $\{x_1^+, x_2^+, \ldots, x_k^+\}$ have a linear dependency, then the subgraph induced on the corresponding vertices $\{x_1, x_2, \ldots, x_k\}$ of H_n has a vertex of degree at least $\sqrt{n}$.*

Proof. Suppose $\sum a_i x_i^+ = 0$ and let $|a_j|$ be a largest coefficient among all non-zero $|a_i|$'s. For the row corresponding to vertex x_j to vanish when viewing x_i^+ as a column vector, there must be at least $\lceil \sqrt{n} \rceil$ other vectors in the linear dependency that are nonzero at the coordinate x_j. Since those vectors can only correspond to neighbours of x_j each of which can contribute at most $|a_j|$ to the sum, x_j must have at least $\sqrt{n}$ neighbours. □

Proposition 6.4.6. *The subspaces V^+ and V^- are each of dimension 2^{n-1}.*

To prove this proposition we use the following observation about sets of vectors x^+ (or x^-) that correspond to independent sets.

Observation 6.4.7. *For any independent set I of vertices of H_n, the sets $\{x^+|x \in I\}$ and $\{x^-|x \in I\}$ are linearly independent.*

Proof. Since I is an independent set and the vector x^+ (resp. x^-) is non-zero only at the coordinate x and its neighbours in the hypercube, x^+ (resp. x^-) is the only vector in $\{y^+|y \in I,\ y \neq x\}$ (resp. in $\{y^-|y \in I,\ y \neq x\}$) which is nonzero at the coordinate x. □

We can now prove Proposition 6.4.6 using the above observation as follows:

Proof of Proposition 6.4.6. Since the set of odd vertices (similarly even vertices) forms an independent set of size 2^{n-1} in H_n, the dimension of vector spaces V^+ and V^- are at least 2^{n-1}, i.e. $dim(V^+) \geq 2^{n-1}$, $dim(V^-) \geq 2^{n-1}$ using Observation 6.4.7. Since the subspaces V^+ and V^- are orthogonal to each other by Proposition 6.4.4, we see that equality holds. □

As an immediate corollary we have that for any set of $2^{n-1}+1$ vertices, there must be a linear dependency among some of the corresponding vectors which by Observation 6.4.5 implies the existence of a vertex of degree at least $\sqrt{n}$. This concludes the proof of Theorem 6.3.1.

Chapter 7

Structural information from linear dependencies

Huang's proof of the sensitivity conjecture yields a lower bound on the graph degree when the number of vertices is large enough. We can strengthen this result in two ways. Firstly, we can weaken the hypothesis to any graph presenting a linear dependency regardless of the number of vertices in the linear dependency (as shown in Observation 6.4.5). Secondly, we can exploit the linear dependency further to extract more structural information about the graph other than its largest degree.

In this chapter, we will see some of the structural relations between vertices that are at distance 2 (Theorem 7.1.1) and distance 3 (Theorem 7.2.2) in the hypercube. We will also look at their implications on the degree-sensitivity relation of a function (Corollary 7.1.4).

We introduce the following terminology before going into our results. We work with the vectors v^+ defined in Definition 6.4.1 for vertices $v \in U^{\text{even}} \cup U^{\text{odd}}$, where U^{odd} (U^{even}) denotes the vertices of the hypercube with an odd (even) Hamming weight. A subset F of the vertices of H_n is said to be *linearly dependent* if we have

$$\sum_{u \in F \cap U^{\text{odd}}} a_u u^+ = \sum_{v \in F \cap U^{\text{even}}} b_v v^+$$

where $a_u \neq 0$ for every $u \in F \cap U^{\text{odd}}$ and $b_v \neq 0$ for every $v \in F \cap U^{\text{even}}$.

Let $H_n[F]$ denote the subgraph of H_n induced by a set of vertices F. Let $N^F(x)$ be the set of all neighbours of a vertex x in the induced subgraph $H_n[F]$. The size of this set, which is the degree of x in this subgraph, is denoted by $d_F(x)$. Recall that in a hypercube, there exist two paths of length 2 between two vertices that are at Hamming distance 2 from each other. We take $N_2^F(x)$ to be the set of vertices $y \in F$ that are at a Hamming distance 2 from x such that there is a unique path of length 2 in the subgraph $H_n[F]$ from y to x. Let $N_3^F(x)$ be the set of vertices in F that are at a Hamming distance 3 from x.

We start by looking at the properties implied on the structure of induced subgraphs at distance at most 2 by linear dependencies.

7.1 Structural relations at distance 2

The subspace V^+, which is spanned by vectors x^+ corresponding to vertices x of the Boolean hypercube H_n, has dimension 2^{n-1} by Proposition 6.4.6. This implies that for any set $K \subseteq V(H_n)$ with $|K| \geq 2^n + 1$, there exists a set of linearly dependent vertices F such that $F \subseteq K$. We have the following theorem that (roughly) guarantees the existence of an odd vertex (and an even vertex) that has a large number of neighbours at distance 1 or 2 from it in the subgraph $H_n[F]$.

Theorem 7.1.1. *Given a linearly dependent set F of vertices in H_n, there exist vertices $u \in U^{odd}$ and $v \in U^{even}$ in $H_n[F]$ such that $|N^F(u)| + |N_2^F(u)| \geq n$ and $|N^F(v)| + |N_2^F(v)| \geq n$.*

Before going into the proof of this theorem, let us look at the following corollary. This corollary is observed by taking a vertex u given by the theorem and considering its neighbour which has the largest degree in $H_n[F]$.

Corollary 7.1.2. *If F is a set of linearly dependent vertices of H_n, there exists an edge (u, v) in $H_n[F]$ such that $d_F(u) \times d_F(v) \geq n$.*

The key tool in the proof of Theorem 7.1.1 is Lemma 7.1.3 stated below. For any pair of vertices x and y at Hamming distance 2 in H_n, there are exactly two paths of length 2 connecting them. When there is a unique 2-path connecting x and y in $H_n[F]$, we extend the signature σ to $\hat{\sigma}$ such that $\hat{\sigma}_F(x, y) = \sigma(x, z)\sigma(z, y)$ where z is the unique common neighbour of x and y in $H_n[F]$.

Lemma 7.1.3. *Given a linearly dependent set F of vertices in H_n such that*

$$\sum_{u \in F \cap U^{odd}} a_u u^+ = \sum_{v \in F \cap U^{even}} b_v v^+,$$

for every vertex $x \in F \cap U^{odd}$ and $y \in F \cap U^{even}$ we have

$$(n - d_F(x))a_x = \sum_{z \in N_2^F(x)} \hat{\sigma}_F(x, z)a_z, \quad \text{and} \quad (n - d_F(y))b_y = \sum_{t \in N_2^F(y)} \hat{\sigma}_F(t, y)b_t.$$

Proof of Lemma 7.1.3. We look at the vectors u^+ and v^+ as column vectors. In the linear dependency, by considering the row corresponding to a vertex $x \in F \cap U^{\text{odd}}$ we get

$$\sqrt{n}a_x = \sum_{v \in N^F(x)} \sigma(x, v)b_v. \tag{7.1}$$

Similarly, by considering the row corresponding to a vertex $v \in F \cap U^{\text{even}}$ we get

$$\sqrt{n}b_v = \sum_{u \in N^F(v)} \sigma(u, v)a_u. \tag{7.2}$$

Multiplying both sides of equation (7.1) by $\sqrt{n}$ we get

$$na_x = \sum_{v \in N^F(x)} \sigma(x, v)\sqrt{n}b_v. \tag{7.3}$$

Substituting equation (7.2) in equation (7.3), we get

$$na_x = \sum_{v \in N^F(x)} \sum_{u \in N^F(v)} \sigma(x, v)\sigma(v, u)a_u. \tag{7.4}$$

On examining the right hand side of this identity we make two key observations. The first is that a_x appears for each v in its neighbourhood with a coefficient $\sigma(x, v)^2 = 1$. The second is that if a vertex $u \neq x$ appears on the right hand side twice, then the sum of its coefficients is 0. This is based on the main property of the signature we have chosen to work with, namely the product of signs of all the edges in a 4-cycle is negative, i.e. $\sigma(x, v_1)\sigma(v_1, u) = -\sigma(x, v_2)\sigma(v_2, u)$ where $x - v_1 - u - v_2$ is a 4-cycle

in $H_n[F]$. Upon rearranging and simplifying, we get

$$(n - d_F(x))a_x = \sum_{u \in N_2^F(x)} \hat{\sigma}_F(x,u)a_u \tag{7.5}$$

as claimed in Lemma 7.1.3. The proof of the identity for the vertices in $F \cap U^{\text{even}}$ is analogous. □

We now complete the proof of Theorem 7.1.1 which is about the second neighbourhood of vertices in $H_n[F]$.

Proof of Theorem 7.1.1. For a set F of linearly dependent vertices,

$$\sum_{u \in F \cap U^{\text{odd}}} a_u u^+ = \sum_{v \in F \cap U^{\text{even}}} b_v v^+$$

and let $|a_x| = \max_{z \in F \cap U^{\text{odd}}} \{|a_z|\}$. Note that $a_x \neq 0$ by the definition of a linearly dependent set F. From the identity (7.5), we see that there should be at least $n - d_F(x)$ values of a_z which are nonzero since $|a_z| \leq |a_x|$. An analogous argument follows for the even vertices by taking the maximum value over $|b_v|$. □

7.1.1 From linear dependency to sensitivity

We obtain a stronger upper bound on the degree of a function in terms of 0-sensitivity and 1-sensitivity from the structural relations at distance 2 [69]. Later, Aaronson et al. [4] gave an alternate proof of our result using Huang's theorem as a black-box.

Corollary 7.1.4. *For any Boolean function f, we have* $\deg(f) \leq \mathsf{s}_0(f)\mathsf{s}_1(f)$.

Proof. If the degree of a function f is d, we can work with a function f' of degree d on d variables by setting the variables outside of the largest monomial to 0 (as was done in the proof of the Gotsman-Linial Theorem 6.2.1). This can only decrease the sensitivity. As was shown in the proof of the Gotsman-Linial Theorem 6.2.1, we know that the sensitivity with respect to f of an input x is the graph degree of a vertex x in the bipartite subgraph induced by $T_1 = (f'^{-1}(0) \cap U^{\text{even}}) \cup (f'^{-1}(1) \cap U^{\text{odd}})$ or $T_2 = (f'^{-1}(1) \cap U^{\text{even}}) \cup (f'^{-1}(0) \cap U^{\text{odd}})$ depending on $f'(x)$ and the parity of $|x|$. Since f' has full polynomial degree, it is not parity-balanced. This implies that one of these subgraphs has at least $2^{d-1} + 1$ vertices. Since the dimension of V^+ is 2^{d-1}, any set of $2^{d-1} + 1$ vectors from V^+ has a linear dependency and the larger of the two induced subgraphs, say T_1 without loss of generality, has a linearly dependent set of vertices F. Recall Observation 6.4.5 which states that any linearly dependent set F of vertices implies the existence of a vertex with graph degree at least $\sqrt{d}$ in $H_n[F]$. Therefore, there exists some vertex in the subgraph induced on T_1 that has graph degree at least $\sqrt{d}$. This proves that the sensitivity of the function is at least $\sqrt{d} = \sqrt{\deg(f)}$. We can use the structural relations at distance 2 obtained above to get a stronger upper bound on the polynomial degree of f. Recall from Corollary 7.1.2 that given a linearly dependent set of vertices F, there exists an edge (u, v) in the subgraph induced on F such that $d_F(u)d_F(v) \geq d$. By definition of T_1, $f'(u) \neq f'(v)$ and without loss of generality, let $f'(u) = 0$. This gives $\mathsf{s}_0(f', u)\mathsf{s}_1(f', v) \geq d$ and we get

$$\mathsf{s}_0(f)\mathsf{s}_1(f) \geq \mathsf{s}_0(f')\mathsf{s}_1(f') \geq \mathsf{s}_0(f', u)\mathsf{s}_1(f', v) \geq d = \deg(f).$$

□

Since by the Nisan-Szegedy Theorem 6.1.1 $\mathsf{bs}(f) \leq \deg(f)^2$, we get the following polynomial relation between sensitivity and block sensitivity.

Corollary 7.1.5. *For any Boolean function f, $\mathsf{bs}(f) \leq \mathsf{s}_0(f)^2\mathsf{s}_1(f)^2$.*

7.2 Structural relations at distance 3

We will now analyse the structure of a subgraph induced by linearly dependent vertices at distance at most 3. Without loss of generality, let us assume that there is a linear dependency among the vectors $x^+ \in V^+$ corresponding to a subset F of vertices in the hypercube. This linear dependency can equivalently be viewed in terms of matrices as follows.

Observation 7.2.1. *Given a set F of linearly dependent vertices such that $\sum_{u\in\{0,1\}^n} z_u u^+ = 0$ where $z_u = 0$ if $u \notin F$, we have*

$$\left(A_F - \sqrt{n}I\right) z = 0,$$

where A_F is the signed adjacency matrix A with columns and rows corresponding to a vertex set to zero if it does not belong to F, and z is a column vector with entries being the coefficients z_u, i.e.

$$A_F(x,y) = \begin{cases} A(x,y) & \text{if } x,y \in F \\ 0 & \text{otherwise} \end{cases}$$
$$z = \begin{bmatrix} z_0 & z_1 & \cdots & z_{2^n-1} \end{bmatrix}^T$$

Proof. Recall that by definition a vector x^+ is the column corresponding to the vertex x in $A + \sqrt{n}I$. This implies that a linear dependency can be expressed in matrix form as,

$$(A + \sqrt{n}I)z = 0. \tag{7.6}$$

For any row i, the non zero contributions in $\sum_{j\in\{0,1\}^n} A(i,j)z(j)$ are unaffected if we set $A(i,j) = 0$ when $z(j) = 0$. This implies that a modified matrix obtained from A by setting $A(i,j)$ to zero when $j \notin F$ also satisfies (7.6). Similarly, for row i such that $z(i) = 0$, we have $\sum_{j\in\{0,1\}^n} A(i,j)z(j) = 0$. This implies that (7.6) is satisfied when we set $A(i,j) = 0$ when $i \notin F$. □

We have the following theorem describing structural relations at distance 3 in the subgraph $H_n[F]$.

