KUNGL
TEKNISKA
HÖGSKOLAN

On Probabilistic Proof Systems and Hardness of Approximation

Jonas Holmerin

Stockholm 2002

Doctoral Dissertation

Stockholm University

Department of Numerical Analysis and Computer Science

Akademisk avhandling som med tillstånd av Stockholms universitet framlägges till offentlig granskning för avläggande av filosofie doktorsexamen 9 December 2002 i Kollegiesalen, Valhallavägen 79.

ISBN 91-7283-395-5
TRITA-NA-0224
ISSN 0348-2952
ISRN KTH/NA/R--02/24--SE

Universitetsservice US AB, Stockholm 2002

Abstract

In this thesis we study the approximability of combinatorial optimization problems whose decision versions are **NP**-complete. These problems cannot be solved exactly in polynomial time, unless $\mathbf{P} = \mathbf{NP}$. However, it may be possible to solve them approximately in polynomial time, i.e., there might exist a polynomial time algorithm which produces a solution whose value is always close to the optimum.

An important question is then: how close to optimum can we get in polynomial time?

For a maximization problem, we say that an algorithm approximates within a factor c if it produces a solution where the optimum is at most c times the value of the produced solution, and for a minimization problem we say that an algorithm approximates within c if it finds a solution whose value is at most c times the optimum solution.

The focus of this thesis is to use probabilistic proof systems to prove, for various problems, that if we can approximate within a certain factor, then $\mathbf{P} = \mathbf{NP}$ (or in other words, it is **NP**-hard to approximate within this factor).

The connection between lower bounds for approximation and so called *probabilistically checkable proofs* or PCPs was discovered by Feige, Goldwasser, Lovász, Safra and Szegedy. In the PCP model, a proof is checked by a probabilistic verifier which only looks at a few randomly chosen positions in the proof. The famous PCP theorem by Arora, Lund, Motwani, Sudan and Szegedy gives us a starting point for proving **NP**-hardness of approximation results. Håstad introduced techniques which make it possible to prove strong lower bounds by using the Fourier transform as a tool to analyze PCPs.

In this thesis, we extend the techniques of Håstad to prove that it is **NP**-hard to approximate several problems. Among the problems we study is a generalization of the Minimum Bisection problem. We prove that this generalization is **NP**-hard to approximate within any constant.

We also study Minimum Vertex Cover generalized to k-uniform hypergraphs, For this problem, we prove for any $\epsilon > 0$, the problem is **NP**-hard to approximate within $\Omega(k^{1-\epsilon})$. When $k = 3$ and $k = 4$ we prove that the problem is **NP**-hard to approximate within $1.5 - \epsilon$ and $2 - \epsilon$ respectively, for any $\epsilon > 0$.

On a slightly different note, we look at what parameters are possible to achieve for a certain class of PCPs. More precisely, we look at PCPs that have perfect completeness (i.e., a correct proof is always accepted) and read three positions from the proof. For the case when the proof consists of values from a domain of size d for some integer constant $d \geq 2$, for any constant $\epsilon > 0$ we construct a PCP with perfect completeness and soundness $d^{-1} + d^{-2} + \epsilon$ (i.e., incorrect proofs are accepted with at most this probability),

Finally we consider the problem of simultaneously satisfying a family of equations over any fixed finite group G and show that it is **NP**-hard to approximate the number of simultaneously satisfiable equations to within $|G| - \epsilon$ for any $\epsilon > 0$.

ISBN 91-7283-395-5 • TRITA-NA-0224 • ISSN 0348-2952 • ISRN KTH/NA/R--02/24--SE

Sammanfattning

Ämnet för denna avhandling är approximerbarhet hos kombinatoriska optimeringsproblem som är **NP**-fullständiga i sin beslutsvariant. Sådana problem kan inte lösas exakt i polynomisk tid om inte $\mathbf{P} = \mathbf{NP}$. Däremot kan det vara möjligt att lösa dem approximativt i polynomisk tid, det vill säga det kan mycket väl finnas en polynomisk algoritm som alltid hittar en lösning vars värde ligger nära optimum.

Frågan som ställer sig är då: hur nära optimum kan vi komma i polynomisk tid?

För maximeringsproblem säger vi att en algoritm approximerar inom en faktor c om den hittar en lösning där maximum är högst c gånger större än värdet av den funna lösningen. För ett minimeringsproblem säger vi att en algoritm approximerar inom en faktor c om värdet av den funna lösningen är högst c gånger större än minimum.

Denna avhandling fokuserar på att visa *undre gränser* med hjälp av probabilistiska bevissystem. Att visa en undre gräns innebär att visa att om man kan approximera inom en viss faktor i polynomisk tid så är $\mathbf{P} = \mathbf{NP}$. Vi säger att det är **NP**-svårt att approximera inom denna faktor.

Att det finns ett samband mellan undre gränser för approximerbarhet och så kallade *probabilistiskt verifierbara bevis* upptäcktes av Feige m.fl. [18]. Ett probabilistiskt verifierbart bevis (den engelska förkortningen är PCP) är ett bevis som kan kontrolleras av en verifierare som slumpmässigt tittar på några få positioner i beviset, och därefter bestämmer sig för att förkasta eller acceptera beviset.

Den berömda PCP-satsen som bevisades av Arora m.fl. [2] ger oss en första byggsten för att kunna bevisa att det är **NP**-svårt att approximera problem. Håstad [27, 28] har utvecklat en teknik att med hjälp av den diskreta fouriertransformen analysera verifieraren för ett probabilistiskt verifierbart bevis. På detta sätt kan man visa mycket starka undre gränser.

I denna avhandling bevisar vi att det är **NP**-svårt att approximera diverse problem med hjälp av utvidgningar av Håstads teknik. Bland de problem vi studerar är en generalisering av problemet minimalt balancerat snitt. Vi visar att det är **NP**-svårt att approximera denna generalisering inom varje konstant.

Ett annat problem vi studerar är hörntäckning generaliserat till k-likformiga hypergrafer. Vi visar att för varje $\epsilon > 0$ så är problemet **NP**-svårt att approximera inom $\Omega(k^{1-\epsilon})$. Vi betraktar också fallen $k = 3$ och $k = 4$, och visar att dessa problem är **NP**-svåra att approximera inom respektive $1.5 - \epsilon$ och $2 - \epsilon$, för varje $\epsilon > 0$.

Ett något annorlunda problem som vi behandlar är vilka parametrar som är möjliga att få för en verifierare som har perfekt fullständighet (d.v.s. ett korrekt bevis måste alltid accepteras) och enbart tittar på 3 positioner i beviset. I fallet när beviset består av en tabell med d möjliga värden i varje position, så visar vi att för varje $\epsilon > 0$ går det att få sundhet $d^{-1} + d^{-2} + \epsilon$ (d.v.s. ett inkorrekt bevis accepteras högst med denna sannolikhet).

Slutligen betraktar vi problemet att givet en mängd ekvationer över en godtycklig ändlig grupp G samtidigt satisfiera så många ekvationer som möjligt. Vi visar

att det är **NP**-svårt att approximera detta problem inom $|G|^{-1} + \epsilon$, för varje $\epsilon > 0$.

ISBN 91-7283-395-5 • TRITA-NA-0224 • ISSN 0348-2952 • ISRN KTH/NA/R--02/24--SE

Acknowledgments

First of all I would like to thank my supervisor, *Johan Håstad*. He is always a source of good ideas, quick counterexamples, and constructive criticism.

Among Johan's many good qualities is the ability of being able to understand, extremely quickly, the most incoherent explanation of a proof. Sometimes, however, it is useful to try to explain your ideas to someone who does not know your area of research at least twice as well as yourself. In this regard, I would like to thank *Mikael Goldmann* and *Lars Engebretsen* for listening to my ramblings.

Lars is also co-author of two of the papers on which this thesis is based, and I would like to thank him and my other co-authors, *Subhash Khot* and *Alex Russell* for collaborating with me on the corresponding projects.

The theory group at NADA has been a stimulating environment in which to work. For this I would like to thank the faculty and past and present PhD students. In particular I would like to thank my current and former office mates: *Gunnar Andersson*, *Johnny Bigert*, *Joel Brynielsson*, *Johan Carlberger*, *Lars Engebretsen* (again), *Anna Redz* and *Staffan Ulfberg*.

Contents

Chapter 1

Introduction

What problems can be efficiently solved by computer? It is well known that many problem can be solved very efficiently by computer, for example a computer can multiply (even very large) numbers quickly, or search through large amounts of text in a short amount of time. However, many important problems appear difficult to solve efficiently by computer.

Let us give an example of a hard problem. When developing software one often wants to reuse previously written code. This may lead to a situation in which one has a large collection of software components which implement possibly overlapping sets of features. Now suppose that we have a specification of which features are required, and we want to use as few software components as possible to implement this specification.

The problem can be formalized as follows: We have a set S of components and for a given specification we form for each feature f which is required by the specification a set S_f which contain all the components which implement that feature. We now want to find a set of components of minimum size such that we have at least one component from each set S_f, i.e., that all the features which we want are implemented by at least one component. How do we find a set of components of minimum size? One approach is to try all possible subsets of S. There are, however, $2^{|S|}$ of those, so for large S this approach is infeasible. If $|S| = 100$, a fairly small instance, there are 1267650600228229401496703205376 different subsets of S, which is far to many to try them all.

The problem appears infeasible to solve by computer. However, it may be the case that if we think about it enough, we might come up with a way to solve the problem efficiently. On the other hand, it might be the case that there is no better way than trying all the sets. If there appears to be no way of solving the problem efficiently, perhaps we may settle on a reasonably good solution instead of the very best solution. In the example above, perhaps we are satisfied with a set which is at most 10% larger than the optimal size. Of course we would like to be as close to optimal as possible. The natural question then is: "how close to optimal can we

get by an efficient algorithm". This is the kind of question we study in this thesis.

1.1 NP-completeness

When we believe we have found a hard computational problem, we would like to be able to prove that this is so. That is, we would like to prove that any method of solving a certain problem must take a certain number of computational steps. However, so far theoretical computer science has made very little progress towards proving any useful results of this kind. The theory of ***NP-completeness*** gives us an alternative.

A *decision problem* is a problem where the answer is "yes" or "no". For example, consider the problem of finding a optimal set of components which we described in the previous section. This problem is known as the *Minimum Hitting Set* problem.

As formulated this is an optimization problem; we want to find a solution which minimizes the value of the object function (i.e., the number of elements in the solution). However, there is a decision variant of the problem in which are given a goal K, and the question is whether or not there is a solution with at most K elements.

Suppose someone claims that the answer is "yes", there is a solution of size K, and he hands us a set T of components which he claims contains K elements and implements all the features in the specification, i.e., contains at least one element from each of the sets S_f. Of course we can then easily tell if he is lying: just check for each set S_f that there is some element from S_f in T, count the number of elements in T and check if they were more than K.

Informally, **NP** is the class of decision problems where checking a claim that the answer is "yes" can be done efficiently if we are given access to a solution, while **P** is the class of problems where finding out for yourself what the answer is can be done efficiently.

What do we mean by "efficiently"? The definition we use is that as a function of the number of bits in the input, the running time should be bounded by a polynomial. That is, if the number of bits used to represent the input is n, then for some constants c and k, the number of steps performed by the algorithm should be at most cn^k.

The most important open question in theoretical computer science is whether $\mathbf{P} = \mathbf{NP}$. It is widely believed that it is the case that $\mathbf{P} \neq \mathbf{NP}$, and a large body of work in theoretical computer science, including this thesis, is done under the assumption that $\mathbf{P} \neq \mathbf{NP}$. There are several reasons to believe that such an assumption is reasonable. For one thing, intuitively it *should* be much easier to verify a solution than to find the solution. Also, many important problems in **NP** has been studied extensively, without any efficient algorithms being found.

A problem is said to be ***NP-complete*** if it is among the hardest problems in **NP**. If we have a polynomial time algorithm for any **NP**-complete problem, we can solve *any* problem in **NP** in polynomial time. Thus for any **NP**-complete

problem to be in **P** would imply **P** = **NP**. The notion of **NP**-completeness was introduced by Cook who in a breakthrough paper [7] defined the class **NP** and **NP**-completeness and proved that the satisfiably problem for boolean formulas is **NP**-complete. Several problems, including the Minimum Hitting Set problem, were proven **NP**-complete by Karp [36]. Since then, extremely many problems have been proven **NP**-complete, see for example the book by Garey and Johnson [20] for a long list of **NP**-complete problems.

We say that a problem, decision problem or not, is ***NP-hard*** if a polynomial time algorithm for this problem would imply **P** = **NP**. For example, the optimization version of Minimum Hitting Set is **NP**-hard, since if we can compute the minimum, we can solve the decision version.

1.2 Approximation algorithms

How do we cope with **NP**-hard problems? For an optimization problem, the news that it is hard to find the best solution should not lead us to despair completely. It might be possible that all we need is a solution that is good enough. This naturally leads us to the concept of *approximation algorithms*. An approximation algorithm is an algorithm which outputs a solution which is guaranteed to be within a certain factor of the optimal solution. Specifically, for a minimization problem we say that an algorithm approximates within a factor c if it always finds a solution which is at most c times larger than the optimum solution, and for a maximization problem we say that the algorithm approximates within a factor c if the algorithm always finds a solution where the optimum solution is at most c times larger.

By this definition, $c \geq 1$ and we would of course like an algorithm which approximates within a factor which is as close to 1 as possible. Thus an important question to ask is, given an **NP**-hard optimization problem, how well this problem can be approximated in polynomial time. A somewhat surprising fact is that even for the class of optimization problems where the decision version is **NP**-complete, and where the exact solution in some sense is equally difficult to find, the approximability may vary wildly.

There are two parts to establishing a problems approximability. We need to establish that it can be approximated within a certain factor, and that it is not possible to do better.

To get an *upper bound*, i.e., to prove that for some c, we can approximate within this factor c, we exhibit an algorithm and prove that it runs in polynomial time and produces a solution within the required factor.

To get a *lower bound* we should like to prove that no polynomial time algorithm can approximate the problem better than a certain factor. However, we do not know how to prove this kind of lower bound, and instead we settle for proving that if there is an algorithm which approximates the problem better than a certain factor, then **P** = **NP**. For optimization problems where the decision version is **NP**-complete, this actually means that if we can approximate the problem within a certain factor

in polynomial time, then we can solve the problem exactly in polynomial time as well.

1.2.1 Algorithmic techniques

Although the main focus of this thesis is proving **NP**-hardness results for approximating various problems, we briefly review some techniques for constructing approximation algorithms.

One simple way of constructing approximation algorithms which works surprisingly well is to pick a solution at random and hope for the best. Of course, not all problems are possible to attack in this way, since the solution space may be hard to describe (and thus difficult to sample). A surprising fact is that for many problems, picking a solution at random is essentially the best possible algorithm, in that it is **NP**-hard to approximate within a better constant than that achieved by the randomized algorithm.

A more sophisticated way of constructing approximation algorithms is to *relax* the problem to another problem which may be solved efficiently. Many optimization problems can be stated as integer programs. If this integer program is linear, the program can be relaxed to a linear program over the real numbers, which can be solved in polynomial time. Since relaxation means that we expand the solution space, the solution to the linear program may not be a solution to the original integer program, so we need to somehow round the solution back to an integer solution without losing too much of the quality of the solution.

A similar use of relaxation which has been used very successfully in recent years is the relaxation of a quadratic integer program to a semidefinite program, which is solved in polynomial time and then rounded back to an integer solution. This technique was introduced by Goemans and Williamson in the breakthrough paper [21] in which they proved that the Maximum Cut problem can be approximated within 1.139. Semidefinite programming has since been used to get improved approximation algorithms for many problems, for example graph coloring [35], and several constraint satisfaction problems [54].

1.2.2 Techniques for proving lower bounds

The fact that we can prove lower bounds for approximation under the assumption $\mathbf{P} \neq \mathbf{NP}$ and so relate exact computation to approximate computation is somewhat surprising. Indeed, prior to the early 90's, work in the field of approximability mostly consisted of classifying problems into equivalence classes by their approximability properties.

However, the discovery of the surprising power of *interactive proof systems* lead to new techniques which made it possible to prove **NP**-hardness results. An interactive proof system may be viewed as a game between two players, the prover, which is computationally unbounded, and the verifier, which is a polynomial time algorithm with access to a source of random bits. The prover wants to convince the

verifier of some fact. The verifier makes queries to the prover and wants to decide probabilistically whether this fact is true. We want this game to have the property that if the prover is telling the truth, he always convinces the verifier, but if he is lying, then there is a good probability that he is caught. The probability should be only over the random bits in the verifier.

We define the class of problems **IP** which consist of decision problems which admits interactive proofs. That is, the prover tries to convince the verifier that the answer is "yes", and if this is the case then the prover always convinces the verifier, and if this is not is the case, then the verifier is convinced with probability at most $1/2$.

Shamir, building on work by Lund et al. [42], showed in a breakthrough paper that **IP** = **PSPACE** [48], that is, all the decision problems which may be solved using a polynomial amount of memory, admits interactive proofs, and vice versa. Following this result, Babai et al. further established the power of interactive proof by showing that **MIP** = **NEXP** [4], i.e., that the decision problems which admits probabilistic proof systems where the verifier interacts with multiple provers are the same which have the property that "yes" solutions may be verified in exponential time in the input to the decision problem.

Another model of proof checking is that of probabilistically checkable proofs, or PCPs. Instead of the verifier interacting with the prover, the prover writes down the entire proof in a large table. The verifier is a probabilistic polynomial time algorithm which has random access to this table. The verifier checks the proof by looking at a few randomly chosen positions in the proof, and then decides whether to accept or reject. Parameters of interest for such a proof system are the number of positions queried in the proof, the number of random bits used by the verifier, and the probabilities with which correct and incorrect proofs are accepted (the *completeness* and *soundness* of the proof system).

It was discovered by Feige et al. [18] that this kind of proof system is connected to approximability. Using scaled down techniques from [4], they derived a PCP system for **NP** and used this characterization of **NP** to prove that approximating the Maximum Clique problem within any constant is almost **NP**-hard.

Following the discovery of this connection, improved characterizations of **NP** in terms of the parameters of the proof system were proven by Arora and Safra [3] and by Arora et al. [2], who proved that any problem in **NP** admits a PCP where the verifier uses $O(\log n)$ random bits and looks only at a constant number of positions in the proof. The first of these results lead to that the **NP**-hardness of approximating Maximum Clique within any constant could be established, and the second result lead to that the hardness result for Maximum Clique could be further strengthened and it also implies many other hardness of approximation results.

It turns out to get strong hardness of approximation result it is often useful to design a specific proof system for **NP** with a specific problem in mind. This technique was introduced by Bellare et al [5] and has since been refined and used to get improved lower bounds, in many cases matching the best known algorithms (and thus proving the optimality of this algorithm). A few notable papers are the

paper by Feige [16], which proves that the Minimum Hitting Set problem which we described earlier is "almost" **NP**-hard to approximate within $\ln n$ where n is the number of sets, and the papers by Håstad which proves an optimal result for Maximum Clique [27] and for the Maximum E3-Sat problem and several other important problems [28]. We also mention a relatively new result by Dinur and Safra [10] which improves the lower bound for the Minimum Vertex Cover problem using several new ideas which may prove influential.

The only general techniques which we know for proving inapproximability is either, as mentioned above, to invent a special purpose PCP, or to *reduce* a problem which is known to be hard to the problem at hand. This means that we want to translate the hard problem into our problem in such a way that we can use an approximation algorithm for our problem to approximate the hard problem. Since the hard problem was hard to approximate, so must our problem be as well. Using reductions, one can prove lower bounds for many problems. However, for many problems no really strong lower bounds were achieved until a special purpose PCP was constructed.

For some problems it appears difficult to construct special purpose PCPs; if this is due to limitations in current techniques or due to more fundamental reasons is not clear. Hence it makes sense to try to find the best possible reductions. A general technique for constructing optimal reductions using local substructures, so called *gadgets*, was found by Trevisan et al. [53] who used linear programming to find optimal gadgets, which when combined with the results of Håstad [28] made it possible to get improved lower bounds for several problems.

1.3 Topic and Organization of this thesis

The focus of this thesis is proving lower bounds on the approximability of various problems through the use of probabilistic proof systems.

Part I provides mathematical background and introduces techniques which we use throughout the thesis.

Part II contains the new result.

In Chapter 6 we prove that a generalization of the Minimum Bisection problem cannot be approximated within any constant in polynomial time, unless $\mathbf{P} = \mathbf{NP}$. In the Minimum Bisection problem, we are given a graph and asked to partition the vertices into two equal sized parts so as to minimize the number of edges between the parts. This chapter is based on a previously unpublished note co-authored with Subhash Khot [33].

In Chapters 7, 8 and 9 we consider vertex cover generalized to k-uniform hypergraphs. This problem can be viewed as the Minimum Hitting Set problem where all the sets have size exactly k.

In Chapter 7, we prove that, for any $\epsilon > 0$, vertex cover on k-uniform hypergraphs cannot be approximated within $\Omega(k^{1-\epsilon})$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$.

In Chapter 8 we prove that for $k = 3$, for any $\epsilon > 0$ the problem cannot be approximated within $1.5 - \epsilon$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$. Chapters 7 and 8 are based on the paper [31].

In Chapter 9 the main result is that for any $\epsilon > 0$ vertex cover on 4-uniform hypergraphs cannot be approximated within $2 - \epsilon$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$. This chapter is based on the paper [32].

In Chapter 10 we study PCPs with perfect completeness (i.e., a correct proof must be accepted with probability 1) where the answers from the proof is from a domain of size $d > 2$ and achieve the best currently known soundness for a PCP making three queries. This chapter is based on a paper co-authored with Lars Engebretsen [14].

Finally, in Chapter 11, we show that for any finite group G, the problem of given a system of equation over G finding an assignment which satisfies the maximum number of equations is cannot be approximated within $|G|^{-1} + \epsilon$ in polynomial time unless $\mathbf{P} = \mathbf{NP}$. This chapter is based on a paper co-authored with Lars Engebretsen and Alexander Russell [15].

Part I

Background

Chapter 2

Preliminaries

2.1 Mathematical background

In this section we give mathematical background and definitions which we use throughout the thesis.

2.1.1 Basic inequalities

Proposition 2.1 The Cauchy–Schwartz inequality. For any complex numbers $\{a_1, \ldots, a_n\}$ and $\{b_1, \ldots, b_n\}$,

$$\left|\sum_{i=1}^{n} a_i b_i\right|^2 \leq \left(\sum_{i=1}^{n} |a_i|^2\right)\left(\sum_{i=1}^{n} |b_i|^2\right).$$

As a special case, corresponding to $b_i = 1$,

$$\left|\sum_{i=1}^{n} a_i\right|^2 \leq n \sum_{i=1}^{n} |a_i|^2.$$

Proposition 2.2 Markov's inequality. If X is a random variable which takes only non-negative values, then for any $a > 0$

$$\Pr[X \geq a] \leq \frac{\mathrm{E}[X]}{a}$$

Proposition 2.3 Jensen's inequality. If $f(x)$ is a convex function, then

$$\mathrm{E}[f(x)] \geq f(\mathrm{E}[X])$$

2.2 Problems considered

There are two different definitions for algorithm what is means for an algorithm to solve an optimization problem. Either we deem it sufficient if we know what the value of the optimum solution is, or we require the algorithm to actually give us solution which achieves the optimum value. A lower bound is stronger if it holds for algorithms of the first kind, while an algorithm is of course is better if it produces a solution. Since we only prove lower bounds in this thesis, we use the first definition.

The first group of problems are all variants on the problem of given a system of linear equations over $\boldsymbol{Z}_d$, finding an assignment which satisfies as many equations as possible.

Definition 2.4. Maximum Lin-2 is the following maximization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$ find the maximum number of simultaneously satisfiable equations.

Definition 2.5. Maximum Ek-Lin-2 is the following maximization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$, each equation having exactly k variables (i.e., k non-zero coefficients), find the maximum number of simultaneously satisfiable equations.

Definition 2.6. An assignment to a set of n $\{0,1\}$-variables $x_1, \ldots, x_n$ is *p-balanced* if exactly a fraction p of the variables are 1. We say that an assignment is *balanced* if it is $1/2$-balanced.

Definition 2.7. Maximum Balanced-Lin-2 is the following maximization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$ with all right hand sides 0, find the maximum number of equations satisfied by balanced assignments to the variables.

Definition 2.8. Minimum UnSat Balanced-Lin-2 is the following minimization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$, with all right hand sides 0, find the minimum number of equations unsatisfied by balanced assignments to the variables.

Definition 2.9. Maximum Ek-Balanced-Lin-2 is the following maximization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$ each containing exactly k variables and with all right hand sides 0, find the maximum number of equations satisfied by balanced assignments to the variables.

Definition 2.10. Minimum Ek-UnSat Balanced-Lin-2 is the following minimization problem: Given a system of m linear equations on n variables over $\boldsymbol{Z}_2$ each containing exactly k variables and with all right hand sides 0, find the minimum number of equations unsatisfied by balanced assignments to the variables.

The problem of solving sets of equations can be generalized to arbitrary groups, as follows: an *equation* in variables $x_1, \ldots, x_n$ over a group G is an expression of

form $w_1 \dots w_k = 1_G$, where each w_i is either a variable, an inverted variable, or a group constant and 1_G denotes the identity element. A *solution* is an assignment of the variables to values in G that realizes the equality.

Definition 2.11. For any finite group G, EQ_G is the following maximization problem: Given a system of m equations on n variables over G, find the maximum number simultaneously satisfiable equations.

Definition 2.12. For any finite group G, EQ_G^1 is the following maximization problem: Given a system of m equations on n variables over G, where each variable occurs at most once in each equation, find the maximum number simultaneously satisfiable equations.

Definition 2.13. For any finite group G, $\mathrm{EQ}_G^1[k]$ is the following maximization problem: Given a system of m equations on n variables over G, where each variable occurs at most once in each equation and each equation contains exactly k variables, find the maximum number simultaneously satisfiable equations.

We next define some problem having to do with the satisfiability of logical formulas.

Definition 2.14. A propositional formula ϕ is on *conjunctive normal form* (CNF) if it is on the form $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$, where each *clause* C_i is on the form $C_i = (l_1 \vee l_2 \vee \ldots \vee l_{k_i})$ where each l_i is a *literal*, i.e., either l_i is a variable or a negated variable. A variable may not occur both negated and non-negated in the same clause.

Definition 2.15. A formula ϕ is a Ek-CNF formula if it is on conjunctive normal form, and all clauses contain exactly k distinct literals.

Definition 2.16. A formula ϕ is a Ek-CNF(l) formula if it is a Ek-CNF formula, and furthermore, all variables occur in exactly k clauses.

Definition 2.17. Sat is the following decision problem: given a CNF formula ϕ, is ϕ satisfiable or not?

Definition 2.18. E3-Sat is the following maximization problem: Given a E3-CNF formula ϕ, find the maximum number of clauses in ϕ which are simultaneously satisfiable.

The last group of problems are problems on graphs and hypergraphs.

Recall that an undirected graph $G = (V, E)$ consist of a set of vertices V and a set of edges E where each $e \in E$ is a subset of the vertices of size 2. A *vertex cover* of G is a set $S \subseteq V$ such that $S \cap e \neq \emptyset$ for each $e \in E$.

Definition 2.19. Minimum Vertex Cover is the following minimization problem: given a graph G, find the size of the smallest vertex cover of G.

Definition 2.20. A *hypergraph* consist of consists of a set of vertices, V and a set of edges E. An edge $e \in E$ is a subset of the vertices.

Definition 2.21. A hypergraph is *k-uniform* if each $e \in E$ is a set of size k.

Thus an undirected graph is a 2-uniform hypergraph.

Definition 2.22. Let $H = (V, E)$ be a k-uniform hypergraph. A *vertex cover of* H is a subset S of the vertices such that for all edges $e \in E$, $e \cap S \neq \emptyset$.

For any subset S of the vertices, we say that an edge e is *covered by* S if $S \cap e \neq \emptyset$.

Definition 2.23. Let $H = (V, E)$ be a k-uniform hypergraph. An independent set of H is a subset S of the vertices such that no edge $e \in E$ is contained entirely in S.

Thus the complement of an independent set is a vertex cover.

Definition 2.24. Let $H = (V, E)$ be be a k-uniform hypergraph. A r-coloring of H is a function $f : V \to \{1, \ldots, r\}$ such that for no $e \in E$, f takes the same value on all elements of e.

Definition 2.25. Minimum Ek-Vertex Cover is the following minimization problem: given a k-uniform hypergraph H, find the size of the smallest vertex cover of H.

Definition 2.26. Maximum Ek-Vertex Cover is the following maximization problem: given a k-uniform hypergraph H, in which we allow E to be a multiset, and an integer $B > 0$, find the maximum number of edges covered by a subset $S \subseteq V$ where $|S| = B$.

Definition 2.27. Maximum Ek-Vertex Cover-p is the following maximization problem: given a k-uniform hypergraph $H = (V, E)$, in which we allow E be a multiset, find the maximum number of edges covered by a subset $S \subseteq V$ where $|S| = p|V|$.

2.3 Optimization and approximation

Definition 2.28. Let P be a maximization problem. For an instance x of P let $\text{opt}(x)$ be the optimal value. A solution y, with weight $w(x, y)$, is *c-approximate* if it is feasible and $w(x, y) \geq \text{opt}(x)/c$.

Definition 2.29. Let P be a minimization problem. For an instance x of P let $\text{opt}(x)$ be the optimal value. A solution y, with weight $w(x, y)$, is *c-approximate* if it is feasible and $w(x, y) \leq c \cdot \text{opt}(x)$.

Definition 2.30. A *c-approximation algorithm* for an optimization problem is a polynomial time algorithm that for any instance x of the problem and any input y outputs a c-approximate solution.

We use the wording *to approximate within c* as a synonym for *to compute a c-approximate solution.*

Chapter 3

Probabilistically Checkable Proofs

A language L is in the class **NP** if there exists a polynomial time Turing machine M, with the properties that

1. For $x \in L$, there exists a proof π, of size polynomial in $|x|$, such that M accepts (x, π).
2. For $x \notin L$, M does not accept (x, π) for any proof π of size polynomial in $|x|$.

Arora and Safra [3] used a generalization of the above definition of **NP** to define the class $\mathbf{PCP}[r, q]$ (which was implicit in previous work), consisting of languages which have witnesses (or membership proofs) that can be checked by a probabilistically verifier which has oracle access to the membership proof, is allowed to use r random bits, and allowed to query q bits from the oracle.

Definition 3.1. A probabilistic polynomial time Turing machine V with oracle access to π is an (r, q, d)-*restricted verifier* if it, for every oracle π and every input of size n, uses at most $r(n)$ random bits, makes at most $q(n)$ queries to the oracle, and the answers from the oracle are from a domain of size d. We denote by V^π the verifier V with the oracle π fixed.

We say that V is *non-adaptive* if the queries V makes only depend on the input x and the random string ρ. More formally,

Definition 3.2. A (r,q,d)-restricted verifier V is *non-adaptive* if it works as follows: Let V_1 and V_2 be deterministic polynomial time Turing machines which makes no oracle queries. The verifier V on input x and oracle π produces a random string σ of length at most r, runs $V_1(x, \sigma)$ which produces a sequence of queries $p_1, \ldots, p_q$. Then V queries the oracle π in positions $p_1, \ldots, p_q$, runs $V_2(\pi(p_1), \ldots, \pi(p_q), \sigma)$, accepts if V_2 accepts and rejects otherwise.

Definition 3.3. A language L belongs to the class $\mathbf{PCP}_{c,s}[r, q, d]$ if there exists an (r, q, d)-restricted verifier V with the properties that

1. For $x \in L$, $\Pr_\rho[V^\pi \text{ accepts } (x, \rho)] \geq c$ for some oracle π.

2. For $x \notin L$, $\Pr_\rho[V^\pi \text{ accepts } (x, \rho)] < s$ for all oracles π.

where ρ is the random string of length r.

We call c and s the completeness and the soundness of the verifier, respectively. When $c = 1$, we say that the verifier has *perfect completeness*.

Definition 3.4. A language L belongs to the class $\mathbf{naPCP}_{c,s}[r, q, d]$ if all requirements of Definition 3.3 holds, and furthermore the verifier V is non-adaptive.

As a shorthand, we write $\mathbf{PCP}[q, r]$ for $\mathbf{PCP}_{1,1/2}[q, r, 2]$ and $\mathbf{naPCP}[q, r]$ for $\mathbf{naPCP}_{1,1/2}[q, r, 2]$.

We need one other kind of probabilistic proof system:

Definition 3.5. A two-prover one-round (2P1R) game consists of a probabilistic polynomial time verifier V and two computationally unbounded provers P_1 and P_2. The verifier V is allowed to send one query to each prover, which can be query for several bits.

A 2P1R game for a language L has completeness c and soundness s if the following holds:

- If $x \in L$ then there is two provers P_1 and P_2 such that Pr[V accepts] $\geq c$.

- If $x \notin L$ then for any two provers P_1 and P_2, that Pr[V accepts] $< s$.

The probabilities are over the random bits of the verifier.

It is sometimes convenient to allow the provers to be probabilistic as well. We note that probabilistic provers can always be converted to deterministic provers since it it always possible for the provers to fix their random bits in such as way as to not decrease the acceptance probability of the verifier.