Theorem 7.2.2. *Given a linearly dependent set of vertices F in H_n, there exists a vertex $x \in F$ such that,*

$$|N_3^F(x)| \geq n^{3/2} - \sum_{\substack{y\in F \\ y:|x-y|=1}} c(x,y)$$

where $c(x,y)$ denotes the number of indices $j \in [n]$ such that either x^j or y^j belongs to F i.e,

$$c(x,y) \coloneqq \left|\{j : x^j \in F \vee y^j \in F\}\right|.$$

and $N_3^F(x)$ is the set of vertices $y \in F$ at distance exactly 3 from x.

For studying the structure of $H_n[F]$ at distance 3, we will analyse A_F^3 as it contains information about paths of length 3 in the subgraph $H_n[F]$. In particular, $A_F^3(x,y)$ is the sum of the signs of all the length-3 paths from x to y for vertices x and y in F, where the sign of a path is the product of the signs of all the edges in it.
The following lemma will be useful in proving Theorem 7.2.2.

Lemma 7.2.3. *Given a linearly dependent set of vertices F in H_n, for vertices $x, y \in F$,*

$$A_F^3(x,y) = \begin{cases} 0 & \text{if } |x-y| > 3 \text{ or } |x-y| \text{ is even} \\ 0/-1/+1 & \text{if } |x-y| = 3 \\ \sigma(x,y)\ c(x,y) & \text{if } |x-y| = 1 \end{cases}$$

where $c(x,y)$ denotes the number of indices $j \in [n]$ such that either x^j or y^j belongs to F.

Proof. We have the following cases for $A_F^3(x,y)$ based on the Hamming distance between x and y where $x, y \in F$.

Case 1: $|\mathbf{x}-\mathbf{y}| > 3$ or $|\mathbf{x}-\mathbf{y}|$ is even
In this case there exists no path of length 3 that connects x and y, and we have $A_F^3(x,y) = 0$.

Case 2: $|\mathbf{x}-\mathbf{y}| = 3$
If $y = x^{\{i,j,k\}}$ where $i, j, k \in [n]$, we can write $A_F^3(x,y)$ as the following

$$A_F^3(x,y) = (A_F \times A_F^2)(x,y) = \sum_{l \in \{i,j,k\}} A_F(x,x^l) A_F^2(x^l,y).$$

In the proof of Lemma 7.1.3 for structural relations at distance 2, we had observed that for any two vertices u and v at distance 2 from each other, $A_F^2(u,v)$ is either 0, -1 or $+1$. If there were two length-2 paths between u and v, the signs of the two paths would cancel each other, due to the property of the signature we have chosen (i.e. all 4-cycles are negative). This implies that there can be at most 1 non-zero path of length 2 from each of the x^i, x^j and x^k to y, and we get $|A_F^3(x,y)| \leq 3$. If there are more than one paths of length 3 between x and y, they will form an *induced 6-cycle*. An induced cycle is a cycle that is an induced subgraph of H_n, i.e. no two vertices in the cycle are connected by an edge in H_n that does not belong to the cycle. By Lemma 7.2.4 stated and proven below, every induced 6-cycle in the signed hypercube that we consider is assigned a negative sign by the property of our signature. This gives us that $|A_F^3(x,y)| \leq 1$ when $|x-y| = 3$.

Lemma 7.2.4. *In a signed hypercube H_n with a signature that satisfies the property that every 4-cycle is negative, every induced 6-cycle will also be negative.*

Proof. Let a, b, c, d, e and f be the signs of the edges of an induced 6-cycle in the hypercube as illustrated in Figure 7.1. The edges with signs g, h and i are the other edges that appear in our analysis as they are a part of some of the 4-cycles that share edges with the 6-cycle we consider. By the property of our signature that every 4-cycle is assigned a negative sign: $bchg = -1$, $afig = -1$ and $dhie = -1$. On multiplying all of these we get $abcdefg^2h^2i^2 = abcdef = -1$. □

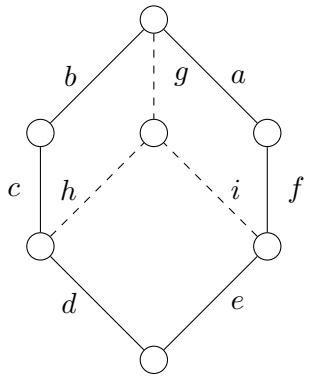


FIGURE 7.1: An induced 6-cycle in the signed hypercube H_n. The edges are labelled by their signs.

Case 3: $|\mathbf{x} - \mathbf{y}| = \mathbf{1}$

Let us suppose that $y = x^i$ for an $i \in [n]$. There are 4 types of possible length-3 paths from x to y which are shown in Figure 7.2. The length-3 paths from x to y are marked in red in the figure.

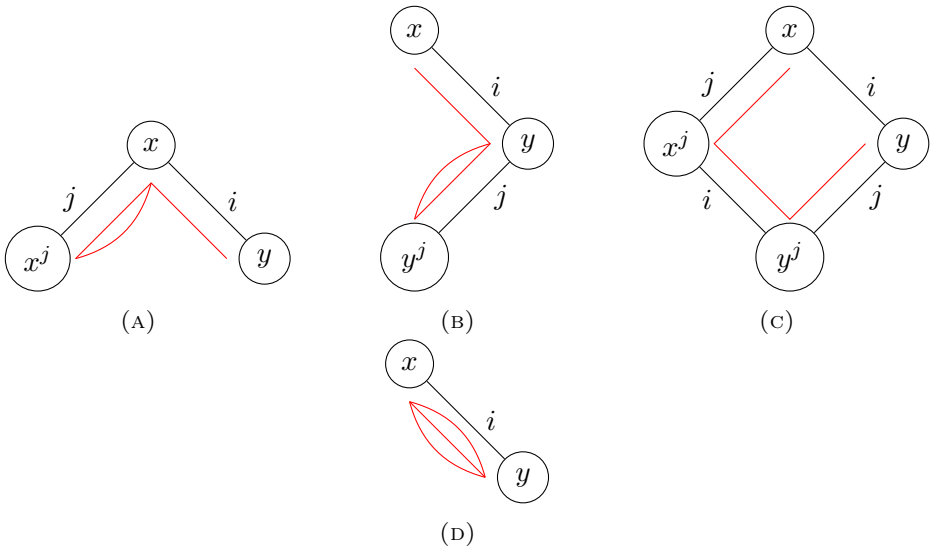


FIGURE 7.2: All length-3 paths from x to y when $|x, y| = 1$. The edges are labelled by the indices at which the endpoints differ.

To count all of these possible paths from x to y in $H_n[F]$, we study all the paths between them using edges corresponding to an index $j \in [n]$. For each index j, we have the following sub-cases: when both x^j and y^j are vertices in the subgraph $H_n[F]$, and when only one of x^j and y^j belong to $H_n[F]$.

Case 3a: Both $\mathbf{x^j}$ and $\mathbf{y^j}$ belong to F for an index $\mathbf{j} \in [\mathbf{n}]$

There are three paths that use edges corresponding to an index $j \neq i$ such that both x^j and y^j belong to F (see Figure 7.3: the first path being $x - x^j - x - y$, the second $x - y - y^j - y$, and the third $x - x^j - y^j - y$. Both the first path and the second path have the sign $\sigma(x, y)$. The sign of the third path is $-\sigma(x, y)$ due to the property of the signature that all 4-cycles are negative. This gives rise to a contribution of $\sigma(x, y)$ in $A^3_F(x, y)$ from each index $j \neq i$ that has both x^j and y^j as vertices of $H_n[F]$. When the index $j = i$, both x^j and y^j belong to F trivially and the path $x - y - x$

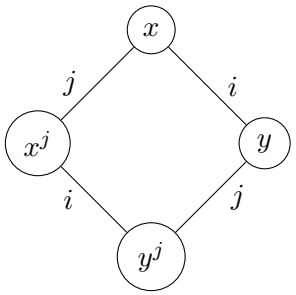


FIGURE 7.3: The subgraph when x, y, x^j and y^j belong to F. Edge labels are the indices at which the endpoints differ.

– y has a sign $\sigma(x, y)$. This implies that we have a contribution of $\sigma(x, y)$ from each index j such that both x^j and y^j belong to F.

Case 3b: Only one of x^j and y^j belong to F for an index j

In this case, we have one of the following length-3 paths from x to y using

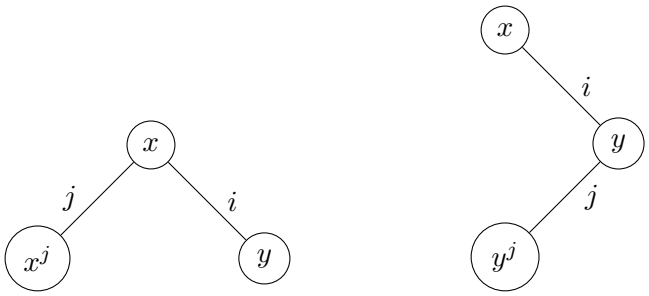


FIGURE 7.4: The subgraph when only one of x^j and y^j belong to F. Edge labels are the indices on which the endpoints differ.

edges corresponding to the index j: either $x - x^j - x - y$ or $x - y - y^j - y$ (see Figure 7.4). Since each of these paths has a signature $\sigma(x, y)$, an index j such that only one of x^j and y^j belong to F contributes $\sigma(x, y)$ to $A^3_F(x, y)$.

This implies that every index in the set $\{j : x^j \in F \vee y^j \in F\}$ contributes $\sigma(x, y)$ to $A^3_F(x, y)$ when $|x - y| = 1$.

This completes the proof of the lemma. □

We can now prove Theorem 7.2.2 which shows the structural relations between vertices in $H_n[F]$ that are at distance 3.

Proof of Theorem 7.2.2. Recall that a linear dependency among vertices in F can be viewed in terms of matrices as $A_F z = \sqrt{n}z$, following Observation 7.2.1. Since A_F has an eigenvector z with eigenvalue $\sqrt{n}$, we get

$$A^3_F z = n^{3/2} z.$$

Let z_x be the largest coefficient in the linear dependency, i.e. $z_x = \max_i |z_i|$. At the row labelled by x we have,

$$n^{3/2} z_x = (A_F^3 \times z)(x) = \sum_{y \in F} A_F^3(x,y)\ z_y$$

$$n^{3/2}\,|z_x| \leq \sum_{y \in F} \left|A_F^3(x,y)\ z_y\right| \leq \sum_{y \in F} \left|A_F^3(x,y)\right|\ |z_x|\,.$$

This gives us the following inequality where we use Lemma 7.2.3 to get

$$n^{3/2} \leq \sum_{y \in F} \left|A_F^3(x,y)\right| \leq \sum_{y:|x-y|=3} 1 + \sum_{y:|x-y|=1} c(x,y).$$

This shows that the size of the neighbourhood of x at distance 3, $|N_3^F(x)| \geq n^{3/2} - \sum_{y:|x-y|=1} c(x,y)$ which completes the proof of the theorem. □

We have the following corollary from Theorem 7.2.2 that gives a lower bound on the number of neighbours of a vertex at distance 3 in $H_n[F]$ in terms of the maximum degree d_F.

Corollary 7.2.5. *Given a linearly dependent set of vertices F in H_n, there exists a vertex in $H_n[F]$ that has at least $n^{3/2} - d_F(2d_F - 1)$ neighbours at distance 3 from it.*

Proof. From Theorem 7.2.2, we get a vertex x that has at least $n^{3/2} - \sum_{y:|x-y|=1} c(x,y)$ neighbours at distance 3 from it. The number of indices i such that x^i belongs to $H_n[F]$ is at most d_F for any vertex $x \in F$. For any two vertices x and y in F such that $x = y^i$ for some $i \in [n]$, the size of the set of indices j such that either x^j or y^j belongs to F is at most $2d_F - 1$. This holds as there is a vertex i such that both x^i and y^i belong to F. This implies that $c(x,y) \leq 2d_F - 1$ for any x and y such that $|x - y| = 1$. Since the number of y that are at distance 1 from x is bounded above by d_F, we have the above corollary. □

7.3 Linear dependency

It is interesting to see what subsets of vertices give rise to a linear dependency among their vectors in V^+. Characterising them might help us understand when the sensitivity of a function can be large. If a high value for some complexity measure (other than degree) implies the existence of a linear dependency, this could lead to new bounds between sensitivity and other measures of complexity.

Problem 7.3.1. *What are the (minimal) subsets of V^+ that are linearly dependent?*

It can be checked that the smallest linear dependency is among a vertex and all its neighbours:

$$x^+ = \frac{1}{\sqrt{n}} \sum_{y \sim x} \sigma(x,y)\ y^+.$$

On the other hand for linearly independent sets, the easiest examples are sets I of vectors in which for every vector x^+ there exists an index $u \in V(H_n)$ such that x^+ is the only vector in I that is nonzero at u. We call such a linearly independent set a *basic linearly independent set*. A main example of a basic linearly independent

set is the set $\{u^+ \mid u \in U^{\text{odd}}\}$ or $\{v^+ \mid v \in U^{\text{even}}\}$. Each of these sets provides an orthogonal basis for V^+.

Another example of a basic linearly independent set is the set of all u^+ such that the i^{th} coordinate of the vertex u is 1 for a fixed i. For each u^+ of this set, the vector u^+ is the only vector of the set that is not 0 at the coordinate u^i. Thus taking all such vectors provides another basis for V^+, although this basis is no longer an orthogonal one. The proof of Huang's result given by Knuth in [63] uses one such basis with $i = n$.