There is a strong connection between PCPs and hardness of approximation for the following reason: A PCP naturally gives rise to a *proof optimization problem*, namely the problem of given an input x, compute the proof π which maximizes the probability that the verifier accepts.

Suppose we prove that some language $L \in \mathbf{PCP}_{c,s}[r, q]$. Then when given input x, we know that if $x \in L$, the best proof has probability c of being accepted, while if $x \notin L$, the best proof has at most probability s of being accepted. Now suppose there is an approximation algorithm for the proof optimization problem of this PCP which approximates within c/s. We get two cases:

1. $x \in L$. The approximation algorithm returns a value which is $\geq \frac{c}{c/s} = s$.

2. $x \notin L$. The approximation algorithm must return a value $< s$.

So we can distinguish between these two cases and thus decide L. In particular, if L is an **NP**-complete problem, the proof optimization problem is **NP**-hard to approximate within c/s. Thus a result on the form that $\mathbf{NP} \subseteq \mathbf{PCP}_{c,s}[r,q]$ gives us a **NP**-hard optimization problem which we can reduce from. It turns out that having the parameter $r = O(\log n)$ is important for making such reductions run in polynomial time.

The connection between PCPs and approximability was first discovered by Feige et al. [18], who showed that

$$\mathbf{NP} \subseteq \mathbf{PCP}[O(\log n \log\log n), O(\log n \log\log n)] \tag{3.1}$$

and used this characterization of **NP** and a reduction to show that unless **NP** admits algorithms which run in slightly super-polynomial time, Maximum Clique cannot be approximated within any constant in polynomial time.

Results giving stronger characterizations of **NP** in terms of $\mathbf{PCP}[r,q]$ were proven by Arora and Safra [3] and Arora et al. [2]. The following result is by [2]:

Theorem 3.6 The PCP theorem.

$$\mathbf{NP} = \mathbf{PCP}[O(\log n), O(1)].$$

In other words, this remarkable theorem says that membership for **NP**-languages can be probabilistically checked by a verifier which uses logarithmic randomness, always accepts a correct proof, rejects incorrect proof with probability at least $1/2$, and looks only at a constant number of bits of the proof.

Chapter 4

A simple PCP and Hardness of approximating Maximum E3-Lin-2

As mentioned in the previous chapter, from the PCP characterization of **NP** given by the PCP theorem we automatically get an optimization problem which is hard to approximate, namely the problem of computing a proof with maximum acceptance probability. If we want to prove hardness of approximation for some problem, we could try to reduce this proof optimization problem to our problem. However, it turns out that by specially designing the PCP to make this reduction more efficient for the specific problem, we can get stronger results. This technique was introduced by Bellare, Goldreich and Sudan [5] who were able to get improved results for many problems, and Håstad [28] improved previous techniques to get almost optimal bounds for several problems.

Throughout this thesis this is the technique which we use: get strong inapproximability results by constructing a PCP specially tailored for the problem at hand.

All the PCPs which we construct in this thesis are based on techniques introduced by Håstad [27, 28]. In this chapter, as an introduction to these techniques, we describe and analyze the simplest PCP construction from [28].

Recall that Maximum E2-Lin-2 is the optimization problem where we are given a system of linear equations over $\boldsymbol{Z}_2$ where all the equations contain exactly three variables, and we want to find an assignment which satisfies as many equations as possible.

It is possible to approximate this problem within a factor 2; just choosing an assignment uniformly at random satisfies an expected fraction of $1/2$ of all equations, which gives a factor 2 approximation. It is possible to remove the randomness and achieve the same ratio deterministically by the method of conditional expectations [1, Section 15.1].

Since our goal in this section is proving a lower bound for Maximum E2-Lin-2 we want to construct a PCP so that the reduction to this problem is very efficient.

Suppose that we can prove that any language in **NP** is decided by a PCP which has completeness c, soundness s, uses $O(\log n)$ random bits, queries three bits from the proof, and accepts depending on the sum modulo 2 of these three bits.

Then we may create a system of linear equations by for each bit $\pi(i)$ in the proof introducing a variable x_i, and for each outcome of the random string introducing an equation. For example suppose the verifier on random string σ checks that $\pi(42) + \pi(17) + \pi(4711) = 1 \pmod 2$. Then we introduce the equation $x_{24} + x_{17} + x_{4711} = 1 \pmod 2$. Since the verifier uses $O(\log n)$ random bits, this will be a polynomial number of equations.

Now let L be an **NP**-complete language. If the input x to the verifier is in L, there is a proof which the verifier accepts with probability at least c, which translates to an assignment satisfying a fraction c of the equations. On the other hand, if the input is not in the language, there is no proof which the verifier accepts with probability greater than s, so there is no assignment which satisfies more than a fraction s of the equations. Hence we may conclude that Maximum E2-Lin-2 is **NP**-hard to approximate within c/s.

Many PCP constructions, starting with Arora and Safra [3] use *verifier composition*. The principle is that we have two verifiers, one which uses few random bits (and thus have a small proof), but where the values read from the proof are large, and another which uses many random bits but reads very few bits from the proof. The idea is to get the best of both worlds by combining the two verifiers.

In our case, we start with a two-prover one-round proof system which can be made to have arbitrarily low soundness, but in which the size of the answers is a large constant. On top of this a PCP is created in which the proof is supposed to contain for each possible query to the two-prover one-round proof system, the corresponding answer encoded with an error-correcting code. This proof is then checked by a verifier which reads very few bits. Intuitively, the verifier checks that the first verifier would have accepted.

4.1 A 2-prover 1-round protocol

A consequence of the PCP theorem is that there exists some constant Γ such that it is **NP**-hard to distinguish between satisfiable E3-CNF formulas and formulas in which at most a fraction Γ of the the clauses can be satisfied. Feige [16] proved that this implies the following theorem:

Theorem 4.1. There is a constant $\mu < 1$, such that: For any $L \in \mathbf{NP}$, there is a polynomial time reduction which given x produces a E3-CNF(5) formula $\phi(x)$ such that

1. If $x \in L$, then $\phi(x)$ is satisfiable,
2. If $x \notin L$, at most a fraction μ of the clauses in $\phi(x)$ are satisfiable.

2P1R protocol.
Input: E3-CNF(5) formula ϕ.

1. Choose uniformly at random a clause C from ϕ.
2. Choose uniformly at random a variable x from C.
3. Send C as a query to P_1, receiving an answer in $y \in \mathrm{SAT}^C$.
4. Send x to P_2, receiving an answer $z \in \{0,1\}$.
5. Accept if $\pi^{x,C}(y) = z$.

Figure 4.1. Basic 2P1R game.

In other words, it is **NP**-hard to distinguish between E3-CNF(5) formulas where all clauses may be satisfied, and E3-CNF(5) formulas where at most a fraction μ of the clauses are satisfiable.

There is a well-known two-prover one-round (2P1R) interactive proof system for **NP** which makes use of the previous theorem.

For a clause $C = (l_1 \vee l_2 \vee l_3)$ where l_1, l_2 and l_3 are literals, there are 7 satisfying assignments. We denote by SAT^C the set $\{1, \ldots, 7\}$, where $k \in \mathrm{SAT}^C$ is interpreted as the k:th satisfying assignment to C, in lexicographic order. For a clause C and a variable x in C we denote by $\pi^{x,C}$: $SAT^C \to \{0,1\}$ the function such that $\pi^{x,C}(y)$ is the value of x when $y \in \{1, \ldots, 7\}$ is interpreted as the y:th satisfying assignment to C.

The algorithm for the verifier in the two-prover one-round proof system is given in Figure 4.1. When ϕ is satisfiable, if the provers answers according to a satisfying assignment of ϕ, the verifier accepts with probability 1, i.e., the proof system has *completeness* 1, or *perfect completeness*.

Let us show that if ϕ is at most μ-satisfiable, the provers can fool the verifier with probability at most $(2 + \mu)/3$, or other words, that the above proof system has *soundness* $(2 + \mu)/3$.

The answers from the prover P_2 defines an assignment to the variables in ϕ. Since ϕ is at most μ-satisfiable, this assignment satisfies at most a fraction μ of the clauses. If the verifier chooses a clause C satisfied by this assignment, the verifier accepts. On the other hand, if C is not satisfied, the best P_1 can do is to change the value of one of the variables in C to create a satisfying assignment. The probability that this variable was sent to P_2 is $1/3$. Thus the provers fool the verifier with probability at most $\mu + 2(1 - \mu)/3 = (2 + \mu)/3$.

The soundness can be lowered to $((2+\mu)/3)^u$ by repeating the protocol u times independently, but it is also possible to construct a one-round proof system with lower soundness by repeating u times in parallel. We extend the notation SAT^C to the u-parallel case by for a sequence W of u clauses denoting by SAT^W the set

Parallel 2P1R protocol.
Input: An E3-CNF(5) formula ϕ.

1. Choose uniformly a sequence $W = (C_1, \ldots, C_u)$ of u clauses from ϕ.
2. From each C_i choose uniformly a variable x_i. Let $U = (x_1, \ldots, x_u)$ be the sequence of variables so constructed.
3. Send W to P_1, receiving a string $y \in \mathrm{SAT}^W$.
4. Send U to P_2, receiving a string $x \in \{0,1\}^U$.
5. Accept if $\pi^{U,W}(y) = x$

Figure 4.2. Parallel 2P1R game.

of all 7-ary strings of length u which we interpret as satisfying assignments to the clauses in W. That is, for $W = (C_1, \ldots, C_u)$, any $y \in \mathrm{SAT}^W$ can be written $y = (y_1, \ldots, y_u)$ where $y_i \in \mathrm{SAT}^{C_i}$. For any $U = (x_1, \ldots, x_u)$ and $W = (C_1, \ldots, C_u)$ such that x_i is a variable of C_j, we denote by $\pi^{U,W}$ the function which maps a $y \in \mathrm{SAT}^W$ to $x \in \{0,1\}^U$ by restricting y to the variables in U. More precisely, for $y = (y_1, \ldots, y_u)$, $\pi^{U,W}(y) = (\pi^{x_1,C_1}(y_1), \ldots, \pi^{x_u,C_u}(y_u))$.

The parallel two-prover one-round proof system is given in Figure 4.2.

The completeness of this proof system is 1, and it follows by a general theorem of Raz [43] that the soundness is at most c_μ^u, where $c_\mu < 1$ is some constant depending on μ but not on u or the size of the instance.

4.2 The long code and the Fourier transform

The proof in our PCP contains a purported encoding of a pair of strategies for the provers in the above u-parallel game.

For any satisfying assignment z to all the variables in ϕ, we denote by $z|_U$ and $z|_W$ the elements in $\{0,1\}^U$ and SAT^W respectively obtained by assigning the variables values according to z.

For $\beta \subseteq \mathrm{SAT}^W$, we will, by abuse of notation, write $\pi^{U,W}(\beta)$, by which we mean the set $\{\pi^{U,W}(y) \mid y \in \beta\}$. For any set S, we denote the set of functions $\{f\colon S \to \{-1,1\}\}$ by 2^S.

We also need another form of projection. For U, W as above and $\beta \subseteq \mathrm{SAT}^W$, we define *the modulo 2 projection of β on U*, $\pi_2^{U,W}(\beta)$ by $x \in \pi_2^{U,W}(\beta)$ iff there is an odd number of $y \in \beta$ such that $\pi^{U,W}(y) = x$.

The proof is supposed to be encoded with the *long code*, introduced by Bellare et al. [5]. The long code of $x \in \{0,1\}^U$ consists of the values of all functions from $\{0,1\}^U$ to $\{-1,1\}$ evaluated on x. In general:

Definition 4.2. Let S be a finite set. For $x \in S$, the *long code of* x, $\mathrm{LC}(x)$ is the function from 2^S to $\{-1,1\}$ defined as $\mathrm{LC}(x)(f) = f(x)$.

Definition 4.3. A *Standard Written Proof with parameter* u for a formula ϕ consists of a table $A_U \colon 2^{\{0,1\}^U} \to \{-1,1\}$ for each sequence U of u variables from ϕ and a table $A_W : 2^{\mathrm{SAT}^W} \to \{-1,1\}$ for each sequence W of u clauses from ϕ.

Definition 4.4. A Standard Written Proof with parameter u is a *correct proof for a formula* ϕ if there is an assignment x, satisfying ϕ, such that A_U is the long code of $x|_U$ for all sequences U of u variables, and A_W is the long code of $x|_W$ sequences W of u clauses.

A long code $\mathrm{LC}(x)$ has the property that $A_U(-f) = -A_U(f)$. It will sometimes be useful for a verifier to access the proof in a certain way to ensure that this property holds. We do this by *folding over* ***true***, as introduced by Bellare et al. [5].

Definition 4.5. Given a function $A \colon 2^S \to \boldsymbol{C}$ we define A_{true}, A folded over true, by for each pair of functions $(f, -f)$ choosing one and (if f is chosen), define $A_{\mathrm{true}}(f) = A(f)$, and $A_{\mathrm{true}}(-f) = -A(f)$.

When analyzing a purported long code, the Fourier transform will be our main tool.

Note that the space of functions $\{A \colon 2^S \to \boldsymbol{C}\}$ forms a vector space of dimension $2^{|S|}$. We define an inner product on this vector space by

$$\langle A \mid B \rangle = 2^{-|S|} \sum_{f \in 2^S} A(f)\overline{B(f)}. \tag{4.1}$$

Now for $\alpha \subseteq S$, we define $\chi_\alpha \colon 2^S \to \{-1,1\}$ by

$$\chi_\alpha(f) = \prod_{x \in \alpha} f(x). \tag{4.2}$$

Note that χ_α is the point-wise product of the long codes of all $x \in \alpha$. For $\alpha, \beta \subseteq S$, let $\alpha \Delta \beta$ be the symmetric difference of α and β, i.e.,

$$\alpha \Delta \beta = (\alpha \cup \beta) \setminus (\alpha \cap \beta)$$

In other words, $\alpha \Delta \beta$ is the set of elements which occur in exactly one of α and β. For $f, g \in 2^S$, let fg denote the function $(fg)(x) = f(x)g(x)$.

Proposition 4.6.

$$\chi_\alpha(f)\chi_\beta(f) = \chi_{\alpha \Delta \beta}(f), \tag{4.3}$$

$$\chi_\alpha(f)\chi_\alpha(g) = \chi_\alpha(fg), \tag{4.4}$$

$$\sum_{f \in 2^S} \chi_\alpha(f) = \begin{cases} 2^{|S|} & \text{if } \alpha = \emptyset \\ 0 & \text{if } \alpha \neq \emptyset \end{cases}. \tag{4.5}$$

Proof.

$$\begin{aligned}\chi_\alpha(f)\chi_\beta(f) &= \prod_{x\in\alpha} f(x) \prod_{x\in\beta} f(x)\\ &= \prod_{x\in\alpha\cap\beta} f(x)^2 \prod_{x\in\alpha\Delta\beta} f(x)\\ &= \prod_{x\in\alpha\Delta\beta} f(x)\\ &= \chi_{\alpha\Delta\beta}(f).\end{aligned}$$

Next,

$$\begin{aligned}\chi_\alpha(f)\chi_\alpha(g) &= \prod_{x\in\alpha} f(x) \prod_{x\in\alpha} g(x)\\ &= \prod_{x\in\alpha} f(x)g(x)\\ &= \chi_\alpha(fg).\end{aligned}$$

Finally,

$$\begin{aligned}\sum_{f\in 2^S} \chi_\alpha(f) &= \sum_{f\in 2^S} \prod_{x\in\alpha} f(x)\\ &= \sum_{f\in 2^S} \prod_{x\in\alpha} f(x) \prod_{x\in S\setminus\alpha} f(x)^0\\ &= \prod_{x\in\alpha} \left(\sum_{f(x)\in\{-1,1\}} f(x) \right) \prod_{x\in S\setminus\alpha} \left(\sum_{f(x)\in\{-1,1\}} f(x)^0 \right)\end{aligned}$$

this product is clearly 0 unless $\alpha = \emptyset$, in which case the product is $2^{|S|}$. ∎

Lemma 4.7. The functions χ_α form an orthogonal basis for the vector space $\{A\colon 2^S \to \boldsymbol{C}\}$. That is, there are $2^{|S|}$ of them and

$$\langle \chi_\alpha \mid \chi_\beta \rangle = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases} . \tag{4.6}$$

Proof.

$$\begin{aligned}\langle \chi_\alpha \mid \chi_\beta \rangle &= 2^{-|S|} \sum_{f\in 2^S} \chi_\alpha(f)\overline{\chi_\beta(f)}\\ &= 2^{-|S|} \sum_{f\in 2^S} \chi_\alpha(f)\chi_\beta(f)\\ &= 2^{-|S|} \sum_{f\in 2^S} \chi_{\alpha\Delta\beta}(f)\end{aligned}$$

By proposition 4.6, this last sum is 0 unless $\alpha \Delta \beta = \emptyset$, which happens only when $\alpha = \beta$. ∎

Since the functions χ_α form an orthogonal basis for the vector space of functions from 2^S to $\boldsymbol{C}$, any function $A\colon 2^S \to \boldsymbol{C}$ may be written as a linear combination of the basis functions, where the coefficients are simply projections of A onto the basis functions. More formally,

Theorem 4.8 Fourier-inversion. For any $A\colon 2^S \to \boldsymbol{C}$, let $\hat{A}_\alpha = \langle A \mid \chi_\alpha \rangle$. Then

$$A(f) = \sum_{\alpha \subseteq S} \hat{A}_\alpha \chi_\alpha(f).$$

Proof. Since the χ_α are orthogonal and there are $2^{|S|}$ of them, they form a basis, so any $A\colon 2^S \to \boldsymbol{C}$ may be written as

$$A(f) = \sum_{\alpha \subseteq S} a_\alpha \chi_\alpha(f).$$

We have that, for any $\beta \subseteq S$,

$$\begin{aligned}
\langle A \mid \chi_\beta \rangle &= 2^{-|S|} \sum_f \sum_{\alpha \subseteq S} a_\alpha \chi_\alpha(f) \overline{\chi_\beta(f)} \\
&= \sum_{\alpha \subseteq S} a_\alpha 2^{-|S|} \sum_f \chi_\alpha(f) \overline{\chi_\beta(f)} \\
&= \sum_{\alpha \subseteq S} a_\alpha \langle \chi_\alpha \mid \chi_\beta \rangle \\
&= a_\beta,
\end{aligned}$$

where the last equality holds by Lemma 4.7. ∎

We call $\hat{A}_\alpha$ the *Fourier-coefficient of A at α*. Note that, for a function A which takes values in $\{-1, 1\}$,

$$\hat{A}_\alpha = \Pr_f[A(f) = \chi_\alpha(f)] - \Pr_f[A(f) \neq \chi_\alpha(f)] = 2 \Pr[A(f) = \chi_\alpha(f)] - 1$$

Hence in this case $\hat{A}_\alpha$ is a measure of the correlation of A with χ_α, with $\hat{A}_\alpha = 1$ meaning that they are totally correlated (i.e., $A = \chi_\alpha$), and $\hat{A}_\alpha = -1$ meaning that they are totally negatively correlated (i.e., $A = -\chi_\alpha$).

Theorem 4.9 Plancherel Theorem or Parseval's equality. For any $A\colon 2^S \to \boldsymbol{C}$,

$$\sum_{\alpha \subseteq S} |\hat{A}_\alpha|^2 = 2^{-|S|} \sum_{f \in 2^S} |A(f)|^2$$

In particular, when $A\colon 2^S \to \{-1, 1\}$

$$\sum_{\alpha \subseteq S} |\hat{A}_\alpha|^2 = 1$$

Proof.

$$\begin{aligned}
\sum_{f\in 2^S} |A(f)|^2 &= 2^{-|S|} \sum_{f\in 2^S} A(f)\overline{A(f)} \\
&= 2^{-|S|} \sum_{f\in 2^S} \sum_{\alpha\subseteq S} \sum_{\beta\subseteq S} \hat{A}_\alpha \overline{\hat{A}_\beta} \chi_\alpha(f)\overline{\chi_\beta(f)} \\
&= \sum_{\alpha\subseteq S} \sum_{\beta\subseteq S} \hat{A}_\alpha \overline{\hat{A}_\beta} 2^{-|S|} \sum_{f\in 2^S} \chi_\alpha(f)\overline{\chi_\beta(f)} \\
&= \sum_{\alpha\subseteq S} \sum_{\beta\subseteq S} \hat{A}_\alpha \overline{\hat{A}_\beta} \langle \chi_\alpha \mid \chi_\beta \rangle \\
&= \sum_{\alpha\subseteq S} |\hat{A}_\alpha|^2
\end{aligned}$$

∎

The following lemma shows what effect folding over **true** has on the Fourier-coeffiecents.

Lemma 4.10. Let $A\colon 2^S \to C$ have the property that for all f, $A(-f) = -A(f)$. Then if $\hat{A}_\alpha$ is not 0, then $|\alpha|$ is odd (and in particular α is not empty).

Proof. Clearly, for any $\alpha \subseteq S$,

$$\chi_\alpha(-f) = \prod_{x\in\alpha} (-f(x)) = (-1)^{|\alpha|}\chi_\alpha(f)$$

Thus if $|\alpha|$ is even, $\chi_\alpha(-f) = \chi_\alpha(f)$. Hence

$$\begin{aligned}
\hat{A}_\alpha &= 2^{-|S|} \sum_f A(f)\chi_\alpha(f) \\
&= 2^{-|S|} \frac{1}{2} \sum_f (A(f)\chi_\alpha(f) + A(-f)\chi_\alpha(-f)) \\
&= 2^{-|S|-1} \frac{1}{2} \sum_f (A(f) + A(-f))\chi_\alpha(f).
\end{aligned}$$

The last sum is clearly 0 since $A(-f) = -A(f)$. ∎

4.2.1 Testing of the long code

As a warm-up to PCP construction, we consider how to test that a function $A\colon 2^S \to \{-1, 1\}$ is indeed a long code. Since we only want to look at very few bits in A we can't hope to have a test which always rejects if A is not a long code. Instead the best we can expect is a test such that if it accepts with high enough probability, then we know that the function A is close to a long code in some sense.

LongcodeTest.
Input: A purported long code A.

1. Choose $f, g \in 2^S$ uniformly.
2. Choose $e \in 2^S$ by for each $x \in S$ setting $e(x) = 1$ with probability $1 - \delta$ and $e(x) = -1$ with probability δ.
3. Accept if $A(f)A(g)A(fge) = 1$.

Figure 4.3. Testing a long code.

Since a long code A has the property that $A(fg) = A(f)A(g)$, a possible first try for a test would be to choose $f, g \in 2^S$ uniformly and accept if $A(f)A(g)A(fg) = 1$. This is essentially the linearity test of Blum et al. [6]. However, any product of long codes will also pass this test; let $\alpha \subseteq S$ and $B(f) = \prod_{x \in \alpha} f(x)$. Then $B(fg) = \prod_{x \in \alpha} f(x)g(x) = B(f)B(g)$.

As an attempt to fix this, we modify the test as shown in Figure 4.3.

If $A = \mathrm{LC}(x)$, then A will pass this test with probability $1 - \delta$, while for B a product of long codes the probability of acceptance is

$$\begin{aligned} \frac{1 + \mathrm{E}_{f,g,e}[B(f)B(g)B(fge)]}{2} &= \frac{1 + \mathrm{E}_e[\prod_{x \in \alpha} e(x)]}{2} \\ &= \frac{1 + (1 - 2\delta)^{|\alpha|}}{2} \end{aligned}$$

which approaches $1/2$ when $|\alpha|$ goes to infinity. Thus this takes care of large products of long codes, while small products of long codes will turn out to be "close enough" to real long code for our purposes.

Intuitively, the above test should reject any function $A\colon 2^S \to \{-1, 1\}$ which depends strongly the value of f on many different x. This is because when we multiply with e, the more x on which A depends, the greater the chance that $f(x)e(x)$ is different from $f(x)$ for some such x, and thus the chance that $A(f) \neq A(fe)$ increases.

Let us proceed to analyze this test formally. As we saw above, the test was actually easy to analyze in the case when A was a long code or a product of long codes. Luckily for us, the Fourier inversion formula tells us that any function $A\colon 2^S \to \boldsymbol{C}$ may be written as a sum of products of long codes. Thus we expand A in its Fourier-series and proceed with the analysis. Using Theorem 4.8, we have

that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,g,e}[B(f)B(g)B(fge)] \\
&= \mathop{\mathrm{E}}_{f,g,e}\left[\sum_{\alpha_1\subseteq S}\sum_{\alpha_2\subseteq S}\sum_{\alpha_3\subseteq S}\hat{B}_{\alpha_1}\chi_{\alpha_1}(f)\hat{B}_{\alpha_2}\chi_{\alpha_2}(g)\hat{B}_{\alpha_3}\chi_{\alpha_3}(fge)\right] \\
&= \sum_{\alpha_1\subseteq S}\sum_{\alpha_2\subseteq S}\sum_{\alpha_3\subseteq S}\hat{B}_{\alpha_1}\hat{B}_{\alpha_2}\hat{B}_{\alpha_3}\mathop{\mathrm{E}}_{f,g,e}\left[\chi_{\alpha_1}(f)\chi_{\alpha_2}(g)\chi_{\alpha_3}(fge)\right]. \qquad (4.7)
\end{aligned}$$

Thus we need to compute

$$\mathop{\mathrm{E}}_{f,g,e}\left[\chi_{\alpha_1}(f)\chi_{\alpha_2}(g)\chi_{\alpha_3}(fge)\right]. \qquad (4.8)$$

Using (4.3) and (4.4) from Proposition 4.6, we get that (4.8) is equal to

$$\mathop{\mathrm{E}}_{f,g,e}\left[\chi_{\alpha_1\Delta\alpha_3}(f)\chi_{\alpha_2\Delta\alpha_3}(g)\chi_{\alpha_3}(e)\right] = \mathop{\mathrm{E}}_{f}\left[\chi_{\alpha_1\Delta\alpha_3}(f)\right]\mathop{\mathrm{E}}_{g}\left[\chi_{\alpha_2\Delta\alpha_3}(g)\right]\mathop{\mathrm{E}}_{e}\left[\chi_{\alpha_3}(e)\right].$$

By (4.5) from Proposition 4.6, the expectation is 0 unless $\alpha_1 = \alpha_2 = \alpha_3$, while the expectation over e is $(1-2\delta)^{|\alpha_3|}$. Thus (4.7) is equal to

$$\sum_{\alpha\subseteq S}\hat{A}_\alpha^3(1-2\delta)^{|\alpha|} \le \sum_{\alpha\subseteq S}(\max_{\beta\subseteq S}(\hat{A}_\beta(1-2\delta)^{|\beta|})\hat{A}_\alpha^2) = \max_{\beta\subseteq S}(\hat{A}_\beta(1-2\delta)^{|\beta|}).$$

Where the last equality follows by Parseval's equality (Theorem 4.9).

We conclude that if the test accepts with high probability there must be some reasonably small α such that the input is close to χ_α, i.e., the input is close to a small product of long codes. Of course, this product may be empty, since the constant 1 function passes LongcodeTest with probability 1. However, if A is folded over **true** $\hat{A}_\emptyset = 0$, and in this case the product must be non-empty.

4.3 The basic protocol

In this section we describe and analyze the PCP which we use to prove inapproximability of Maximum E2-Lin-2. Recall that the input to the test is an E3-CNF(5) formula, and that the proof is a Standard Written Proof, which means that the verifier has access to tables A_U and A_W for sequences of variables and clauses respectively. The verifier simulates the verifier in the 2P1R game, choosing a sequence W of u clauses and a sequence U of u variables with the same distribution. The verifier needs to check the corresponding tables for consistency. To get an inapproximability result for Maximum E2-Lin-2 the verifier should accept on a linear condition. The way we construct the test is to take the long code test from the previous section and move one query to the table A_U.

The test is given in Figure 4.4.

LinTest.
Input: An E3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Standard Written Proof with parameter u, and all the tables folded over **true**.

1. Select uniformly at random a sequence $W = (C_1, \ldots, C_u)$ of u clauses and a sequence U of u variables by for each clause C_k choosing uniformly at random a variable occurring in C_k.
2. Select uniformly at random $f \in 2^{\{0,1\}^U}$ and $g \in 2^{SAT^W}$.
3. Select $e \in 2^{SAT^W}$ by for each y independently setting $e(y) = 1$ with probability $1 - \delta$ and $e(y) = -1$ with probability δ.
4. Accept if $A_{U,\mathbf{true}}(f)A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}((f \circ \pi^{U,W})ge) = 1$, else reject.

Figure 4.4. The above PCP is parameterized by the positive integer u and the positive real δ and tests if a E3-CNF(5) formula ϕ is satisfiable by querying three positions in a Standard Written Proof with parameter u. With suitable choices of the parameters u and δ, the test has completeness $1 - \epsilon$ and soundness $\frac{1}{2} + \epsilon$ for any $\epsilon > 0$.

Why would we expect this to work?

If we used the methods which were used prior to the work of Håstad, we would proceed as follows:

1. Run a test to ensure that the tables are close to correct long codes.
2. Run a test to ensure that the values encoded by these long codes are satisfying assignments.
3. Run a test to ensure that the values are consistent.

Step 2 is unnecessary in our case since by definition we interpret all possible strings as satisfying assignments. The test in figure 4.4 may be viewed as a combination of step 1 and step 3.

Another way of viewing the test is to consider the following error correction procedure: if A is close enough to a long code $\mathrm{LC}(x)$, then for any h we can get the value $h(x)$ with good probability by picking a random g and returning the value $A(g)A(gh)$ [6]. In our case, we do not know that A is close to a long code, however, it turns out that this procedure still can be viewed as a sort of error correction even when A is not close to any particular long code. In fact, if A is somewhat close to several products of long codes χ_α we get one of the values $\chi_\alpha(h)$ with probability proportional to how close A is to χ_α. Adding the "noise" function e gives a bias against χ_α where α is large. Thus LinTest may be viewed as reading the corrected value of $f \circ \pi^{U,W}$ from A_W and checking against $A_U(f)$. We expect that if they

often have consistent values, then A_U and A_W should be close to small products of long codes of consistent values. From these, it should be possible to construct a strategy for the 2P1R game with good success probability. In fact, we prove that it is enough that the probability of success is even slightly better than what would be expected from randomly chosen tables.

Note that error correction would be superfluous when reading f from A since f is uniformly distributed in that table. Intuitively, the reason we need error correction when reading $f \circ \pi^{U,W}$ from A_W is that functions on this form is a extremely small proportion of all functions in 2^{SAT^W}. Thus it is possible for A_W to be very close to a long code while having incorrect values for all functions on the form $f \circ \pi^{U,W}$.

Before analyzing the soundness of the test, we turn to the completeness, which is easy:

Lemma 4.11. If ϕ is satisfiable, there is a proof which LinTest accepts with probability $1 - \delta$.

Proof. Let x be a satisfying assignment to ϕ. Let the tables be as in a correct proof, i.e., $A_U = \mathrm{LC}(x|_U)$ and $A_W = \mathrm{LC}(x|_W)$. Then

$$A_{U,\mathbf{true}}(f)A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}((f \circ \pi^{U,W})ge) = \\ f(x|_U)g(x|_W)(f \circ \pi^{U,W})(x|_W)g(x|_W)e(x|_W).$$

Since $(f \circ \pi^{U,W})(x|_W) = f(x|_U)$, this expression reduce to $e(x|_W)$, which is 1 with probability $1 - \delta$. ∎

We now turn to the analysis of the soundness.

Lemma 4.12. If LinTest accepts with probability at least $\frac{1+\delta}{2}$, there is a strategy for the 2P1R game with success probability at least $2\delta^3$.