Chapter 8

Signed Hypercubes with only negative 4-cycles

The proof of the sensitivity conjecture relied crucially on signed hypercubes. In particular, it relied on signatures on the hypercube H_n with the property that every 4-cycle is negative, i.e. the product of the signs on the edges of the 4-cycle is negative. A signature with this property was used to show the existence of a signed hypercube with exactly two eigenvalues for the corresponding weighted adjacency matrix by Ahmadi et al. [6, Theorem 6.7 and Corollary 6.9]. This was the same signature given by Huang inductively. In this chapter, we shall look at how one can find all the signatures of the hypercube that satisfy this property. We will also study the frustration index of these signatures which is the minimum number of negative edges needed for any signature that has only negative 4-cycles. We shall see how the frustration index of these signatures are connected to the problem of the largest C_4-free subgraph of a hypercube, which was a question posed by Erdős [44]. We will give new proofs for a lower bound on the frustration index in Theorem 8.2.2 although an upper bound on the Erdős' problem shown by Bialostocki [24] naturally implies such a lower bound on the frustration index. We will also use a construction of C_4-free graph given by Brass et al. [25] as a lower bound for the Erdős' problem to show a construction of a signed hypercube with the number of negative edges nearly matching the lower bound on the frustration index in Theorem 8.2.4.

8.1 All signatures of hypercubes with only negative 4-cycles

We start with the signature used by Huang in his proof [56] and by Ahmadi et al. [6]. A signature with every 4-cycle being negative can easily be constructed on H_2 which consists of a single C_4: assign to one or three edges a negative sign, and to the rest a positive sign. Recall that H_n is built recursively from two disjoint copies of H_{n-1} by adding a matching between corresponding vertices. Having found a signature σ_{n-1} for H_{n-1}, proceed as follows: in the first copy of H_{n-1} assign signs as in σ_{n-1}, and in the other copy assign signs complementary to that in σ_{n-1}. Finally, all edges in the matching are assigned the same sign. We can observe that 4-cycles in each of the two copies inherit the property. The 4-cycles formed using two edges of the matching (which have the same sign) are also negative since the other edges in the cycle come from the two copies and are of opposite signs. Let us call this signed graph (H_n, σ^*).

Given a signature, one can get another with the same set of positive and negative cycles by a technique called *switching*. Informally, a *switch* at a vertex x in a signed graph of H_n flips the signs of edges incident on x. In particular, we can see that a switch at a vertex x is equivalent to multiplying both the row and column of the

adjacency matrix corresponding to x by -1. This operation does not change the eigenvalues, and the corresponding eigenvectors are obtained by switching the sign at the x^{th} coordinate. One may apply a series of switches on all the vertices in a set X. Formally, we can define switching as follows:

Definition 8.1.1 (Switching)**.** *A switching function $\zeta : V(G) \mapsto \{+, -\}$ associated to a signed graph G with signature σ assigns signs to the vertices of the graph. The switched signature on an edge e connecting vertices u and v, denoted $\sigma^{\zeta}(e)$, is defined as $\sigma^{\zeta}(e) \coloneqq \zeta(u)\sigma(e)\zeta(v)$.*

We now make some simple observations about switched signatures. One is that when switching the signature on a set X of vertices of the graph, only the edges in the cut-set of X and $V(G) \setminus X$ switch signs, i.e. only edges going between X and $V(G) \setminus X$ are affected by the switching operation. Another observation is that the switching operation does not change the sign of a cycle in the signed graph. This holds as every vertex has degree 2 in a cycle and if a vertex has been switched, the signs of both the edges going out of it are flipped, nullifying the effect of switching.

This technique of switching was found to be very integral due to a result by Zaslavsky [100] which showed that for any two signatures that have the same set of negative and positive cycles, there exists a switching function such that one is a switch of the other.

Theorem 8.1.2 ([100, Theorem 3.2])**.** *Given two signatures σ_1 and σ_2 of a graph G, σ_1 is a switching of σ_2 if and only if the sets of positive (or equivalently negative) cycles of (G, σ_1) and (G, σ_2) are the same.*

A simple introduction to signed graphs and a proof of this theorem appear in [101]. We reproduce the proof here for completeness. Note that the sign of a cycle, a closed walk, or in general a structure W in a signed graph (G, σ) is the product of the signs of its edges with its multiplicity being considered.

Proof of Theorem 8.1.2. ($\Rightarrow$) This direction holds as the switching operation does not change the sign of a cycle in a signed graph.

($\Leftarrow$) For the other direction, given two signatures σ_1 and σ_2 that have the same set of negative and positive cycles, we define a switching function ζ such that σ_2 will be obtained by switching according to ζ in σ_1, i.e. $\sigma_1^{\zeta} = \sigma_2$. Assume that the graph G is connected, and pick a spanning tree T and a vertex $v_0 \in V(G)$. Consider the following switching function:

$$\zeta(v) \coloneqq \sigma_1(T_{v_0v})\sigma_2(T_{v_0v})$$

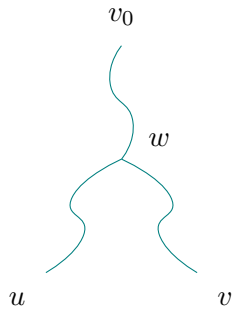
where T_{v_0v} is the path in the spanning tree T from v_0 to v. This switching function gives a new signature σ_1^{ζ} that assigns signs to an edge e between vertices u and v as,

$$\sigma_1^{\zeta}(e) = \zeta(u)\sigma_1(e)\zeta(v) = \sigma_1(T_{v_0u})\sigma_2(T_{v_0u})\sigma_1(e)\sigma_1(T_{v_0v})\sigma_2(T_{v_0v})$$

Let us assume that the path T_{v_0u} from v_0 to u and the path T_{v_0v} from v_0 to v in T diverge at some vertex w as shown in Figure 8.1, where w could be the same as u or v. We will now look at the switched signature and split the paths at w.

$$\begin{aligned}\sigma_1^{\zeta}(e) &= \sigma_1(T_{v_0w})\sigma_1(T_{wu})\sigma_2(T_{v_0w})\sigma_2(T_{wu})\sigma_1(e)\sigma_1(T_{v_0w})\sigma_1(T_{wv})\sigma_2(T_{v_0w})\sigma_2(T_{wv}) \\ &= \sigma_1(T_{wu})\sigma_2(T_{wu})\sigma_1(e)\sigma_1(T_{wv})\sigma_2(T_{wv})\end{aligned}$$

FIGURE 8.1: Path in the spanning tree T from v_0 to u and v



where the second equality holds as $\sigma^2(p) = 1$ for any path p and for any signature σ. Since signs of cycles are the same in the signed graphs corresponding to both the signatures, we have $\sigma_1(T_{wu})\sigma_1(e)\sigma_1(T_{wv}) = \sigma_2(T_{wu}\sigma_2(e)\sigma_2(T_{wv})$ for the cycle consisting of T_{wv}, e and T_{uw}. This gives us that $\sigma_1^{\zeta}(e) = \sigma_2(e)$. □

We point out that there are exactly 2^{2^n-1} signatures with the property that all 4-cycles on H_n are negative and these can be obtained using the switching technique. Given one such signature, one can get another by switching a set of vertices $X \subseteq V(H_n)$ of the hypercube. A series of switches on the complement of X results in the same assignment and hence there are 2^{2^n-1} signatures. We also have that any signature with the property that all 4-cycles are negative is one of the signatures discussed above from Zaslavsky's Theorem 8.1.2 as the 4-cycles generate the cycle space of H_n.

8.2 Frustration index of signed hypercube with only negative C_4

Now that we have the set of all signatures with only negative 4-cycles, we will try to find the signature with the smallest number of negative edges from this set. The minimum number of negative edges in a signature over all its switches is called its frustration index. If we were to remove all the negative edges in a signed hypercube with only negative 4-cycles, we get a C_4-free subgraph of the hypercube. As an interesting open problem in combinatorics, Erdős asked for the number of edges in the largest C_4-free subgraph of the hypercube [44] and this question has been studied extensively [24, 18, 42, 38, 27, 25]. We will see in this section how the best known upper and lower bounds on Erdős' question help us understand the frustration index of signed hypercubes with only negative 4-cycles. The frustration index is formally defined as follows.

Definition 8.2.1 (Frustration Index [52, 100])**.** *We define the frustration index of a signed graph (G, σ), denoted $F(G, \sigma)$, as the smallest number of negative edges of (G, σ') over all signatures σ' that are a switch of σ.*

To begin with, we will take another look at the signature used by Huang in his proof (which we refer to as (H_n, σ^*)) and bound the number of negative edges in it to get a trivial upper bound on the frustration index of (H_n, σ^*). This signature takes two copies of H_{n-1}, assigns a signature to a copy that satisfies every C_4 being negative and assigns the complementary signature to the other copy. All the matching edges are assigned positive signs. This gives that the number of negative edges equals

the number of edges in H_{n-1} as every edge in H_{n-1} is negative in one copy or the other but not both. Thus it is easy to see that the number of negative edges in this signature is $(n-1)\cdot 2^{n-2}$ and this gives us a trivial upper bound on the frustration index $F(H_n,\sigma^*) \leq (n-1)\cdot 2^{n-2}$. We will first get a lower bound on the frustration index $F(H_n,\sigma^*)$ and then construct a signature that tries to match the lower bound.

8.2.1 A lower bound on the frustration index

Although the problem of finding the frustration index of (H_n,σ^*) has not been looked at in the literature to the best of our knowledge, a related problem of finding the minimum number of edges that need to be removed from an H_n to make it C_4-free has received a lot of attention. A lower bound on this problem would easily translate to a lower bound on (H_n,σ^*). Such a lower bound was shown by Bialostocki [24]. We present two proofs of this lower bound, both of which are different from the original proof. They focus on the easier problem of a lower bound on frustration index. Nonetheless, they are useful since they help understand better the negative 4-cycles in the signed hypercube.

Theorem 8.2.2 ([24])**.** *For a signed hypercube (H_n,σ^*) which satisfies the property that every C_4 is negative, the frustration index $F(H_n,\sigma^*) \geq (n-\sqrt{n})\cdot 2^{n-2}$.*

We use the following definition of negative and positive degrees for vertices in our proof.

Definition 8.2.3 (Positive and negative degree)**.** *Given a signed hypercube (H_n,σ^*) and a vertex $v \in V(H_n)$, we define the* positive degree *of v, denoted by $d^+(v)$, as the number of positive edges incident on v, and* negative degree *of v, denoted by $d^-(v)$, as the number of negative edges incident on v*

Proof. We begin with a very simple lower bound on $F(H_n,\sigma^*)$ using the fact that each edge in H_n belongs to $(n-1)$ 4-cycles. The frustration index is at least the number of edges needed to cover all 4-cycles. This is given by the number of 4-cycles divided by the maximum number of 4-cycles covered by an edge and we have

$$F(H_n,\sigma) \geq \frac{2^{n-2}\binom{n}{2}}{n-1} = n\cdot 2^{n-3}.$$

From $F(H_n,\sigma) \geq n\cdot 2^{n-3}$, we can see that the average negative degree over all vertices $\overline{d^-} = \frac{2|E^-|}{|V|} \geq \frac{2\cdot n\cdot 2^{n-3}}{2^n} = \frac{n}{4}$. This implies that in a hypercube H_n for $n > 4$, there exist negative 4-cycles with three negative edges. When we count the total number of negative edges, we would have to count at least one negative edge to cover every C_4 and an extra contribution of 2 edges for every C_4 with three negative edges. Any C_4 formed using two negative edges at a vertex would lead to a C_4 with three negative edges, but such a cycle is counted twice as each such C_4 has two vertices in the cycle with two negative edges leading out of it. Thus we have,

$$\overline{d^-}\cdot 2^{n-1} \geq \frac{2^{n-2}\binom{n}{2} + 2^n\binom{\overline{d^-}}{2}\cdot 2/2}{n-1},$$

which simplifies as, $4\overline{d^-}^2 - 4n\overline{d^-} + (n^2-n) \leq 0$. This gives $\overline{d^-} \in [\frac{n-\sqrt{n}}{2}, \frac{n+\sqrt{n}}{2}]$, and hence $F(H_n,\sigma^*) \geq \overline{d^-}\cdot 2^{n-1} = (n-\sqrt{n})\cdot 2^{n-2}$. □

We now present an alternate proof of the theorem. Let us consider the following types of vertices in a 4-cycle in (H_n,σ^*) :

- *type 00* point: if it has two positive edges in a 4-cycle.
- *type 11* point: if it has two negative edges in a 4-cycle.
- *type 10* point: if it has one negative edge and one positive edge in a 4-cycle.

A vertex could be counted repeatedly as different types of points with respect to different 4-cycles. Every 4-cycle in (H_n, σ^*), regardless of it having one or three negative edges, has two points of type 10. Thus the number of points of type 10 is equal to twice the number of 4-cycles in H_n, i.e. $\sum_v (d^+(v) \cdot d^-(v)) = 2\binom{n}{2} \cdot 2^{n-2}$. Thus we have,

$$\sum_v d^-(v)(n - d^-(v)) = n\sum_v d^-(v) - \sum_v d^-(v)^2 = n(n-1)2^{n-2}.$$

On applying the Cauchy-Schwarz inequality,

$$\sum_v d^-(v)^2 \geq \frac{(\sum_v d^-(v))^2}{2^n},$$

we get

$$n \cdot \left(\frac{\sum_v d^-(v)}{2^n}\right) - \left(\frac{\sum_v d^-(v)}{2^n}\right)^2 \geq \frac{n(n-1)}{4}$$

which leads to the same result as in Theorem 8.2.2. □

Interestingly, both the lower bound and the upper bound on the frustration index of (H_n, σ^*) are closely related to the problem of largest C_4-free subgraph of H_n which was posed by Erdős [44]. In fact, he conjectured that the number of edges in the largest C_4-free subgraph of H_n is at most $(n+c) \cdot 2^{n-2}$ for some constant c but this was later disproved in a work by Brass, Harborth and Nienborg [25]. Brass et al. showed that it is at least $(n + 0.9\sqrt{n}) \cdot 2^{n-2}$ for $n \geq 9$ and showed a construction that achieves $(n + \sqrt{n}) \cdot 2^{n-2}$ for $n = 4^k$ where $k \geq 1$. We will now present the construction used in a slightly different language and obtain a signature on the hypercube that satisfies every 4-cycle being negative. In particular, we will prove the following theorem.