Proof. The probability that the verifier accepts may be written as

$$\frac{1 + \mathrm{E}_{U,W,f,g,e}[A_{U,\mathbf{true}}(f)A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}((f \circ \pi^{U,W})ge)]}{2}.$$

Thus by the assumption in the statement of the lemma, the expectation

$$\mathop{\mathrm{E}}_{U,W,f,g,e}[A_{U,\mathbf{true}}(f)A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}((f \circ \pi^{U,W})ge)] \tag{4.9}$$

is at least δ. We first analyze this expression for fixed U and W. We introduce the shorthands $A = A_{U,\mathbf{true}}$ and $B = A_{W,\mathbf{true}}$. Applying the Fourier-inversion formula (Theorem 4.8), to A and B, and using the linearity of expectation, we get that

$$\mathop{\mathrm{E}}_{f,g,e}[A_{U,\mathbf{true}}(f)A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}((f \circ \pi^{U,W})ge)] = \\ \sum_{\alpha \in 2^{\{0,1\}^U}} \sum_{\substack{\beta_1 \in 2^{\mathrm{SAT}^W} \\ \beta_2 \in 2^{\mathrm{SAT}^W}}} \hat{A}_\alpha \hat{B}_{\beta_1} \hat{B}_{\beta_2} \mathop{\mathrm{E}}_{f,g,e} \left[\chi_\alpha(f)\chi_{\beta_1}(g)\chi_{\beta_2}((f \circ \pi^{U,W})ge)\right]. \tag{4.10}$$

We next compute the expectation over f, g and e. We have that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,g,e}\left[\chi_\alpha(f)\chi_{\beta_1}(g)\chi_{\beta_2}((f\circ\pi^{U,W})ge)\right] = \\
&\quad \mathop{\mathrm{E}}_{f,g,e}\left[\chi_\alpha(f)\chi_{\beta_1}(g)\chi_{\beta_2}(f\circ\pi^{U,W})\chi_{\beta_2}(g)\chi_{\beta_2}(e)\right] = \\
&\quad \mathop{\mathrm{E}}_{f}[\chi_\alpha(f)\chi_{\beta_2}(f\circ\pi^{U,W})]\mathop{\mathrm{E}}_{g}[\chi_{\beta_1}(g)\chi_{\beta_2}(g)]\mathop{\mathrm{E}}_{e}[\chi_{\beta_2}(e)]
\end{aligned}$$

where the first equality follows by the multiplicative property of the basis functions, (Proposition 4.6, equation (4.4)), and the second by the fact that f, g and e are independent. We proceed to compute the expectation over g. By Proposition 4.6, equation (4.3),

$$\mathop{\mathrm{E}}_{g}[\chi_{\beta_1}(g)\chi_{\beta_2}(g)] = \mathop{\mathrm{E}}_{g}[\chi_{\beta_1\Delta\beta_2}(g)]$$

which by Proposition 4.6, equation (4.5) is 0 unless $\beta_1 = \beta_2 = \beta$, in which case the expectation is 1. Next consider the expectation over e. We have that

$$\mathop{\mathrm{E}}_{e}[\chi_\beta(e)] = \mathop{\mathrm{E}}_{e}\left[\prod_{y\in\beta} e(y)\right] = \prod_{y\in\beta}\mathop{\mathrm{E}}_{e(y)}[e(y)] = (1-2\delta)^{|\beta|}.$$

What remains to compute is the expectation over f. Using the definitions of χ_α and χ_β,

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f}[\chi_\alpha(f)\chi_\beta(f\circ\pi^{U,W})] \\
&\quad = \mathop{\mathrm{E}}_{f}\left[\prod_{x\in\alpha} f(x)\prod_{y\in\beta}(f\circ\pi^{U,W})(y)\right]. \qquad (4.11)
\end{aligned}$$

We rewrite this expression by grouping $y \in \beta$ according to which x they project onto. Thus (4.11) becomes

$$\mathop{\mathrm{E}}_{f}\left[\prod_{x\in\alpha}\left(f(x)\prod_{\substack{y\in\beta\\ \pi^{U,W}(y)=x}}(f\circ\pi^{U,W})(y)\right)\prod_{x\in\{0,1\}^U\setminus\alpha}\left(\prod_{\substack{y\in\beta\\ \pi^{U,W}(y)=x}}(f\circ\pi^{U,W})(y)\right)\right]. \quad (4.12)$$

For $x \in \{0,1\}^U$, let s_x be the number of $y \in \beta$ such that $\pi^{U,W}(y) = x$. Then (4.12) may be written as

$$\mathop{\mathrm{E}}_{f}\left[\prod_{x\in\alpha} f(x)^{s_x+1}\prod_{x\in\{0,1\}^U\setminus\alpha} f(x)^{s_x}\right] = \prod_{x\in\alpha}\mathop{\mathrm{E}}_{f(x)}[f(x)^{s_x+1}]\prod_{x\in\{0,1\}^U\setminus\alpha}\mathop{\mathrm{E}}_{f(x)}[f(x)^{s_x}].$$

For this expression to be non-zero, it must be the case that s_x is odd for $x \in \alpha$ and even for $x \in \{0,1\}^U \setminus \alpha$. It follows that the expression is 0 unless $\alpha = \pi_2^{U,W}(\beta)$, in

which case it is 1. To sum up, we have proven that the only non-zero terms in (4.10) are those where $\beta_1 = \beta_2 = \beta$ and $\alpha = \pi_2^{U,W}(\beta)$. Combined with the expression for the expectation over e, this means that (4.10) reduces to

$$\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{|\beta|}.$$

By the Cauchy–Schwartz inequality (Proposition 2.1) with $a_i = \hat{A}_{\pi_2^{U,W}(\beta)} \hat{B}_\beta (1-2\delta)$ and $b_i = \hat{B}_\beta$, we have that

$$\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{|\beta|} \leq$$
$$\sqrt{\left(\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}^2_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{2|\beta|}\right)\left(\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{B}_\beta^2\right)} =$$
$$\sqrt{\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}^2_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{2|\beta|}}.$$

Where the equality follows because the sum $\sum_\beta \hat{B}_\beta^2$ is equal to 1 by Parseval's equality (Theorem 4.9). Taking the expectation over U and W, it follows that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{U,W}\left[\sqrt{\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}^2_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{2|\beta|}}\right] \\
&\geq \mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{|\beta|}\right] \\
&\geq \delta,
\end{aligned}$$

since by assumption (4.9) was at least δ. By Jensen's inequality (Proposition 2.3), $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$, so

$$\mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}^2_{\pi_2^{U,W}(\beta)} \hat{B}_\beta^2 (1-2\delta)^{2|\beta|}\right] \geq \delta^2. \tag{4.13}$$

Now we describe randomized strategies for the provers in the 2P1R game. As mentioned previously, these may be converted to deterministic strategies by fixing the random strings of the provers such that the acceptance probability is not decreased.

The prover P_1 upon receiving W picks a β with probability $\hat{B}_\beta^2 = \hat{A}^2_{W,\mathbf{true},\beta}$ and returns a random $y \in \beta$.

The prover P_2 upon receiving U picks an α with probability $\hat{A}_\alpha^2 = \hat{A}^2_{U,\mathbf{true},\alpha}$ and returns a random $x \in \alpha$.

Note that by Lemma 4.10, $\hat{A}_{U,\mathbf{true},\emptyset} = \hat{A}_{W,\mathbf{true},\emptyset} = 0$, so neither P_1 nor P_2 will ever pick a empty α or β.

Let us analyze the probability that the provers succeed in convincing the verifier when following this strategy.

For any $\alpha \subseteq \{0,1\}^U$ and $\beta \subseteq \mathrm{SAT}^W$, the probability that P_2 picks α and P_1 picks β is $\hat{A}^2_{U,\mathbf{true},\alpha}\hat{A}^2_{W,\mathbf{true},\beta}$. When $\alpha \subseteq \pi^{U,W}(\beta)$, the probability that the provers picks x and y such that $\pi^{U,W}(y) = x$ is at least $|\beta|^{-1}$. Thus the probability that the provers succeed is at least

$$\mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta\subseteq\mathrm{SAT}^W}\sum_{\alpha\subseteq\pi^{U,W}(\beta)}\hat{A}^2_{U,\mathbf{true},\alpha}\hat{A}^2_{W,\mathbf{true},\beta}|\beta|^{-1}\right] \geq \mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta\subseteq\mathrm{SAT}^W}\hat{A}^2_{U,\mathbf{true},\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\mathbf{true},\beta}|\beta|^{-1}\right]. \tag{4.14}$$

Since $e^x \geq 1 + x \geq x$ for real x, $\frac{e^{2\delta|\beta|}}{2\delta} \geq \frac{2\delta|\beta|}{2\delta} = |\beta|$, thus we have that $|\beta|^{-1} \geq 2\delta e^{-2\delta|\beta|} \geq 2\delta(1-2\delta)^{|\beta|}$, where we again used $e^x \geq 1 + x$ in the last inequality (with negative x). It follows that (4.14) is at least

$$2\delta \mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta\subseteq\mathrm{SAT}^W}\hat{A}^2_{U,\mathbf{true},\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\mathbf{true},\beta}(1-2\delta)^{2|\beta|}\right]$$

which is at least $2\delta^3$ by (4.13). Thus the probability that the provers convince the verifier is at least $2\delta^3$. ∎

Why is expanding the tables in their Fourier-series the right approach to analyzing the test? Suppose that ϕ is not satisfiable, and consider a prover which wants to convince us otherwise. If the prover only uses correct long codes, it is easy to see that for a pair U, W, if the long codes for U and W are consistent then the test will accept with probability $1 - \epsilon$, while if they are inconsistent, that is if $A_U = \mathrm{LC}(x)$ and $A_W = \mathrm{LC}(y)$ where $\pi^{U,W}(y) \neq x$, then the test will succeed with probability at most $1/2$. Thus if we have acceptance probability $\frac{1+\delta}{2}$ we must have consistent long codes with probability at least δ, and we can get a strategy for the 2P1R game. Thus the test is easy to analyze in the case when the tables are correct long codes, and as in the case of the long code test, it turns out that the test is also easy to analyze when the tables are products of long codes.

Expanding the tables in their Fourier-series means that we essentially can reduce to these easy cases. If we look back at the actual analysis, we see that what we reduce to is actually slightly more general; since a table can be correlated with more than one product of long codes, we also need to analyze the "cross terms". This means that we need to be able to analyze the case that there are three tables, which are products of long codes $\chi_\alpha, \chi_{\beta_1}$ and χ_{β_2}, and we read one value from each table. Since different products of long codes are minimally correlated (i.e., the χ_β

are orthogonal), it turns out that it is never a good idea for the prover to have β_1 different from β_2.

4.4 Hardness of approximating Maximum E3-Lin-2

Theorem 4.13 [28]. For any $\epsilon > 0$, Maximum E2-Lin-2 is **NP**-hard to approximate within $2 - \epsilon$.

Proof. In LinTest, choose the parameter δ so that

$$\frac{1-\delta}{(1+\delta)/2} \geq 2 - \epsilon$$

and u such that the soundness of the 2P1R game c_μ^u is less than $2\delta^3$. For the resulting verifier, introduce one variable for each position in the proof, and write down for each random string the an equation corresponding to test made by the verifier. Since the verifier uses $O(\log n)$ random bits, this will be a polynomial number of variables and equations. By Lemma 4.11, if ϕ is satisfiable there is a proof which makes the verifier accept with probability at least $1 - \delta$. Thus there is an assignment which satisfies at least a fraction $1 - \delta$ of all equations.

By Lemma 4.12, if ϕ is at most μ satisfiable, there is no proof which makes the verifier accept with probability more than $(1 + \delta)/2$, since that would mean that there is a strategy for the provers in the 2P1R game with success probability at least $2\delta^3$, which is impossible by the choice of u. Thus in this case, at most a fraction $(1 + \delta)/2$ of all equations can be satisfied.

Since by Theorem 4.1 it is **NP**-hard to distinguish between the case that ϕ is satisfiable and the case that ϕ is at most μ satisfiable, the theorem follows by the choice of δ. ∎

Chapter 5

Two generalizations of the framework

5.1 Generalization modulo d

5.1.1 The Fourier transform and the long code over $\boldsymbol{Z}_d$

In this section we generalize the long code and the Fourier transforms from Section 4.2 to $\boldsymbol{Z}_d$ (i.e., the cyclic group with d elements). In general, the Fourier series of a function from an Abelian group to $\boldsymbol{C}$ is a linear combination of the characters of that group, i.e., homomorphisms from the group to the multiplicative group of complex number with norm 1. For a general treatment of this theory, we refer the reader to Terras's book [49]. For the the further generalization to general finite groups, see also Chapter 11 in this thesis.

Since we only work with powers of $\boldsymbol{Z}_d$, we use the following conventions to simplify the framework: The group $\boldsymbol{Z}_d$ is represented by the powers of $\omega = e^{2\pi i/d}$ with multiplication as the group operator. The characters of that group also form a group, $\hat{\boldsymbol{Z}}_d$, which we identify with the integers $\{0, 1, 2, \ldots, d-1\}$ with addition modulo d as the group operator.

For $g \in \boldsymbol{Z}_d$ and $\gamma \in \hat{\boldsymbol{Z}}_d$ we denote the action of γ on g by $\chi_\gamma(g) = g^\gamma$.

Lemma 5.1. Let $\gamma \in \hat{\boldsymbol{Z}}_d$ be arbitrary. Then

$$\frac{1}{d} \sum_{g \in \boldsymbol{Z}_d} \chi_\gamma(g) = \begin{cases} 1 & \text{if } \gamma = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. If $\gamma = 0$, then $\chi_\gamma(g) = 1$, and since there are d terms, the sum is clearly 1.

If $\gamma \neq 0$,

$$\frac{1}{d}\sum_{g\in \boldsymbol{Z}_d} \chi_\gamma(g) = \frac{1}{d}\sum_{k=0}^{d-1} \omega^{k\gamma} = \frac{1}{d}\frac{1-\omega^{\gamma d}}{1-\omega^\gamma} = 0$$

where the last equality holds since $\omega^d = 1$. ∎

The proof of the following lemma is similar:

Lemma 5.2. Let $g \in \boldsymbol{Z}_d$ be arbitrary. Then

$$\frac{1}{d}\sum_{\gamma=0}^{d-1} \chi_\gamma(g) = \begin{cases} 1 & \text{if } g = 1, \\ 0 & \text{otherwise.} \end{cases}$$

We generalize the notation 2^S: for any set S, denote by Z_d^S the set $\{f : S \to \boldsymbol{Z}_d\}$. The space Z_d^S can be viewed as a power of $\boldsymbol{Z}_d$ since we can identify a function with the table of its values. The characters of Z_d^S, $\hat{\boldsymbol{Z}}_d^S$ may be identified with $\hat{\boldsymbol{Z}}_d^{|S|}$, or equivalently as the set of functions $\{\alpha : S \to \{0,\ldots,d-1\}$ with the group operation begin pointwise addition modulo d. Denote the identically 0 function by $\mathbf{0} \in \hat{\boldsymbol{Z}}_d^S$. The long code generalize naturally:

Definition 5.3. Let S be a finite set. For $x \in S$, the *long d code of x*, $\mathrm{LC}_d(x)$ is the function from Z_d^S to $\boldsymbol{Z}_d$ defined as $\mathrm{LC}_d(x)(f) = f(x)$.

Definition 5.4. A *Standard Written d-Proof with parameter u* for a formula ϕ consists of a table A_U: $Z_d^{\{0,1\}^U} \to \boldsymbol{Z}_d$ for each sequence U of u variables from ϕ and a table A_W: $Z_d^{\mathrm{SAT}^W} \to \boldsymbol{Z}_d$ for each sequence W of u clauses from ϕ.

Definition 5.5. A Standard Written d-Proof with parameter u is a *correct proof for a formula ϕ* if there is an assignment x, satisfying ϕ, such that A_U is the long d code of x restricted to U for all sequences U of u variables, and A_W is the long d code of x restricted to W for all sequences W of u clauses

Again we want to expand a purported long code A: $Z_d^S \to \boldsymbol{Z}_d$ in a orthogonal basis. We define an inner product

$$\langle A \mid B\rangle = d^{-|S|} \sum_{f\in \hat{\boldsymbol{Z}}_{\boldsymbol{Z}_d}^S} A(f)\overline{B(f)}$$

In the binary case, the basis functions were given by subsets $\alpha \subseteq S$. Clearly this won't be enough in this case since the underlying vector space is of dimension $d^{|S|}$. Instead, the basis functions are given by the characters for the group Z_d^S, which we

have one of for each element in the group $\hat{\boldsymbol{Z}}^S_{\boldsymbol{Z}_d}$. This gives us our generalization: for $\alpha \in \hat{\boldsymbol{Z}}^S_{\boldsymbol{Z}_d}$ the action of α is defined by:

$$\chi_\alpha(f) = \prod_{x \in S} f(x)^{\alpha(x)}$$

This generalizes the case modulo 2, since a subset $\alpha \subseteq S$ may be identified with a function $\alpha\colon S \to \{0,1\}$. For an $\alpha \in \hat{\boldsymbol{Z}}^S_{\boldsymbol{Z}_d}$ we define the *weight* of α, $|\alpha|$, to be number of $x \in S$ such that $\alpha(x) \neq 0$.

Proposition 5.6.

$$\chi_\alpha(f)\chi_\alpha(g) = \chi_\alpha(fg) \tag{5.1}$$

$$\chi_\alpha(f)\chi_\beta(f) = \chi_{\alpha+\beta}(f) \tag{5.2}$$

$$\chi_\alpha(f^{-1}) = \chi_{-\alpha}(f) = \overline{\chi_\alpha(f)} \tag{5.3}$$

$$\sum_{f \in \hat{\boldsymbol{Z}}^S_{\boldsymbol{Z}_d}} (\chi_\alpha(f)) = \begin{cases} 0 & \text{if } \alpha \neq \mathbf{0} \\ d^{|S|} & \text{if } \alpha = \mathbf{0}. \end{cases} \tag{5.4}$$

Proof.

$$\begin{aligned} \chi_\alpha(f)\chi_\alpha(g) &= \prod_{x \in S} f(x)^{\alpha(x)} \prod_{x \in S} g(x)^{\alpha(x)} \\ &= \prod_{x \in S} (f(x)g(x))^{\alpha(x)} \\ &= \chi_\alpha(fg) \end{aligned}$$

Next,

$$\begin{aligned} \chi_\alpha(f)\chi_\beta(f) &= \prod_{x \in S} f(x)^{\alpha(x)} \prod_{x \in S} f(x)^{\beta(x)} \\ &= \prod_{x \in S} f(x)^{\alpha(x)+\beta(x)} \\ &= \prod_{x \in S} f(x)^{\alpha(x)+\beta(x)} \pmod d \end{aligned}$$

where the last equality holds since $f(x)$ is a d:th root of unity. Thus

$$\begin{aligned}\chi_\alpha(f)\chi_\beta(g) &= \prod_{x\in S} f(x)^{(\alpha+\beta)(x)} \\ &= \chi_{\alpha+\beta}(f)\end{aligned}$$

To continue,

$$\chi_\alpha(f^{-1}) = \prod_{x\in S} f(x)^{-\alpha(x) \pmod d} = \chi_{-\alpha}(f)$$

Also

$$\chi_{-\alpha}(f)\chi_\alpha(f) = \chi_{\mathbf{0}}(f) = 1,$$

and

$$\chi_\alpha(f)\overline{\chi_\alpha(f)} = |\chi_\alpha(f)|^2 = 1.$$

So $\chi_{-\alpha}(f) = \overline{\chi_\alpha(f)} = (\chi_\alpha(f))^{-1}$

Finally,

$$\begin{aligned}\sum_{f\in Z_d^S} (\chi_\alpha(f)) &= \sum_{f\in Z_d^S} \prod_{x\in S} f(x)^{\alpha(x)} \\ &= \prod_{x\in S} \left(\sum_{f(x)\in \boldsymbol{Z}_d} f(x)^{\alpha(x)} \right).\end{aligned}$$

By Lemma 5.1, the inner sum is 0 unless $\alpha(x) = 0$, in which case it is d. Thus the product is 0 unless $\alpha = \mathbf{0}$, in which case it is $d^{|S|}$. ∎

Lemma 5.7. The functions χ_α form an orthogonal basis for the vector space $\{A\colon Z_d^S \to \boldsymbol{C}\}$. That is, there are $d^{|S|}$ of them, and

$$\langle \chi_\alpha \mid \chi_\beta \rangle = \begin{cases} 1 & \text{if } \alpha = \beta \\ 0 & \text{if } \alpha \neq \beta \end{cases}. \tag{5.5}$$

Proof.

$$\begin{aligned}\langle \chi_\alpha \mid \chi_\beta \rangle &= d^{-|S|} \sum_{f\in Z_d^S} \chi_\alpha(f)\overline{\chi_\beta(f)} \\ &= d^{-|S|} \sum_{f\in Z_d^S} \chi_\alpha(f)\chi_{-\beta}(f) \\ &= d^{-|S|} \sum_{f\in Z_d^S} \chi_{\alpha-\beta}(f)\end{aligned}$$

where the last two equalities holds by (5.3) and (5.3) from Proposition 5.6. Using (5.4) from the same proposition, $\langle \chi_\alpha \mid \chi_\beta \rangle = 0$ unless $\alpha = \beta$, in which case $\langle \chi_\alpha \mid \chi_\beta \rangle = 1$ ∎

As in the modulo 2 case, we have the Fourier inversion formula and Parseval's equality. We omit the proofs since they are essentially identical to the modulo 2 case.

Theorem 5.8. Let

$$\hat{A}_\alpha = \langle A \mid \chi_\alpha(f) \rangle$$

Then

$$A(f) = \sum_{\alpha \in \hat{\boldsymbol{Z}}_d^S} \hat{A}_\alpha \chi_\alpha(f) \tag{5.6}$$

Theorem 5.9.

$$d^{-|S|} \sum_f |A(f)|^2 = \sum_\alpha |\hat{A}_\alpha|^2 \tag{5.7}$$

For a function $A\colon Z_d^S \to \boldsymbol{C}$, we define $A_{\boldsymbol{Z}_d}$, A *folded over* $\boldsymbol{Z}_d$ by partitioning Z_d^S into sets of functions $\{f, \omega f, \ldots, \omega^{d-1} f\}$, for each part picking one representative f (by e.g, letting f be the function with a 1 in the first non-zero coordinate), and defining $A_{\boldsymbol{Z}_d}(\gamma f) = \gamma A(f)$.

Definition 5.10. A function A from Z_d^S to$\boldsymbol{Z}_d$ is k-homogeneous for $k \in \hat{\boldsymbol{Z}}_{\boldsymbol{Z}_d}^S$ if $A(gf) = g^k A(f)$.

We have the following lemma:

Lemma 5.11. If $B\colon Z_d^S \to \boldsymbol{C}$ is k-homogeneous then $\hat{B}_\alpha = 0$ unless $\sum_x \alpha(x) = k \pmod d$.

Proof.

$$\begin{aligned}
\hat{B}_\alpha &= d^{-|S|} \sum_{f \in Z_d^S} B(f)\overline{\chi_\alpha(f)} \\
&= d^{-|S|} \sum_{f \in Z_d^S} \frac{1}{d} \sum_{\gamma \in \boldsymbol{Z}_d} B(\gamma f)\overline{\chi_\alpha(\gamma f)} \\
&= d^{-|S|} \sum_{f \in Z_d^S} \frac{1}{d} \sum_{\gamma \in \boldsymbol{Z}_d} \gamma^k B(f)\overline{\chi_\alpha(\gamma f)} \\
&= d^{-|S|} \sum_{f \in Z_d^S} \frac{1}{d} B(f)\chi_\alpha(f) \sum_{\gamma \in \boldsymbol{Z}_d} \gamma^k \overline{\chi_\alpha(\gamma \mathbf{0})}
\end{aligned}$$

Now,

$$\begin{aligned}
\sum_{\gamma \in \boldsymbol{Z}_d} \gamma^k \overline{\chi_\alpha(\gamma \mathbf{0})} &= \sum_{\gamma \in \boldsymbol{Z}_d} \gamma^k \prod_{x \in S} \gamma^{-\alpha(x)} \\
&= \sum_{\gamma \in \boldsymbol{Z}_d} \gamma^k \gamma^{-\sum_x \alpha(x)}.
\end{aligned}$$

The last sum is 0 unless $\sum_x \alpha(x) = k \pmod d$. ∎

Note that in particular, if B is as in the lemma, $\hat{B}_{\mathbf{0}} = 0$.

5.2 A generalization of the two-prover game

In this thesis, we sometimes use a generalization of the u-parallel repetition of the basic 2P1R protocol that was applied by Khot [39] to Håstad's PCP for E3-Sat [28], thus simplifying Håstad's proof. In this section we review Khot's construction.

It turns out that a problem in the analysis of the particular type of PCP verifier that Håstad had in his PCP for **NP** with perfect completeness and soundness 7/8 is that a large set of satisfying assignments to the clauses $(C_1, \ldots, C_u)$ from Section 4.1 may project down to a very small set of assignments to the variables $(x_1, \ldots, x_u)$. Håstad solved this problem by making a very careful analysis of the PCP verifier. Khot recently obtained a simpler analysis by a modification of the basic 2P1R protocol from Section 4.1.

The generalized protocol is parameterized by both u and T (Setting $T = 0$ gives us the protocol from 4.1). In this version, the verifier selects at random a sequence W consisting of $(T+1)u$ clauses. It then selects at random a sequence of Tu clauses and u variables by selecting, uniformly at random from W, a sequence of u clauses and then selecting, independently and uniformly at random, a variable from each of those clauses.

Definition 5.12. Given a E3-CNF(5) formula ϕ, a (T,u)-*block selected from* ϕ is a sequence of $(T+1)u$ clauses from ϕ.

Definition 5.13. Given a (T,u)-block W, a *random u-projection of* W is a sequence formed by first selecting u clauses $C_{i_1}, \ldots, C_{i_u}$ uniformly at random from W, choosing a variable $x_k \in C_{i_k}$ uniformly at random from each of these u clauses, and constructing $U = (x_1, \ldots, x_u, C_{i_{u+1}}, \ldots, C_{i_{(T+1)u}})$ where $x_1, \ldots, x_u$ are the u variables so chosen, and $C_{i_1}, \ldots, C_{i_{Tu}}$ are the remaining Tu clauses (in their original order).

Similarly to the original parallel two-prover one-round protocol, we denote by SAT^W the satisfying assignments to the clauses in W, and by SAT^U the set of strings $\{0,1\}^{U'} \times \mathrm{SAT}^{W'}$, where U' is the sequence of variables in U and W' is the sequence of clauses in U. For a (T,u)-block $W = (C_1, \ldots C_{(T+1)u})$ and a u projection $U = (x_1, \ldots, x_u, C_{i_{u+1}}, \ldots, C_{i_{(T+1)u}})$ of W, constructed by choosing the clauses $C_{i_1}, \ldots, C_{i_u}$ from W and the variable x_k from C_{i_k}, we denote by $\pi^{U,W}$ the natural projection function from SAT^W to SAT^U, defined by

$$\pi^{U,W}(y_1, \ldots, y_{(T+1)u}) = (\pi^{x_1, C_{i_1}}(y_{i_1}), \ldots, \pi^{x_u, C_{i_u}}(y_{i_u}), y_{i_{u+1}}, \ldots, y_{i_{(T+1)u}})$$

The Generalized 2P1R game is given in Figure 5.1 .

Generalized Parallel 2P1R protocol.
Input: An E3-CNF(5) formula ϕ.

1. Choose uniformly at random a (T,u)-block $W = (C_1,\ldots,C_{(T+1)u})$ from ϕ.
2. Choose a random u-projection U from W.
3. Send W to P_1, receiving a string $y \in \mathrm{SAT}^W$.
4. Send U to P_2, receiving a string $x \in \mathrm{SAT}^U$.
5. Accept if $\pi^{U,W}(y) = x$

Figure 5.1. Generalized Parallel 2P1R game.

Lemma 5.14. The protocol described Figure 5.1 has perfect completeness and soundness c_μ^u, where $c_\mu < 1$ is some constant depending on μ but not on u or the size of the instance.

Proof. For a satisfiable formula, the verifier always accepts if the two provers answer according to the same satisfying assignment.

To prove that the verifier accepts unsatisfiable formulae with probability at most c_μ^u, we reduce from the protocol described in Section 4.1. It is known [43] that this protocol has soundness c_μ^u, where $c_\mu < 1$ is some constant depending on μ but not on u or the size of the instance.

Suppose that there exists provers Q_1 and Q_2 for the protocol from this section such that the verifier accepts an unsatisfiable formula with probability $s > c_\mu^u$. Then the provers P_1 and P_2 from the protocol described in Section 4.1 can use Q_1 and Q_2 to construct strategies that make the verifier in that protocol accept an unsatisfiable formula with probability s.

The essence of the proof is that, by an averaging argument, there is some way of fixing the Tu extra clauses given to Q_1 and Q_2 so as to not decrease the acceptance probability.

Recall that the verifier in our generalized protocol proceeds by picking $(T+1)u$ clauses and among these choosing u, from which it then chooses a variable each.

For clauses $C_1,\ldots,C_{tu}$ from the formula ϕ, denote by $A(C_1,\ldots,C_{tu})$ the event that the clauses from which the verifier *do not* choose any variables are $C_1,\ldots,C_{tu}$.

Then there must be some choice of $C'_1,\ldots,C'_{tu}$ such that

$$\Pr[\text{the verifier accepts} \mid A(C'_1,\ldots,C'_{tu})] \geq s.$$

Fix such a choice $C'_1,\ldots,C'_{tu}$.

Given a sequence of W of clauses $(C_1,\ldots,C_u)$, P_1 constructs a sequence W' of $(T+1)u$ clauses by inserting $C_1,\ldots,C_u$ randomly among $C'_1,\ldots,C'_{tu}$, (but

preserving the internal order of the sequences). The thereby obtained (T,u)-block is sent to Q_1 and P_1 then returns the assignment to $(C_1,\ldots,C_u)$ obtained from Q_1. Given a sequence of variables $(x_1,\ldots,x_u)$, P_2 constructs u-projection $U'=(x_1,\ldots,x_u,C'_1,\ldots,C'_{tu})$, which is sent to Q_2 and P_2 then returns the assignment to $(x_1,\ldots,x_u)$ obtained from Q_2.

Since the distribution of W' and U' so constructed is the same as the distribution of the queries to Q_1 and Q_2 in the generalized two-prover one-round protocol conditioned on the event $A(C_1,\ldots,C_{tu})$, the answers sent back by P_1 and P_2 make the verifier accept with probability at least s. But this is a contradiction since $s > c_\mu^u$. ∎

Definition 5.15. A *Generalized Standard Written d-Proof with parameters u and T* contains for each sequence U containing u variables and T clauses a string of length $d^{2^u 7^{Tu}}$, which we interpret as the table of a function $A_U\colon \boldsymbol{Z}_d^{\mathrm{SAT}^U} \to \boldsymbol{Z}_d$. It also contains for each sequence W of $(T+1)u$ clauses a string of length $d^{7^{(T+1)u}}$ which we interpret as the table of a function $A_W\colon \boldsymbol{Z}_d^{\mathrm{SAT}^W} \to \boldsymbol{Z}_d$.

Definition 5.16. A Generalized Standard Written d-Proof with parameters u and T is a *correct proof for a formula* ϕ if there is an assignment x, satisfying ϕ, such that A_U is the long code of x restricted to U for all sequences U containing u variables and Tu clauses, and A_W is the long code of x restricted to W for any sequences W of $(T+1)u$ clauses.

Projections

As we mentioned in Section 5.2, the 2P1R for E3-CNF(5) was modified to handle certain difficulties in the analysis of certain kinds of PCP verifiers. In this section, we formalize the properties needed in the analysis. The facts from this section are straightforward generalizations of the corresponding definitions and lemmas from Khot's paper [39].

Lemma 5.17. Let W be a (T,u)-block and consider two different assignments $y,y' \in \mathrm{SAT}^W$. Then

$$\Pr[\pi^{U,W}(y) \neq \pi^{U,W}(y')] > 1 - \frac{1}{T}$$

where the probability is over the selection of a random u-projection of W.

Proof. Since $y \neq y'$, there is at least one variable that is assigned different values by y and y'. If this clause is present in the u-projection—which happens with probability $T/(T+1) > 1-1/T$—the projections are certainly different. ∎

Corollary 5.18. Let W be a (T,u)-block and consider a $\beta\colon \mathrm{SAT}^W \to \hat{\boldsymbol{Z}}_d$. Then, with probability at least $1-|\beta|/T$ over the choice of a random u-projection U of W, it holds for an arbitrary $y \in \mathrm{SAT}^W$ that there is no $y' \neq y$ such that $\beta(y') \neq 0$ and $\pi^{U,W}(y) = \pi^{U,W}(y')$.

Proof. Take an arbitrary $y \in \mathrm{SAT}^W$. By Lemma 5.17, for any $y' \in \mathrm{SAT}^W \setminus \{y\}$,

$$\Pr[\pi^{U,W}(y) = \pi^{U,W}(y')] < \frac{1}{T}$$

where the probability is over the selection of a random u-projection of W and π is the projection induced by the u-projection of W. By the union bound,

$$\Pr\left[\bigcup_{\substack{y' \in \mathrm{SAT}^W \setminus \{y\} \\ \beta(y) \neq 0}} \{\pi^{U,W}(y) = \pi^{U,W}(y')\}\right] < \frac{|\beta|}{T}$$

where the probability is over the same probability space. ∎

We define the projection function $\pi^{U,W} : \hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W} \to 2^{\mathrm{SAT}^U}$, by $x \in \pi^{U,W}(\beta)$ if there is an y such that $\beta(y) \neq 0$ and $\pi^{U,W}(y) = x$.

Lemma 5.19. Let W be a (T,u)-block and β be a function from SAT^W to $\hat{\boldsymbol{Z}}_d$. Then

$$\mathrm{E}\left[\frac{1}{|\pi^{U,W}(\beta)|}\right] \leq \frac{1}{|\beta|} + \frac{1}{T}$$

where the probability is over the selection of a random u-projection of W.