Theorem 8.2.4. *For a signed hypercube (H_n, σ^*) which satisfies the property that every C_4 is negative, the frustration index is bounded above as*

$$F(H_n, \sigma^*) \leq \begin{cases} (n - \sqrt{n}) \cdot 2^{n-2} & \text{if } n = 4^k \text{ for an integer } k \geq 1 \\ (n - 0.9\sqrt{n}) \cdot 2^{n-2} & \text{otherwise.} \end{cases}$$

8.2.2 Construction of a signed hypercube achieving optimal frustration index for powers of 4

The following signature of a hypercube is largely inspired by the construction of a large 4-cycle free subgraph of H_{m+k-1} given by Brass, Harborth and Nienborg [25] from smaller hypercubes H_m and H_k. In their construction, the edges of the smaller hypercubes are coloured green, red or blue such that every 4-cycle has at least one red and one blue edge. This helps in maintaining a C_4-free subgraph when all edges of a colour, say red, are deleted. In addition to this edge colouring, we add a vertex colouring of white and black to the vertices of the hypercube which helps to maintain the following property: every 4-cycle in the hypercube has an odd number of blue and red coloured edges and an even number of green coloured edges. This helps to ensure that every 4-cycle is negative if all the edges of a colour, say red, are made negative

edges and the rest positive. The hypercube H_{m+k-1} is constructed by splitting H_k into two smaller hypercubes H_{k-1} along an index i, and replacing each black vertex of H_m by an H_{k-1} and white vertices by the other H_{k-1}. The edge relation in the new hypercube H_{m+k-1} is defined using the edge colouring and vertex colouring of the smaller hypercubes. Formally, the hypercube is constructed as follows.

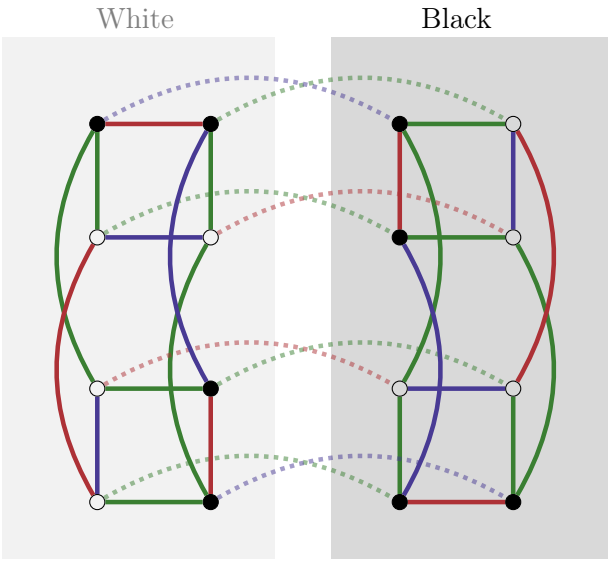


FIGURE 8.2: White and Black halves of a fully coloured $\widetilde{H}_4$.

A fully coloured hypercube of dimension k is a triplet (H_k, f, g) where the function f is a 2-colouring of the vertices of the hypercube H_k. This vertex colouring induces two types of edges: monochromatic edges and 2-coloured edges. The function g is a 3-colouring of the edges using Green, Red, Blue and the colouring satisfies the following constraints:

1. Red and blue edges are monochromatic under f.
2. Each 4-cycle of H_k has an odd number of red edges and an odd number of blue edges.

Let S_1 and S_2 be sets of indices such that S_1 is of order k, S_2 is of order m and they have a single element i in common, i.e. $S_1 \cap S_2 = \{i\}$. Let (H_k, f_1, g_1) be a fully coloured hypercube of dimension k whose coordinates are labeled by S_1, and let (H_m, f_2, g_2) be a fully coloured hypercube of dimension m whose coordinates are labeled by S_2. We define a fully coloured $(H_{m+k-1}, f, g) \coloneqq (H_m, f_2, g_2)\square(H_k, f_1, g_1)$ as follows:

- H_{m+k-1} is a hypercube of dimension $m+k-1$ and the coordinates of its vertices are labeled by elements in $S_1 \cup S_2$.
- To define the vertex colouring f on the new hypercube for a vertex $x \in V(H_{m+k-1})$, we take x_2 to be the restriction of x to the coordinates labelled by S_2. We also define x_1^* to be the vector obtained by restricting x to the coordinates labelled by S_1 and setting the coordinate labelled by i to be $f_2(x_2)$. The vertex colouring f on x is defined as $f(x) \coloneqq f_1(x_1^*)$.

  We can think of this as replacing each vertex of H_m by a half of the hypercube H_k that is split into two halves across the index i. For simplicity, we can think of H_m as the outer hypercube and H_k as the inner hypercube. The colour of

the vertex in the outer hypercube decides which half of the inner hypercube replaces it, and the new vertex in the larger hypercube inherits the colour from the vertex corresponding to it in this half of the inner hypercube.

- The edge-colouring g is defined for strings $x, y \in V(H_{m+k-1})$ such that x and y differ only at an index j i.e. $x \sim_j y$ where $j \in S_1 \cup S_2$ as follows:
  - **If $\mathbf{j \in S_1 \setminus \{i\}}$**: we define the edge colouring as $g(x, y) = g_1(x_1^*, y_1^*)$, i.e. if two vertices differ at an index corresponding to the inner hypercube (other than the index i which is common to both), they lie in the same half of the inner hypercube and the edge colours are inherited from that of the inner hypercube.
  - **If $\mathbf{j \in S_2}$** : we decide on the colouring based on $g_2(x_2, y_2)$ which is the edge colouring of the projection of x and y on the outer hypercube H_m. If $g_2(x_2, y_2) =$ Green, and since green edges are bichromatic (i.e. one of its endpoints is white and the other black), they are replaced by the two halves of the inner hypercube. In this case, the edge between x and y inherits the colour from the parallel edges that split the inner hypercube into two halves i.e $g(x, y) = g_1(x_1^*, y_1^*)$. Otherwise, if $g_2(x_2, y_2) =$ Red (and resp. Blue) and if the restriction of x to the coordinates labelled by $S_1 - \{i\}$, denoted x_1, is of an even weight we set $g(x, y) =$ Blue (and resp. Red). Note that the restriction of x or that of y onto $S_1 - \{i\}$ are the same since x and y do not differ on any index in $S_1 - \{i\}$. Similarly, if x_1 is of an odd weight, we set $g(x, y) =$ Red (and resp. Blue).

We note that in the original construction of Brass et al. [25], there was no vertex colouring but the C_4-free graph constructed eventually leads to the same subgraph as the one with a vertex colouring and all red edges removed. The addition of vertex colouring is helpful in arguing that the parities of red and blue edges in a 4-cycle are odd. The proofs described below regarding the number of red edges differ from that of Brass et al. [25].

We will now see that this vertex and edge colouring satisfy the conditions that blue and red coloured vertices are monochromatic and green edges are bichromatic. We will also prove that every 4-cycle of H_{m+k-1} has an odd number of red edges and an odd number of blue edges.

Proposition 8.2.5. *The construction of the hypercube (H_{m+k-1}, f, g) from fully coloured hypercubes (H_m, f_2, g_2) and (H_k, f_1, g_1) is well defined, and the result is a fully coloured hypercube that satisfies properties (1) and (2).*

Proof. Let us first check if edges between two vertices of the same colour are coloured either red or blue and that between vertices of different colours are coloured green i.e, they satisfy (1). We consider $x, y \in V(H_{m+k-1})$ such that x and y differ only at an index j i.e. $x \sim_j y$ and look at the following cases:

- $\mathbf{j \in S_1 \setminus \{i\}}$: In this case, both x and y correspond to the same vertex in the outer hypercube which has been replaced by a half of the inner hypercube (H_k, f_1, g_1). Since both the vertex and edge colours are inherited from the inner hypercube and since it is properly coloured, the colouring constraints (1) are satisfied in this case.
- $\mathbf{j \in S_2}$: In this case, if the vertices belong to different halves of the inner hypercube i.e. $f_2(x_2) \neq f_2(y_2)$, the colouring constraints are satisfied since the outer

edge is coloured green, and both the edge and vertex colouring are inherited from the inner hypercube. If the two vertices belong to the same half of the hypercube, they are coloured the same and the edges between them are coloured either red or blue in our construction.

The only thing that is left to show is that each 4-cycle of (H_{m+k-1}, f, g) has an odd number of red and an odd number of blue edges. Let $C = xywz$ be a 4-cycle in this fully coloured hypercube. Let p and q be the indices corresponding to the edges of C. We consider three possibilities:

- **p and q are both in $S_1 - \{i\}$:** In this case, since the vertices and edges in C inherit their colours from the inner hypercube (H_k, f_1, g_1) and C is a 4-cycle in it, property (2) holds.
- **p is in $S_1 - \{i\}$, q is in S_2:** Without loss of generality, let us assume that $y = x^q$. Here we consider two cases: when the edge corresponding to the outer hypercube $g_2(x_2, y_2)$ is coloured green and when it is not coloured green. If $g_2(x_2, y_2) =$ Green, C is a 4-cycle in an isomorphic copy of (H_k, f_1, g_1) in this construction in which the number of red and blue edges remain odd, and the green edges are even in number. If $g_2(x_2, y_2) \neq$ Green, there is one red edge and one blue edge in C corresponding to the index q. Since the edge corresponding to the outer hypercube $g_2(x_2, y_2)$ is either red or blue, the end points x_2 and y_2 must be of the same colour i.e. $f_2(x_2) = f_2(y_2)$. The edges corresponding to index p have the same colour since x_2 and y_2 have been replaced by the same copy of H_{k-1}. Property (2) is satisfied in this case as there is an odd number of green and blue edges and the number of green edges is either 0 or 2.
- **p and q are both in S_2:** The 4-cycle C corresponds to a 4-cycle C' with vertices x_2, y_2, w_2 and z_2 in (H_m, f_2, g_2). Since all the vertices in C are the same on all coordinates labelled by $S_1 \setminus \{i\}$, the parity of their restriction onto $S_1 \setminus \{i\}$ is the same. The red and blue edges in the 4-cycle C' remain the same as in C if the parity of their restriction onto $S_1 \setminus \{i\}$ is odd and are swapped otherwise. In any case, the number of red and blue edges they contribute remain odd. Note that for vertices x, y, w and z, only the coordinate at i changes and remaining coordinates are the same. The green edges in the cycle C' contribute edges of the same colour in C since the edge colours are inherited from the same edge in (H_k, f_1, g_1). Since the number of green edges in C' are even, the number of edges it contributes (of any colour) remains even.

□

We will now show an explicit construction of a fully coloured hypercube in k dimensions, denoted $\widetilde{H}_k$, from a fully coloured hypercube in 4 dimensions. Let us denote the fully coloured hypercube given in Figure 8.3 as $\widetilde{H}_4$. For each of the indices $i \in [4]$ of $\widetilde{H}_4$, there are 4 green edges, 2 blue and 2 red edges.

From two copies of $\widetilde{H}_4$, using the construction described above, we get $\widetilde{H}_7$ in which the number of green edges is 2^4 for four of the indices and 2^5 for three other indices.

In general, we use $\widetilde{H}_4$ to build a fully coloured hypercube $\widetilde{H}_{3l+1}$ using $\widetilde{H}_{3l-2}$ and $\widetilde{H}_4$ iteratively as follows

$$\widetilde{H}_{3l+1} = \widetilde{H}_4 \square \widetilde{H}_{3l-2}$$

for $l \geq 2$. In other words, each vertex of $\widetilde{H}_4$ is replaced by a half of $\widetilde{H}_{3l-2}$ depending on the colour of the vertex in $\widetilde{H}_4$.

For a fully coloured hypercube $\widetilde{H}_k$ constructed as above, let $\mathcal{R}_k[j]$, $\mathcal{B}_k[j]$ and $\mathcal{G}_k[j]$ be the number of edges coloured red, blue and green, respectively for an index j. Let the total number of red, blue and green edges in $\widetilde{H}_k$ be denoted $\mathcal{R}_k, \mathcal{B}_k$ and $\mathcal{G}_k$ respectively. It follows from the construction of $\widetilde{H}_{3l+1} = \widetilde{H}_4 \,\square\, \widetilde{H}_{3l-2}$ that if the number of red edges equals the number of blue edges in every index of the constituent $\widetilde{H}_4$ and $\widetilde{H}_{3l-2}$, the same would hold for the resulting $\widetilde{H}_{3l+1}$. Similarly, one can also see that if the constituent hypercubes $\widetilde{H}_4$ and $\widetilde{H}_{3l-2}$ have equal number of black and white vertices, $\widetilde{H}_{3l+1}$ also has an equal number of black and white vertices.

Observation 8.2.6. *For the fully coloured hypercube $\widetilde{H}_{3l+1}$ built using $\widetilde{H}_{3l-2}$ and $\widetilde{H}_4$:*

1. *The number of white vertices equals the number of black vertices.*
2. *The number of blue edges equals the number of red edges corresponding to an index j, i.e. $\mathcal{B}_{3l+1}[j] = \mathcal{R}_{3l+1}[j]$.*

This follows since fully coloured hypercubes are constructed iteratively from $\widetilde{H}_4$ which has an equal number of white and black vertices, and has an equal number of red and blue edges along every coordinate.

Recall that all the edges of $\widetilde{H}_k$ are coloured either red, blue or green, and that there are 2^{k-1} edges corresponding to any given index. Thus, calculating $\mathcal{G}_k[j]$ would also determine $\mathcal{R}_k[j]$ and $\mathcal{B}_k[j]$ when $k = 3l+1$ for some integer $l \geq 1$. The number of green edges in a fully coloured hypercube using the construction above is given by the following theorem.

Theorem 8.2.7. *There exists a construction of a fully coloured hypercube $\widetilde{H}_n$ such that there is a set of indices $\mathcal{L}_n$ of size $\frac{4}{3}(n - 4^{\lceil \log_4 n \rceil - 1})$ with $2^{n - \lceil \log_4 n \rceil - 1}$ green edges, and a set of indices $\mathcal{M}_n$ of size $\frac{1}{3}(4^{\lceil \log_4 n \rceil} - n)$ with $2^{n - \lceil \log_4 n \rceil}$ green edges where $n = 3l + 1$ for an integer $l \geq 1$.*

We prove this theorem by induction. The importance of the choice of the index i along which the inner hypercube is split into two halves is illustrated by the following lemma.

Lemma 8.2.8. *In the construction of a fully coloured hypercube $(H_{m+k-1}, f, g) = (H_m, f_2, g_2)\square(H_k, f_1, g_1)$ from two fully coloured hypercubes $\widetilde{H}_m$ and $\widetilde{H}_k$, if we choose to split the inner hypercube H_k along an index i, the number of green edges in the new hypercube along an index $j \in S_1 \cup S_2$ is given as follows.*

$$\mathcal{G}_{m+k-1}[j] = \begin{cases} 2^{m-1} \times \mathcal{G}_k[j], & \text{if } j \in S_1 \setminus \{i\} \\ \mathcal{G}_k[i] \times \mathcal{G}_m[j] & \text{otherwise,} \end{cases}$$

where $\mathcal{G}_l[j]$ is the number of green edges in a hypercube $\widetilde{H}_l$ along an index j.

Proof. We will prove this lemma by cases:

- For a $j \in S_1 \setminus \{i\}$, all the edges along this coordinate inherit their colours from the corresponding half of the fully coloured hypercube $\widetilde{H}_k$ by construction. Since there are an equal number of white and black vertices in $\widetilde{H}_m$ by Observation 8.2.6, there are 2^{m-1} copies of $\widetilde{H}_k$ split into two which replace every vertex of $\widetilde{H}_m$. Thus the number of green edges corresponding to this index $\mathcal{G}_{m+k-1}[j] = 2^{m-1} \times \mathcal{G}_k[j]$.

- For a $j \in S_2$, notice that a fully coloured hypercube $\widetilde{H}_{m+k-1}$ can have a green edge along an index j only if the corresponding edge in the outer hypercube (when restricted to indices in S_2) is coloured green. If the outer edge is coloured red or blue, the resulting edge could only be of one of those colours. In the construction, the edges corresponding to a green outer edge inherit their colours from the edges along the index i of $\widetilde{H}_k$. This gives that the number of green edges along the index j, $\mathcal{G}_{m+k-1}[j] = \mathcal{G}_k[i] \times \mathcal{G}_m[j]$ since every green edge of the outer hypercube contributes to $\mathcal{G}_k[i]$ green edges in $\widetilde{H}_{m+k-1}$.

□

With the above lemma, we will now prove Theorem 8.2.7.

Proof of Theorem 8.2.7. Let us choose the outer hypercube to be $\widetilde{H}_4$ with 4 green edges along each of its indices. We will now prove the theorem by induction for $n = 3l + 1$ for some integer $l \geq 1$.

Base Case: When $n = 4$, we have shown an example of a fully coloured hypercube with 4 green edges along each index which trivially satisfies the theorem.

Induction Case: By induction hypothesis, let us assume that there exists a fully coloured hypercube $\widetilde{H}_n$ such that there is a set of indices $\mathcal{L}_n$ of size $\frac{4}{3}(n - 4^{\lceil \log_4 n \rceil - 1})$ with $2^{n - \lceil \log_4 n \rceil - 1}$ green edges, and a set of indices $\mathcal{M}_n$ of size $\frac{1}{3}(4^{\lceil \log_4 n \rceil} - n)$ with $2^{n - \lceil \log_4 n \rceil}$ green edges. We will now show that we can construct an $\widetilde{H}_{n'}$ where $n' = n + 3$ from $\widetilde{H}_n$ and $\widetilde{H}_4$. We pick $\widetilde{H}_4$ as the outer hypercube and pick the index i from the inner hypercube $\widetilde{H}_n$ such that it has a large number of green edges. The set S_2 is chosen to work for our choice of i, i.e. $S_1 = \mathcal{L}_n \cup \mathcal{M}_n$ and $S_1 \cap S_2 = \{i\}$. By Lemma 8.2.8, the number of green edges along an index $j \in S_1 \setminus \{i\}$ equals $2^3 \times \mathcal{G}_n[j]$ for our choice of outer and inner hypercubes and the index i. We also have $4 \times \mathcal{G}_n[i]$ green edges along an index $j \in S_2$. We have the following cases:

Case 1: n is a power of 4.

In this case, $\widetilde{H}_n$ has $\mathcal{M}_n = \emptyset$ and $|\mathcal{L}_n| = n$ with each index in $\mathcal{L}_n$ having $2^{n - \lceil \log_4 n \rceil - 1}$ green edges by induction hypothesis. We pick any one of the indices in $\mathcal{L}_n$ as the index i to split $\widetilde{H}_n$ into two halves. By Lemma 8.2.8, the number of green edges along an index $j \in S_1 \setminus \{i\}$ (or equivalently $j \in \mathcal{L}_n \setminus \{i\}$) is $2^{n - \lceil \log_4 n \rceil - 1 + 3} = 2^{n + 3 - \lceil \log_4 (n+3) \rceil}$ since $\lceil \log_4 n \rceil + 1 = \lceil \log_4 (n+3) \rceil$ when n is a power of four. This gives $n - 1$ coordinates with $2^{n' - \lceil \log_4 n' \rceil}$ green edges. Since $\frac{1}{3}(4^{\lceil \log_4 n' \rceil} - n') = \frac{1}{3}(4n - n - 3) = n - 1$, we have a set of indices $\mathcal{M}_{n'} = S_1 \setminus \{i\}$ with $2^{n' - \lceil \log_4 n' \rceil}$ green edges. Note that in this case every index in $\mathcal{L}_n$ except i is now in $\mathcal{M}_{n'}$.

By Lemma 8.2.8, the number of green edges along each index $j \in S_2$ is $4 \times 2^{n - \lceil \log_4 n \rceil - 1} = 2^{n' - \lceil \log_4 n' \rceil - 1}$ and there are 4 indices in S_2. Since $\frac{4}{3}(n' - 4^{\lceil \log_4 n' \rceil - 1}) = \frac{4}{3}(n + 3 - 4^{\lceil \log_4 n \rceil}) = 4$, we have a set of indices $\mathcal{L}_{n'} = S_2$ with $2^{n' - \lceil \log_4 n' \rceil - 1}$ green edges. This proves that the induction case holds when $n' = n + 3$ is a power of 4. We note here that the chosen index i now belongs to $\mathcal{L}_{n'}$.

Case 2: When $n = 3l + 1$ for some integer $l > 1$ but is not a power of 4.

In this case, we will assume that the induction hypothesis for n and argue that the induction case holds for an $n' = n + 3$. Since n is not a power of 4 and since powers of 4 are 1 modulo 3, we have $\lceil \log_4 n \rceil = \lceil \log_4 n' \rceil$.

We choose to split $\widetilde{H}_n$ into two halves across an index $i \in \mathcal{M}_n$ that has the largest number of green edges (which is $2^{n-\lceil \log_4 n \rceil}$). There are $\frac{1}{3}(4^{\lceil \log_4 n \rceil} - n) - 1$ indices in $\mathcal{M}_n \setminus \{i\}$ with $2^{n-\lceil \log_4 n \rceil}$ green edges in $\widetilde{H}_n$. By Lemma 8.2.8, these indices have $2^{n-\lceil \log_4 n \rceil+3} = 2^{n'-\lceil \log_4 n' \rceil}$ green edges and they are $\frac{1}{3}(4^{\lceil \log_4 n \rceil} - n) - 1 = \frac{1}{3}(4^{\lceil \log_4 n' \rceil} - n')$ in number. These indices now form $\mathcal{M}_{n'}$.
Since $\mathcal{L}_n$ has $\frac{4}{3}(n - 4^{\lceil \log_4 n \rceil - 1})$ indices with $2^{n-\lceil \log_4 n \rceil - 1}$ green edges and the i chosen does not belong to $\mathcal{L}_n$, these indices have $2^{n-\lceil \log_4 n \rceil - 1+3} = 2^{n'-\lceil \log_4 n' \rceil - 1}$ green edges in $\widetilde{H}_{n'}$ by Lemma 8.2.8. These indices now belong to $\mathcal{L}_{n'}$. We also have $4\mathcal{G}_n[i] = 2^{n-\lceil \log_4 n \rceil+2} = 2^{n'-\lceil \log_4 n' \rceil - 1}$ green edges along an index $j \in S_2$ which are also in $\mathcal{L}_{n'}$.
In total, we have a set of indices $\mathcal{L}_{n'}$ of size $\frac{4}{3}(n - 4^{\lceil \log_4 n \rceil - 1}) + 4 = \frac{4}{3}(n' - 4^{\lceil \log_4 n' \rceil - 1})$ with $2^{n'-\lceil \log_4 n' \rceil - 1}$ green edges and a set of indices $\mathcal{M}_{n'}$ of size $\frac{1}{3}(4^{\lceil \log_4 n' \rceil} - n')$ with $2^{n'-\lceil \log_4 n' \rceil}$ green edges which proves that the theorem holds for this case. We note that the indices in $\mathcal{M}_n$ except the chosen index i are now in $\mathcal{M}_{n'}$ and that the indices in $\mathcal{L}_n$ and S_2 form the set $\mathcal{L}_{n'}$. In particular, the chosen index i belongs to $\mathcal{L}_{n'}$.

This proves the theorem by induction for all $n = 3l + 1$ for an integer $l \geq 1$.

□

We compile the following observations about the sets $\mathcal{L}_{n'}$ and $\mathcal{M}_{n'}$ made in the above proof that will prove to be useful later to characterise the vertex colouring of the fully coloured hypercubes.

Observation 8.2.9. *In a fully coloured hypercube $\widetilde{H}_{n'} = \widetilde{H}_n \square \widetilde{H}_4$ constructed as in the proof of Theorem 8.2.7, we have the following:*

- *The index i, chosen to split $\widetilde{H}_n$ into two halves, belongs to $\mathcal{L}_{n'}$.*
- *Every index $j \in \mathcal{M}_n$ except i belongs to $\mathcal{M}_{n'}$.*
- *Every index $j \in S_2$ belongs to $\mathcal{L}_{n'}$.*
- *When n is a power of 4, the set $S_1 \setminus \{i\}$ belongs to $\mathcal{M}_{n'}$.*

We use Theorem 8.2.7 to obtain an upper bound on the frustration index of a signature that assigns a negative sign to every 4-cycle in a hypercube.

Proof of Theorem 8.2.4. Since every 4-cycle in the fully coloured hypercube constructed in Theorem 8.2.7 has an odd number of red and blue edges, setting the signs of all the edges of a colour, say blue, to negative and the rest to positive ensures that every 4-cycle will be negative under this signature. Since the number of red edges equals the number of blue edges along each index as seen in Observation 8.2.6, we will count the number of green edges to find the number of blue edges. We analysis this in various cases as below:

Case 1. n is a power of 4.
Let $n = 4^k$ for an integer $k \geq 1$. From Theorem 8.2.7, we have n indices with 2^{n-k-1} green edges each. Since $\sqrt{n} = 2^k$, the total number of green edges is $\sqrt{n}\left(2^{n-1}\right)$. All the edges in a fully coloured hypercube are coloured green, red or blue and there are an equal number of red and blue edges. Hence the total number of edges coloured blue is $\frac{n2^{n-1} - \sqrt{n}2^{n-1}}{2}$. The frustration index when n is a power of 4 is,

$$F(H_n, \sigma^*) \leq (n - \sqrt{n}) \cdot 2^{n-2} .$$

Case 2: $n = 3^{l+1}$ for an integer $l \geq 1$.

Let $4^k \leq n < 4^{k+1}$. From Theorem 8.2.7, the number of green edges in $\widetilde{H}_n$ is

$$\mathcal{G}_n = \frac{4}{3}\left(n - 4^k\right)2^{n-k-2} + \frac{1}{3}\left(4^{k+1} - n\right)2^{n-k-1} = \frac{2^{n-1}}{3 \cdot 2^k}\left(n + 2 \cdot 4^k\right).$$

The number of blue edges is

$$\mathcal{B}_n = \frac{n \cdot 2^{n-1} - \frac{2^{n-1}}{3 \cdot 2^k}\left(n + 2 \cdot 4^k\right)}{2} = 2^{n-2}\left(n - \frac{\left(n + 2 \cdot 4^k\right)}{3 \cdot 2^k}\right).$$

To get an upper bound on the number of blue edges and to get it to the form $(n - c\sqrt{n}) \cdot 2^{n-2}$, we minimise $\frac{\left(n+2\cdot 4^k\right)}{\sqrt{n}\cdot 3\cdot 2^k}$ and see that the minimum occurs at $n = 2^{2k+1}$ and

$$\min \frac{\left(n + 2 \cdot 4^k\right)}{\sqrt{n} \cdot 3 \cdot 2^k} = \frac{2\sqrt{2}}{3}.$$

Thus we have an upper bound on the frustration index when $n = 3l + 1$ for an integer $l \geq 1$:

$$F(H_n, \sigma^*) \leq (n - 0.9428\sqrt{n})2^{n-2}.$$

Case 3: $n \neq 1 \mod 3$.