Proof. Since every $y \in \mathrm{SAT}^W$ projects down to exactly one $x \in \mathrm{SAT}^U$,

$$|\beta| = \sum_{x \in \pi^{U,W}(\beta)} |\{y \in \mathrm{SAT}^W : \pi^{U,W}(y) = x \wedge \beta(y) \neq 0\}|.$$

Now apply the Cauchy–Schwartz inequality to the above equation.

$$\begin{aligned}
|\beta|^2 &= \left(\sum_{x \in \pi^{U,W}(\beta)} 1 \cdot |\{y \in \mathrm{SAT}^W : \pi^{U,W}(y) = x \wedge \beta(y) \neq 0\}|\right)^2 \\
&\leq \left(\sum_{x \in \pi^{U,W}(\beta)} 1^2\right)\left(\sum_{x \in \pi^{U,W}(\beta)} |\{y \in \mathrm{SAT}^W : \pi^{U,W}(y) = x \wedge \beta(y) \neq 0\}|^2\right) \\
&= |\pi^{U,W}(\beta)| \sum_{x \in \pi^{U,W}(\beta)} |\{y \in \mathrm{SAT}^W : \pi^{U,W}(y) = x \wedge \beta(y) \neq 0\}|^2 \\
&= |\pi^{U,W}(\beta)| N_{\pi^{U,W}}(\beta),
\end{aligned}$$

where $N_{\pi^{U,W}}(\beta)$ is the numbers of pairs $(y_1, y_2) \in \mathrm{SAT}^W \times \mathrm{SAT}^W$ such that $\pi^{U,W}(y_1) = \pi^{U,W}(y_2)$, $\beta(y_1) \neq 0$ and $\beta(y_2) \neq 0$. Hence,

$$\frac{1}{|\pi^{U,W}(\beta)|} \leq \frac{N_{\pi^{U,W}}(\beta)}{|\beta|^2}.$$

Now introduce for each pair $(y_1, y_2) \in \mathrm{SAT}^W \times \mathrm{SAT}^W$ an indicator random variable $I_{(y_1,y_2)}$ that is one if $\pi^{U,W}(y_1) = \pi^{U,W}(y_2)$, $\beta(y_1) \neq 0$ and $\beta(y_2) \neq 0$. Then

$$\begin{aligned}
\mathrm{E}\left[\frac{1}{|\pi^{U,W}(\beta)|}\right] &= \frac{1}{|\beta|^2} \sum_{\substack{y_1 \in \mathrm{SAT}^W \\ \beta(y_1)\neq 0}} \sum_{\substack{y_2 \in \mathrm{SAT}^W \\ \beta(y_2)\neq 0}} \mathrm{E}[I_{(y_1,y_2)}] \\
&= \frac{1}{|\beta|^2}\left(|\beta| + \sum_{\substack{y_1 \in \mathrm{SAT}^W \\ \beta(y_1)\neq 0}} \sum_{\substack{y_2 \in \mathrm{SAT}^W \\ \beta(y_2)\neq 0 \\ y_1 \neq y_2}} \mathrm{E}[I_{(y_1,y_2)}]\right) \\
&\leq \frac{1}{|\beta|^2}\left(|\beta| + \sum_{\substack{y_1 \in \mathrm{SAT}^W \\ \beta(y_1)\neq 0}} \sum_{\substack{y_2 \in \mathrm{SAT}^W \\ \beta(y_2)\neq 0 \\ y_1 \neq y_2}} \frac{1}{T}\right) \\
&= \frac{1}{|\beta|} + \frac{|\beta|-1}{|\beta|T} \leq \frac{1}{|\beta|} + \frac{1}{T},
\end{aligned}$$

where the first inequality follows from Lemma 5.17. ∎

Corollary 5.20. Let W be a (T,u)-block and β be a function from SAT^W to $\hat{\boldsymbol{Z}}_d$. Then $|\pi^{U,W}(\beta)| \geq \delta \min\{|\beta|, T\}$ with probability $1 - 2\delta$, where the probability is over the selection of a random u-projection of W and $\pi^{U,W}$ is the projection induced by the u-projection of W.

Proof. Apply Markov's inequality to the conclusion of Lemma 5.19. ∎

We also have another kind of projection: For $\beta \in \hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}$ and $\alpha \in \hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^U}$ we define *the modulo d projection of* β, $\pi_d^{U,W} \in \hat{\boldsymbol{Z}}_{\boldsymbol{Z}_d}^S$ as

$$\pi_d^{U,W}(x) = \sum_{y:\pi^{U,W}(y)=x} \beta(x) \pmod d \tag{5.8}$$

Lemma 5.21. Let $f \in Z_d^{\mathrm{SAT}^U})$ and $\beta \in \hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}$.

$$\chi_\beta(f \circ \pi^{U,W}) = \chi_{\pi_d^{U,W}(\beta)}(f). \tag{5.9}$$

Proof. We expand χ_β according to its definition:

$$\chi_\beta(f \circ \pi^{U,W}) = \prod_{y \in \mathrm{SAT}^W} (f \circ \pi^{U,W})(y)^{\beta(y)}.$$

Grouping the $y \in \mathrm{SAT}^W$ according to their projection, we have that

$$\begin{aligned}\prod_{y\in\mathrm{SAT}^W}(f\circ\pi^{U,W})(y)^{\beta(y)} &= \prod_{x\in\mathrm{SAT}^U}\prod_{y:\pi^{U,W}(y)=x} f(x)^{\beta(y)}\\ &= \prod_{x\in\mathrm{SAT}^U} f(x)^{\sum\beta(y) \pmod d}\\ &= \chi_{\pi_d^{U,W}(\beta)}(f).\end{aligned}$$

■

Part II

New Results

Chapter 6

A generalization of Minimum Bisection

6.1 Introduction

PCP techniques have been more successful in some areas than others. Notably, they have been extremely successful for constraint satisfaction problems (CSPs) where we are given a set of boolean constraints over some variables and asked to find an assignment satisfying as many constraints as possible. For example, we saw in Chapter 4 an optimal hardness result for Maximum E3-Lin-2, which is a constraint satisfaction problem.

One class of problems where PCPs have failed so far to get any results at all is CSPs where the solution space is restricted, and the problem is easy to solve without the restriction. The most well-known examples are Minimum Bisection, where we are given a graph and are asked to divide the vertices in two *equal sized* parts such that the number of edges between the parts are minimized, and Maximum Edge Subgraph in which we are given a graph and are asked to pick a set of k vertices such that the number of edges in the induced subgraph is maximized. These problems become trivial to solve if we remove the restrictions, i.e., if we are allowed parts of unequal size and if we are allowed to pick any number of vertices.

For Minimum Bisection, the problem is not known to be **NP**-hard to approximate even within a constant factor, while the best known algorithm approximates the problem within $O(\log^2 n)$ [19] (Several special cases of Minimum Bisection admit better approximation algorithms, see [37] for a survey.)

Using a stronger assumption that $\mathbf{P} \neq \mathbf{NP}$, namely that in a certain sense, refuting CNF formulas is hard on average, Feige [17] recently proved a lower bound of 4/3 on the approximability of Minimum Bisection.

In this chapter we prove, under standard complexity theoretic assumptions, that a generalization of Minimum Bisection is very hard to approximate. More

specifically, we prove that Minimum E3-UnSat Balanced-Lin-2 is **NP**-hard to approximate within any constant. Recall that Minimum E3-UnSat Balanced-Lin-2 is the problem where we are given a set of homogeneous equations modulo 2, with three variables in each equation, and want to find a balanced assignment to the variables which minimizes the number of unsatisfied equations. For the case of four equations in each equation, i.e., Minimum E4-UnSat Balanced-Lin-2, it has been pointed out to us by Ran Raz that the weaker result that this problem is **NP**-hard to approximate within some constant can be proven by a reduction from Maximum E3-Lin-2.

We also consider the maximization version, where we want to maximize the number of satisfied equations. We prove that this problem is **NP**-hard to approximate within $4/3 - \delta$.

The proof introduces some ideas which we will later use to prove inapproximability of vertex cover generalized to hypergraphs.

We use the approach described in Chapter 4. That is, we construct a specially tailored PCP for **NP** such that computing the maximum acceptance probability is exactly the problem Maximum Balanced-Lin-2. This means that the proof presented to the verifier will be required to contain an equal number of 0 and 1, and the test used by the verifier will be that an exclusive or of three variables is zero.

6.2 The PCP

The verifier will be essentially the same as Håstad's verifier described in Chapter 4, however, since we want to ensure that the right-hand side of each equation is always 1 (in $\{1, -1\}$-notation), we cannot use folding. Since Håstad's verifier accesses folded tables, it will for half of the functions use the value $-A_U(-f)$ instead of $A_U(f)$. This means that, as a function of the actual bits in the proof, the test is not always $A_U(f)A_W(g)A_W(h) = 1$. It may equally well be that the test is (for instance) $A_U(-f)A_W(g)A_W(h) = -1$.

Folding ensures that every table A_U and A_W contain an equal number of -1 and 1; we no longer have that guarantee. In its place we will use that the prover is restricted to proofs which contain an equal number of -1 and 1 globally. Since the tables corresponding to sequences of variables U contains an insignificant fraction of the bits, the prover can have all tables A_U constant 1, almost $1/2$ of the W-tables constant 1 and the others constant -1. This will always fool the test as described above, since two of the bits are queried from the same A_W, and thus their product is always 1 when the table is constant. We fix this by modifying the Standard Written Proof so that the U-tables instead contain almost all of the bits.

6.2.1 A variant of the Standard Written Proof

We need to modify the Standard Written Proof as described previously somewhat, since we have two classes of tables (corresponding to sets U of variables and sets

W of clauses), and we want to be able to control how many bits go into each class.

Recall that the input to the verifier is a E3-CNF(5) formula ϕ. If there are n variables, this means that the number of clauses $m = \frac{5n}{3}$. Let $D(u) = K(2^{7^u}(5/3)^u)/2^{2^u}$, where K is a constant to be determined later. For each U we have $D(u)$ tables will we denote by $A_{U,\kappa}$ for $1 \leq \kappa \leq D(u)$. In the protocol when making a query we will choose a κ at random and look in the corresponding table.

Definition 6.1. A *Standard Written Proof with parameter u and K* for a E3-CNF(5) formula ϕ consists of, for each sequence U of u variables from ϕ and for each κ where $1 \leq \kappa \leq D(u)$, a table $A_{U,\kappa} \colon 2^{\{0,1\}^U} \to \{-1,1\}$, where $D(u) = K(2^{7^u}(5/3)^u)/2^{2^u}$, and a table $A_W \colon 2^{\mathrm{SAT}^W} \to \{-1,1\}$ for each sequence W of u clauses from ϕ.

Definition 6.2. For $0 \leq p \leq 1$, a proof is p-balanced if a fraction p of the bits are -1. We say that a proof is *balanced* if it is $1/2$-balanced.

Definition 6.3. A Standard Written Proof with parameter u and K is a *correct proof* for a formula ϕ of n variables if there is an assignment x, satisfying ϕ, such that $A_{U,\kappa}$ is the long code of $x|_U$ for all sequences U of u variables, and A_W is the long code $x|_W$ for all sequences W of u clauses.

Since each table is a long code, and a long code contain the same number of -1 as 1, it follows that a correct proof is always balanced.

Since the tables corresponding to sequences U of variables contains almost all of the bits (depending on the choice of K), we have that, for a p-balanced proof, the tables corresponding to sequences U must be almost p-balanced. The next lemma make this precise, reformulated in terms of the discrete Fourier transform:

Lemma 6.4. For a p-balanced proof,

$$\left| E_{U,\kappa}\left[\hat{A}_{U,\kappa,\emptyset}\right] - (1-2p)\right| \leq \frac{2}{K+1} \tag{6.1}$$

Proof. Recall that

$$\hat{A}_{U,\kappa,\emptyset} = \mathop{\mathrm{E}}_f[A_{U,\kappa}(f)] = \Pr_f[A_{U,\kappa}(f) = 1] - \Pr_f[A_{U,\kappa}(f) = -1].$$

Since the proof is p-balanced, we have

$$\frac{2^{2^u}\sum_{U,\kappa}\hat{A}_{U,\kappa,\emptyset} + 2^{7^u}\sum_W \hat{A}_{W,\emptyset}}{2^{2^u}n^u D(u) + 2^{7^u}(5n/3)^u} = 1 - 2p. \tag{6.2}$$

Thus

$$\left|\frac{\sum_{U,\kappa}\hat{A}_{U,\kappa,\emptyset}}{n^u D(u)} - (1-2p)\right| =$$

$$\left|\frac{\sum_{U,\kappa}\hat{A}_{U,\kappa,\emptyset}}{n^u D(u)} - \frac{2^{2^u}\sum_{U,\kappa}\hat{A}_{U,\kappa,\emptyset} + 2^{7^u}\sum_W \hat{A}_{W,\emptyset}}{2^{2^u}n^u D(u) + 2^{7^u}(5n/3)^u}\right|$$

which, by the definition of $D(u)$, is equal to

$$\left| \frac{2^{2^u} \sum_{U,\kappa} \hat{A}_{U,\kappa,\emptyset}}{K 2^{7^u} (5/3)^u n^u} - \frac{2^{2^u} \sum_{U,\kappa} \hat{A}_{U,\kappa,\emptyset}}{(K+1) 2^{7^u} (5/3)^u n^u} - \frac{2^{7^u} \sum_{W} \hat{A}_{W,\emptyset}}{(K+1) 2^{7^u} (5/3)^u n^u} \right|.$$

Now, $\sum_W |\hat{A}_{W,\emptyset}| \leq (5/3)^u n^u$, so

$$\left| \frac{2^{7^u} \sum_{W} \hat{A}_{W,\emptyset}}{(K+1) 2^{7^u} (5/3)^u n^u} \right| \leq \frac{1}{K+1}$$

and

$$\frac{2^{2^u} \sum_{U,\kappa} \hat{A}_{U,\kappa,\emptyset}}{K 2^{7^u} (5/3)^u n^u} - \frac{2^{2^u} \sum_{U,\kappa} \hat{A}_{U,\kappa,\emptyset}}{(K+1) 2^{7^u} (5/3)^u n^u} = \frac{2^{2^u} \sum_{U,\kappa} \hat{A}_{U,\kappa,\emptyset}}{K(K+1) 2^{7^u} (5/3)^u n^u}$$

where the last expression has absolute value at most $1/(K+1)$, since

$$\sum_{U,\kappa} |\hat{A}_{U,\kappa,\emptyset}| \leq \frac{K 2^{7^u} (5/3)^u n^u}{2^{2^u}}.$$

■

6.2.2 The verifier

Now we are ready to describe our verifier. It is given in Figure 6.1. The completeness of the test is straightforward:

Lemma 6.5. If ϕ is satisfiable, there is a balanced proof which HomLinTest accepts with probability at least $1 - \delta$.

The soundness of the test is not-so-obvious. We have to prove that if verifier accepts with large enough probability, then there is a good strategy for the provers P_1 and P_2 in the 2P1R game for E3-CNF(5) from Sec. 4.1. The probability that the verifier accepts can be written as

$$\frac{1 + \mathrm{E}_{U,W,\kappa,f,g,e}[A_U(f) A_W(g) A_W((f \circ \pi^{U,W}) g)]}{2}. \tag{6.3}$$

Hence we need to compute

$$\mathop{\mathrm{E}}_{U,W,\kappa,f,g,e}[A_{U,\kappa}(f) A_W(g) A_W((f \circ \pi^{U,W}) g)]. \tag{6.4}$$

Applying the Fourier-transform, we get that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,g,e}[A_{U,\kappa}(f) A_W(g) A_W((f \circ \pi^{U,W}) g)] \\
&= \sum_{\beta \subseteq \mathrm{SAT}^W} \hat{A}_{U,\kappa,\pi_U^2(\beta)} \hat{A}_{W,\beta} (1-2\delta)^{|\beta|} \\
&= \sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta) = \emptyset}} \hat{A}_{U,\kappa,\emptyset} \hat{A}_{W,\beta} (1-2\delta)^{|\beta|} + \sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta) \neq \emptyset}} \hat{A}_{U,\kappa,\pi_2^{U,W}(\beta)} \hat{A}_{W,\beta} (1-2\delta)^{|\beta|}.
\end{aligned}$$

HomLinTest .
Input: An E3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Standard Written Proof with parameters u and K.

1. Select uniformly at random a sequence $W = (C_1, \ldots, C_u)$ of u clauses and a sequence U of u variables by for each clause C_k choosing uniformly at random a variable occurring in C_k.
2. Select uniformly at random $\kappa \in \{1, \ldots, D(n)\}$.
3. Select uniformly at random $f \in 2^{\{0,1\}^U}$ and $g \in 2^{SAT^W}$.
4. Select $e \in 2^{SAT^W}$ by for each y independently setting $e(y) = 1$ with probability $1 - \delta$ and $e(y) = -1$ with probability δ.
5. Accept if $A_{U,\kappa}(f)A_W(g)A_W((f \circ \pi^{U,W})g) = 1$, else reject.

Figure 6.1. The above PCP is parameterized by the positive integers u and K and the positive real δ and tests if a E3-CNF(5) formula ϕ is satisfiable by querying three positions in a Standard Written Proof with parameters u and K. With suitable choices of the parameters u, K and δ, the above PCP has completeness $1 - \epsilon$ and soundness $\frac{2-p}{2} + \epsilon$ for any $\epsilon > 0$ and a p-balanced proof.

Hence we need to consider expressions on the form

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta)=\emptyset}} \hat{A}_{U,\kappa,\emptyset}\hat{A}^2_{W,\beta}(1-2\delta)^{|\beta|}\right]$$

and

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta)\neq\emptyset}} \hat{A}_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\beta}(1-2\delta)^{|\beta|}\right].$$

From the proof of soundness of Håstad's verifier in Chapter 4, we expect that we can extract a strategy from the latter terms, and this will indeed turn out to be the case. We first concentrate on bounding the terms of the first kind.

Lemma 6.6. For a p-balanced proof,

$$\mathop{\mathrm{E}}_{U,\kappa}[\max(0, \hat{A}_{U,\kappa,\emptyset})] \leq (1-p) + \frac{1}{1+K}.$$

Proof.

$$\mathop{\mathrm{E}}_{U,\kappa}[\hat{A}_{U,\kappa,\emptyset}] = \mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})] + \mathop{\mathrm{E}}_{U,\kappa}[\min(0,\hat{A}_{U,\kappa,\emptyset})].$$

Now,

$$\mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})] - \mathop{\mathrm{E}}_{U,\kappa}[\min(0,\hat{A}_{U,\kappa,\emptyset})] = E_{U,\kappa}[|\hat{A}_{U,\kappa,\emptyset}|] \leq 1$$

and by Lemma 6.4

$$\mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})] + \mathop{\mathrm{E}}_{U,\kappa}[\min(0,\hat{A}_{U,\kappa,\emptyset})] \leq (1-2p) + \frac{2}{K+1},$$

it follows that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})] \\
&\quad\leq\ 1 + \mathop{\mathrm{E}}_{U,\kappa}[\min(0,\hat{A}_{U,\kappa,\emptyset})] \\
&\quad\leq\ 1 + (1-2p) + \frac{2}{K+1} - \mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})].
\end{aligned}$$

So

$$\mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})] \leq (1-p) + \frac{1}{1+K}.$$

■

Lemma 6.7. For a p-balanced proof,

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta\subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta)=\emptyset}} \hat{A}_{U,\kappa,\emptyset}\hat{A}^2_{W,\beta}(1-2\delta)^{|\beta|}\right] \leq (1-p) + 1/(1+K). \tag{6.5}$$

Proof. Since by Parseval's equality, $0 \leq \sum_{\pi_2^{U,W}(\beta)=\emptyset} \hat{A}^2_{W,\beta} \leq 1$, it follows that

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta\subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta)=\emptyset}} \hat{A}_{U,\kappa,\emptyset}\hat{A}^2_{W,\beta}(1-2\delta)^{|\beta|}\right] \leq \mathop{\mathrm{E}}_{U,\kappa}[\max(0,\hat{A}_{U,\kappa,\emptyset})].$$

By Lemma 6.6, the lemma follows. ■

Lemma 6.8. If HomLinTest accepts a p-balanced proof with probability at least $\frac{2-p}{2} + \frac{1}{2(K+1)} + \delta$, there is a strategy for the 2P1R game with success probability at least $8\delta^3$.

Proof. Recall that the probability that the verifier accepts may be written as $\frac{1+q}{2}$, where q is

$$\begin{aligned}&\mathop{\mathrm{E}}_{U,W,\kappa,f,g,e}[A_{U,\kappa}(f)A_W(g)A_W((f\circ\pi^{U,W})g)]\\ &\quad=\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\beta\subseteq\mathrm{SAT}^W}\hat{A}_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{B}_{W,\beta}(1-2\delta)^{|\beta|}\right].\end{aligned}$$

By Lemma 6.7, the terms where $\pi_2^{U,W}(\beta)=\emptyset$ contribute at most $(1-p)+\frac{1}{1+K}$, so the terms where $\pi_2^{U,W}(\beta)\neq\emptyset$ must contribute at least 2δ.

First note that, for fixed U,W and κ, by the Cauchy–Schwartz inequality,

$$\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\beta}(1-2\delta)^{|\beta|}\leq \tag{6.6}$$

$$\sqrt{\left(\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}^2_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\beta}(1-2\delta)^{2|\beta|}\right)\left(\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}^2_{W,\beta}\right)}. \tag{6.7}$$

Now,

$$\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}^2_{W,\beta}\leq\sum_{\beta\subseteq\mathrm{SAT}^W}\hat{A}^2_{W,\beta}=1 \tag{6.8}$$

by Parseval's equality, so the right-hand side of (6.7) is at most

$$\sqrt{\left(\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}^2_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\beta}(1-2\delta)^{2|\beta|}\right)}. \tag{6.9}$$

Thus, using $\mathrm{E}[X^2]\geq\mathrm{E}[X]^2$ (Jensen's inequality), we have

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta\subseteq\mathrm{SAT}^W\\ \pi_2^{U,W}(\beta)\neq\emptyset}}\hat{A}^2_{U,\kappa,\pi_2^{U,W}(\beta)}\hat{A}^2_{W,\beta}(1-2\delta)^{2|\beta|}\right]\geq 4\delta^2. \tag{6.10}$$

Now consider the following strategy for the provers P_1 and P_2: The prover P_1 receives a sequence W of u clauses whereupon P_1 picks a β with probability $\hat{A}^2_{W,\beta}$.

Finally P_1 picks a $y \in \beta$ uniformly at random, and returns it. The prover P_2 receives a sequence U of u variables, whereupon P_2 chooses a random $\kappa \in \{1, \ldots, D(u)\}$ and a random α with probability $\hat{A}^2_{U,\kappa,\alpha}$. Then P_2 picks a $x \in \alpha$ uniformly at random, and returns this x as its answer. If either P_1 or P_2 picks an empty α or β, they give up. The probability of convincing the verifier is at least

$$\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta) \neq \emptyset}} \hat{A}^2_{U,\kappa,\pi_2^{U,W}(\beta)} \hat{A}^2_{W,\beta} |\beta|^{-1}\right]. \tag{6.11}$$

Since $e^x \geq 1 + x \geq x$ for real x, $\frac{e^{2\delta|\beta|}}{2\delta} \geq \frac{2\delta|\beta|}{2\delta} = |\beta|$, thus we have that $|\beta|^{-1} \geq 2\delta e^{-2\delta|\beta|} \geq 2\delta(1-2\delta)^{|\beta|}$, where we again used $e^x \geq 1 + x$ in the last inequality (with negative x). We have that (6.11) is at least

$$2\delta \mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\substack{\beta \subseteq \mathrm{SAT}^W \\ \pi_2^{U,W}(\beta) \neq \emptyset}} \hat{A}^2_{U,\kappa,\pi_2^{U,W}(\beta)} \hat{A}^2_{W,\beta} (1-2\delta)^{2|\beta|}\right]. \tag{6.12}$$

By (6.10), this expression is at least $8\delta^3$. ∎

The analysis above is essentially tight; consider the case of $p = 0$ and the following strategy for a cheating prover: Color a fraction q of the variables blue where q is chosen so that exactly $1/2$ of the sequences U contains at least one blue variable. Call these U blue, and the sequences U which contain no blue variables white. Similarly we call sequences W of clauses blue if they contain any blue variables, and white otherwise. For each blue U, let $A_{U,\kappa}$ be constant -1 for $1 \leq \kappa \leq D(u)$, and for each white U let $A_{U,\kappa}$ be constant 1. For each blue A_W, let $y \in \mathrm{SAT}^W$ be any assignment such that the assignment y', obtained by flipping the blue variables is still in SAT^W (such an y will almost always exist). Let $A_W = \chi_{\{y,y'\}}$. For the white W let A_W be constant 1. Now, if the verifier pick a white U and a white W, this strategy always convinces the verifier. If the verifier picks a white U and a blue W, the strategy succeeds with probability at least $(1-\delta)^2$, since $A_{U,\kappa}(f) = 1$ and

$$\begin{aligned} A_W(g)A_W((f \circ \pi^{U,W})ge) &= \chi_{\{y,y'\}}(g)\chi_{\{y,y'\}}((f \circ \pi^{U,W})ge) \\ &= \chi_{\{y,y'\}}((f \circ \pi^{U,W})e) \\ &= (f \circ \pi^{U,W})(y)(f \circ \pi^{U,W})(y')e(y)e(y'). \end{aligned}$$

Now $\pi^{U,W}(y) = \pi^{U,W}(y')$ since y and y' only differ in blue variables, of which there are none in U, so $A_W(g)A_W((f \circ \pi^{U,W})ge) = 1$ with probability at least $(1-\delta)^2$. If the verifier picks a blue U and a blue W, $\pi^{U,W}(y) \neq \pi^{U,W}(y')$, since U contains

at least one variable in which y and y' differ, so $A_W(g)A_W((f \circ \pi^{U,W})ge)$ will be uniformly distributed, and the strategy will succeed with probability $1/2$. Since all variables in U occur in the clauses of W, the verifier will never choose a blue U and a white W. Hence the probability of success is at least

$$\Pr[U \text{ blue and } W \text{ blue}]\frac{1}{2} + (1 - \Pr[U \text{ blue and } W \text{ blue}])(1-\delta)^2.$$

Since the verifier chooses U uniformly and U is blue with probability $1/2$, $\Pr[U \text{ blue and } W \text{ blue}] = 1/2$, and thus the strategy succeeds with probability at least $\frac{2(1-\delta)^2+1}{4}$, which is essentially $3/4$ when δ is small.

6.3 Hardness of Approximating Balanced Homogeneous Equations

In this section we prove inapproximability results for Maximum Balanced-Lin-2 and Minimum UnSat Balanced-Lin-2 by reducing from the proof system in the previous section.

Lemma 6.9. Let $L \in \mathbf{NP}$. Then for any $\delta_0 > 0$, given x it is possible to construct a system of linear equations modulo 2 on n variables (of polynomial size), with all right-hand sides 0 and where each equation contains exactly three variables, such that the following holds:

- If $x \in L$, then there is a balanced assignment to the variables which satisfies at least a fraction $1 - \delta_0$ of all equations
- If $x \notin L$ then any p-balanced assignment satisfies at most a fraction $\frac{2-p}{2} + \delta_0$ of all equations.

Proof. Choose the parameters for HomLinTest so that $\delta = \frac{\delta_0}{2}$, $\frac{1}{2(K+1)} \leq \frac{\delta_0}{2}$, and the parameter u so that the soundness in the 2P1R game $c_\mu^u < 8\delta^3$. For the resulting verifier, write down for each random string the corresponding equation. Since the verifier uses $O(\log n)$ random bits, this will be a polynomial number of equations. By Lemma 6.5, if ϕ is satisfiable there is a balanced assignment which satisfies at least a fraction $1 - \delta = 1 - \delta_0/2 \geq 1 - \delta_0$ of all equations. By Lemma 6.8, if ϕ is at most μ-satisfiable, and there is a p-balanced assignment which satisfies as least a fraction $\frac{2-p}{2} + \frac{1}{2(K+1)} + \delta \leq \frac{2-p}{2} + \delta_0/2 + \delta_0/2$ of all equations, there is a strategy for the 2P1R game which makes the verifier accept with probability at least $8\delta^3$, a contradiction. The lemma follows since by Theorem 4.1 it is **NP**-hard to distinguish between the cases that ϕ is satisfiable and that ϕ is at most μ-satisfiable. ∎

Now we can prove inapproximability of Minimum E3-UnSat Balanced-Lin-2 .

Theorem 6.10. For any $c \geq 1$, Minimum E3-UnSat Balanced-Lin-2 cannot be approximated within c in polynomial time unless $\mathbf{P} = \mathbf{NP}$.

Proof. In Lemma 6.9 choose the parameter $\delta_0 < \frac{1}{8c}$.

In the case when the input $x \in L$, there is a balanced assignment which satisfies all but at most a fraction δ_0 of all equations, while in case $x \notin L$, at least $\frac{1}{8}$ of all equations are unsatisfied by any balanced assignment. Hence if we could approximate Minimum E3-UnSat Balanced-Lin-2 within c, we could decide an **NP**-complete problem. ∎

For the maximization version, we have the following:

Theorem 6.11. For any $\epsilon > 0$, Maximum E3-Balanced-Lin-2 cannot be approximated within $4/3 - \epsilon$ in polynomial time, unless $\mathbf{P} = \mathbf{NP}$.

Proof. Choose δ in Lemma 6.9 such that

$$\frac{1-\delta_0}{\frac{3}{4}-\delta_0} \leq \frac{4}{3} - \epsilon.$$

∎

There are "pseudo-approximation" algorithms for Minimum Bisection which relax the condition that the parts should be equal. For example it is possible to find a solution where the smallest part is at least $1/3$ and the number of edges cut is within a factor $O(\log n)$ of the minimum bisection [41]. Because of this, it is interesting with negative results of the same kind. Indeed, for the minimization problem we can prove that the problem is still hard to approximate within any constant for this kind of algorithm:

Theorem 6.12. For any $0 < p < 1$, for any $c \geq 1$ there is no polynomial time algorithm with finds a p balanced solution which approximates the optimum of Minimum E3-UnSat Balanced-Lin-2 within a factor c, unless $\mathbf{P} = \mathbf{NP}$.

Proof. Choose $\delta_0 < \frac{p}{2(c+1)}$ in Lemma 6.9. Then if $x \in L$ there are at most $\frac{p}{2(c+1)}$ unsatisfied equations for a balanced assignment, while if $x \notin L$, for a p-balanced assignment there are at least a fraction

$$\frac{p}{2} - \delta_0 \geq \frac{p}{2} - \frac{p}{2(c+1)} = \frac{pc}{2(c+1)}$$

unsatisfied equations. Thus if the p-balanced solution approximates within

$$\frac{pc}{2(c+1)} \Big/ \frac{p}{2(c+1)} = c$$

we could decide an **NP**-complete language. ∎

Theorem 6.13. For any $0 < p < 1$, for any $\epsilon > 0$, there is no polynomial time algorithm which finds a p-balanced solution which approximates the optimum of Maximum E3-Balanced-Lin-2 within $\frac{2}{2-p} - \epsilon$.

Proof. Choose the parameter δ_0 in Lemma 6.9 so that

$$\frac{1-\delta_0}{\frac{2-p}{2}+\delta_0} \leq \frac{2}{2-p} - \epsilon.$$

■

6.4 Conclusions and open problems

The most interesting question is whether these techniques can be applied to Minimum Bisection, and other similar problems. However, designing a PCP for Minimum Bisection seems difficult. It has to read two bits and check that they are equal, and we don't know how to do this with current techniques. An alternative would be to try find a reduction from our problem to Minimum Bisection, since they are somewhat similar, this might be a workable approach.

We have proven lower bounds; what about upper bounds? For Maximum Ek-Balanced-Lin-2 there is a 2-approximation algorithm – the algorithm which just partitions the variables into two equal sized parts at random will satisfy a fraction $1/2 - o(1)$ of all variables. For Minimum Ek-UnSat Balanced-Lin-2 we only know of an approximation algorithm for the case $k = 2$ i.e., Minimum Bisection. As mentioned previously, there is an $O(\log^2 n)$-approximation algorithm for Minimum Bisection [19]. It seems very unlikely that this algorithm would extend to $k > 2$, since it is very graph-theoretical in nature.

Chapter 7

Vertex cover on k-uniform hypergraphs

7.1 Introduction

Consider Minimum Vertex Cover, i.e., the problem in which we are given a graph and are asked to find a minimum set of vertices S such that each edge has an end-point in S.

It is well-known that this problem is **NP**-hard to solve exactly, and it was recently proven by Dinur and Safra [10] that it is **NP**-hard to approximate within $10\sqrt{5} - 21 - \epsilon \approx 1.3606$. The best known algorithm approximates the problem within a factor $2 - o(1)$

Now consider the generalization to k-uniform hypergraphs. We call this problem Minimum Ek-Vertex Cover. Since Minimum Vertex Cover on general hypergraphs is equivalent to Minimum Set Cover, it follows by a result of Feige [16] that the problem is "almost" **NP**-hard to approximate within a factor $(1 - \epsilon) \ln n$ for any $\epsilon > 0$. This result is essentially tight since there is an $1 + \ln n$-approximation algorithm [34]. It is also known that the problem is **NP**-hard to approximate to within $\Omega(\log n)$ [44].

A less well studied case is what happens when k is a constant not equal to 2. There is a trivial approximation preserving reduction from Minimum Ek-Vertex Cover to Minimum E(k+1)-Vertex Cover (just add one unique element to each set), so the hardness result for Minimum Vertex Cover also applies to Minimum Ek-Vertex Cover for $k > 2$. The best known algorithm is by Halperin and approximates Minimum Ek-Vertex Cover within $k - \frac{(k-1) \ln \ln n}{\ln n}$ [26], so this is even farther from tight than for Minimum Vertex Cover. Since the general problem where the size of the sets is unbounded is hard to approximate within a logarithmic factor, we would expect Minimum Ek-Vertex Cover to get harder to approximate as k grows. Indeed, Trevisan [52] proved that asymtoptically Minimum Ek-Vertex Cover is **NP**-hard to

approximate within $\Omega(k^{1/19})$

In this chapter we improve upon the result of Trevisan and prove that Minimum Ek-Vertex Cover is **NP**-hard to approximate within $\Omega(k^{1-\epsilon})$, for any $\epsilon > 0$.

There are two sources of our improvement over Trevisan's result. Firstly, we give an improved reduction from a PCP system to Minimum Vertex Cover on hypergraphs. By this reduction, we have that if **NP** can be characterized by a q query PCP with answers from a domain of size d, with perfect completeness and soundness $s(d)$ (i.e., correct proofs are accepted with probability 1 and incorrect proofs are accepted with probability at most $s(d)$), then Minimum Vertex Cover on $q(d-1)$-uniform hypergraphs is **NP**-hard to approximate to within $s(d)^{1/q}$.

Secondly, we use a different PCP characterization of **NP**. Trevisan constructs a 3 query PCP with perfect completeness, answer domain of size $d = 2^l$ and soundness $(3/4)^l = d^{\log(3/4)} \approx d^{-0.415}$. We instead use a PCP with properties similar to the query efficient PCP of Håstad and Khot [29]. Our PCP has perfect completeness, makes $2t + t^2$ queries from a domain of size $d \geq 3$ and has soundness $(d/2)^{-t^2}$.

7.1.1 Subsequent results

The $\Omega(k^{1-\epsilon})$ lower bound for Minimum Ek-Vertex Cover has subsequently been improved to $k - 3 - \epsilon$ by Dinur, Guruswami and Khot [8].

It has been communicated to me by Dinur, Guruswami, Khot and Regev that they have since improved this to $k - 1 - \epsilon$. This result has not yet been published, however.

7.2 A reduction from PCP to Minimum Ek-Vertex Cover

In this section we prove that if $\mathbf{NP} \subseteq \mathbf{naPCP}_{1,s(d)}[O(\log n), q, d]$, then, for $k = q(d-1)$, Minimum Ek-Vertex Cover is **NP**-hard to approximate within $s(d)^{-1/q}$. Our reduction is inspired by Goldreich [23], and its analysis uses techniques similar to those used by Trevisan [52].

Assume that $\mathbf{NP} \subseteq \mathbf{naPCP}_{1,s(d)}[O(\log n), q, d]$ and for some **NP**-complete language L let V be the verifier which thus exists. First we modify the verifier to make sure that each of the q queries are uniformly distributed in the proof. We do this by first making all positions equally likely by if necessary duplicating some positions in the proof and when querying a duplicated position choosing one of the duplicates randomly. If some positions are never queried then we simply omit them from the proof. We also randomly permute the order of the queries. This is possible since the verifier is non-adaptive. By doing this each query will be uniformly distributed (albeit of course not independent from each other).

We now describe the reduction to Minimum Ek-Vertex Cover. On input x we construct the hypergraph H. The vertices in H are pairs (p, a) for all positions p

in the proofs and all the possible answers a. Hence if P is the number of positions in the proof, the graph H consists of P *layers* S_p, each layer of size d.

For each layer p we add an edge S_p containing the vertices in layer p. We call these edges *layer-edges*. The purpose of the layer-edges is to force a vertex cover to contain at least one answer for each position.

Now we add edges for all the random strings σ. Suppose the verifier queries the positions $p_1, \ldots, p_q$ on random string σ. Consider the corresponding q layers $S_{p_1}, \ldots, S_{p_q}$. We want to ensure that for any vertex cover of H, there is at least one way of picking $(p_1, a_2), \ldots, (p_q, a_q)$ from the cover such that $a_1, \ldots a_q$ is accepted by the verifier. We can achieve this by for each set for which this is not the case, simply adding, as an edge in H, the complement in the subgraph induced by the layers $S_{p_1}, \ldots, S_{p_q}$

More formally, for all non-empty subsets $S_1 \subseteq S_{p_1}, \ldots, S_q \subseteq S_{p_q}$ such that there is no way of choosing $(p_1, a_1) \in S_1, \ldots, (p_q, a_q) \in S_q$, such that $a_1, \ldots, a_q$ is accepted by the verifier on random string σ, we add $(S_{p_1} \setminus S_1) \cup \ldots \cup (S_{p_q} \setminus S_q)$ as an edge in H. We call these edges *non layer-edges*.

Lemma 7.1. If $x \in L$ then there is a vertex cover of size P in H.

Proof. If $x \in L$ then there is a proof π which makes the verifier accept with probability 1. Construct S by for each layer p putting $(p, \pi(p))$ in S. The layer-edges S_p are clearly covered. Suppose we have an uncovered non layer-edge $e = (S_{p_1} \setminus S_1) \cup \ldots \cup (S_{p_q} \setminus S_q)$. Then we must have $(p_1, \pi(p_1)) \in S_1, \ldots, (p_q, \pi(p_q)) \in S_q$. But then $\pi(p_1), \ldots, \pi(p_q)$ must be rejected by the verifier on random string σ, a contradiction. Thus all edges must be covered. ∎

Lemma 7.2. If $x \notin L$, then no set of size less than $s(d)^{-1/q}P$ is a vertex cover of H.

Proof. Suppose there is a cover S of size tP. To begin with, all layer-edges S_p must be covered, so each layer contains at least one vertex from the cover.

For layer p let $L(p) = S_p \cap S$. Now define a random proof π by for each position p picking uniformly at random a such that $(p, a) \in L(p)$ and letting $\pi(p) = a$. Consider a random string σ, and note that since S is a cover, there must be at least one way of choosing $(p_1, a_1) \in L(p_1), \ldots, (p_q, a_q) \in L(p_q)$ such that the verifier accepts on random string σ when seeing the answers $a_1, \ldots, a_q$, since otherwise $(S_{p_1} \setminus L(p_1)) \cup \ldots \cup (S_{p_q} \setminus L(p_q))$ would be an edge in H which is not covered by S. The probability that we picked these answers when constructing π is $\prod_{i=1}^{q} |L(p_i)|^{-1}$. Also, $\mathrm{E}[|L(p_i)|] = \mathrm{E}[|L(p_1)|] = t$, since all the queries are uniformly distributed in the proof. Hence the expected number of σ for which the verifier accepts on π is at least

$$\mathop{\mathrm{E}}_{\sigma}\left[\prod_{i=1}^{q} |L(p_i)|^{-1}\right] \geq e^{-\mathrm{E}_\sigma[\sum_{i=1}^{q} \ln |L(p_i)|]} = e^{-q\,\mathrm{E}_\sigma[\ln |L(p_1)|]} \geq$$
$$e^{-q \ln \mathrm{E}[|L(p_1)|]} = e^{-q \ln t} = t^{-q}$$

where we used that $\mathrm{E}[e^X] \geq e^{\mathrm{E}[X]}$ and $\mathrm{E}[-\ln X] \geq -\ln \mathrm{E}[X]$, by Jensen's inequality.

If follows that there must be some π which makes the verifier accept with at least probability t^{-q}

However, we know that no proof makes the verifier accept with probability greater than $s(d)$, so we must have $t \geq s(d)^{-1/q}$. ∎

We combine the two previous lemmas into the following theorem:

Theorem 7.3. Suppose that $\mathbf{NP} \subseteq \mathbf{naPCP}_{1,s(d)}[O(\log n), q, d]$. Then Minimum E(q(d-1))-Vertex Cover is **NP**-hard to approximate within $s(d)^{-1/q}$.

Proof. On input x, create the hypergraph H as described above. The edges in this hypergraph has size at most $q(d-1)$, and we can make sure that all edges has exactly this size by padding them with unique elements. By Lemma 7.1 and Lemma 7.2 if we could distinguish between the case that there is a cover of size P and the case that no cover of size less than $s(d)^{-1/q}P$ is a cover, then we could decide whether $x \in L$. Hence it is **NP**-hard to approximate Minimum Vertex Cover on $q(d-1)$-uniform hypergraphs within $s(d)^{-1/q}$. ∎

7.3 A query efficient PCP for NP

In this section we give an simplified construction as compared to that of Håstad and Khot [29] of a query efficient non-adaptive PCP with domain d. We then use it to prove the $\Omega(k^{1-\epsilon})$ lower bound for Minimum Ek-Vertex Cover. The test is based on simple 3-query PCP with perfect completeness and domain d. Similarly to Samorodnitsky and Trevisan [45] this test is then iterated in a query-efficient way. The resulting test is similar to the test of Samorodnitsky and Trevisan, and in particular to the generalization of this test to larger domains by Engebretsen [11]. The test is analyzed using techniques from Håstad and Wigderson's [30] simplified analysis of the PCP of Samorodnitsky and Trevisan.

7.3.1 The basic 3-query protocol

The input to the verifier is a Standard Written d-Proof. Instead of accessing the tables A_U and A_W directly, the verifier will access folded tables, $A_{U,\mathbf{Z}_d}$ and $A_{W,\mathbf{Z}_d}$. The verifier is given in Figure 7.1. The test is very similar to the basic PCP by Håstad described in Chapter 4. The difference is that the "noise"-function e has been replaced by h which uniformly takes values in $\{1, \omega\}$, and instead of only accepting when the product is 1, we make use of the larger domain and accept both when the product is 1 and when the product is ω. We note at this point that this test is similar to the test used by Engebretsen and Guruswami [12] to

BasicTest.
Input: An E3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Standard Written d-Proof with parameter u, with all the tables folded over $\boldsymbol{Z}_d$.

1. Select uniformly at random a sequence $W = (C_1, \ldots, C_u)$ of u clauses and a sequence U of u variables by for each clause C_k choosing uniformly at random a variable occurring in C_k.
2. Select uniformly at random $f \in Z_d^{\{0,1\}^U}$
3. Select uniformly at random $g \in Z_d^{\mathrm{SAT}^W}$
4. Select $h \in Z_d^{\mathrm{SAT}^W}$ by choosing $h(y) \in \{1, \omega\}$ uniformly and independently.
5. Accept if $A_{U,\boldsymbol{Z}_d}(f) A_{W,\boldsymbol{Z}_d}(g) A_{W,\boldsymbol{Z}_d}(((f \circ \pi^{U,W})g)^{-1}h) \in \{1, \omega\}$.

Figure 7.1. The above PCP is parameterized by the positive integers u and d and tests if a E3-CNF(5) formula ϕ is satisfiable by querying three positions in a Standard Written d-Proof with parameter u. With suitable choice of the parameter u, the test has perfect completeness and soundness $\frac{2}{d} + \epsilon$ for any $\epsilon > 0$.

prove inapproximability of linear in-equations over $\boldsymbol{Z}_d$. In their test $h(y)$ is chosen uniformly in $\boldsymbol{Z}_d \setminus \{1\}$ and the test accepts if

$$A_{U,\boldsymbol{Z}_d}(f) A_{W,\boldsymbol{Z}_d}(g) A_{W,\boldsymbol{Z}_d}(((f \circ \pi^{U,W})g)^{-1}h) \neq 1.$$

The completeness of the test is straightforward:

Lemma 7.4. If ϕ is satisfiable, a correct proof is accepted by BasicTest with probability 1.

We proceed to analyze the soundness.

Lemma 7.5. If BasicTest accepts with probability $\frac{2}{d} + \delta$, then there is a strategy for the 2P1R game with success probability at least $(2d)^{-2}\delta^2$.

Proof. Fix U and W and let $A = A_{U,\boldsymbol{Z}_d}$ and $B = A_{W,\boldsymbol{Z}_d}$. As a shorthand, let

$$Acc(\gamma) = \gamma^{-1} A(f) B(g) B((f \circ \pi^{U,W})g)^{-1}h),$$

for $\gamma \in \{1, \omega\}$. We have that the acceptance probability can be written

$$\begin{aligned}
&\mathop{\mathrm{E}}_{U,W,f,g,h}\left[\frac{1}{d}\sum_{k=0}^{d-1} Acc(1)^k + Acc(\omega)^k\right] = \\
&\frac{1}{d}\left(\sum_{k=0}^{d-1}\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(1)^k] + \sum_{k=0}^{d-1}\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(w)^k]\right) \leq \\
&\frac{1}{d}\left(\sum_{k=0}^{d-1}\left|\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(1)^k]\right| + \sum_{k=0}^{d-1}\left|\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(w)^k]\right|\right) = \qquad (7.1)\\
&\frac{2}{d}\sum_{k=0}^{d-1}\left|\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(1)^k]\right|. \qquad (7.2)
\end{aligned}$$

Where the last equality holds since $\mathrm{E}_{U,W,f,g,h}[Acc(w)^k] = w^k\,\mathrm{E}_{U,W,f,g,h}[Acc(w)^k]$ and $|w^k| = 1$. Now if the acceptance probability, and hence (7.2) is greater that $2/d + \delta$ there must be some term

$$\left|\mathop{\mathrm{E}}_{U,W,f,g,h}[Acc(1)^k]\right| = \left|\mathop{\mathrm{E}}_{U,W,f,g,h}[A(f)^k B(g)^k B(((f\circ\pi^{U,W})g)^{-1}h)^k]\right| \qquad (7.3)$$

which is at least $\delta/2$. For notational simplicity we may assume that $k = 1$, since if this is not the case we can just replace each table with its k:th power. Thus consider

$$\mathop{\mathrm{E}}_{U,W,f,g,h}[A(f)B(g)B(((f\circ\pi^{U,W})g)^{-1}h)] \qquad (7.4)$$

using the Fourier inversion formula (Theorem 5.8) we get that this can be written as

$$\mathop{\mathrm{E}}_{U,W}\left[\sum_{\substack{\alpha\in\hat{\boldsymbol{Z}}_d^{\{0,1\}^U}\\ \beta_1,\beta_2\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}}} \hat{A}_\alpha\hat{B}_{\beta_1}\hat{B}_{\beta_2}\mathop{\mathrm{E}}_{f,g,h}[\chi_\alpha(f)\chi_{\beta_1}(g)\chi_{\beta_2}(((f\circ\pi^{U,W})g)^{-1}h)]\right] \qquad (7.5)$$

and we have that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,g,h}[\chi_\alpha(f)\chi_{\beta_1}(g)\chi_{\beta_2}(((f\circ\pi^{U,W})g)^{-1}h)] = \\
&\quad \mathop{\mathrm{E}}_{f}[\chi_\alpha(f)\chi_{\beta_2}((f\circ\pi^{U,W})^{-1})]\mathop{\mathrm{E}}_{g}[\chi_{\beta_1}(g)\chi_{\beta_2}(g^{-1})]\mathop{\mathrm{E}}_{h}[\chi_{\beta_2}(h)]
\end{aligned}$$

where we used the multiplicative property (5.1). We have that, by (5.2), (5.3) and (5.4), $\mathrm{E}_g[\chi_{\beta_1}(g)\chi_{\beta_2}(g^{-1})] = \mathrm{E}_g[\chi_{\beta_1-\beta_2}(g)] = 0$ unless $\beta_1 = \beta_2 = \beta$. Similarly by in

addition using Lemma 5.21, $\mathrm{E}_f[\chi_\alpha(f)\chi_\beta((f\circ\pi^{U,W})^{-1})] = 0$ unless $\pi_d^{U,W}(\beta) = \alpha$. Also note that

$$\mathop{\mathrm{E}}_h[\chi_\beta(h)] = \mathop{\mathrm{E}}_h\left[\prod_{y\in\beta}(h(y)^{\beta(y)}\right] = \prod_y \frac{1}{2}(1+\omega^{\beta(y)})$$

where a factor is 1 if the corresponding $\beta(y)$ is 0 and otherwise has absolute value at most $|(1+\omega)|/2$. We have

$$\begin{aligned}|1+\omega|^2 &= (1+\cos(2\pi/d))^2 + \sin^2(2\pi/d)\\ &= 2+2\cos(2\pi/d).\end{aligned}$$

For $d\in\{3,4\}$, we have $2+2\cos(2\pi/d) < 2$, and for $d\geq 5$, we use the MacLaurin expansion of $\cos(x)$ to get that

$$\cos(2\pi/d) \leq 1-(2\pi/d)^2/2 + (2\pi/d)^4/6 \leq 1-(\pi/d)^2.$$

Thus $2+2\cos(2\pi/d) \leq 4-(\pi/d)^2$ i.e., $|(1+\omega)|/2 \leq (1-1/d^2)^{1/2}$. It follows that (7.5) has absolute value at most

$$\mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|\hat{B}_\beta^2(1-1/d^2)^{|\beta|/2}\right]. \tag{7.6}$$

By the Cauchy–Schwartz inequality, this is as most

$$\mathop{\mathrm{E}}_{U,W}\left[\sqrt{\left(\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2(1-1/d^2)^{|\beta|}\right)\left(\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{B}_\beta|^2\right)}\right] =$$

$$\mathop{\mathrm{E}}_{U,W}\left[\sqrt{\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2(1-1/d^2)^{|\beta|}}\right]. \tag{7.7}$$

We also have that $(1-1/d^2)^{|\beta|} \leq e^{-d^{-2}|\beta|} \leq (d^{-2}|\beta|)^{-1}$, so if we insert this in (7.7) we have proven that

$$\frac{\delta}{2} \leq d \mathop{\mathrm{E}}_{U,W}\left[\sqrt{\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}}\right]. \tag{7.8}$$

Now we describe the strategies for the provers in the 2P1R game. P_2 upon receiving U picks an α with probability $|\hat{A}_\alpha|^2$ and returns a random x such that

$\alpha(x) \neq 0$. P_1 upon receiving W picks a β with probability $|\hat{B}_\beta|^2$ and returns a random y such that $\beta(y) \neq 0$. This is always possible since by Lemma 5.11 $\hat{A}_{\mathbf{0}} = \hat{B}_{\mathbf{0}} = 0$. The probability of success is at least

$$\begin{aligned}
&\mathop{\mathrm{E}}_{U,W}\left[\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}\right] \\
&\quad= \mathop{\mathrm{E}}_{U,W}\left[\left(\sqrt{\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}}\right)^2\right] \\
&\quad\geq \left(\mathop{\mathrm{E}}_{U,W}\left[\sqrt{\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}} |\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}}\right]\right)^2 . \qquad (7.9)
\end{aligned}$$

Where we used $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$. Now by (7.8), we have that (7.9) is a least $(2d)^{-2}\delta^2$, and we are done. ∎

Theorem 7.6. For any $d \geq 3$, for any $\epsilon > 0$,

$$\mathbf{NP} = \mathbf{naPCP}_{1,2/d+\epsilon}[O(\log n), 3, d]$$

Proof. By Theorem 4.1, deciding any language in **NP** can be reduced to distinguishing between the cases that an E3-CNF(5) formula ϕ is satisfiable of at most μ satisfiable.

Choose the number of repetitions in the 2P1R game such that the soundness $c_\mu^u < (2d)^{-2}\epsilon^2$. Clearly BasicTest makes 3 queries, and uses $O(\log n)$ random bits. By Lemma 7.4, if ϕ is satisfiable, there is a proof which is accepted with probability 1. Thus the verifier has completeness 1. By Lemma 7.5, if ϕ is at most μ satisfiable and there is a proof which is accepted with probability at least $2/d + \epsilon$, there would be a strategy for the 2P1R game with success probability greater than c_μ^u, a contradiction. Thus the soundness is at most $2/d + \epsilon$ ∎

7.3.2 The iterated test

In this section we iterate the basic test from the previous section. The iterated test is given in Figure 7.2. Again the completeness is straightforward, and we omit the proof of the following lemma:

Lemma 7.7. If ϕ is satisfiable, a correct proof is accepted by IterTest with probability 1.

We proceed to analyze the soundness.

IterTest.
Input: An E3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Standard Written d-Proof with parameter u, and all the tables folded over $\boldsymbol{Z}_d$.

1. Select uniformly at random a sequence $U = (x_1, \ldots, x_u)$ of u variables.
2. Select t sequences $W_1, \ldots W_t$ by independently for each j selecting W_j by for each $x_l \in U$ picking uniformly at random a clause C_l in which x_l occurs, and letting $W_j = (C_1, \ldots, C_u)$.
3. For $1 \leq i \leq t$ select uniformly at random $f_i \in Z_d^{\{0,1\}^U}$.
4. For $1 \leq j \leq t$ select uniformly at random $g_j \in Z_d^{\mathrm{SAT}^{W_j}}$.
5. For $1 \leq i, j \leq t$ select $h_{ij} \in Z_d^{\mathrm{SAT}^{W_j}}$ by choosing $h_{ij}(y) \in \{1, \omega\}$ uniformly and independently.
6. Accept if, for all $1 \leq i, j \leq t$,

$$A_{U,\boldsymbol{Z}_d}(f_i) A_{W_j,\boldsymbol{Z}_d}(g_j) A_{W_j,\boldsymbol{Z}_d}(((f_i \circ \pi^{U,W_j}) g_j)^{-1} h_{ij}) \in \{1, \omega\}.$$

Figure 7.2. The above PCP is parameterized by the positive integers u, t and d and tests if a E3-CNF(5) formula ϕ is satisfiable by querying $2t+t^2$. With suitable choice of the parameter u, the test has perfect completeness and soundness $2^{t^2} d^{-t^2} + \epsilon$, for any $\epsilon > 0$.

Lemma 7.8. If IterTest accepts with probability $2^{t^2} d^{-t^2} + \delta$, then there is a strategy for the 2P1R game with success probability at least $2^{-2t^2} d^{-2} \delta^2$.

Proof. We define the shorthand

$$Acc_{ij}(\gamma) = \gamma^{-1} A_{U,\boldsymbol{Z}_d}(f_i) A_{W_j,\boldsymbol{Z}_d}(g_j) A_{W_j,\boldsymbol{Z}_d}(((f_i \circ \pi^{U,W_j}) g_j)^{-1} h_{ij}). \tag{7.10}$$

Then the acceptance probability can be written as

$$\mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_{ij}} \left[\prod_{(i,j) \in [t] \times [t]} \frac{\sum_{k=0}^{d-1} Acc_{ij}(1)^k + \sum_{k=0}^{d-1} Acc_{ij}(\omega)^k}{d} \right]. \tag{7.11}$$

Now let $S \in \{0, \ldots, d-1\}^{t^2}$ and $T \in \{0,1\}^{t^2}$ be vectors indexed by (i,j). Then (7.11) can be written as

$$\mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_j} \left[d^{-t^2} \sum_{S \in \{0..d-1\}^{t^2}} \sum_{T \in \{0,1\}^{t^2}} \prod_{(i,j) \in [t] \times [t]} Acc_{ij}(w^{T(i,j)})^{S(i,j)} \right]. \tag{7.12}$$

Now,

$$\mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_j}\left[\prod_{(i,j)\in[t]\times[t]} Acc_{ij}(w^{T(i,j)})^{S(i,j)}\right] =$$

$$\omega^c \mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_j}\left[\prod_{(i,j)\in[t]\times[t]} Acc_{ij}(1)^{S(i,j)}\right]$$

where c is some integer constant, so the absolute value of (7.12) is at most

$$d^{-t^2}2^{t^2}\sum_{S\in\{0..d-1\}^{t^2}}\left|\mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_j}\left[\prod_{(i,j)\in[t]\times[t]} Acc_{ij}(1))^{S(i,j)}\right]\right|$$

Hence if the acceptance probability is at least $d^{-t^2}2^{t^2}+\delta$, there must be some $S\neq\mathbf{0}$ so that

$$\left|\mathop{\mathrm{E}}_{U,W_i,f_i,g_j,h_j}\left[\prod_{(i,j)\in[t]\times[t]} Acc_{ij}(1))^{S(i,j)}\right]\right| \geq \frac{\delta}{2^{t^2}}.$$

Fix such an S. Without loss of generality we may assume that $S(1,1)\neq 0$. Fix U and consider

$$\prod_{(i,j)\in[t]\times[t]} (A_{U,\boldsymbol{Z}_d}(f_i)A_{W_j,\boldsymbol{Z}_d}(g_j)A_{W_j,\boldsymbol{Z}_d}(((f_i\circ\pi^{U,W_j})g_j)^{-1}h_j))^{S(i,j)} \qquad (7.13)$$

fix the values of $f_2,\ldots,f_t$, $W_2,\ldots,W_2$, $g_2,\ldots,g_t$ and h_{ij} except for h_{11} in such a way as to not decrease the expectation (over W_1, f_1, g_1 and h_{11}). These values depend only on U. Then (7.13) can be written as

$$A(f_1)C(g_1)B((f_1g_1)^{-1}h_{11})$$

where

$$A(f_1) = \prod_{(i,j)\in[t]\times[t]} A_{U,\boldsymbol{Z}_d}(f_i)\prod_{\substack{(i,j)\in[t]\times[t]\\ j\neq 1}} A_{W_j,\boldsymbol{Z}_d}(((f_i\circ\pi^{U,W_j})g_j)^{-1}h_{ij}))^{S(i,j)}$$

is a function which depends on the values chosen above, $B(g)=A_{W_1}(g)^{S(1,1)}$ and $C(g)$ contains the rest of the factors. Applying the Fourier transform, we get that

$$\left|E_{U,W_1,f_1,g_1,h_{11}}A(f_1)C(g_1)B(((f_1\circ\pi^{U,W_1})g_1)^{-1}h_{11})]\right| \leq$$

$$\mathop{\mathrm{E}}_{U,W_1}\left[\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^{W_1}}} |\hat{A}_{\pi_d^{U,W_1}(\beta)}\hat{B}_\beta\hat{C}_\beta|(d^{-2}|\beta|)^{-1/2}\right]$$

Using the Cauchy–Schwartz inequality, this is as most

$$\mathop{\mathrm{E}}_{U,W_1}\left[\sqrt{\left(\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}W_1}}|\hat{A}_{\pi_d^{U,W}(\beta)}|^2|\hat{B}_\beta|^2(d^{-2}|\beta|)^{-1}\right)\left(\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}W_1}}|\hat{C}_\beta|^2\right)}\right] =$$

$$\mathop{\mathrm{E}}_{U,W_1}\left[\sqrt{\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}W_1}}|\hat{A}_{\pi_d^{U,W_1}(\beta)}|^2|\hat{B}_\beta|^2(d^{-2}|\beta|)^{-1}}\right]$$

Now we define the strategies for the P_1 upon receiving W chooses a β with probability $|\hat{B}_\beta|^2$ and returns a random y such that $\beta(y)\neq 0$. The probability of success is at least P_2 upon receiving U computes the function A whose value depends only on U, and picks and α with probability $|\hat{A}_\alpha|^2$ and returns a random x such that $\alpha(x)\neq 0$. If no such x exist, P_2 gives up. The probability of success is at least

$$\sum_{\substack{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}W_1}\\ \pi_d^{U,W_1}(\beta)\neq\mathbf{0}}}|\hat{A}_{\pi_d^{U,W_1}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}=\sum_{\beta\in\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}W_1}}|\hat{A}_{\pi_d^{U,W_1}(\beta)}|^2|\hat{B}_\beta|^2|\beta|^{-1}. \quad (7.14)$$

Where the equality follows since if $\pi_d^{U,W}(\beta)=\mathbf{0}$, then $\sum_y\beta(y)=0 \pmod d$, and if this is the case we have $\hat{B}_\beta=0$, since $B(g)=A_{W_1,\boldsymbol{Z}_d}(g)^{S(1,1)}$, $S(1,1)\neq 0$, and A_{W_1} is folded over $\boldsymbol{Z}_d$, By the above we have that the right hand side of (7.14) is at least $2^{-2t^2}d^{-2}\delta^2$. ∎

Combining the two previous lemmas, we get:

Theorem 7.9. For any $t\geq 1$, $d\geq 2$, for any $\delta>0$,

$$\mathbf{NP}=\mathbf{naPCP}_{1,2^{t^2}d^{-t^2}+\delta}[O(\log n),2t+t^2,d]$$

Proof. By Theorem 4.1, deciding any language in **NP** can be reduced to distinguishing between the cases that an E3-CNF(5) formula ϕ is satisfiable of at most μ satisfiable. Choose the parameter u such that the soundness of the 2P1R game $c_\mu < 2^{-2t^2}d^{-2}\delta^2$. The verifier IterTest makes $2t+t^2$ queries and uses $O(\log n)$ random bits. By Lemma 7.7, the verifier has completeness 1, and by Lemma 7.8, if ϕ is at most μ satisfiable and there is a proof which is accepted with probability at least $2^{t^2}d^{-t^2}+\delta$, there is a strategy for the 2P1R game with success probability at least $2^{-2t^2}d^{-2}\delta^2$, a contradiction. ∎

7.4 Hardness of approximating Minimum Ek-Vertex Cover

Now we can put the PCP characterization of **NP** of the previous section into the reduction of Section 7.2 and get the following inapproximability result:

Theorem 7.10. For any $\epsilon > 0$, Minimum Ek-Vertex Cover is **NP**-hard to approximate within $\Omega(k^{1-\epsilon})$

Proof. Choose $t = 2\epsilon^{-1}$. Combining Theorem 7.9 and Theorem 7.3, we have that for any $d \geq 2$, Minimum Vertex Cover on $k = (2t + t^2)(d-1)$-uniform hypergraphs is **NP**-hard to approximate to within

$$\begin{aligned} ((d/2)^{-t^2})^{-1/(t^2+2t)} &= (d/2)^{1-2/t} \\ &= (d/2)^{1-\epsilon} \end{aligned}$$

Since t depends only on ϵ which is a fixed constant, $d = \Omega(k)$ and $(d/2)^{1-\epsilon} = \Omega(k^{1-\epsilon})$. ∎

Chapter 8

Vertex cover on 3-uniform hypergraphs

8.1 Introduction

In the previous chapter we studied what happened with the approximability of Minimum Ek-Vertex Cover when the parameter k goes to infinity. In this chapter and the next, we instead consider the case when k is small constant greater than 2. In particular, in this chapter we prove that Minimum E3-Vertex Cover is **NP**-hard to approximate within $3/2 - \epsilon$. We use the same reduction as in Chapter 7, but instead of using the PCP from Chapter 7 we use a non-adaptive 3-query boolean PCP with perfect completeness and soundness $3/4 + \epsilon$, which is essentially the same PCP which Håstad [28] uses to prove inapproximability of E3-Sat on satisfiable instances.

When analyzing the reduction we do not use the PCP as a black box, instead we do a careful analysis of the number of covered edges.

Compared to the original proof in [31], the analysis is simplified by the use of the generalized two prover game which we reviewed in Section 5.2, and is in part very similar to Khot's simplified analysis of Håstad's PCP [39].

8.1.1 Subsequent results

As mentioned in Chapter 7, Dinur, Guruswami, Khot and Regev have proven a $k - 1 - \epsilon$ lower bound for Minimum Ek-Vertex Cover, thus giving a lower-bound of $2 - \epsilon$ for the case of 3-uniform hypergraphs.

8.2 Hardness of approximating Minimum E3-Vertex Cover

We need to modify the basic setup as described in Section 5.2 somewhat, since we have a bipartite situation where the verifier queries tables of two classes (corresponding to sequences U and sequences W), it is important that the total number of positions in the two classes are equal. To this end, let $D(u)$ be the number of copies we need of each position corresponding to an U to make the classes the same size. We make the classes the same size by for $1 \leq \kappa \leq D(u)$ for each U having alternate tables $A_{U,\kappa}$ and picking one at random when making a query. Again, positions which are never queried are omitted from the proof.

The verifier is essentially Khot's version of Håstad's PCP for E3-Sat [28, 39]. Since we are interested in the simplest possible analysis rather than to get a result for E3-Sat we use a different acceptance predicate. We give the verifier in Figure 8.1.

Recall that the verifier in LinTest could be viewed as reading the error corrected value of $(f \circ \pi^{U,W})$ from $A_{W,\mathbf{true}}$ and comparing with the value of f read from $A_{U,\mathbf{true}}$.

While in LinTest the function $(f \circ \pi^{U,W})e$ is essentially $(f \circ \pi^{U,W})$ with uniformly added "noise", in Vc3Test the function h is essentially $-(f \circ \pi^{U,W})$ with some noise added. However, the noise is not uniformly added, but is only added at points y where $(f \circ \pi^{U,W})(y) = -1$. This makes is possible to get a test with perfect completeness, since in the case when $A_{U,\mathbf{true}}(f) = 1$, we know that in a correct proof $A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}(gh)$ must have the value -1. In the case when $A_{U,\mathbf{true}}(f) = -1$, the verifier accepts immediately.

We proceed by applying the reduction described in Section 7.2 to this PCP. The vertices of the hypergraph consists of pairs (p, a) where p is a position in the proof and $a \in \{-1, 1\}$. Remember that because of folding not all f and g are queried by the verifier. Denote by $\tilde{f}$ and $\tilde{g}$ the functions actually queried when the verifier chooses f and g respectively. Note that, since we omit positions in the proofs which are never queried, a position in the proof is given either by the triple $(U, \kappa, \tilde{f})$ or by the pair $(W, \tilde{g})$.

This means that a layer looks either as $S_{(U,\kappa,\tilde{f})} = \{((U, \kappa, \tilde{f}), -1), ((U, \kappa, \tilde{f}), 1)\}$, or as $S_{(W,\tilde{g})} = \{((W, \tilde{g}), -1), ((W, \tilde{g}), 1)\}$.

For a random string σ for which the verifier makes queries $(U, \kappa, \tilde{f})$ $(W, \tilde{g})$ and $(W, \widetilde{gh})$ the non-layer edges are the complements of the rejecting views. For example, in the case when $\tilde{f} = f$, $\tilde{g} = g$ and $\widetilde{gh} = gh$, the rejecting views are $(1, 1, 1)$ and $(1, -1, -1)$, so we add the edges corresponding to the answers $(-1, -1, -1)$ and $(-1, 1, 1)$. Hence the size of the edges will be at most 3.