In this case, we take the fully coloured hypercube $\widetilde{H}_{n+m}$ constructed as above such that $n + m = 3l + 1$ for an integer $l \geq 1$ and $m \in \{1, 2\}$. The main idea is to remove all the edges corresponding to m coordinates in $\widetilde{H}_{n+m}$, which results in 2^m components and we choose the component with the least number of blue edges. We will now formalise this intuition and count the number of blue edges.

Let $n + m = 1 \mod 3$ be such that $4^k < n + m \leq 4^{k+1}$ for an $m \in \{1, 2\}$. From Theorem 8.2.7, $\widetilde{H}_{n+m}$ has $\frac{4}{3}(n + m - 4^k)$ indices with $2^{n+m-k-2}$ green edges and $\frac{1}{3}(4^{k+1} - n - m)$ indices with $2^{n+m-k-1}$ green edges. We delete all the edges from m indices that have $2^{n+m-k-2}$ green edges. This is possible since $\frac{4}{3}(n + m - 4^k) \geq 4 > m$. The total number of green edges along n indices, after m coordinates have been deleted is

$$\frac{1}{3}\left(4n + m - 4^{k+1}\right)2^{n+m-k-2} + \frac{1}{3}\left(4^{k+1} - n - m\right)2^{n+m-k-1} = \frac{2^{n+m-k-2}}{3}\left(2n - m + 4^{k+1}\right).$$

This subgraph of $\widetilde{H}_{n+m}$, formed after deleting all the edges in m indices, consists of 2^m components each of which is a fully coloured hypercube of dimension n. We pick a component of this subgraph with the largest number of green edges and there exists a component with the number of green edges being

$$\mathcal{G}_n \geq \frac{1}{2^m}\left(\frac{2^{n+m-k-2}}{3}\left(2n - m + 4^{k+1}\right)\right) = \frac{2^{n-k-2}}{3}\left(2n - m + 4^{k+1}\right).$$

The number of blue edges,

$$\mathcal{B}_n \leq \frac{n \cdot 2^{n-1} - \frac{2^{n-k-2}}{3}\left(2n - m + 4^{k+1}\right)}{2} = 2^{n-2}\left(n - \frac{\left(n + 2 \cdot 4^k - m/2\right)}{3 \cdot 2^k}\right).$$

As was done in the previous case, we minimise $\frac{\left(n+2\cdot 4^k - m/2\right)}{\sqrt{n}\cdot 3\cdot 2^k}$ to get an upper

bound on the number of blue edges and see that the minimum occurs at $n = 2^{2k+1} - 1$. We get

$$\min \frac{\left(n + 2 \cdot 4^k - m/2\right)}{\sqrt{n} \cdot 3 \cdot 2^k} = \frac{2}{3 \cdot 2^k} \sqrt{2^{2k+1} - 1} \,,$$

which for large values of k becomes $2\sqrt{2}/3 = 0.9428$. In particular, we see that when $n \geq 2^{17} - 1$,

$$F(H_n, \sigma^*) \leq (n - 0.9428\sqrt{n})2^{n-2} \,.$$

We can also see from the above bound that $F(H_n, \sigma^*) \leq (n - 0.9\sqrt{n})2^{n-2}$ when $n \geq 11$, and see that this bound also holds for values of n between 9 and 11 by a brute force check.

□

We now turn our attention to the function that is used for vertex colouring in the construction of $\widetilde{H}_n$. We see that this function corresponds to the Ambainis function (see Definition 1.2.6) which has been well-studied in the literature [8, 67, 55]. It was introduced by Ambainis as a function whose positive adversary bound is larger than its polynomial degree, which showed that the positive adversary bound is a very useful lower bound on quantum query complexity [8].

Ambainis function

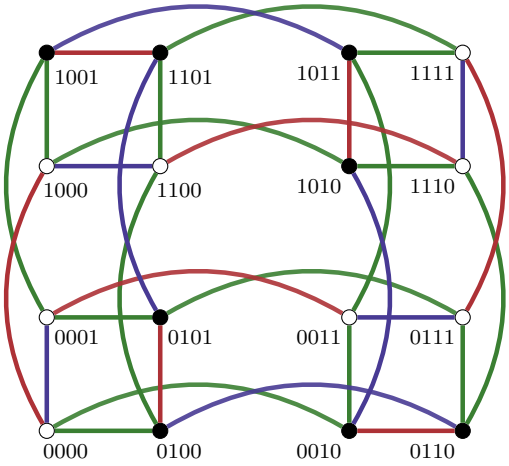


FIGURE 8.3: A fully coloured $\widetilde{H}_4$ with vertices being coloured according to Ambainis function.

We observe that the recursive function we used to determine the vertex colouring in $\widetilde{H}_{3l+1}$ for any integer l is the Ambainis function. This is easy to see in $\widetilde{H}_4$ (Figure 8.3) where a vertex x is coloured "white" if the corresponding string is evaluated to 1 by the Ambainis function (Definition 1.2.6), i.e. $f_A(x) = 1$.

We now show that the Ambainis function is composed with itself to get the vertex colouring for fully coloured hypercubes with larger dimensions. Let the vertex colouring of $\widetilde{H}_k$ be denoted f_k. Recall that a fully coloured hypercube $\widetilde{H}_{3l+4}$ is constructed from $\widetilde{H}_{3l+1}$ and $\widetilde{H}_4$ for any integer $l \geq 1$ by splitting $\widetilde{H}_{3l+1}$ across an index i and

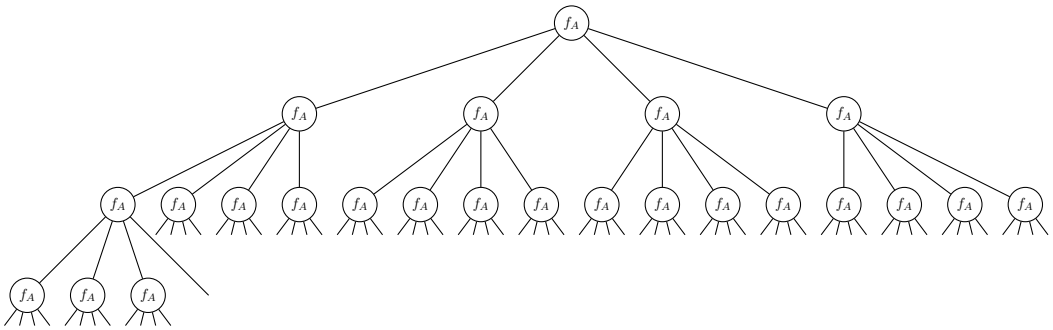


FIGURE 8.4: Vertex Colouring function for $\widetilde{H}_{73}$.

replacing each vertex of $\widetilde{H}_4$ by a half of $\widetilde{H}_n$ which depends on the vertex colouring f_4. We get f_{3l+4} from f_{3l+1} after replacing the i-th input by an f_4. This is because f_4 determines the half of the hypercube from which the vertex inherit colours and the two halves differ only at the ith bit. We show that such a construction yields a vertex colouring function that can be represented by a 4-ary tree with each internal node replaced by the Ambainis function f_A which acts on its 4 input bits.

Theorem 8.2.10. *For a fully coloured hypercube $\widetilde{H}_k$ of dimension k where $k = 3l+1$ for an integer $l \geq 1$, the vertex colouring function f_k can be represented by a 4-ary tree whose leaves correspond to input bits of the function. Every internal node in the tree corresponds to the Ambainis function f_A and has exactly 4 children. The leaves lie at either the last or the penultimate level and the height of this tree is $\lceil \log_4 k \rceil + 1$.*

This tree corresponds to the Ambainis function f_A^d when $k = 4^d$ for any integer $d \geq 1$. We denote such a tree with k leaves by T_k. The tree T_{73} is given in Figure 8.4.

Proof of Theorem 8.2.10. The proof of this theorem can be shown by induction on integers $l \geq 1$. The induction hypothesis is as follows.

Induction Hypothesis: For a $k = 3l+1$, the vertex colouring of $\widetilde{H}_k$ is represented by the tree T_k. If the leaf corresponding to an index j lies in the last level of the tree T_k, $j \in \mathcal{L}_k$ and if it lies in the penultimate level, $j \in \mathcal{M}_k$.

Base Case: For the base case, the hypercube of dimension $k = 4$ when $l = 1$ is given in Figure 8.3 whose vertex colouring is according to the Ambainis function f_A on 4 bits. This is represented by a T_4 that takes as input 4 bits and outputs according to f_A. In this case, all the leaves lie in the same level and we also know that all the indices lie in $\mathcal{L}_4$ since k is a power of 4.

Induction Case: For the induction case, let us assume that the theorem holds for $k = 3l + 1$. The hypercube corresponding to $3l + 4$ is constructed from $\widetilde{H}_{3l+1}$ and $\widetilde{H}_4$ where $\widetilde{H}_{3l+1}$ is split into halves across an index i. The vertex colouring f_{3l+4} is given by f_{3l+1} with the i-th input bit replaced by an f_4 by construction. By induction hypothesis, f_{3l+1} is given by a 4-ary tree T_{3l+1} with the internal nodes being f_A. By the construction of $\widetilde{H}_{3l+4}$, the tree corresponding to f_{3l+4} is T_{3l+1} with the leaf corresponding to the i-th index replaced by an f_A node with 4 children. We will make use of Observation 8.2.9 to show that the tree corresponding to f_{3l+4} increases in height from that of T_{3l+1} only if all the leaves of T_{3l+1} are at the same level i.e., the last level. We consider the following two cases:

- $3l+1$ is a power of 4: all the indices in S_1 lie in $\mathcal{L}_{3l+1}$ and are in the last level of T_{3l+1} by induction hypothesis and Observation 8.2.9. In the construction, one of these indices is chosen as i and if T_{3l+1} is modified by changing the i-th leaf to f_A, this tree matches the description of T_{3l+4}. Note that the new leaves of the f_A node that replaced a leaf of T_{3l+1} are now indices in S_2. From Observation 8.2.9, we have that every index in $S_1 \setminus \{i\}$ lies in the penultimate level and are in $\mathcal{M}_{3l+4}$. We also have that every index in S_2 that lies in the last level belongs to $\mathcal{L}_{3l+4}$, thus proving the induction case.
- $3l+1$ is not a power of 4: the set $\mathcal{M}_{3l+1}$ is non-empty and by the induction hypothesis, there are leaves in the penultimate level of T_{3l+1}. In the construction of $\widetilde{H}_{3l+4}$, the tree representing the vertex colouring f_{3l+4} can be obtained by modifying T_{3l+1} by taking the leaf corresponding to an index $i \in \mathcal{M}_{3l+1}$ and replacing it by a node f_A with its leaves now being labelled by S_2. Such a tree is consistent with the description of T_{3l+4} as the leaf being replaced lies in the penultimate level. Since all the other leaves in the penultimate level remain at the same level and since the leaves corresponding to S_2 now lie in the last level, these are consistent with the new sets of indices $\mathcal{M}_{3l+4}$ and $\mathcal{L}_{3l+4}$ by Observation 8.2.9. This proves the induction case.

This shows that the tree T_k represents the vertex colouring for $\widetilde{H}_k$ and that the vertex colouring is given by the Ambainis function. □

The frustration index of a signed hypercube with only negative 4-cycles, $F(H_n, \sigma^*)$, is not yet fully characterised: we have a tight bound when n is a power of 4, but in other cases there is a gap between the best known upper and lower bounds. It would be interesting to see if the construction as detailed above can lead to a better upper bound when the constituent graphs are chosen differently, or if this is the best this method can do. It would also be illuminating to see if an improvement in the frustration index problem would lead to an improvement in the bounds for Erdős' problem, as it would help see how well connected these two problems are.

Conclusion

In the first part of this thesis, we looked at a new measure of complexity which we termed "certificate game complexity" and its variants when allowed to have shared randomness, quantum, or non-signalling strategies. We obtained upper and lower bounds on these variants in terms of certificate complexity C, randomised query complexity R, positive Adversary bound MM, fractional certificate complexity FC etc. The following are interesting research questions to pursue:

Relations between variants of certificate game complexity: We do not know of any separation between the non-signalling, quantum or shared randomness variants of certificate game complexity. Such a result might indicate how non-signalling strategies fare better than shared randomness strategies in the context of zero-communication protocols. If a total function can be shown to separate these variants of certificate game complexity, they would help separate FC from EC. Another relevant question would be to ask if the private coin variant CG is at most quadratic in $\mathsf{CG}^{\mathrm{pub}}$.

A better upper bound on CG in terms of the single-bit variant $\mathsf{CG}^{\mathbf{pub}}_{[1]}$: Since $\mathsf{CG}^{\mathrm{pub}}_{[1]}$ is asymptotically equal to sensitivity s, a better upper bound might bring us closer to the quadratic sensitivity-block sensitivity conjecture, i.e. $\mathsf{bs}(f) \leq O(\mathsf{s}(f)^2)$. The current upper bound on $\mathsf{CG}^{\mathrm{pub}}(f)$ is $O(\mathsf{CG}^{\mathrm{pub}}_{[1]}(f)^5)$ and an upper bound of $O(\mathsf{CG}^{\mathrm{pub}}_{[1]}(f)^2)$ would prove the conjecture.

In the second part of the thesis, we looked at linear dependencies among vectors chosen using a signature on the hypercube with only negative 4-cycles. We studied the various structural relations that can be guaranteed in the induced subgraph produced from these linear dependencies at distances 2 and 3. We also studied the signatures on the hypercube that have only negative 4-cycles, and tried to find the signature that minimises the number of negative edges used. We have the following open questions from this part:

Characterising linear dependencies: When the above mentioned linear dependencies among vertices are created the sensitivity is large. Understanding how these linear dependencies arise would shed light on how the sensitivity of a function can be large.