We note as this point that the same edge may be added several times by this procedure. In the lemmas below which mentions the fraction of edges which is uncovered, edges are counted with this multiplicity. However, since we are interested in whether all edges are covered, the multiplicity doesn't matter. Thus we may consider the edges to be a set and not a multiset, and the graph which we construct

Vc3Test.
Input: AnE3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Generalized Standard Written 2-Proof with parameters u and T and all the tables folded over true.

1. Select a (T,u)-block W uniformly at random, and choose a random u projection U of W.
2. Select uniformly at random $f \in 2^{\mathrm{SAT}^U}$ and $g \in 2^{\mathrm{SAT}^W}$.
3. Select $h \in 2^{\mathrm{SAT}^W}$, by for each y, if $f(\pi^{U,W}(y)) = 1$ set $h(y) = -1$. If $f(\pi^{U,W}(y)) = -1$, set $h(y) = 1$ with probability $1-\delta$ and $h(y) = -1$ with probability δ.
4. Accept if $(1 + A_{U,\kappa,\mathbf{true}}(f))(1 + A_{W,\mathbf{true}}(g)A_{W,\mathbf{true}}(gh)) = 0$, else reject.

Figure 8.1. The above PCP is parameterized by the positive integers u and T and the positive real $\delta \leq 1/2$ and tests if a E3-CNF(5) formula ϕ is satisfiable by querying three positions in a Generalized Standard Written 2-Proof with duplicate U-tables. With suitable choices of the parameters u, T and δ as functions of ϵ, the above PCP has perfect completeness and soundness $3/4 + \epsilon$ for any constant $\epsilon > 0$.

is a hypergraph as per definition 2.20.

Let P be the total number of positions in the proof. By Lemma 7.1 it follows that if ϕ is satisfiable, then there is a cover of size P.

Let Q_1 be the set of all triples $(U,\kappa,\tilde{f})$ and Q_2 be the set of all pairs (W,g). By construction, $P/2 = |Q_1| = |Q_2|$.

Suppose there is a cover S of size tP in H. Now since S is a cover, all layer-edges must be covered. We may thus view S as a function $S : Q_1 \cup Q_2 \to \{\{-1\},\{1\},\{1,-1\}\}$ defined by the intersection of S and S_p for a position p. Define tables $B_{U,\kappa}$ and B_W by (for $f = \tilde{f}$ i.e., f which are actually in the proof)

$$B_{U,\kappa}(\tilde{f}) = \begin{cases} -1 & \text{if } S(U,\kappa,\tilde{f}) = \{-1\} \\ 1 & \text{if } S(U,\kappa,\tilde{f}) = \{1\} \\ 0 & \text{if } S(U,\kappa,\tilde{f}) = \{1,-1\}. \end{cases}$$

For the f which are not in the proof $-f$ must be in the proof so we define $B_{U,\kappa}(f) = -B_{U,\kappa}(-f)$ (i.e., folding over **true**). Define $B_W(\tilde{g})$ in the same way and extend the table to all g by folding over **true**.

Lemma 8.1. The fraction of non-layer edges which is covered is

$$1 - \frac{\mathrm{E}\left[B_{U,\kappa}(f)^2 B_W(g)^2 B_W(gh)^2 (1 + B_{U,\kappa}(f))(1 + B_W(g)B_W(gh))\right]}{8} \qquad (8.1)$$

where the expectation is over the choice of U,κ,W,f,g and h in the verifier.

Proof. Since for each random string σ there are two non-accepting views there is exactly two non layer-edges e_1 and e_2 for each random string. Suppose that the verifier picks functions f, g and h on random string σ. Note that if at least one of $B_{U,\kappa}(f), B_W(g)$ or $B_W(hg)$ is 0, this means at least one of the corresponding layers contains two elements from the cover, which means that both e_1 and e_2 are covered. On the other hand, if none of $B_{U,\kappa}(f), B_W(g)$ or $B_W(gh)$ is 0, there is one uncovered edge if $B_{U,\kappa}(f), B_W(g)$ and $B_W(gh)$ would make the verifier reject, and otherwise both edges are covered. In other words, the number of covered edges is

$$2 - B_{U,\kappa}(f)^2 B_W(g)^2 B_W(gh)^2 \frac{(1 + B_{U,\kappa}(f))(1 + B_W(g)B_W(hg))}{4}$$

Taking the expectation, we get that the fraction of covered non layer-edges is

$$1 - \frac{\mathrm{E}\left[B_{U,\kappa}(f)^2 B_W(g)^2 B_W(hg)^2 (1 + B_{U,\kappa}(f))(1 + B_W(g)B_W(hg))\right]}{8}$$

∎

Lemma 8.2. Let $C_{U,\kappa} : 2^{\mathrm{SAT}^U} \to [-1,1]$, and let $C_W : 2^{\mathrm{SAT}^W} \to [-1,1]$, Let U, κ, W, f, g, h be chosen as in Vc3Test with $0 < \delta \leq 1/2$ and $T = \frac{\log(1/\delta^2)}{\delta^3}$. Suppose that

$$\left| \mathop{\mathrm{E}}_{U,\kappa,W,f,g,h} [C_{U,\kappa}(f) C_W(g) C_W(hg)] - \mathop{\mathrm{E}}_{U,\kappa,W} \left[\hat{C}_{U,\kappa,\emptyset} \hat{C}^2_{W,\emptyset}\right] \right| \geq 9\delta. \tag{8.2}$$

Then there is a strategy for the Generalized 2P1R game with success probability at least

$$\frac{\delta^4}{\log(1/\delta)}.$$

Proof. Using Fourier-expansion,

$$\mathop{\mathrm{E}}_{U,\kappa,W,f,g,h} [C_{U,\kappa}(f) C_W(g) C_W(hg)] =$$

$$\mathop{\mathrm{E}}_{U,\kappa,W} \left[\sum_{\substack{\alpha \subseteq \mathrm{SAT}^U \\ \beta_1, \beta_2 \subseteq \mathrm{SAT}^W}} \hat{C}_{U,\kappa,\alpha} \hat{C}_{W,\beta_1} \hat{C}_{W,\beta_2} \mathop{\mathrm{E}}_{f,g,h} [\chi_\alpha(f) \chi_{\beta_1}(g) \chi_{\beta_2}(hg)] \right]. \tag{8.3}$$

We have that

$$\mathop{\mathrm{E}}_{f,g,h} [\chi_\alpha(f) \chi_{\beta_1}(g) \chi_{\beta_2}(hg)] = \mathop{\mathrm{E}}_{g} [\chi_{\beta_1}(g) \chi_{\beta_2}(g)] \mathop{\mathrm{E}}_{f,h} [\chi_\alpha(f) \chi_{\beta_2}(h)].$$

The expectation $\mathrm{E}_g[\chi_{\beta_1}(g)\chi_{\beta_2}(g)] = \mathrm{E}_g[\chi_{\beta_1 \Delta \beta_2}(g)]$ is 0 unless $\beta_1 = \beta_2 = \beta$. Let us next compute

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,h}[\chi_\alpha(f)\chi_\beta(h)] \\
&= \mathop{\mathrm{E}}_{f,h}\left[\prod_{x\in\alpha} f(x) \prod_{y\in\beta} h(y)\right] \\
&= \mathop{\mathrm{E}}_{f,h}\left[\prod_{x\in\alpha}\left(f(x) \prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}} h(y)\right) \prod_{x\in \mathrm{SAT}^U\setminus\alpha}\left(\prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}} h(y)\right)\right] \\
&= \prod_{x\in\alpha}\left(\frac{1}{2}\prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}}(-1) - \frac{1}{2}\prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}}(1-2\delta)\right)\times \\
&\qquad \prod_{x\in\mathrm{SAT}^U\setminus\alpha}\left(\frac{1}{2}\prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}}(-1) + \frac{1}{2}\prod_{\substack{y\in\beta \\ y:\pi^{U,W}(y)=x}}(1-2\delta)\right).
\end{aligned}$$

For $x \in \mathrm{SAT}^U$, let s_x be the number of $y \in \beta$ such that $\pi^{U,W}(y) = x$. Then

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,h}[\chi_\alpha(f)\chi_\beta(h)] = \\
&\qquad \prod_{x\in\alpha}\frac{1}{2}((-1)^{s_x} - (1-2\delta)^{s_x}) \prod_{x\in\mathrm{SAT}^U\setminus\alpha}\frac{1}{2}((-1)^{s_x} + (1-2\delta)^{s_x}). \qquad (8.4)
\end{aligned}$$

This expression is 0 if, for any $x \in \alpha$, $s_x = 0$, that is, it is 0 unless $\alpha \subseteq \pi^{U,W}(\beta)$. When $\alpha \subseteq \pi^{U,W}(\beta)$, we have that (8.4) may be written

$$\prod_{x\in\pi^{U,W}(\beta)\cap\alpha}\frac{1}{2}((-1)^{s_x} - (1-2\delta)^{s_x}) \prod_{x\in\pi^{U,W}(\beta)\setminus\alpha}\frac{1}{2}((-1)^{s_x} + (1-2\delta)^{s_x}).$$

Let us write $\nu(\alpha,\beta)$ for this quantity. Then (8.3) can be written

$$\mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{\beta\subseteq\mathrm{SAT}^W}\sum_{\alpha\subseteq\pi^{U,W}(\beta)}\hat{C}_{U,\kappa,\alpha}\hat{C}^2_{W,\beta}\nu(\alpha,\beta)\right].$$

We split this sum into two parts depending on whether α is empty or not.

$$\mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{\beta\subseteq \mathrm{SAT}^W}\sum_{\alpha\subseteq\pi^{U,W}(\beta)}\hat{C}_{U,\kappa,\alpha}\hat{C}^2_{W,\beta_1}\nu(\alpha,\beta)\right] =$$

$$\mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{\beta\subseteq \mathrm{SAT}^W}\hat{C}_{U,\kappa,\emptyset}\hat{C}^2_{W,\beta}\nu(\emptyset,\beta)\right] + \tag{8.5}$$

$$\mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{\beta\subseteq \mathrm{SAT}^W}\sum_{\substack{\alpha\neq\emptyset\\ \alpha\subseteq\pi^{U,W}(\beta)}}\hat{C}_{U,\kappa,\alpha}\hat{C}^2_{W,\beta}\nu(\alpha,\beta)\right]. \tag{8.6}$$

We first consider (8.5). We claim that

$$\left|\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\beta\neq\emptyset}\hat{C}^2_{W,\beta}\nu(\emptyset,\beta)\right]\right| \leq 5\delta.$$

First consider the terms where $|\beta| \leq \delta T$. Then by Corollary 5.18 with probability at least $1 - |\beta|/T \geq 1-\delta$ over the choice of U, there is an x such that $s_x = 1$, in which case $|\nu(\emptyset,\beta)| \leq \delta$. Thus the terms where $|\beta| \leq \delta T$ contribute at most 2δ.

Next consider the terms where $|\beta| > \delta T$. Applying Corollary 5.20, we get that the probability over U that $|\pi^{U,W}(\beta)| < \frac{\log(1/\delta^2)}{\delta}$ is at most 2δ. Next, we have that

$$\begin{aligned}|\nu(\emptyset,\beta)| &= \left|\prod_{x\in\pi^{U,W}(\beta)}\frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x})\right| \\ &\leq (1-\delta)^{|\pi^{U,W}(\beta)|}.\end{aligned}$$

Thus when $|\pi^{U,W}(\beta)| \geq \frac{log(1/\delta^2)}{\delta}$, we have that

$$(1-\delta)^{|\pi^{U,W}(\beta)|} \leq e^{-\delta|\pi^{U,W}(\beta)|} \leq \delta^2$$

and thus the contribution is bounded by δ. It follows that the contribution of the terms where at $|\beta| > \delta T$ is at most 3δ. Hence

$$\left|\mathop{\mathrm{E}}_{U,W,\kappa}\left[\sum_{\beta\neq\emptyset}\hat{C}^2_{W,\beta}\nu(\emptyset,\beta)\right]\right| \leq 5\delta.$$

as claimed. It follows that (8.5) differ at most 5δ from $\mathrm{E}_{U,\kappa,W}\left[\hat{C}_{U,\kappa,\emptyset}\hat{C}^2_{W,\emptyset}\right]$

We next consider (8.6). Again we consider the sum where $|\beta| < \delta T$ and $|\beta| \geq \delta T$ separately. When $|\beta| \geq \delta T$, we apply the Cauchy–Schwartz inequality and get that

$$\begin{aligned}
&\left| \mathop{\mathrm{E}}_{U,\kappa,W} \left[\sum_{|\beta| \geq \delta T} \sum_{\substack{\alpha \neq \emptyset \\ \alpha \subseteq \pi^{U,W}(\beta)}} \hat{C}_{U,\kappa,\alpha} \hat{C}^2_{W,\beta} \nu(\alpha,\beta) \right] \right| \\
&\leq \mathop{\mathrm{E}}_{U,\kappa,W} \left[\sqrt{\left(\sum_{|\beta| \geq \delta T} \sum_{\substack{\alpha \neq \emptyset \\ \alpha \subseteq \pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha} \hat{C}^2_{W,\beta} \right) \left(\sum_{|\beta| \geq \delta T} \hat{C}^2_{W,\beta} \sum_{\alpha \subseteq \pi^{U,W}(\beta)} \nu(\alpha,\beta)^2 \right)} \right] \\
&\leq \mathop{\mathrm{E}}_{U,\kappa,W} \left[\sqrt{\sum_{|\beta| \geq \delta T} \hat{C}^2_{W,\beta} \sum_{\alpha \subseteq \pi^{U,W}(\beta)} \nu(\alpha,\beta)^2} \right]. \qquad (8.7)
\end{aligned}$$

A calculation gives us that

$$\sum_{\alpha \subseteq \pi^{U,W}(\beta)} \nu(\alpha,\beta)^2 = \prod_{x \in \pi^{U,W}(\beta)} \left(\left(\frac{1}{2}((-1)^{s_x} - (1-2\delta)^{s_x}) \right)^2 + \left(\frac{1}{2}((-1)^{s_x} + (1-2\delta)^{s_x}) \right)^2 \right).$$

The factor corresponding to x in this product is on the form a^2+b^2 where $|a|+|b| = 1$ and $\max(|a|,|b|) \leq 1-\delta$, hence the factor corresponding to x is bounded by $(1-\delta)^2 + \delta^2 \leq 1-\delta$ for $\delta \leq 1/2$. Thus $\sum_{\alpha \subseteq \pi^{U,W}(\beta)} \nu(\alpha,\beta)^2 \leq (1-\delta)^{|\pi^{U,W}(\beta)|}$. Applying Corollary 5.20, we get that with probability at least $1-2\delta$ over the choice of U, $|\pi^{U,W}(\beta)| \geq \frac{\log(1/\delta^2)}{\delta}$, in which case

$$\sqrt{\sum_{|\beta| \geq \delta T} \hat{C}^2_{W,\beta} \sum_{\alpha \subseteq \pi^{U,W}(\beta)} \nu(\alpha,\beta)^2} \leq \sqrt{(1-\delta)^{|\pi^{U,W}(\beta)|}} \leq \delta$$

since in any case the left hand side of the above expression is at most 1, we have that (8.7) is at most 3δ. Next consider the terms in (8.6) where $|\beta| < \delta T$. Again

we apply the Cauchy–Schwartz inequality to get that

$$\begin{aligned}
&\left| \mathop{\mathrm{E}}_{U,\kappa,W}[\sum_{|\beta|<\delta T} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}_{U,\kappa,\alpha}\hat{C}^2_{W,\beta}\nu(\alpha,\beta)] \right| \\
&\leq \mathop{\mathrm{E}}_{U,\kappa,W}\left[\sqrt{\left(\sum_{|\beta|<\delta T} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha}\hat{C}^2_{W,\beta} \right)\left(\sum_{|\beta|<\delta T} \hat{C}^2_{W,\beta} \sum_{\alpha\subseteq\pi^{U,W}(\beta)} \nu(\alpha,\beta)^2 \right)} \right] \\
&\leq \mathop{\mathrm{E}}_{U,\kappa,W}\left[\sqrt{\left(\sum_{|\beta|<\delta T} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha}\hat{C}^2_{W,\beta} \right)\left(\sum_{|\beta|<\delta T} \hat{C}^2_{W,\beta}(1-\delta)^{|\pi^{U,W}(\beta)|} \right)} \right] \\
&\leq \mathop{\mathrm{E}}_{U,\kappa,W}\left[\sqrt{\sum_{|\beta|<\delta T} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha}\hat{C}^2_{W,\beta}} \right]. \qquad (8.8)
\end{aligned}$$

Hence if (8.2) differ more than 9δ from $\mathrm{E}_{U,\kappa,W}\left[\hat{C}_{U,\kappa,\emptyset}\hat{C}^2_{W,\emptyset}\right]$, it must be that (8.8) is at least δ.

We now describe a strategy for the provers in the Generalized 2P1R game: The prover P_1 upon receiving W picks a β with probability proportional to $\hat{C}^2_{W,\beta}$, and returns a random $y \in \beta$. The prover P_2 upon receiving U picks κ uniformly and α with probability proportional to $\hat{C}^2_{U,\kappa,\alpha}$, and returns a random $x \in \alpha$ as its answer. (if α or β is empty, the corresponding prover gives up). The probability of success is at least

$$\begin{aligned}
&\mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{\beta\subseteq \mathrm{SAT}^W} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha}\hat{C}^2_{W,\beta}|\beta|^{-1} \right] \\
&\geq (\delta T)^{-1} \mathop{\mathrm{E}}_{U,\kappa,W}\left[\sum_{|\beta|\leq S} \sum_{\substack{\alpha\neq\emptyset \\ \alpha\subseteq\pi^{U,W}(\beta)}} \hat{C}^2_{U,\kappa,\alpha}\hat{C}^2_{W,\beta} \right] \\
&\geq (\delta T)^{-1}\delta^2 = \frac{\delta^4}{\log(1/\delta^2)}.
\end{aligned}$$

Where we in the last inequality used $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$ (which follows from Jensen's inequality) and the fact that (8.8) is at least δ. ∎

Lemma 8.3. Let $t \leq 3/2 - \epsilon$. Then there is a choice of the parameters u, T, and δ such that if ϕ is at most μ-satisfiable, then for any set of size tP which covers all layer-edges, the fraction of non-layer edges which is covered is at most $1 - \frac{\epsilon^2}{2}$

Proof. Choose $T = \frac{\log(1/\delta^2)}{\delta^3}$, choose δ such that $9\delta \leq \frac{\epsilon^2}{2}$. and u such that the soundness of the Generalized 2P1R game $c_\mu^u \leq \frac{\delta^4}{\log(1/\delta^2)}$ Let

$$p_1 = \Pr_{U,\kappa,f}[B_{U,\kappa}(f) = 0],$$

and

$$p_2 = \Pr_{W,g}[B_W(g) = 0],$$

Then $tP = (p_1 + 1)|Q_1| + (p_2 + 1)|Q_2| = ((p_1 + p_2)/2 + 1)P$. So $p_1 + p_2 \leq 1 - 2\epsilon$. Let

$$\begin{aligned} D_{U,\kappa}(f) &= B_{U,\kappa}^2(f) \\ D_W(g) &= B_W^2(g). \end{aligned}$$

By construction, the tables $B_{U,\kappa}$ and B_W are folded over **true**. We also have that

$$\begin{aligned} \hat{D}_{U,\kappa,\emptyset} &= \Pr_f[B_{U,\kappa}(f) \neq 0] \\ \hat{D}_{W,\emptyset} &= \Pr_g[B_W(g) \neq 0] \end{aligned}$$

so $\mathrm{E}_{U,\kappa}[\hat{D}_{U,\kappa,\emptyset}] = 1 - p_1$, and $E_W[\hat{D}_{W,\emptyset}] = 1 - p_2$. Now consider the terms in (8.1). By the choice of u and Lemma 8.2 with $C_{U,\kappa} = B_{U,\kappa}$, $C_W = B_W$, the term

$$\mathrm{E}_{U,\kappa,W,f,g,h}[B_{U,\kappa}(f)^3 B_W(g)^3 B_W(hg)^3] = \mathrm{E}_{U,\kappa,W,f,g,h}[B_{U,\kappa}(f) B_W(g) B_W(hg)]$$

is at least

$$-9\delta + \mathrm{E}_{U,\kappa,W}\left[\hat{B}_{U,\kappa,\emptyset}\hat{B}_{W,\emptyset}^2\right].$$

Since B_W is folded over **true**, $\hat{B}_{U,\kappa,\emptyset} = 0$, so this is equal to -9δ. Similarly, since $B_{U,\kappa}$ is folded over **true**, the term

$$\mathrm{E}_{U,\kappa,W,f,g,h}[B_{U,\kappa}(f) B_W(g)^2 B_W(hg)^2]$$

is at least -9δ. For the term

$$\mathrm{E}_{U,\kappa,W,f,g,h}[B_{U,\kappa}(f)^2 B_W(g) B_W(hg)]$$

we again apply Lemma 8.2 and by this time using that B_W is folded over **true** we again get that this term is at least -9δ. Finally we have the term

$$\mathop{\mathrm{E}}_{U,\kappa,W,f,g,h}[B_{U,\kappa}(f)^2 B_W(g)^2 B_W(hg)^2]. \tag{8.9}$$

By Lemma 8.2 with $C_{U,\kappa} = D_{U,\kappa}$ and $C_W = D_W$, this is at least

$$-9\delta + \mathop{\mathrm{E}}_{U,\kappa,W}\left[\hat{D}_{U,\kappa,\emptyset}\hat{D}^2_{W,\emptyset}\right]. \tag{8.10}$$

Let $S_{U,\kappa} = \mathrm{E}_W[\hat{D}_{W,\emptyset} \mid U]$. Then $E_{U,\kappa}[S_{U,\kappa}] = 1 - p_2$, and $E_W[\hat{D}^2_{W,\emptyset} \mid U] \geq S^2_{U,\kappa}$. Furthermore we have that

$$\begin{aligned}\mathop{\mathrm{E}}_{U,\kappa}[\hat{D}_{U,\kappa,\emptyset} + S_{U,\kappa}] &= (1-p_1) + (1-p_2)\\ &\geq 1 + 2\epsilon.\end{aligned}$$

Note that, when X, Y are random variables taking values in $[0,1]$, we have that $(X-1)(Y-1) \geq 0$, and thus $XY \geq X + Y - 1$. Taking the expectation, $\mathrm{E}[XY] \geq \mathrm{E}[X] + \mathrm{E}[Y] - 1$. Also, $\mathrm{E}[XY^2] \geq \mathrm{E}[X^2Y^2] \geq \mathrm{E}[XY]^2$. Combining these two inequalities, we get that $\mathrm{E}[XY^2] \geq (\mathrm{E}[X+Y]-1)^2$. Hence

$$\begin{aligned}\mathop{\mathrm{E}}_{U,\kappa,W}[\hat{D}_{U,\kappa,\emptyset}\hat{D}^2_{W,\emptyset}] &\geq \mathop{\mathrm{E}}_{U,\kappa}[\hat{D}_{U,\kappa,\emptyset}S^2_{U,\kappa}]\\ &\geq (\mathop{\mathrm{E}}_{U,\kappa}[\hat{D}_{U,\kappa,\emptyset} + S_{U,\kappa}] - 1)^2\\ &\geq 4\epsilon^2.\end{aligned}$$

To sum up, we have that the fraction of covered non-layer edges is at most

$$1 - \frac{4\epsilon^2 - 36\delta}{4} = 1 - \epsilon^2 + 9\delta \leq 1 - \frac{\epsilon^2}{2}.$$

■

Now we are ready to prove the inapproximability result for Minimum E3-Vertex Cover .

Theorem 8.4. For any $\epsilon > 0$, Minimum E3-Vertex Cover is **NP**-hard to approximate within $3/2 - \epsilon$.

Proof. By Theorem 4.1, deciding any language in **NP**can be reduced to distinguishing between the cases that an E3-CNF(5) formula ϕ is satisfiable of at most μ satisfiable. Choose the parameters u and δ as in Lemma 8.3. Construct the graph H as described above. Then if ϕ is satisfiable, there is a vertex cover of size tP of H. If ϕ is at most μ-satisfiable, for each set S of size at most $(3/2-\epsilon)P$ there must be some uncovered edge, since either some layer-edge is uncovered, or some fraction of non layer-edges is uncovered. Hence if we could distinguish between these two cases, we could decide any language in **NP**. ■

Chapter 9

Vertex cover on 4-uniform hypergraphs

9.1 Introduction

In this chapter we continue the study of Minimum Ek-Vertex Cover and prove that Minimum E4-Vertex Cover is **NP**-hard to approximate within $2 - \epsilon$.

As usual our technique is to construct a specially tailored PCP. There is a twist, however. The standard approach is to construct a PCP such that the optimization problem of computing the proof which has maximum acceptance probability is easy to reduce to the problem at hand. In our case, we instead look at the optimization problem of the minimum number of -1 in the proof which makes the verifier accept with probability 1, and we design our PCP so that this problem is almost trivial to reduce to Minimum E4-Vertex Cover. We use the same tools as in the previous chapters to analyze the acceptance probability of the verifier as a function of the proportion of -1 and 1 in the proof.

This also makes it possible to prove **NP**-hardness of approximation for the maximization version in which we are given a 4-uniform hypergraph and a number B and we are asked to find a subset of the vertices of size B which intersects as many edges as possible. We call this problem Maximum E4-Vertex Cover (or Maximum Ek-Vertex Cover for the general problem on k-uniform hypergraphs). We prove that Maximum E4-Vertex Cover is **NP**-hard to approximate within $16/15 - \epsilon$, for any $\epsilon > 0$.

Our analysis also gives us a proof of the result of Guruswami, Håstad and Sudan [24] that it is **NP**-hard to color a two-colorable hypergraph with a constant number of colors.

9.1.1 Subsequent results

The result by Dinur, Guruswami, Khot and Regev mentioned in Chapter 7 and 8 that Minimum Ek-Vertex Cover is **NP**-hard to approximate within $k-1-\epsilon$ gives a $3-\epsilon$ lower bound for Minimum E4-Vertex Cover. However, as far as I am aware the results for Maximum E4-Vertex Cover presented in this chapter are still the best known.

9.2 PCP for Minimum E4-Vertex Cover

Consider a verifier V which non-adaptively queries k bits. There is a natural k-uniform hypergraph H associated with V, namely the one in which we have a vertex for each position in the proof and for each outcome of the random bits have an edge consisting of the k vertices which correspond to the positions queried. We wish to construct a 4-query PCP so that the vertex cover problem on this hypergraph is hard.

Note that a proof π is naturally associated with a subset S of the vertices of H, defined by $i \in S$ if the corresponding bit $\pi(i) = -1$. This means that if the test in the verifier is that at least one of the queried bits is -1, the acceptance probability is exactly the fraction of edges covered by S in H, if we count the edges with multiplicity.

We mention at this point that there is a slight technicality in that the same set of queries may occur for different random strings. When talking about the fraction of covered edges, we count the edges with this multiplicity. As mentioned in the previous chapter, for minimum vertex cover we do not need to consider the multiplicity of the edges (since whether or not all edges are covered is not affected by how many times each edge occur). The same holds for hypergraph coloring. For the maximization version of minimum vertex cover, however, we need to count the edges with their multiplicity.

If the verifier has perfect completeness and the long code is used to encode correct proofs, a correct proof is balanced, and thus corresponds to a vertex cover of $1/2$ of all vertices. We want the verifier to have the property that if there is no correct proof, then no p-balanced proof makes the verifier accept with probability 1, for p as close to 1 as possible. This implies that when there is no correct proof, there is no vertex cover of size a fraction p of all elements. The following lemma formalizes the above discussion.

Lemma 9.1. Let $1/2 < c < 1$ and suppose there is a $(O(\log n), O(1), 2)$-restricted verifier for some **NP**-complete language L with the following properties:

1. The verifier V non-adaptively reads exactly k bits and accepts if not all are 1.

2. If the input is in L, then there is a balanced proof which V accepts with probability 1.

3. If the input is not in L, then for $p \leq c$, there is no p-balanced proof which V accepts with probability 1.

Then Minimum Ek-Vertex Cover is impossible to approximate within $\frac{2}{c}$ in polynomial time, unless $\mathbf{P} = \mathbf{NP}$.

Proof. On input x, create the following instance of Minimum Ek-Vertex Cover : For each bit $\pi(i)$ in the proof we have an element $x_i \in X$. For each random string σ of the verifier, we have a set S_σ containing the elements corresponding to the bits the verifier reads. A proof π corresponds to a $S \subset X$ by $x_i \in S$ iff the corresponding bit $\pi(i)$ is -1. By the second property of V, if $x \in L$ is satisfiable, then there is a hitting set S of size less than or equal to $|X|/2$. By the third property of V, if $x \notin L$, no set of size smaller than $|X|c$ is a hitting set. Thus it is **NP**-hard to approximate Minimum Ek-Vertex Cover within $\frac{2}{c}$. ∎

The verifier has access to a Generalized Standard Written 2-Proof but only uses the tables corresponding to (T, u)-block W. Hence when considering the proportion of -1 and 1 bits we only count the tables corresponding to (T, u)-blocks W. The verifier is given in Figure 9.1.

This is essentially the same verifier which Håstad [28] uses to prove hardness of approximation of Maximum Set Splitting on satisfiable instances. The difference is in the acceptance predicate and in the use of the Generalized 2P1R game instead of the basic 2P1R game. Instead of having the parameter δ a fixed constant, Håstad picks δ according to a certain distribution. Using the Generalized 2P1R game makes it possible to get by without this complication.

Similarly to the verifier Vc3Test in the previous chapter, this verifier can be viewed as essentially reading an error corrected value of $(f \circ \pi^{U,W_1})$ from the table A_{W_1}, and reading an error corrected value of $-(f \circ \pi^{U,W_2})$ from the table A_{W_2}. Since the "noise" is again not uniform, but only added to points where $(f \circ \pi^{U,W_1})(y) = 1$ and $(f \circ \pi^{U,W_2})(y) = -1$ for the tables A_{W_1} and A_{W_2} respectively, we know that for a correct proof at least one of the products $A_{W_1}(g_1)A_{W_1}(g_1h_1)$ and $A_{W_2}(g_2)A_{W_2}(g_2h_2)$ is -1.

Lemma 9.2. If ϕ is satisfiable, then for $p \geq 1/2$ there is a p-balanced proof which Vc4Test accepts with probability 1. Furthermore, if $p = 1/2$, there is a proof such that the values queried from the proof does not all have the same value (neither -1 or 1).

For $p < 1/2$, there is a proof which Vc4Test accepts with probability at least $1 - (1-p)^4 + p^4 - \delta$.

Proof. We first consider the case $p = 1/2$.

Let x be a satisfying assignment. Then there is a correct proof such that A_W is the long code of $x|_W$. The tables A_{W_1} and A_{W_2} picked by the verifier are long codes of $x|_{W_1}$ and $x|_{W_2}$ respectively. With probability $1/2$, $f(x|_U) = -1$, in which case $A_{W_1}(g_1) = g_1(x|_{W_1})$ and $A_{W_1}(g_1h_1) = -g_1(x|_{W_1})$. Similarly with probability

Vc4Test.
Input: An E3-CNF(5) formula $\phi = C_i \wedge \ldots \wedge C_m$ with n variables and m clauses. The proof is a Generalized Standard Written 2-Proof with parameters u and T.

1. Select a (T,u)-block W_1 uniformly at random, and choose a random u projection U of W_1. Then choose W_2 uniformly at random among the (T,u)-blocks projecting to U.
2. Select uniformly at random $f \in 2^{SAT^U}$ and $g_i \in 2^{SAT^{W_i}}$.
3. Select $h_1 \in 2^{\mathrm{SAT}^{W_1}}$, by for each y, if $f(\pi^{U,W_1}(y)) = -1$ set $h_1(y) = -1$. If $f(\pi^{U,W_1}(y)) = 1$, set $h_1(y) = 1$ with probability $1-\delta$ and $h_1(y) = -1$ with probability δ.
4. Select $h_2 \in 2^{\mathrm{SAT}^{W_2}}$, by for each y, if $f(\pi^{U,W_2}(y) = 1$ set $h_2(y) = -1$. If $f(\pi^{U,W_2}(y)) = -1$, set $h_2(y) = 1$ with probability $1-\delta$ and $h_2(y) = -1$ with probability δ.
5. Reject if

$$A_{W_1}(g_1) = A_{W_1}(g_1h_1) = A_{W_2}(g_2) = A_{W_2}(g_2h_2) = 1$$

else accept.

Figure 9.1. The above PCP is parameterized by the positive integers u and T and the positive real δ and tests if a E3-CNF(5) formula ϕ is satisfiable by querying four positions in a Generalized Standard Written 2-Proof. With suitable choices of the parameters u, T and δ as functions of ϵ, the above PCP has perfect completeness and soundness $1-(1-p)^4+\epsilon$ for any constant $\epsilon > 0$ and a p-balanced proof.

$1/2$, $f(x|_U) = 1$, in which case $A_{W_1}(g_2) = g_2(x|_{W_2})$ and $A_{W_2}(h_2g_2) = -g_2(x_{W_2})$. Hence in each case at least one queried bit is -1, and at least one queried bit is 1.

The case when $p > 1/2$ follows, since changing 1 entries to -1 entries in a correct proof will not decrease the acceptance probability.

For the case $p < 1/2$, we consider the proof where each table is as close to a correct proof as possible. That is, the prover begins with a satisfying assignment x to ϕ and A_W is made as close to $\mathrm{LC}(x|_W)$ as possible by changing a fraction $1-2p$ randomly chosen occurrences of -1 to 1. Note that in a correct proof with probability at least $\frac{1-\delta}{2}$ one of the queried bit is -1 in which case the probability that this bit has been changed is $1-2p$ and with probability at least $\frac{1-\delta}{2}$ three of the queried bits are -1 in which case the probability that all these bits have been

changed is at most $(1-2p)^3$. Thus the verifier accepts with probability at least

$$\begin{aligned}
&\frac{1-\delta}{2}(1-(1-2p)+1-(1-2p)^3) \\
&\geq \quad \frac{1-(1-2p)+1-(1-2p)^3}{2}-\delta \\
&= \quad 4p-6p^2+4p^4-\delta \\
&= \quad 1-(1-p)^4+p^4-\delta.
\end{aligned}$$

■

For the soundness analysis, it turns out to be convenient to change from $\{1,-1\}$ notation to $\{0,1\}$ notation for the proof tables. This idea is due to Khot [39] and greatly simplifies the analysis as compared to the original proof in [32]. The soundness analysis is also simpler than the original in that we use the generalized two prover protocol described in Section 5.2, instead of the original u parallel protocol.

We define for each (T,u)-block W k new table B_W:

$$B_W(g) = \begin{cases} 1 & \text{if } A_W(g)=1 \\ 0 & \text{if } A_W(g)=-1. \end{cases}$$

We note some properties of B_W:

$$\sum_{\alpha \subseteq \mathrm{SAT}^W} \hat{B}^2_{W,\alpha} = \Pr_g[B_W(g)=1] \tag{9.1}$$

$$\hat{B}_{W,\emptyset} = \Pr_g[B_W(g)=1]. \tag{9.2}$$

The probability that the verifier accepts may be written as

$$1 - \mathop{\mathrm{E}}_{W_1,W_2,f,g_1,g_2,h_1,h_2}[B_{W_1}(g_1)B_{W_1}(g_1h_1)B_{W_2}(g_2)B_{W_2}(g_2h_2)]. \tag{9.3}$$

We next rewrite this expression using Fourier-expansion:

Lemma 9.3. The acceptance probability of Vc4Test may be written as

$$1 - \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\alpha \subseteq \mathrm{SAT}^{W_1}} \sum_{\beta \subseteq \mathrm{SAT}^{W_2}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right]$$

where

$$\nu(\alpha,\beta) = \prod_{x \in \mathrm{SAT}^U} \frac{1}{2}\left((-1)^{s_x}(1-2\epsilon)^{t_x} + (1-2\epsilon)^{s_x}(-1)^{t_x}\right)$$

where s_x is the number of $y \in \alpha$ such that $\pi^{U,W_1}(y) = x$ and t_x is the number of $y \in \beta$ such that $\pi^{U,W_2}(y) = x$.

Proof. For fixed W_1 and W_2, we apply Fourier-inversion on B_{W_1} and B_{W_2}. By the usual calculations, we get that

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,g_1,g_1,g_2,h_1,h_2}[B_{W_1}(g_1)B_{W_1}(g_1h_1)B_{W_2}(g_2)B_{W_2}(g_2h_2)] \\
&\quad = \sum_{\alpha\subseteq \mathrm{SAT}^{W_1}} \sum_{\beta\subseteq \mathrm{SAT}^{W_2}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta} \mathop{\mathrm{E}}_{f,h_1,h_2}[\chi_\alpha(h_1)\chi_\beta(h_2)].
\end{aligned}$$

We proceed to calculate the expectation over f, h_1 and h_2.

$$\begin{aligned}
&\mathop{\mathrm{E}}_{f,h_1,h_2}[\chi_\alpha(h_1)\chi_\beta(h_2)] \\
&\quad = \mathop{\mathrm{E}}_{f,h_1,h_2}\left[\prod_{y_1\in\alpha} h_1(y) \prod_{y_2\in\alpha_2} h_2(y)\right] \\
&\quad = \mathop{\mathrm{E}}_{f,h_1,h_2}\left[\prod_{x\in\mathrm{SAT}^U} \prod_{\substack{y\in\alpha \\ \pi^{U,W_1}(y)=x}} h_1(y) \prod_{\substack{y\in\beta \\ \pi^{U,W_2}(y)=x}} h_2(y)\right] \\
&\quad = \prod_{x\in\mathrm{SAT}^U} \frac{1}{2}\left(\left(\prod_{\substack{y\in\alpha \\ \pi^{U,W_1}(y)=x}} (-1) \prod_{\substack{y\in\beta \\ \pi^{U,W_2}(y)=x}} (1-2\delta)\right) + \left(\prod_{\substack{y\in\alpha \\ \pi^{U,W_1}(y)=x}} (1-2\delta) \prod_{\substack{y\in\beta \\ \pi^{U,W_2}(y)=x}} (-1)\right)\right).
\end{aligned}$$

This can be seen to be equal to $\nu(\alpha,\beta)$. Hence the acceptance probability may be written as

$$1 - \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\alpha,\beta} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right].$$

■

Thus we need to bound from below the expression

$$\begin{aligned}
&\mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\alpha,\beta}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right] = \\
&\quad \mathop{\mathrm{E}}_{W_1,W_2}[\hat{B}^2_{W_1,\emptyset}\hat{B}^2_{W_2,\emptyset}] + \\
&\quad \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{|\alpha|\geq\delta T \text{ or } |\beta|\geq\delta T}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right] + \\
&\quad \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ \alpha\neq\emptyset \text{ or } \beta\neq\emptyset \\ |\alpha|,|\beta|\leq\delta T}}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right] + \\
&\quad \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset \\ |\alpha|,|\beta|\leq\delta T}}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right].
\end{aligned}$$

One intuition for this way of breaking up the sum is that the different types of terms corresponds to different kinds of strategies for a (possibly cheating) prover. The first type of term corresponds to strategies in which the prover try to fool us by manipulating the concentration of 1 and -1 for different tables (but distribute the bits randomly within each table). It turns out that the best strategy is to distribute the bits evenly over all tables.

The second type of term corresponds to strategies which are based on making the tables close to large products of long codes. These strategies will fail because of Corollary 5.20, i.e., that a large $|\beta|$ has a large projection with large probability.

The third type of term corresponds to strategies which are based on making the tables close to small products of long codes which are mutually inconsistent. It turns out that these will fail because of Corollary 5.18, i.e., that for small β, with large probability there is some $y \in \beta$ with unique projection.

Finally, the fourth type of term corresponds to strategies in which the prover tries to make the tables close to small products of long codes where there is some consistency between the tables. It turns out that these strategies must fail, because otherwise we could construct a strategy for the Generalized 2P1R game with good success probability.

Lemma 9.4. For a p-balanced proof,

$$\mathop{\mathrm{E}}_{W_1,W_2}[\hat{B}^2_{W_1,\emptyset}\hat{B}^2_{W_2,\emptyset}] \geq (1-p)^4$$

Proof. Fix U and let

$$p_U = \mathop{\mathrm{E}}_{W_1}[\Pr_g[B_{W_1} = 0] \mid U]. \tag{9.4}$$

Clearly $\mathrm{E}_{W_1}[\hat{B}_{W_1,\emptyset} \mid U] = \mathrm{E}_{W_2}[\hat{B}_{W_2,\emptyset} \mid U] = (1 - p_U)$, and since W_1 and W_2 are independent and identically distributed once we have conditioned on U, we have that

$$\mathop{\mathrm{E}}_{W_1,W_2}[\hat{B}^2_{W_1,\emptyset}\hat{B}^2_{W_2,\emptyset} \mid U] = \mathop{\mathrm{E}}_{W}[\hat{B}^2_{W,\emptyset} \mid U]^2 \geq (1 - p_U)^4$$

where we in the inequality used that $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$ by Jensen's inequality. Taking the expectation over U and again using Jensen's inequality to get that that $\mathrm{E}[X^4] \geq \mathrm{E}[X]^4$, we get

$$\mathop{\mathrm{E}}_{U}[(1 - p_U)^4] \geq \mathop{\mathrm{E}}_{U}[(1 - p_U)]^4 = (1 - p)^4.$$

■

Next consider the terms where at least one of α and β contains at least δT elements.

Lemma 9.5. For a Generalized Standard Written 2-Proof with parameter $T = \frac{\log(1/\delta)}{\delta^3}$,

$$\left| \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{|\alpha|\geq\delta T \text{ or} |\beta|\geq\delta T} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta) \right] \right| \leq 6\delta.$$

Proof. We have that

$$\begin{aligned}
&\left| \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{|\alpha|\geq\delta T \text{ or} |\beta|\geq\delta T} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta) \right] \right| \\
&\quad\leq \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{|\alpha|\geq\delta T \text{ or} |\beta|\geq\delta T} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}|\nu(\alpha,\beta)| \right] \\
&\quad\leq \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{|\alpha|\geq\delta T \\ \beta\subseteq \mathrm{SAT}^{W_2}}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}|\nu(\alpha,\beta)| \right] + \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{|\beta|\geq\delta T \\ \alpha\subseteq \mathrm{SAT}^{W_1}}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}|\nu(\alpha,\beta)| \right] \\
&\quad= 2 \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{|\alpha|\geq\delta T \\ \beta\subseteq \mathrm{SAT}^{W_2}}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}|\nu(\alpha,\beta)| \right].
\end{aligned} \tag{9.5}$$

Furthermore,

$$\begin{aligned}
|\nu(\alpha,\beta)| &= \left|\prod_{x\in \mathrm{SAT}^U} \frac{1}{2}((-1)^{s_x}(1-2\delta)^{t_x} + (1-2\delta)^{s_x}(-1)^{t_x})\right| \\
&\leq \prod_{x\in \mathrm{SAT}^U} \frac{1}{2}(\left|(-1)^{s_x}(1-2\delta)^{t_x}\right| + \left|(1-2\delta)^{s_x}(-1)^{t_x}\right|) \\
&\leq \prod_{x\in \mathrm{SAT}^U} \frac{1}{2}(1+(1-2\delta)^{s_x}) \\
&= (1-\delta)^{|\pi^{U,W_1}(\alpha)|}.
\end{aligned}$$

So (9.5) is at most

$$\begin{aligned}
&2 \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{|\alpha|\geq\delta T \\ \beta\subseteq \mathrm{SAT}^{W_2}}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}(1-\delta)^{|\pi^{U,W_1}(\alpha)|}\right] \\
&= 2 \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{|\alpha|\geq\delta} \hat{B}^2_{W_1,\alpha}(1-\delta)^{|\pi^{U,W_1}(\alpha)|}\right], \qquad (9.6)
\end{aligned}$$

where the equality follows by Parseval's equality. For fixed W_1, by Corollary 5.20, the probability over U that $|\pi^{U,W_1}(\alpha)| < \frac{log(1/\delta)}{\delta}$ is at most 2δ. If $|\pi^{U,W_1}(\alpha)| \geq \frac{\log(1/\delta)}{\delta}$,

$$(1-\delta)^{|\pi^{U,W_1}(\alpha)|} \leq e^{-\delta|\pi^{U,W_1}(\alpha)|} \leq \delta$$

where we used that for real x, $1+x \leq e^x$. It follows that (9.6) is at most 6δ. The lemma follows. ∎

Next we consider the sum where both β and α contains less than δT elements, the projections are disjoint, and at least one of α and β is non-empty.

Lemma 9.6. For a Generalized Standard Written 2-Proof with parameter $T = \frac{\log(1/\delta)}{\delta^3}$,

$$\left|\mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ \alpha\neq\emptyset \text{ or } \beta\neq\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta)\right]\right| \leq 6\delta.$$

Proof. We have that

$$\left| \mathop{\mathrm{E}}_{W_1,W_2} \left[\sum_{\substack{\alpha\neq\emptyset \text{ or } \beta\neq\emptyset \\ \pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\nu(\alpha,\beta) \right] \right| \leq$$

$$2 \mathop{\mathrm{E}}_{W_1,W_2} \left[\sum_{\substack{\alpha\neq\emptyset \\ \pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\left|\nu(\alpha,\beta)\right| \right].$$

Thus consider

$$\mathop{\mathrm{E}}_{W_1,W_2} \left[\sum_{\substack{\alpha\neq\emptyset \\ \pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\left|\nu(\alpha,\beta)\right| \right].$$

This is equal to

$$\mathop{\mathrm{E}}_{W_1} \left[\sum_{\substack{\alpha\neq\emptyset \\ |\alpha|\leq\delta T}} \hat{B}^2_{W_1,\alpha} \mathop{\mathrm{E}}_{W_2} \left[\sum_{\substack{\beta:\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset \\ |\beta|\leq\delta T}} \hat{B}^2_{W_2,\beta}\left|\nu(\alpha,\beta)\right| \right] \right]. \tag{9.7}$$

When $\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)=\emptyset$, $\nu(\alpha,\beta)$ can be written as

$$\prod_{x\in\pi^{U,W_1}(\alpha)} \frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x}) \prod_{x\in\pi^{U,W_2}(\beta)} \frac{1}{2}((-1)^{t_x}+(1-2\delta)^{t_x}).$$

Hence, since $|\frac{1}{2}((-1)^{t_x}+(1-2\delta)^{t_x})|\leq 1$, it follows that (9.7) is less than or equal to

$$\mathop{\mathrm{E}}_{W_1} \left[\sum_{\substack{\alpha\neq\emptyset \\ |\alpha|\leq\delta T}} \hat{B}^2_{W_1,\alpha} \mathop{\mathrm{E}}_{W_2} \left[\left| \prod_{x\in\pi^{U,W_1}(\alpha)} \frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x}) \right| \sum_{|\beta|\leq\delta T} \hat{B}^2_{W_2,\beta} \right] \right] \leq$$

$$\mathop{\mathrm{E}}_{W_1} \left[\sum_{\substack{\alpha\neq\emptyset \\ |\alpha|\leq\delta T}} \hat{B}^2_{W_1,\alpha} \mathop{\mathrm{E}}_{W_2} \left[\left| \prod_{x\in\pi^{U,W_1}(\alpha)} \frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x}) \right| \right] \right].$$

Where the last inequality holds because of Parseval's equality. We next consider

$$\mathop{\mathrm{E}}_{W_2}\left[\left|\prod_{x\in\pi^{U,W_1}(\alpha)}\frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x})\right|\right]. \tag{9.8}$$

By Corollary 5.18, with probability at least $1-|\beta|/T \geq 1-\delta$ over the choice of U, there is an x such that $s_x = 1$, in which case

$$\frac{1}{2}((-1)^{s_x}+(1-2\delta)^{s_x}) = \delta.$$

Since the other factors in (9.8) have absolute value at most 1, the entire product can be bounded by δ in this case. The case that there is no x such that $s_x = 1$ happens with probability at most δ. It follows that (9.8) is at most 2δ. The lemma follows. ∎

Finally we consider the small α and β where the intersection of the projections is non-empty, and prove that if the sum over these terms is large, the is a strategy for the Generalized 2P1R game with good success probability.

Lemma 9.7. There is a strategy for the Generalized 2P1R game with success probability at least

$$\mathop{\mathrm{E}}_{W_1,W_2}\left[(\delta T)^{-2}\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset\\ |\alpha|,|\beta|\leq\delta T}}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\right].$$

Proof. The prover P_1, on receiving W_2 selects a β with probability $\hat{B}^2_{W_2,\beta}$ and returns a random $y_2 \in \beta$. On receiving U, the prover P_2 picks a random (T,u) block W_1 which project to U, and then α with probability $\hat{B}^2_{W_1,\alpha}$. Then P_2 picks a random $y' \in \alpha$ and returns $x = \pi^{U,W_1}(y')$. The probability of picking α and β where the projections on U have non-empty intersection is

$$\sum_{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}.$$

The probability of the provers picking y and y' which projects to the same element in the intersection is at least $|\alpha|^{-1}|\beta|^{-1}$. Thus the probability that the verifier accepts is at least

$$\sum_{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}|\alpha|^{-1}|\beta|^{-1} \geq$$
$$\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset\\ |\alpha|,|\beta|\leq\delta T}}\hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta},(\delta T)^{-2}.$$

The lemma follows. ∎

Lemma 9.8. For any $\epsilon > 0$, there is a choice of the parameters T, u and δ in Vc4Test such that if ϕ is at most μ-satisfiable, a p-balanced proof is accepted with probability at most $1 - (1-p)^4 + \epsilon$

Proof. Choose $\delta = \epsilon/10$, $T = \frac{\log(1/\delta)}{\delta^3}$ and u such that $c_\mu^u < \frac{\delta^5}{\log(1/\delta)^2}$. By Lemma 9.3, the probability that the verifier accepts may be written as

$$1 - \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\alpha,\beta} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta} \prod_{x\in \mathrm{SAT}^U} \nu(s_x, t_x)\right]. \tag{9.9}$$

Now, by Lemma 9.4, Lemma 9.5 and Lemma 9.6, this expectation is at most

$$1 - (1-p)^4 - 9\delta - \mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\right].$$

Furthermore, if

$$\mathop{\mathrm{E}}_{W_1,W_2}\left[\sum_{\substack{\pi^{U,W_1}(\alpha)\cap\pi^{U,W_2}(\beta)\neq\emptyset \\ |\alpha|,|\beta|\leq\delta T}} \hat{B}^2_{W_1,\alpha}\hat{B}^2_{W_2,\beta}\right] \geq \delta,$$

by Lemma 9.7 there is a strategy the Generalized 2P1R game with success probability at least $(\delta T)^{-2}\delta = \frac{\delta^5}{\log(1/\delta)^2} > c_\mu^u$, a contradiction. Thus (9.9) is at most $1 - (1-p)^4 + 10\delta \leq 1 - (1-p)^4 + \epsilon$. ∎

9.3 Hardness of approximating Minimum E4-Vertex Cover

We can now prove our main theorem of this chapter:

Theorem 9.9. For any $\epsilon > 0$, Minimum E4-Vertex Cover is **NP**-hard to approximate within $2 - \epsilon$.

Proof. By Theorem 4.1, deciding any language in **NP** can be reduced to distinguishing between the cases that an E3-CNF(5) formula ϕ is satisfiable of at most μ satisfiable.

By Lemma 9.2 and Lemma 9.8 for any $\delta > 0$ there is a choice of the parameters for the verifier Vc4Test such that for $p \leq 1 - \epsilon$, if ϕ is satisfiable, there is a balanced proof which is is accepted with probability 1, and if ϕ is at most μ-satisfiable, no p-balanced proof is accepted with probability 1. Thus we have that the verifier Vc4Test fulfills the assumptions of Lemma 9.1 with $c = 1 - \epsilon$, and we are done.

∎

For Maximum E4-Vertex Cover, we have the following:

Theorem 9.10. For any $\epsilon > 0$, Maximum E4-Vertex Cover is **NP**-hard to approximate within $16/15 - \epsilon$.

Proof. Create the same hypergraph as in Lemma 9.1, with the difference that we count the edges with multiplicity, i.e., the edges form a multiset. By Lemma 9.2 and Lemma 9.8 with $p = 1/2$ it is **NP**-hard to distinguish between the case that a there is a set of size $B = n/2$ which intersects all sets and the case that all sets of size $n/2$ intersects at most a fraction $15/16 + \epsilon$ of all edges The theorem follows. ∎

The previous statement says that an algorithm which should work for any size B can at best approximate Maximum E4-Vertex Cover within $16/15 + \epsilon$. However, we can also say something about the for the problem Maximum E4-Vertex Cover-p where the relative size of the covering set is fixed.

Theorem 9.11. For any $\epsilon > 0$, when $p \geq 1/2$, it is **NP**-hard to approximate Maximum E4-Vertex Cover-p within

$$\frac{1}{1-(1-p)^4} + \epsilon.$$

When $p < 1/2$, Maximum E4-Vertex Cover-p is **NP**-hard to approximate within

$$\frac{1-(1-p)^4+p^4}{1-(1-p)^4} + \epsilon$$

Proof. Create the same instance of Maximum E4-Vertex Cover as in Theorem 9.10. When $p \geq 1/2$ we have from Lemma 9.2 that if ϕ is satisfiable then there is a set S of size pn which intersects all sets. If ϕ is at most μ-satisfiable, then by Lemma 9.8 there is no set S of size pn which intersects more than a fraction $1-(1-p)^4+\delta$ of the sets. Since δ can be made arbitrarily small, this proves the case $p \geq 1/2$. For the case when $p < 1/2$, we have that if ϕ is satisfiable then there is a set S of size B which intersects a fraction $1-(1-p)^4+p^4-\delta$ of all sets. As before if ϕ is at most μ-satisfiable, there is no set S of size B which intersects more than a fraction $1-(1-p)^4+10\delta$. By an appropriate choice of δ the case $p < 1/2$ follows. ∎

Guruswami, Håstad and Sudan proved that it is **NP**-hard to color a two-colorable hypergraph with a constant number of colors [24]. The analysis of our PCP gives us an alternate proof of hardness of coloring two-colorable hypergraphs:

Theorem 9.12. For any constant c, there is no polynomial time algorithm which colors all two-colorable 4-uniform hypergraph with c colors, unless $\mathbf{P} = \mathbf{NP}$.

Proof. Create the same hypergraph as in Theorem 9.9, with $\epsilon < 1/c$. Call this hypergraph H. Proofs correspond to colorings in the obvious way, i.e., color vertices

corresponding to positions which is -1 red and vertices corresponding to positions which 1 blue. If ϕ is satisfiable, by Lemma 9.2, there is a proof such that each set of queries made by the verifier contains both a 1 and a -1, hence H is two-colorable in this case.

By the proof of Theorem9.9, if ϕ is at most μ-satisfiable, there is no vertex cover of size less than $(1-\epsilon)|V|$. Since the complement to a vertex cover is an independent set, there is no independent set of size $\geq \epsilon|V|$. Since each color-class must form an independent set, it follows that we must use at least $|V|/(|V|\epsilon) = 1/\epsilon > c$ colors to color H

Hence if there was a polynomial time algorithm which colored all two-colorable 4-uniform hypergraph with c colors, it would in the case of ϕ satisfiable, color H with c colors, but would have to use $> c$ colors in the case of ϕ at most μ-satisfiable. Thus we could distinguish between these two cases, and thus decide a **NP**-hard problem in polynomial time. ∎

Chapter 10

Non-Boolean 3 query PCP with perfect completeness

10.1 Introduction

A natural question to ask is: What is the best soundness that can be achieved with a small number of queries? In Chapter 4 we reviewed the construction of Håstad [28] and saw that it is possible to construct a three-query PCP where the soundness is arbitrarily close to 1/2. To be more precise: for any $L \in \mathbf{NP}$ and any given constant $\epsilon > 0$, Håstad's construction gives a proof system that accepts inputs in L with probability at least $1-\epsilon$ and accepts inputs that are not in L with probability at least $1/2+\epsilon$. Since Håstad's proof system accepts inputs in L only with probability $1-\epsilon$, and not always, we say that it has *near-perfect completeness*. By contrast, recall that a proof system is said to have *perfect completeness* if inputs in the language are always accepted.

Samorodnitsky and Trevisan [45] used a clever extension of Håstad's construction to prove that any language in **NP** can be recognized by a (k^2+2k)-query PCP with near-perfect completeness and soundness $2^{-k^2}+\epsilon$ for any $\epsilon > 0$. This result is known to be essentially optimal since it is not possible to get better soundness than 2^{1-q} for a non-adaptive q-query PCP for **NP** unless $\mathbf{P} = \mathbf{NP}$ [50].

For several reasons we would prefer the proof systems to have perfect completeness. Firstly, it is simply esthetically pleasing to have a proof system where a correct proof is always be accepted. Secondly, in several proofs of hardness of approximation, perfect completeness has been important. For example, when proving hardness of hypergraph coloring [24, 40] (see also Chapter 9) and vertex cover on hypergraphs (Chapter 7, 8 and 9) perfect completeness appears to be crucial. Thirdly, it is of theoretical interest to get an understanding of how near-perfect versus perfect completeness and adaptive versus non-adaptive verifiers affect our ability to construct proof systems with low soundness. For non-adaptive 3-query PCPs,

Håstad's construction provides a PCP for **NP** with near-perfect completeness and soundness $1/2+\epsilon$ for every constant $\epsilon > 0$ and it is impossible to achieve soundness better than $5/8$ in proof systems with perfect completeness unless $\mathbf{P} = \mathbf{NP}$ [51, 54]. Using the tools from Håstad's paper, it is possible to construct a three query PCP with perfect completeness and soundness $3/4+\epsilon$ for any constant $\epsilon > 0$ (indeed, we used essentially this PCP in Chapter 8, although we never analyzed its soundness explicitly) and Guruswami et al. [25], building on Håstad's work, proved that there is an adaptive PCP for **NP** with perfect completeness and soundness $1/2 + \epsilon$ for any constant $\epsilon > 0$. For more than 3 queries, Håstad and Khot [29] proved that a non-adaptive verifier can achieve perfect completeness and soundness $2^{-k^2} + \epsilon$ with $k^2 + 4k$ queries, while an adaptive verifier can achieve the same completeness and soundness with only $k^2 + 2k$ queries.

In all PCPs mentioned so far, the proof has been a list of bits, i.e., the proof has been *binary*. Håstad also constructed a PCP where the positions in the proof contain elements from a larger domain in his paper [28]: For the case when each position in the proof contains an element from a domain of size d, where d is an arbitrary integer constant, he constructed a proof system with near-perfect completeness and soundness $d^{-1} + \epsilon$ for any constant $\epsilon > 0$. Engebretsen [11] adapted the construction of Samorodnitsky and Trevisan [45] to domains of size d, thereby constructing a proof system that queries $k^2 + 2k$ positions in the proof, has near-perfect completeness and soundness $d^{-k^2} + \epsilon$ for any $\epsilon > 0$.

This brings up the question that we address in this chapter: How does the soundness behave, as a function of the domain size d, for 3-query PCPs with perfect completeness? In Section 7.3.1 we saw that it is easy to construct a proof system with perfect completeness and soundness $2/d$ for $d \geq 3$. In this chapter, we prove that for a verifier with perfect completeness it is possible to achieve soundness arbitrarily close to $d^{-1} + d^{-2}$ with a non-adaptive PCP, and soundness arbitrarily close to d^{-1} if we allow the verifier to be adaptive.

This generalizes the constructions for the binary case [25, 28] and shows that we can achieve, as a function of the domain size, the same higher order term in the soundness for PCPs with perfect completeness as for the best currently known PCPs with near-perfect completeness.

It appears difficult to get stronger results using current techniques. All of the most efficient PCPs are essentially variants of the simple PCP which we reviewed in Chapter 4 (generalized modulo d in our case). Since this test accepts on a linear condition, a random assignment will always fool the verifier with probability $\frac{1}{d}$. Thus we need another kind of test to get below $\frac{1}{d}$. However, since in the boolean case it is known that a non-adaptive 3-query PCP cannot get soundness below $\frac{1}{2}$ with near perfect completeness and not below $5/8$ with perfect completeness, it might very well be the case that in general it is impossible for a non-adaptive 3-query PCP to get soundness below $\frac{1}{d}$ (with perfect or near-perfect completeness). However, for $d > 2$ it is currently only known that it is impossible to get soundness d^{-2} in a 3-query PCP for **NP** [46] and it is at present unknown if it is possible to get better

NonadaptiveTest .
Input: An E3-CNF(5) formula ϕ. The proof is a Generalized Standard Written d-Proof with parameters u and T, with all the tables folded over $\boldsymbol{Z}_d$:

1. Let W be a random (T,u)-block selected from Φ.
2. Let U be a random u-projection of W.
3. Select $f \in Z_d^{\mathrm{SAT}^U}$ and $g \in Z_d^{\mathrm{SAT}^W}$ uniformly at random.
4. Select $e_f \in Z_d^{\mathrm{SAT}^W}$ by selecting, independently for every $y \in \mathrm{SAT}^W$, $e_f(y)$ such that: $f(\pi^{U,W}(y)) \neq 1 \implies e_f(y) = 1$; $f(\pi^{U,W}(y)) = 1 \implies (\Pr[e_f(y) = 1] = 1 - \delta) \wedge (\Pr[e_f(y) = \omega] = \delta)$.
5. Accept if $A_{U,\boldsymbol{Z}_d}(f)A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f \circ \pi^{U,W})^{-1}e_f) = 1$ or if $A_{U,\boldsymbol{Z}_d}(f) = 1$ and $A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f \circ \pi^{U,W})^{-1}e_f) = \omega$; reject otherwise.

Figure 10.1. The above PCP is parameterized by the positive integers d, u and T and the positive real δ and tests if a E3-CNF(5) formula Φ is satisfiable by querying three positions in a Generalized Standard Written d-Proof with parameter u. With suitable choices of the parameters u, T and δ as functions of ϵ and d, the above PCP has perfect completeness and soundness $d^{-1} + d^{-2} + \epsilon$ for any constant $\epsilon > 0$.

soundness with near-perfect completeness than with perfect completeness.

10.2 Non-adaptive PCP construction

The PCP is shown in Fig. 10.1. The intuition behind the protocol is to take a linearity test—which we know has soundness at most $d^{-1}+\epsilon$ for any constant $\epsilon > 0$ and near-perfect completeness—and modify it a slightly in such a way that we get at test with perfect completeness which is still close enough to a linearity test to have soundness $d^{-1} + d^{-2} + \epsilon$ for any constant $\epsilon > 0$. The following lemma is immediate:

Lemma 10.1. If ϕ is satisfiable, then there is a proof which NonadaptiveTest accepts with probability 1.

To analyze the soundness of the protocol we follow the standard approach. We estimate the acceptance probability of the verifier using Fourier analysis and prove that if the acceptance probability is large there must exist pairs of correlated coefficients whose product is large. We then use these products to devise a strategy for the provers in the balanced version of the 2P1R game from Section 5.2. To simplify

the notation, we let $F = Z_d^{\mathrm{SAT}^U}$ and $H = Z_d^{\mathrm{SAT}^W}$. The groups $\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^U}$ and $\hat{\boldsymbol{Z}}_d^{\mathrm{SAT}^W}$ are denoted $\hat{F}$ and $\hat{H}$ respectively.

The expression $d^{-1}\sum_{\gamma=0}^{d-1} g^\gamma$ is an indicator for the event $g = 1$ when g assumes values in $\boldsymbol{Z}_d$. Hence, the test accepts with probability $\mathrm{E}[I_1 + I_2]$ where

$$\begin{aligned} I_1 &= \frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^\gamma, \\ I_2 &= \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)\Big)^\gamma\right)\times \\ &\quad \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\omega^{-1}\Big)^\gamma\right). \end{aligned}$$

This expression can be rewritten as $d^{-1} + d^{-2} + \mathrm{E}[L + Q + C_1 + C_2]$ where

$$L = \frac{1}{d}\sum_{\gamma=1}^{d-1}(A_{U,\boldsymbol{Z}_d}(f))^\gamma, \tag{10.1}$$

$$Q = \frac{1}{d^2}\sum_{\gamma=1}^{d-1}\Big(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\omega^{-1}\Big)^\gamma, \tag{10.2}$$

$$C_1 = \frac{1}{d}\sum_{\gamma=1}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^\gamma, \tag{10.3}$$

$$\begin{aligned} C_2 &= \frac{1}{d^2}\sum_{\gamma_1=1}^{d-1}\sum_{\gamma_2=1}^{d-1} \\ &\quad \Big(A_{U,\boldsymbol{Z}_d}(f)\Big)^{\gamma_1}\Big(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\omega^{-1}\Big)^{\gamma_2}. \end{aligned} \tag{10.4}$$

We now prove that L and Q have small magnitude and then prove that most of the terms in C_1 and C_2 also have small magnitude. Finally, we prove that the remaining terms can be used to extract a strategy for the provers in the Generalized 2P1R game from Section 5.2.