Applications for structural relations at larger distances: Structural relations at distance 2 gave rise to a better upper bound on polynomial degree in terms of the 0-sensitivity and 1-sensitivity of a function. Although structural relations at larger distances are interesting on their own right, it would be nice to have applications that are relevant from a complexity theory point of view.

Better upper and lower bounds on the frustration index: Since the current upper and lower bounds on the frustration index of a signature with only negative 4-cycles match the bounds on Erdős' problem of the largest C_4-free subgraph of the hypercube, it remains to be seen if an improvement on one problem would translate to an improvement on the other.

Bibliography

[1] Scott Aaronson. "Lower Bounds for Local Search by Quantum Arguments". In: *SIAM Journal on Computing* 35.4 (2006), pp. 804–824. DOI: 10.1137/S0097539704447237. URL: https://doi.org/10.1137/S0097539704447237.

[2] Scott Aaronson. "Quantum certificate complexity". In: *J. Comput. Syst. Sci.* 74 (2008), pp. 313–322.

[3] Scott Aaronson, Shalev Ben-David, and Robin Kothari. "Separations in Query Complexity Using Cheat Sheets". In: *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '16. Cambridge, MA, USA, 2016, pp. 863–876. ISBN: 9781450341325. DOI: 10.1145/2897518.2897644. URL: https://doi.org/10.1145/2897518.2897644.

[4] Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. "Degree vs. approximate degree and Quantum implications of Huang's sensitivity theorem". In: *Symposium on Theory of Computing (STOC)*. ACM, 2021, pp. 1330–1342.

[5] A. Acín, T. Durt, N. Gisin, and J. I. Latorre. "Quantum nonlocality in two three-level systems". In: *Phys. Rev. A* 65 (5 May 2002), p. 052325. DOI: 10.1103/PhysRevA.65.052325. URL: https://link.aps.org/doi/10.1103/PhysRevA.65.052325.

[6] Bahman Ahmadi, Fatemeh Alinaghipour, Shaun Cavers Michael S.and Fallat, Karen Meagher, and Shahla Nasserasr. "Minimum number of distinct eigenvalues of graphs". In: *Electronic Journal of Linear Algebra* 26.45 (2013), pp. 673–691. DOI: 10.13001/1081-3810.1679. URL: https://repository.uwyo.edu/ela/vol26/iss1/45.

[7] Mafalda L. Almeida, Stefano Pironio, Jonathan Barrett, Géza Tóth, and Antonio Acín. "Noise Robustness of the Nonlocality of Entangled Quantum States". In: *Phys. Rev. Lett.* 99 (4 July 2007), p. 040403. DOI: 10.1103/PhysRevLett.99.040403. URL: https://link.aps.org/doi/10.1103/PhysRevLett.99.040403.

[8] Andris Ambainis. "Polynomial degree vs. quantum query complexity". In: *Journal of Computer and System Sciences* 72.2 (2006). JCSS FOCS 2003 Special Issue, pp. 220–238. ISSN: 0022-0000. DOI: https://doi.org/10.1016/j.jcss.2005.06.006. URL: https://www.sciencedirect.com/science/article/pii/S0022000005000899.

[9] Andris Ambainis. "Quantum lower bounds by quantum arguments". In: *Symposium on Theory of Computing (STOC)*. 2000, pp. 636–643. DOI: 10.1145/335305.335394. URL: https://doi.org/10.1145/335305.335394.

[10] Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. "Separations in Query Complexity Based on Pointer Functions". In: *J. ACM* 64.5 (Sept. 2017). ISSN: 0004-5411. DOI: 10.1145/3106234. URL: https://doi.org/10.1145/3106234.

[11] Andris Ambainis, Martins Kokainis, Krisjanis Prusis, and Jevgēnijs Vihrovs. "All Classical Adversary Methods are Equivalent for Total Functions". In: *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*. Vol. 96. Leibniz International Proceedings in Informatics (LIPIcs). 2018, 8:1–8:14. ISBN: 978-3-95977-062-0. DOI: 10.4230/LIPIcs.STACS.2018.8. URL: http://drops.dagstuhl.de/opus/volltexte/2018/8495.

[12] Anurag Anshu, Shalev Ben-David, and Srijita Kundu. "On Query-To-Communication Lifting for Adversary Bounds". In: *36th Computational Complexity Conference (CCC 2021)*. Vol. 200. 2021, 30:1–30:39. DOI: 10.4230/LIPIcs.CCC.2021.30. URL: https://drops.dagstuhl.de/opus/volltexte/2021/14304.

[13] Anurag Anshu, Dmitry Gavinsky, Rahul Jain, Srijita Kundu, Troy Lee, Priyanka Mukhopadhyay, Miklos Santha, and Swagato Sanyal. "A Composition Theorem for Randomized Query Complexity". In: *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2017)*. Vol. 93. Leibniz International Proceedings in Informatics (LIPIcs). 2018, 10:1–10:13. ISBN: 978-3-95977-055-2. DOI: 10.4230/LIPIcs.FSTTCS.2017.10. URL: http://drops.dagstuhl.de/opus/volltexte/2018/8396.

[14] Nikhil Bansal and Makrand Sinha. "K-Forrelation Optimally Separates Quantum and Classical Query Complexity". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 1303–1316. ISBN: 9781450380539. URL: https://doi.org/10.1145/3406325.3451040.

[15] H. Barnum, M. Saks, and M. Szegedy. "Quantum query complexity and semidefinite programming". In: *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.* 2003, pp. 179–193. DOI: 10.1109/CCC.2003.1214419.

[16] Andrew Bassilakis, Andrew Drucker, Mika Göös, Lunjia Hu, Weiyun Ma, and Li-Yang Tan. "The Power of Many Samples in Query Complexity". In: *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Vol. 168. Leibniz International Proceedings in Informatics (LIPIcs). 2020, 9:1–9:18. ISBN: 978-3-95977-138-2. DOI: 10.4230/LIPIcs.ICALP.2020.9. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12416.

[17] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. "Quantum Lower Bounds by Polynomials". In: *J. ACM* 48.4 (July 2001), pp. 778–797. ISSN: 0004-5411. DOI: 10.1145/502090.502097. URL: https://doi.org/10.1145/502090.502097.

[18] Bernd Becker and Hans-Ulrich Simon. "How robust is the n-cube?" In: *Information and Computation* 77.2 (1988), pp. 162–178. ISSN: 0890-5401. DOI: https://doi.org/10.1016/0890-5401(88)90056-9. URL: https://www.sciencedirect.com/science/article/pii/0890540188900569.

[19] Mihir Bellare, Oded Goldreich, and Madhu Sudan. "Free Bits, PCPs, and Nonapproximability—Towards Tight Results". In: *SIAM Journal on Computing* 27.3 (1998), pp. 804–915. DOI: 10.1137/S0097539796302531. URL: https://doi.org/10.1137/S0097539796302531.

[20] Shalev Ben-David. *Sensitivity Conjecture resolved*. Blog Comment. 2019. URL: https://scottaaronson.blog/?p=4229#comment-1813084.

[21] Shalev Ben-David and Eric Blais. "A Tight Composition Theorem for the Randomized Query Complexity of Partial Functions: Extended Abstract". In: *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*. 2020, pp. 240–246. DOI: 10.1109/FOCS46700.2020.00031. URL: https://doi.org/10.1109/FOCS46700.2020.00031.

[22] Shalev Ben-David and Robin Kothari. "Randomized Query Complexity of Sabotaged and Composed Functions". In: *Theory of Computing* 14.5 (2018), pp. 1–27. DOI: 10.4086/toc.2018.v014a005. URL: https://theoryofcomputing.org/articles/v014a005.

[23] A. Bernasconi. "Sensitivity vs. block sensitivity (an average-case study)". In: *Information Processing Letters* 59.3 (1996), pp. 151–157. ISSN: 0020-0190. DOI: https://doi.org/10.1016/0020-0190(96)00105-6. URL: https://www.sciencedirect.com/science/article/pii/0020019096001056.

[24] Arie Bialostocki. "Some Ramsey-type results regarding the graph of the n-cube". In: *Ars Combinatoria* 16-A (1983), pp. 39–48.

[25] Peter Brass, Heiko Harborth, and Hauke Nienborg. "On the maximum number of edges in a c4-free subgraph of qn". In: *Journal of Graph Theory* 19.1 (1995), pp. 17–23. DOI: 10.1002/jgt.3190190104. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190190104.

[26] Gilles Brassard, Richard Cleve, and Alain Tapp. "Cost of Exactly Simulating Quantum Entanglement with Classical Communication". In: *Phys. Rev. Lett.* 83 (9 Aug. 1999), pp. 1874–1877. DOI: 10.1103/PhysRevLett.83.1874. URL: https://link.aps.org/doi/10.1103/PhysRevLett.83.1874.

[27] A E Brouwer, I J Dejter, and C Thomassen. "Highly Symmetric Subgraphs of Hypercubes". In: *Journal of Algebraic Combinatorics* 2.1 (Mar. 1993), pp. 25–29.

[28] Nicolas Brunner, Daniel Cavalcanti, Stefano Pironio, Valerio Scarani, and Stephanie Wehner. "Bell nonlocality". In: *Rev. Mod. Phys.* 86 (2 Apr. 2014), pp. 419–478. DOI: 10.1103/RevModPhys.86.419. URL: https://link.aps.org/doi/10.1103/RevModPhys.86.419.

[29] Harry Buhrman, Richard Cleve, Serge Massar, and Ronald de Wolf. "Nonlocality and communication complexity". In: *Rev. Mod. Phys.* 82 (1 Mar. 2010), pp. 665–698. DOI: 10.1103/RevModPhys.82.665. URL: https://link.aps.org/doi/10.1103/RevModPhys.82.665.

[30] Harry Buhrman, Łukasz Czekaj, Andrzej Grudka, Michał Horodecki, Paweł Horodecki, Marcin Markiewicz, Florian Speelman, and Sergii Strelchuk. "Quantum communication complexity advantage implies violation of a Bell inequality". In: *Proceedings of the National Academy of Sciences* 113.12 (2016), pp. 3191–3196. DOI: 10.1073/pnas.1507647113. URL: https://www.pnas.org/doi/abs/10.1073/pnas.1507647113.

[31] Harry Buhrman and Ronald de Wolf. "Complexity Measures and Decision Tree Complexity: A Survey". In: *Theor. Comput. Sci.* 288.1 (Oct. 2002), pp. 21–43. ISSN: 0304-3975. DOI: 10.1016/S0304-3975(01)00144-X. URL: http://dx.doi.org/10.1016/S0304-3975(01)00144-X.

[32] N. J. Cerf, N. Gisin, S. Massar, and S. Popescu. "Simulating Maximal Quantum Entanglement without Communication". In: *Phys. Rev. Lett.* 94 (22 June 2005), p. 220403. DOI: 10.1103/PhysRevLett.94.220403. URL: https://link.aps.org/doi/10.1103/PhysRevLett.94.220403.

[33] Sourav Chakraborty. "On the Sensitivity of Cyclically-Invariant Boolean Functions". In: vol. 13. July 2005, pp. 163–167. ISBN: 0-7695-2364-1. DOI: 10.1109/CCC.2005.38.

[34] Sourav Chakraborty. "On the sensitivity of cyclically-invariant Boolean functions". In: *Discrete Mathematics & Theoretical Computer Science* Vol. 13 no. 4 (Dec. 2011). DOI: 10.46298/dmtcs.552. URL: https://dmtcs.episciences.org/552.

[35] Sourav Chakraborty, Anna Gál, Sophie Laplante, Rajat Mittal, and Anupa Sunny. "Certificate Games". In: *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Vol. 251. Leibniz International Proceedings in Informatics (LIPIcs). 2023, 32:1–32:24. ISBN: 978-3-95977-263-1. DOI: 10.4230/LIPIcs.ITCS.2023.32. URL: https://drops.dagstuhl.de/opus/volltexte/2023/17535.

[36] Elliott Ward Cheney. *Introduction to approximation theory*. McGraw-Hill Book Company, 1966.

[37] F. R. K. Chung, Zoltán Füredi, R.L Graham, and P. Seymour. "On induced subgraphs of the cube". In: *Journal of Combinatorial Theory, Series A* 49.1 (1988), pp. 180–187. ISSN: 0097-3165. DOI: https://doi.org/10.1016/0097-3165(88)90034-9. URL: http://www.sciencedirect.com/science/article/pii/0097316588900349.

[38] Fan R. K. Chung. "Subgraphs of a hypercube containing no small even cycles". In: *Journal of Graph Theory* 16.3 (1992), pp. 273–286. DOI: https://doi.org/10.1002/jgt.3190160311. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jgt.3190160311.

[39] Richard Cleve. "The Query Complexity of Order-Finding". In: *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*. COCO '00. USA: IEEE Computer Society, 2000, p. 54. ISBN: 0769506747.

[40] Richard Cleve, Peter Hoyer, Ben Toner, and John Watrous. "Consequences and limits of nonlocal strategies". In: vol. 19. July 2004, pp. 236–249. ISBN: 0-7695-2120-7. DOI: 10.1109/CCC.2004.1313847.

[41] Stephen Cook and Cynthia Dwork. "Bounds on the Time for Parallel RAM's to Compute Simple Functions". In: *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*. STOC '82. San Francisco, California, USA: Association for Computing Machinery, 1982, pp. 231–233. ISBN: 0897910702. DOI: 10.1145/800070.802196. URL: https://doi.org/10.1145/800070.802196.

[42] I. J. Dejter and P. Guan. "Square-blocking subsets in hypercubes and vertex avoidance". In: *Graph Theory, Combinatorics, Algorithms and Applications*. Philadelphia, PA: SIAM, 1991, pp. 162–174.

[43] Hartmut Ehlich and Karl Longin Zeller. "Schwankung von Polynomen zwischen Gitterpunkten". In: *Mathematische Zeitschrift* 86.1 (Feb. 1964), pp. 41–44.

[44] Paul Erdös. "Problems and results in combinatorial analysis and combinatorial number theory." In: *Graph Theory, Combinatorics and Applications* 1 (1991), pp. 397–406. ISSN: 0166-218X.

[45] D. J. Foulis and C. H. Randall. "Empirical logic and tensor products". In: *Interpretations and Foundations of Quantum Theory*. Vol. Interpretations and Foundations of Quantum Theory. 1981, pp. 1–20.

[46] Dmitry Gavinsky, Troy Lee, Miklos Santha, and Swagato Sanyal. "A Composition Theorem for Randomized Query Complexity via Max-Conflict Complexity". In: *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Vol. 132. Leibniz International Proceedings in Informatics (LIPIcs). 2019, 64:1–64:13. ISBN: 978-3-95977-109-2. DOI: 10.4230/LIPIcs.ICALP.2019.64. URL: http://drops.dagstuhl.de/opus/volltexte/2019/10640.

[47] Justin Gilmer, Michael Saks, and Srikanth Srinivasan. "Composition Limits and Separating Examples for Some Boolean Function Complexity Measures". In: *Combinatorica* 36.3 (2016), pp. 265–311. ISSN: 0209-9683. DOI: 10.1007/s00493-014-3189-x. URL: https://doi.org/10.1007/s00493-014-3189-x.

[48] N. Gisin and B. Gisin. "A local hidden variable model of quantum correlation exploiting the detection loophole". In: *Physics Letters A* 260.5 (1999), pp. 323–327. ISSN: 0375-9601. DOI: https://doi.org/10.1016/S0375-9601(99)00519-8. URL: https://www.sciencedirect.com/science/article/pii/S0375960199005198.

[49] Mika Göös, T. S. Jayram, Toniann Pitassi, and Thomas Watson. "Randomized Communication versus Partition Number". In: *ACM Trans. Comput. Theory* 10.1 (Jan. 2018). ISSN: 1942-3454. DOI: 10.1145/3170711. URL: https://doi.org/10.1145/3170711.

[50] C Gotsman and N Linial. "The equivalence of two problems on the cube". In: *Journal of Combinatorial Theory, Series A* 61.1 (1992), pp. 142–146. ISSN: 0097-3165. DOI: https://doi.org/10.1016/0097-3165(92)90060-8. URL: https://www.sciencedirect.com/science/article/pii/0097316592900608.

[51] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*. 1996, pp. 212–219.

[52] Frank Harary. "On the notion of balance of a signed graph." In: *Michigan Mathematical Journal* 2.2 (1953), pp. 143–146. DOI: 10.1307/mmj/1028989917. URL: https://doi.org/10.1307/mmj/1028989917.

[53] Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. *Variations on the Sensitivity Conjecture*. Graduate Surveys 4. Theory of Computing Library, 2011, pp. 1–27. DOI: 10.4086/toc.gs.2011.004. URL: http://www.theoryofcomputing.org/library.html.

[54] Wassily Hoeffding. "Probability Inequalities for Sums of Bounded Random Variables". In: *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30. ISSN: 01621459. URL: http://www.jstor.org/stable/2282952.

[55] Peter Hoyer, Troy Lee, and Robert Spalek. "Negative Weights Make Adversaries Stronger". In: *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. STOC '07. San Diego, California, USA: Association for Computing Machinery, 2007, pp. 526–535. ISBN: 9781595936318. DOI: 10.1145/1250790.1250867. URL: https://doi.org/10.1145/1250790.1250867.

[56] Hao Huang. "Induced subgraphs of hypercubes and a proof of the Sensitivity Conjecture". In: *Annals of Mathematics* 190.3 (2019), pp. 949–955. DOI: 10.4007/annals.2019.190.3.6. URL: https://doi.org/10.4007/annals.2019.190.3.6.

[57] Rahul Jain and Hartmut Klauck. "The Partition Bound for Classical Communication Complexity and Query Complexity". In: *Proceedings of the Annual IEEE Conference on Computational Complexity* (Oct. 2009). DOI: 10.1109/CCC.2010.31.

[58] Rahul Jain, Hartmut Klauck, Srijita Kundu, Troy Lee, Miklos Santha, Swagato Sanyal, and Jevgēnijs Vihrovs. "Quadratically Tight Relations for Randomized Query Complexity". In: *Theor. Comp. Sys.* 64.1 (Jan. 2020), pp. 101–119. ISSN: 1432-4350. DOI: 10.1007/s00224-019-09935-x. URL: https://doi.org/10.1007/s00224-019-09935-x.

[59] Mauricio Karchmer and Avi Wigderson. "Monotone Circuits for Connectivity Require Super-Logarithmic Depth." In: *SIAM J. Discrete Math.* 3 (Jan. 1990), pp. 255–265.

[60] Claire Kenyon and Samuel Kutin. "Sensitivity, block sensitivity, and l-block sensitivity of Boolean functions". In: *Information and Computation* 189.1 (2004), pp. 43–53. ISSN: 0890-5401. DOI: https://doi.org/10.1016/j.ic.2002.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0890540103002530.

[61] Subhash Khot. "On the Power of Unique 2-Prover 1-Round Games". In: *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*. STOC '02. Montreal, Quebec, Canada: Association for Computing Machinery, 2002, pp. 767–775. ISBN: 1581134959. DOI: 10.1145/509907.510017. URL: https://doi.org/10.1145/509907.510017.

[62] M. Kläy, C. H. Randall, and D. J. Foulis. "Tensor products and probability weights". In: *Int. J. Theor. Phys.* 26.3 (1987), pp. 199–219.

[63] Donald E. Knuth. *A computational proof of Huang's degree theorem*. Manuscript, 28 July, revised 3 August, https://www.cs.stanford.edu/ knuth/papers/huang.pdf. 2019. URL: https://www.cs.stanford.edu/~knuth/papers/huang.pdf.

[64] Elias Koutsoupias. "Improvements on Khrapchenko's theorem". In: *Theor. Comput. Sci.* 116.2 (1993), pp. 399–403.

[65] Raghav Kulkarni and Avishay Tal. "On Fractional Block Sensitivity". In: *Chicago Journal of Theoretical Computer Science* 2016 (2016). Article 08, pp. 1–16.

[66] Sophie Laplante, Mathieu Laurière, Alexandre Nolin, Jérémie Roland, and Gabriel Senno. "Robust Bell inequalities from communication complexity". In: *Quantum* 2 (June 2018), p. 72. ISSN: 2521-327X. DOI: 10.22331/q-2018-06-07-72. URL: https://doi.org/10.22331/q-2018-06-07-72.

[67] Sophie Laplante, Troy Lee, and Mario Szegedy. "The Quantum Adversary Method and Classical Formula Size Lower Bounds". In: *Comput. Complex.* 15.2 (2006), pp. 163–196.

[68] Sophie Laplante and Frédéric Magniez. "Lower Bounds for Randomized and Quantum Query Complexity Using Kolmogorov Arguments". In: *SIAM Journal on Computing* 38.1 (2008), pp. 46–62. DOI: 10.1137/050639090. URL: https://doi.org/10.1137/050639090.

[69] Sophie Laplante, Reza Naserasr, and Anupa Sunny. "Sensitivity Lower Bounds from Linear Dependencies". In: *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). 2020, 62:1–62:14. DOI: 10.4230/LIPIcs.MFCS.2020.62. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12732.

[70] Sophie Laplante, Reza Naserasr, Anupa Sunny, and Zhouningxin Wang. "Sensitivity Conjecture and Signed Hypercubes". In preparation.

[71] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. "Quantum Query Complexity of State Conversion". In: *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. 2011, pp. 344–353. DOI: 10.1109/FOCS.2011.75.

[72] AA Markov. "Sur une question posée par Mendeleieff". In: *IAN* 62 (1889), pp. 1–24.

[73] Jiří Matoušek and Jan Vondrák. *The Probabilistic Method Lecture Notes*. Mar. 2008.

[74] Tim Maudlin. "Bell's Inequality, Information Transmission, and Prism Models". In: *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association* 1992.1 (1992), pp. 404–417. DOI: 10.1086/psaprocbienmeetp.1992.1.192771.

[75] Gatis Midrijanis. *Exact quantum query complexity for total Boolean functions*. Tech. rep. 2004.

[76] Marvin Minsky and Seymour A. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969. ISBN: 0262130432.

[77] Rajat Mittal, Sanjay S Nair, and Sunayana Patro. *Lower bounds on quantum query complexity for symmetric functions*. Tech. rep. 2021.

[78] Michael Mitzenmacher and Eli Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[79] Ashley Montanaro. "A composition theorem for decision tree complexity". In: *Chicago Journal of Theoretical Computer Science* 2014.6 (July 2014).

[80] Noam Nisan. "CREW PRAMS and Decision Trees". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC '89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 327–335. ISBN: 0897913078. DOI: 10.1145/73007.73038. URL: https://doi.org/10.1145/73007.73038.

[81] Noam Nisan and Mario Szegedy. "On the degree of boolean functions as real polynomials". In: *Computational Complexity* 4.4 (Dec. 1994), pp. 301–313.

[82] Carlos Palazuelos and Thomas Vidick. "Survey on Nonlocal Games and Operator Space Theory". In: *Journal of Mathematical Physics* 57 (Dec. 2015).

[83] S. Pironio. "Lifting Bell inequalities". In: *J. Math. Phys.* 46 (2005), p. 062112.

[84] C. H. Randall and D. J. Foulis. "Operational statistics and tensor products". In: *Interpretations and Foundations of Quantum Theory*. 1981, pp. 21–28.

[85] Ran Raz. "A Parallel Repetition Theorem". In: *SIAM Journal on Computing* 27.3 (1998), pp. 763–803. DOI: 10.1137/S0097539795280895. URL: https://doi.org/10.1137/S0097539795280895.

[86] Ben W. Reichardt. "Reflections for Quantum Query Algorithms". In: *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '11. San Francisco, California: Society for Industrial and Applied Mathematics, 2011, pp. 560–569.

[87] Rüdiger Reischuk. *A lower time-bound for parallel random access machines without simultaneous writes*. Tech. rep. Technical Report RJ3431, IBM, New York, 1982.

[88] Theodore Joseph Rivlin and Elliott Ward Cheney. "A Comparison of Uniform Approximations on an Interval and a Finite Subset Thereof". In: *SIAM Journal on Numerical Analysis* 3.2 (1966), pp. 311–320. ISSN: 00361429. URL: http://www.jstor.org/stable/2949624.

[89] David Rubinstein. "Sensitivity vs. block sensitivity of Boolean functions". In: *Combinatorica* 15.2 (June 1995), pp. 297–299. ISSN: 1439-6912. DOI: 10.1007/BF01200762. URL: https://doi.org/10.1007/BF01200762.

[90] Yaoyun Shi and Yufan Zhu. "Tensor Norms and the Classical Communication Complexity of Nonlocal Quantum Measurement". In: *SIAM Journal on Computing* 38.3 (2008), pp. 753–766. DOI: 10.1137/050644768. URL: https://doi.org/10.1137/050644768.

[91] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM J. Comput.* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 0097-5397. DOI: 10.1137/S0097539795293172. URL: https://doi.org/10.1137/S0097539795293172.

[92] Hans-Ulrich Simon. "A tight Ω(loglog n)-bound on the time for parallel Ram's to compute nondegenerated boolean functions". In: *Foundations of Computation Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 439–444.

[93] Robert Špalek and Mario Szegedy. "All Quantum Adversary Methods are Equivalent". In: *Theory of Computing* 2.1 (2006), pp. 1–18. DOI: 10.4086/toc.2006.v002a001. URL: http://www.theoryofcomputing.org/articles/v002a001.

[94] Michael Steiner. "Towards quantifying non-local information transfer: finite-bit non-locality". In: *Physics Letters A* 270.5 (2000), pp. 239–244. ISSN: 0375-9601. DOI: https://doi.org/10.1016/S0375-9601(00)00315-7. URL: https://www.sciencedirect.com/science/article/pii/S0375960100003157.

[95] Avishay Tal. "Properties and Applications of Boolean Function Composition". In: *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*. ITCS '13. Berkeley, California, USA: ACM, 2013, pp. 441–454. ISBN: 978-1-4503-1859-4. DOI: 10.1145/2422436.2422485. URL: http://doi.acm.org/10.1145/2422436.2422485.

[96] B. F. Toner and D. Bacon. "Communication Cost of Simulating Bell Correlations". In: *Phys. Rev. Lett.* 91 (18 Oct. 2003), p. 187904. DOI: 10.1103/PhysRevLett.91.187904. URL: https://link.aps.org/doi/10.1103/PhysRevLett.91.187904.

[97] Uzi Vishkin and Avi Wigderson. "Trade-offs between depth and width in parallel computation". In: *SIAM J. Discrete Math.* 14 (2 1985), pp. 303–314.

[98] A. Wilce. "Tensor products in generalized measure theory". In: *Int. J. Theor. Phys.* 31.11 (1992), pp. 1915–1928.

[99] Alex Yu. *Boolean Function Complexity Measures*. https://funcplot.com/table/. Adapted from [3]. 2019.

[100] Thomas Zaslavsky. "Signed graphs". In: *Discrete Applied Mathematics* 4.1 (1982), pp. 47–74. ISSN: 0166-218X. DOI: https://doi.org/10.1016/0166-218X(82)90033-6. URL: http://www.sciencedirect.com/science/article/pii/0166218X82900336.

[101] Thomas Zaslavsky. "Signed graphs and geometry". English. In: *J. Comb. Inf. Syst. Sci.* 37.2-4 (2012), pp. 95–143. ISSN: 0250-9628.

[102] Shengyu Zhang. "On the power of Ambainis lower bounds". In: *Theoretical Computer Science* 339.2 (2005), pp. 241–256. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2005.01.019. URL: https://www.sciencedirect.com/science/article/pii/S0304397505001234.