Lemma 10.2. $\mathrm{E}[L] = 0$

Proof. $\mathrm{E}_f[A_{U,\boldsymbol{Z}_d}(f)^\gamma] = \hat{B}_{\mathbf{0}}$, where $B(f) = A_{U,\boldsymbol{Z}_d}(f)^\gamma$. Since $A_{U,\boldsymbol{Z}_d}$ is folded over $\boldsymbol{Z}_d$, B is γ-homogeneous. It follows by Lemma 5.11 that $\hat{B}_{\mathbf{0}} = 0$, and thus $\mathrm{E}[L] = 0$. ∎

Let us now analyze $\mathrm{E}[Q]$:

$$\mathrm{E}[Q] = \mathop{\mathrm{E}}_{U,W,f,g,e_f}\left[\frac{1}{d^2}\sum_{\gamma=1}^{d-1}\omega^{-\gamma}\big(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\big)^\gamma\right].$$

Our aim is to show that

$$\mathop{\mathrm{E}}_{U,W}[Q_\gamma] = \mathop{\mathrm{E}}_{U,W,f,g,e_f}\left[\left(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\right)^\gamma\right] \tag{10.5}$$

always has small magnitude, regardless of γ. To this end, we expand Q_γ in a Fourier series:

$$\begin{aligned} Q_\gamma &= \mathop{\mathrm{E}}_{f,g,e_f}\left[\sum_{\beta_1,\beta_2\in\hat{H}} \hat{B}_{\beta_1}\hat{B}_{\beta_2}\chi_{\beta_1}(g)\chi_{\beta_2}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\right] \\ &= \sum_{\beta_1,\beta_2\in\hat{H}} \hat{B}_{\beta_1}\hat{B}_{\beta_2} \mathop{\mathrm{E}}_{f,e_f}[\chi_{\beta_2}(e_f)\chi_{-\pi_d^{U,W}(\beta_2)}(f)] \mathop{\mathrm{E}}_{g}[\chi_{\beta_1-\beta_2}(g)], \end{aligned}$$

where $B = A^\gamma_{W,\boldsymbol{Z}_d}$, and $\hat{B}_{\beta_1}$ thus is the Fourier coefficient of $A^\gamma_{W,\boldsymbol{Z}_d}$ at β. Since $\mathrm{E}_h[\chi_{\beta_1-\beta_2}(h)] = 0$ unless $\beta_1 = \beta_2$ by Lemma 5.1, the above expression can be simplified to

$$Q_\gamma = \sum_{\beta\in\hat{H}} \hat{B}_\beta^2 \mathop{\mathrm{E}}_{f,e_f}[\chi_\beta(e_f)\chi_{\pi_d^{U,W}(\beta)}(f)]$$

Let us now compute

$$\begin{aligned} \mathop{\mathrm{E}}_{f,e_f}[\chi_\beta(e_f)\chi_{-\pi_d(\beta)}(f)] &= \mathop{\mathrm{E}}_{f,e_f}\left[\prod_y (e_f(y))^{\beta(y)}(f(\pi^{U,W}(y)))^{-\beta(y)}\right] \\ &= \prod_{x\in\pi^{U,W}(\beta)} \mathrm{E}\left[\prod_{y:\pi^{U,W}(y)=x}(e_f(y)f^{-1}(x))^{\beta(y)}\right]. \end{aligned}$$

Consider the factor corresponding to a fixed x in the above product. Since $e_f(y)$ depends on $f(x)$ we condition on $f(x)$:

$$\begin{aligned} &\mathrm{E}\left[\prod_{y:\pi^{U,W}(y)=x}(e_f(y)f^{-1}(x))^{\beta(y)}\right] \\ &\qquad = \frac{1}{d}\sum_{t=0}^{d-1} \mathrm{E}\left[\prod_{y:\pi^{U,W}(y)=x}(e_f(y)f^{-1}(x))^{\beta(y)} \;\middle|\; f(x)=\omega^t\right]. \end{aligned}$$

If $f(x)\neq 1$, $e_f(y)$ is always 1, otherwise $e_f(y)$ is selected according to a biased distribution on $\{1,\omega\}$. Hence

$$\begin{aligned} &E\left[\prod_{y:\pi^{U,W}(y)=x}(e_f(y)f^{-1}(x))^{\beta(y)}\right] \\ &\qquad = \frac{1}{d}\sum_{t=1}^{d-1}\prod_{y:\pi^{U,W}(y)=x}\omega^{-t\beta(y)} + \frac{1}{d}\prod_{y:\pi^{U,W}(y)=x}(1-\delta+\omega^{\beta(y)}\delta) \end{aligned}$$

Recall the notation $\pi_d^{U,W}(\beta)(x) = \sum_{y:\pi^{U,W}(y)=x} \beta(y) \bmod d$. If $\pi_d^{U,W}(\beta)(x) \neq 0$, the first sum above is -1; otherwise it is $d-1$. If we define

$$a_{\beta,x} = \frac{1}{d}\Bigg(\prod_{y:\pi^{U,W}(y)=x} (1-\delta+\omega^{\beta(y)}\delta) + d - 1\Bigg), \tag{10.6}$$

$$b_{\beta,x} = \frac{1}{d}\Bigg(\prod_{y:\pi^{U,W}(y)=x} (1-\delta+\omega^{\beta(y)}\delta) - 1\Bigg), \tag{10.7}$$

we can write

$$\begin{aligned}
|Q_\gamma|^2 &= \Bigg|\sum_{\beta\in\hat{H}} \hat{B}_\beta^2 \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)=0}} a_{\beta,x} \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)\neq 0}} b_{\beta,x}\Bigg|^2 \\
&\leq \Bigg(\sum_{\beta\in\hat{H}} |\hat{B}_\beta|^2\Bigg)\Bigg(\sum_{\beta\in\hat{H}} |\hat{B}_\beta|^2 \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)=0}} |a_{\beta,x}|^2 \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)\neq 0}} |b_{\beta,x}|^2\Bigg) \\
&= \sum_{\beta\in\hat{H}} |\hat{B}_\beta|^2 \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)=0}} |a_{\beta,x}|^2 \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \pi_d^{U,W}(\beta)(x)\neq 0}} |b_{\beta,x}|^2,
\end{aligned}$$

where the inequality is the Cauchy–Schwartz inequality and the last equality follows from Plancherel's equality.

Lemma 10.3. For any integer $b \neq 0 \bmod d$ and any real $\delta \in [0,1]$, $|1-\delta+\omega^b\delta|^2 \leq 1-\delta d^{-2}$ and, consequently, $|1-\delta+\omega^b\delta| \leq 1-\delta d^{-2}/2$.

Proof. Since $\omega = e^{2\pi i/d}$,

$$\begin{aligned}
|1-\delta+\omega^b\delta|^2 &\leq |1-\delta+\omega\delta|^2 \\
&= (1-\delta(1-\cos(2\pi/d)))^2 + \delta^2\sin^2(2\pi/d) \\
&= 1 - 2(\delta-\delta^2)\big(1-\cos(2\pi/d)\big).
\end{aligned}$$

For $d \in \{2,3,4\}$ this expression is at most $1-2\delta(1-\delta) \leq 1-\delta$ while for larger d it is at most

$$1 - 2(\delta-\delta^2)(2\pi^2 d^{-2} - 2\pi^4 d^{-4}/3) \leq 1 - 2\pi^2 d^{-2}\delta(1-\delta) \leq 1-\delta d^{-2}.$$

where we use the MacLaurin expansion of $\cos(x)$ to derive that $\cos(2\pi/d) < 1 - 2\pi^2 d^{-2} + 2\pi^4 d^{-4}/3$. The conclusion of the lemma now follows since $(1-\eta)^{1/2} \leq 1-\eta/2$ for all $\eta \in [0,1]$. ∎

Using separate arguments for terms where $|\beta| = |\{y \in \hat{H} : \beta(y) \neq 0\}|$ is large and terms where $|\beta|$ is small, we now bound the factor multiplying $|\hat{B}_\beta|^2$ by 2δ. For the case when $|\beta|$ is small, we use Corollary 5.18 and when $|\beta|$ is large, we use Corollary 5.20.

Lemma 10.4. Let Q_γ be defined as in equation (10.5) and let $\gamma \in \hat{\boldsymbol{Z}}_d \setminus \{0\}$ and $\delta \in (0,1)$ be arbitrary. Then $\mathrm{E}_{U,W}\left[|Q_\gamma|^2\right] \leq 2\delta$ provided that $T \geq \delta^{-3} d^3 \ln \delta^{-1}$.

Proof. By the above reasoning, we have established that

$$\mathrm{E}\left[|Q_\gamma|^2\right] \leq \mathop{\mathrm{E}}_{W}\left[\sum_{\beta \in \hat{H}} |\hat{B}_\beta|^2 \mathop{\mathrm{E}}_{U}\left[\prod_{\substack{x \in \pi^{U,W}(\beta) \\ \pi_d^{U,W}(\beta)(x)=0}} |a_{\beta,x}|^2 \prod_{\substack{x \in \pi^{U,W}(\beta) \\ \pi_d^{U,W}(\beta)(x)\neq 0}} |b_{\beta,x}|^2\right]\right] \tag{10.8}$$

We now bound the inner expectation above separately for every β and then use Plancherel's equality to bound the entire sum.

Suppose that $|\beta| < \delta T$ and $\hat{B}_\beta \neq 0$. It follows by Lemma 4.10 that $\hat{B}_\beta = 0$ if $\beta(y) = 0$ for all y; hence there exists some y such that $\beta(y) \neq 0$. By Corollary 5.18, with probability at least $1 - |\beta|/T \geq 1 - \delta$ over the choice of U there is no other y' such that $\pi^{U,W}(y) = \pi^{U,W}(y')$. Given that this event happens, $\pi_d^{U,W}(\beta)(\pi^{U,W}(y)) = \beta(y) \neq 0$, and the factor corresponding to $x = \pi^{U,W}(y)$ in the product has magnitude

$$\frac{|1 - \delta + \omega^{\beta(y)}\delta - 1|}{d^2} = \frac{\delta |1 - \omega^{\beta(y)}|}{d^2} \leq \frac{2\delta}{d^2}.$$

Since the other factors have at most unit magnitude, the magnitude of the inner expectation is bounded by $2\delta d^{-2}$ when $|\beta| < \delta T$. Since in any case the entire product has at most unit magnitude, the factor multiplying $|\hat{B}_\beta|^2$ in (10.8) is at most $(1-\delta) \cdot 2\delta d^{-2} + \delta \cdot 1 \leq 3\delta/2$ when $|\beta| \leq \delta T$.

Next, we consider the case when $|\beta| \geq \delta T$. Since $|a_{\beta,x}| \geq |b_{\beta,x}|$, we can upper bound the product inside the inner expectation in (10.8) by $|a_{\beta,x}|^{2|\pi^{U,W}(\beta)|} \leq (1 - \delta d^{-3}/2)^{2|\pi^{U,W}(\beta)|}$, using Lemma 10.3 to bound $|a_{\beta,x}|$. By Corollary 5.20, $|\pi^{U,W}(\beta)| \geq \delta|\beta| = \delta^{-1} d^3 \ln \delta^{-1}$ with probability at least $1 - 2\delta$ over the choice of U. When this holds,

$$(1 - \delta d^{-3}/2)^{2|\pi^{U,W}(\beta)|} \leq e^{-\delta d^{-3} |\pi^{U,W}(\beta)|} = \delta.$$

When $|\pi^{U,W}(\beta)|$ fails to be large, which happens with probability at most 2δ, the product has at most unit magnitude. Therefore, the factor multiplying $|\hat{B}_\beta|^2$ in (10.8) is at most $(1 - 2\delta) \cdot \delta + \delta \cdot 1 \leq 2\delta$ when $|\beta| \geq \delta T$.

To conclude, the term corresponding to an arbitrary β in (10.8) has magnitude at most $2\delta |\hat{B}_\beta|^2$, hence

$$\mathrm{E}\left[|Q_\gamma|^2\right] \leq 2\delta \,\mathrm{E}\left[\sum_{\beta \in \hat{H}} |\hat{B}_\beta|^2\right] = 2\delta.$$

∎

Corollary 10.5. Let Q be defined as in equation (10.2) and $\delta \in (0,1)$ be arbitrary. Then $|\,\mathrm{E}[Q]|^2 \leq \delta$ provided that $T \geq \delta^{-3} d^3 \ln \delta^{-1}$.

Proof. Since $\mathrm{E}[Q] = \mathrm{E}[d^{-2}\sum_{\gamma=1}^{d-1}\omega^{-\gamma}Q_\gamma]$, and the function $z \mapsto |z|^2$ is convex, it follows from Jensen's inequality that

$$|\mathrm{E}[Q]|^2 = \frac{(d-1)^2}{d^4}\left|\frac{1}{d-1}\sum_{\gamma=1}^{d-1}\omega^{-\gamma}\,\mathrm{E}[Q_\gamma]\right|^2 \le \frac{d-1}{d^4}\sum_{\gamma=1}^{d-1}|\,\mathrm{E}[Q_\gamma]|^2$$
$$\le \frac{d-1}{d^4}\sum_{\gamma=1}^{d-1}\mathrm{E}[|Q_\gamma|^2] \le \frac{2\delta(d-1)^2}{d^4},$$

where the last inequality follows from Lemma 10.4 and the first two inequalities follow from Jensen's inequality. ∎

Let us now look at the terms of degree three, i.e., the expectations (10.3) and (10.4). They can be written as

$$\mathrm{E}[C_1] = \mathop{\mathrm{E}}_{U,W}\left[\frac{1}{d}\sum_{\gamma=1}^{d-1}C_{\gamma,\gamma}\right], \qquad \mathrm{E}[C_2] = \mathop{\mathrm{E}}_{U,W}\left[\frac{1}{d^2}\sum_{\gamma_1=1}^{d-1}\sum_{\gamma_2=1}^{d-1}\omega^{-\gamma_2}C_{\gamma_1,\gamma_2}\right]$$

where

$$C_{\gamma_1,\gamma_2} = \mathop{\mathrm{E}}_{f,g,e_f}\Big[\big(A_{U,\boldsymbol{Z}_d}(f)\big)^{\gamma_1}\big(A_{W,\boldsymbol{Z}_d}(g)A_{W,\boldsymbol{Z}_d}(g^{-1}(f\circ\pi^{U,W})^{-1}e_f)\big)^{\gamma_2}\Big]. \quad (10.9)$$

To prove that $\mathrm{E}[C_1]$ and $\mathrm{E}[C_2]$ have small magnitude, it is enough to bound $|C_{\gamma_1,\gamma_2}|$ for arbitrary $\gamma_1,\gamma_2 \neq 0$. To this end, we expand C_{γ_1,γ_2} in a Fourier series:

$$C_{\gamma_1,\gamma_2} = \sum_{\alpha\in\hat{F}^{\boldsymbol{Z}_d}}\sum_{\beta\in\hat{H}}\hat{A}_\alpha\hat{B}_\beta^2\mathop{\mathrm{E}}_{f,e_f}[\chi_\beta(e_f)\chi_{\alpha-\pi_d^{U,W}(\beta)}(f)]$$

where $\hat{A}_\alpha$ is the Fourier coefficient of $A_{U,\boldsymbol{Z}_d}^{\gamma_1}$ at α and $\hat{B}_\beta$ is the Fourier coefficient of $A_{W,\boldsymbol{Z}_d}^{\gamma_2}$ at β. Since e_f and f are dependent, we cannot simply discard all terms where $\alpha \neq \pi_d^{U,W}(\beta)$. However, we can discard some terms. For $\alpha \in \hat{F}$ and $\beta \in \hat{H}$, we let $\alpha \preceq \beta$ denote the relation that for each x such that $\alpha(x) \neq 0$, there is some $y \in (\pi^{U,W})^{-1}(x)$ such that $\beta(y) \neq 0$. Consider

$$\mathop{\mathrm{E}}_{f,e_f}[\chi\beta(e_f)\chi_{\alpha-\pi_d^{U,W}(\beta)}(f)] =$$
$$\prod_{x\in\mathrm{SAT}^U}\mathrm{E}\Big[(f(x))^{\alpha(x)}\prod_{y:\pi^{U,W}(y)=x}(e_f(y)f^{-1}(x))^{\beta(y)}\Big]. \quad (10.10)$$

If $\alpha \preceq \beta$ does not hold, there is some x_0 such that $\alpha(x_0) \neq 0$ but $\beta(y) = 0$ for all $y \in (\pi^{U,W})^{-1}(x_0)$. The factor corresponding to this x_0 will be 0. Thus the only terms which remain are those where $\alpha \preceq \beta$, and

$$C_{\gamma_1,\gamma_2} = \sum_{\beta\in\hat{H}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2\mathop{\mathrm{E}}_{f,e_f}[\chi_\beta(e_f)\chi_{\alpha-\pi_d^{U,W}(\beta)}(f)].$$

Let us now compute 10.10 in the case when $\alpha \preceq \beta$. We consider each factor separately. The factor corresponding to an arbitrary x can be written

$$\begin{aligned}
&\mathrm{E}\left[(f(x))^{\alpha(x)} \prod_{y:\pi^{U,W}(y)=x} (e_f(y)f^{-1}(x))^{\beta(y)}\right] \\
&= \frac{1}{d} \prod_{y:\pi^{U,W}(y)=x} (1-\delta+\omega^{\beta(y)}\delta) + \frac{1}{d}\sum_{t=1}^{d-1} \omega^{t\alpha(x)} \prod_{y:\pi^{U,W}(y)=x} \omega^{-t\beta(y)} \\
&= \frac{1}{d} \prod_{y:\pi^{U,W}(y)=x} (1-\delta+\omega^{\beta(y)}\delta) + \frac{1}{d}\sum_{t=1}^{d-1} \omega^{t(\alpha(x)-\pi_d^{U,W}(\beta)(x))}.
\end{aligned}$$

If $\alpha(x) \neq \pi_d^{U,W}(\beta)(x)$, the second term evaluates to $-d^{-1}$; otherwise it evaluates to $(d-1)/d$. Therefore,

$$C_{\gamma_1,\gamma_2} = \sum_{\beta\in\hat{H}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}_\beta^2 p(\alpha,\beta) \tag{10.11}$$

where

$$p(\alpha,\beta) = \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \alpha(x)=\pi_d^{U,W}(\beta)(x)}} a_{\beta,x} \prod_{\substack{x\in\pi^{U,W}(\beta)\\ \alpha(x)\neq\pi_d^{U,W}(\beta)(x)}} b_{\beta,x} \tag{10.12}$$

and $a_{\beta,x}$ and $b_{\beta,x}$ are defined in equations (10.6) and (10.7) above. We now proceed to bound the terms corresponding to terms with $|\beta| > \delta T$.

Lemma 10.6. For any $\beta \in \hat{H}$,

$$\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 = \prod_{x\in\pi^{U,W}(\beta)} (|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2),$$

where $a_{\beta,x}$, $b_{\beta,x}$ and $p(\alpha,\beta)$ are defined in equations (10.6), (10.7) and (10.12).

Proof. There are $d^{|\pi^{U,W}(\beta)|}$ different α such that $\alpha \preceq \beta$. Of those, exactly one α, namely $\alpha = \pi_d^{U,W}(\beta)$ gives a product with only factors of type a. More generally, let $A \subseteq \pi^{U,W}(\beta)$ be the set of x such that $\alpha(x) = \pi_d^{U,W}(\beta)(x)$. Then there are $(d-1)^{|\pi^{U,W}(\beta)\setminus A|}$ different α such that α and $\pi_d^{U,W}(\beta)$ agree exactly on A and for all those α,

$$|p(\alpha,\beta)|^2 = \prod_{x\in A} |a_{\beta,x}|^2 \prod_{x\in\pi^{U,W}(\beta)\setminus A} |b_{\beta,x}|^2.$$

Hence,

$$\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 = \sum_{A\subseteq\pi^{U,W}(\beta)} (d-1)^{|\pi^{U,W}(\beta)\setminus A|} \prod_{x\in A} |a_{\beta,x}|^2 \prod_{x\in\pi^{U,W}(\beta)\setminus A} |b_{\beta,x}|^2$$
$$= \prod_{x\in\pi^{U,W}(\beta)} (|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2).$$

■

Lemma 10.7. For all integers $d \geq 2$,

$$\sum_{\substack{\alpha\in F\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 \leq \left(1 - \frac{\delta}{d^3}\right)^{|\pi^{U,W}(\beta)|}$$

Proof. By Lemma 10.6,

$$\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 = \prod_{x\in\pi^{U,W}(\beta)} (|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2).$$

We now bound each factor in the above product using the observation that

$$|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2 = \big(a_{\beta,x} + (d-1)b_{\beta,x}\big)\big(a_{\beta,x} + (d-1)b_{\beta,x}\big)^*$$
$$- (d-1)\big(a_{\beta,x}b^*_{\beta,x} + a^*_{\beta,x}b_{\beta,x}\big) - (d-1)(d-2)|b_{\beta,x}|^2, \quad (10.13)$$

where z^* denotes the complex conjugate of z. We introduce the shorthand

$$c_{\beta,x} = \prod_{y:\pi^{U,W}(y)=x} (1 - \delta + \omega^{\beta(y)}\delta).$$

With this shorthand, by the definitions of $a_{\beta,x}$ and $b_{\beta,x}$,

$$a_{\beta,x} = \frac{1}{d}(c_{\beta,x} + d - 1)$$
$$b_{\beta,x} = \frac{1}{d}(c_{\beta,x} - 1).$$

Using these, and the notation $\Re\{z\}$ for the real part of z, we now expand each of the expressions in the right-hand side of (10.13):

$$a_{\beta,x} + (d-1)b_{\beta,x} = c_{\beta,x},$$
$$(a_{\beta,x} + (d-1)b_{\beta,x})(a_{\beta,x} + (d-1)b_{\beta,x})^* = |c_{\beta,x}|^2,$$
$$a^*_{\beta,x}b_{\beta,x} = d^{-2}\big(|c_{\beta,x}|^2 + (d-1)c_{\beta,x} - c^*_{\beta,x} - (d-1)\big),$$
$$a_{\beta,x}b^*_{\beta,x} + a^*_{\beta,x}b_{\beta,x} = 2d^{-2}\big(|c_{\beta,x}|^2 + (d-2)\Re\{c_{\beta,x}\} - (d-1)\big),$$
$$|b_{\beta,x}|^2 = d^{-2}\big(|c_{\beta,x}|^2 - 2\Re\{c_{\beta,x}\} + 1\big),$$

Hence $a_{\beta,x}b^*_{\beta,x} + a^*_{\beta,x}b_{\beta,x} + (d-2)|b_{\beta,x}|^2 = d^{-1}|c_{\beta,x}|^2 - d^{-1}$ and

$$|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2 = |c_{\beta,x}|^2 - \frac{d-1}{d}\Big(|c_{\beta,x}|^2 - 1\Big) = 1 - \frac{1-|c_{\beta,x}|^2}{d}\,.$$

By Lemma 10.3,

$$|c_{\beta,x}|^2 = \prod_{y:\pi^{U,W}(y)=x} |1-\delta+\omega^{\beta(y)}\delta|^2 \leq (1-\delta d^{-2})^{|(\pi^{U,W})^{-1}(x)|} \leq 1-\delta d^{-2}$$

and we can therefore bound each $|a_{\beta,x}|^2 + (d-1)|b_{\beta,x}|^2$ from above by $1-\delta d^{-3}$. Since there are $|\pi^{U,W}(\beta)|$ factors in the product, the proof is complete. ∎

As before, the terms in the Fourier series are bounded in separate ways depending on the size of $|\beta|$. We first prove that the sum of all terms corresponding to large $|\beta|$ must have a magnitude that is upper bounded by an expression linear in $\delta^{1/2}$. Intuitively, this follows since $p(\alpha,\beta)$ has small magnitude with very high probability when $|\beta|$ is large. Here we, again, use Corollary 5.20 from Section 5.2. We then prove that the terms corresponding to small $|\beta|$ must also have a magnitude that is upper bounded by an expression linear in $\delta^{1/2}$, or else there exists a strategy for the provers in the 2P1R game from Section 5.2 that makes the verifier accept with probability larger than c^u_μ.

Lemma 10.8. Suppose that $B\colon Z_d^{\mathrm{SAT}^W} \to \boldsymbol{Z}_d$ is γ_2-homogeneous for some $\gamma_2 \neq 0$, that $A\colon \to^{\mathrm{SAT}^U} \boldsymbol{Z}_d$ is γ_1-homogeneous for some $\gamma_1 \neq 0$. Let $\hat{A}_\alpha$ be the Fourier coefficients of A and $\hat{B}_\beta$ be the Fourier coefficients of B and let $\alpha \preceq \beta$ mean that $\alpha(x) \neq 0 \implies x \in \pi(\beta)$. If

$$\mathrm{E}\Bigg[\sum_{\beta\in\hat{H}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2 |\hat{B}_\beta|^2 |\beta|^{-1}\Bigg] \geq \eta,$$

where the probability is over the selection of a (T,u)-block W uniformly at random, and a random u-projection U of W, there exists a strategy for the provers in the Generalized 2P1R game from Section 5.2 that makes the verifier in that game accepts with probability at least η.

Proof. The strategy is as follows: On receiving a sequence W of clauses, the first prover computes the Fourier coefficients $\hat{B}_\beta$, selects a β according to probability distribution given by $|\hat{B}_\beta|^2$ and then a y such that $\beta(y) \neq 0$ uniformly—by Lemma 4.10 such a y always exists—this y is returned to the verifier.

On receiving a sequence U of clauses and variables, the second prover computes the Fourier coefficients $\hat{A}_\alpha$, selects an α according to probability distribution given by $|\hat{A}_\alpha|^2$ and then an x such that $\alpha(x) \neq 0$ uniformly—by Lemma 4.10 such an x always exists—this x is returned to the verifier.

The assignment y always satisfies the clauses in W and it is guaranteed to be consistent if $\alpha \preceq \beta$ and the second prover happens to select a y that projects onto the x selected by the first prover. Therefore, the success probability of the above strategy is at least

$$\mathrm{E}\Bigg[\sum_{\beta\in\hat{H}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}|\hat{A}_\alpha|^2|\hat{B}_\beta|^2|\beta|^{-1}\Bigg] \geq \eta.$$

∎

Lemma 10.9. Let C_{γ_1,γ_2} be defined as in equation (10.9) and let $\gamma_1,\gamma_2 \in \hat{\boldsymbol{Z}}_d \setminus \{0\}$ be arbitrary. Then, provided that $T \geq \delta^{-3}d^3\ln\delta^{-1}$,

$$|C_{\gamma_1,\gamma_2}|^2 \leq 2\left|\mathrm{E}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\leq\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2 + 6\delta,$$

where $\hat{A}_\alpha$ is the Fourier coefficient of $A^{\gamma_1}_{U,\boldsymbol{Z}_d}$ at α and $\hat{B}_\beta$ is the Fourier coefficient of $A^{\gamma_2}_{W,\boldsymbol{Z}_d}$ at β.

Proof. By equation (10.11),

$$\begin{aligned}\frac{|C_{\gamma_1,\gamma_2}|^2}{2} &= \frac{1}{2}\left|\mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\beta\in\hat{H}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2\\ &\leq \left|\mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\leq\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2 + \left|\mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2\end{aligned}$$

where the inequality follows from Jensen's inequality. We now bound the latter sum above. By the Cauchy–Schwartz inequality and Parseval's equality applied to $\hat{B}_\beta$,

$$\begin{aligned}&\left|\sum_{\substack{\beta\in\hat{H}\\ |\beta|\geq\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}\hat{A}_\alpha\hat{B}_\beta^2 p(\alpha,\beta)\right|^2\\ &\leq \Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}|\hat{A}_\alpha|^2|\hat{B}_\beta|^2\Bigg)\Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}}\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}|\hat{B}_\beta|^2|p(\alpha,\beta)|^2\Bigg)\\ &\leq \sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}}|\hat{B}_\beta|^2\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}}|p(\alpha,\beta)|^2,\end{aligned}$$

hence

$$\left| \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2 \le \mathop{\mathrm{E}}_{W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|>\delta T}} |\hat{B}_\beta|^2 \mathop{\mathrm{E}}_{U}\Bigg[\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2\Bigg]\Bigg].$$

Consider the expression multiplying $|\hat{B}_\beta|^2$. Since $|\beta| > \delta T \ge \delta^{-2} d^3 \ln \delta^{-1}$, Corollary 5.20 implies that $|\pi^{U,W}(\beta)| \ge \delta^{-1} d^3 \ln \delta^{-1}$ with probability at least $1 - 2\delta$ over the choice of U. When this holds,

$$\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 \le \left(1 - \frac{\delta}{d^3}\right)^{|\pi^{U,W}(\beta)|} \le \exp(\delta d^{-3} |\pi^{U,W}(\beta)|) \le \delta.$$

In the other case, when $|\pi^{U,W}(\beta)|$ fails to be large,

$$\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2 \le 1.$$

To conclude, for any fixed W,

$$\mathop{\mathrm{E}}_{U}\Bigg[\sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |p(\alpha,\beta)|^2\Bigg] \le (1-2\delta)\cdot\delta + 2\delta\cdot 1 \le 3\delta$$

and hence

$$\mathop{\mathrm{E}}_{U,W}\Bigg[\Bigg| \sum_{\substack{\beta\in\hat{H}\\ |\beta|\ge T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}_\beta^2 p(\alpha,\beta)\Bigg|^2\Bigg] \le 3\delta \mathop{\mathrm{E}}_{W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\ge T}} |\hat{B}_\beta|^2\Bigg].$$

Since the latter expectation is at most 1 due to Plancherel's equality, the conclusion of the lemma follows. ∎

Lemma 10.10. Let C_{γ_1,γ_2} be defined as in equation (10.9) and let $\gamma_1, \gamma_2 \in \hat{\boldsymbol{Z}}_d \setminus \{0\}$ and $\delta \in (0,1)$ be arbitrary. Then if ϕ is at most μ satisfiable, $\mathrm{E}\big[|C_{\gamma_1,\gamma_2}|^2\big] \le 8\delta$ provided that $T \ge \delta^{-3} d^3 \ln \delta^{-1}$ and $u > (\log T)/\log c_\mu^{-1}$.

Proof. By Lemma 10.9,

$$|C_{\gamma_1,\gamma_2}|^2 \le 2\left| \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}_\beta^2 p(\alpha,\beta)\Bigg]\right|^2 + 6\delta,$$

where $\hat{A}_\alpha$ is the Fourier coefficient of $A^{\gamma_1}_{U,\boldsymbol{Z}_d}$ at α and $\hat{B}_\beta$ is the Fourier coefficient of $A^{\gamma_2}_{W,\boldsymbol{Z}_d}$ at β. Now suppose that $|C_{\gamma_1,\gamma_2}|^2 > 8\delta$. Then

$$\begin{aligned}
\delta &< \left| \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}^2_\beta p(\alpha,\beta)\Bigg]\right|^2 \\
&\le \mathop{\mathrm{E}}_{U,W}\left[\left| \sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} \hat{A}_\alpha \hat{B}^2_\beta p(\alpha,\beta)\right|^2\right] \\
&\le \mathop{\mathrm{E}}_{U,W}\left[\Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2|\hat{B}_\beta|^2\Bigg)\Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{B}_\beta|^2|p(\alpha,\beta)|^2\Bigg)\right] \\
&\le \mathop{\mathrm{E}}_{U,W}\left[\Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2|\hat{B}_\beta|^2\Bigg)\Bigg(\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} |\hat{B}_\beta|^2\Big(1-\frac{\delta}{d^3}\Big)^{|\pi^{U,W}(\beta)|}\Bigg)\right] \\
&\le \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2|\hat{B}_\beta|^2\Bigg] \\
&\le \delta T \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\beta\in\hat{H}\\ |\beta|\le\delta T}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2|\hat{B}_\beta|^2|\beta|^{-1}\Bigg] \\
&\le \delta T \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\beta\in\hat{H}} \sum_{\substack{\alpha\in\hat{F}\\ \alpha\preceq\beta}} |\hat{A}_\alpha|^2|\hat{B}_\beta|^2|\beta|^{-1}\Bigg].
\end{aligned}$$

We can now apply Lemma 10.8 with $\eta = T^{-1}$ to see that there exists a strategy for the provers in Generalized 2P1R game from Section 5.2 with success rate η. Note that the functions $A^{\gamma_1}_{U,\boldsymbol{Z}_d}$ and $A^{\gamma_2}_{W,\boldsymbol{Z}_d}$ are known to the provers in that game since the provers can simply try all possible combinations of γ_1, γ_2, $A_{U,\boldsymbol{Z}_d}$, and $A_{W,\boldsymbol{Z}_d}$ and select the combination that gives the largest expectation. But since u has been selected in such a way that $c^u_\mu < \eta$, we obtain a contradiction, and hence $|C_{\gamma_1,\gamma_2}|^2 \le 8\delta$. ∎

Corollary 10.11. Let C_1 and C_2 be defined as in equations (10.3)–(10.4) and $\delta \in (0,1)$ be arbitrary. Then if ϕ is at most μ-satisfiable, $|\mathrm{E}[C_1]|^2 \le 8(d-1)\delta$ and $|\mathrm{E}[C_2]|^2 \le 8\delta$ provided that $T \ge \delta^{-3}d^3 \ln\delta^{-1}$ and $u > (\log T)/\log c_\mu^{-1}$.

Proof. By the definitions of C_1 and C_2 from equations (10.3)–(10.4) it follows that

$$|\operatorname{E}[C_1]|^2 = \frac{(d-1)^2}{d^2}\left|\frac{1}{d-1}\sum_{\gamma=1}^{d-1}\operatorname{E}[C_{\gamma,\gamma}]\right|^2 \leq \frac{(d-1)}{d^2}\sum_{\gamma=1}^{d-1}|\operatorname{E}[C_{\gamma,\gamma}]|^2$$
$$\leq \frac{(d-1)}{d^2}\sum_{\gamma=1}^{d-1}\operatorname{E}[|C_{\gamma,\gamma}|^2] \leq \frac{8\delta(d-1)^2}{d^2}$$

and similarly that

$$|\operatorname{E}[C_2]|^2 \leq \frac{(d-1)^2}{d^4}\sum_{\gamma_1=1}^{d-1}\sum_{\gamma_2=1}^{d-1}\operatorname{E}[|C_{\gamma_1,\gamma_2}|^2] \leq \frac{8\delta(d-1)^4}{d^4}\,.$$

Hence $|\operatorname{E}[C_1]|^2 \leq 8\delta$ and $|\operatorname{E}[C_2]|^2 \leq 8\delta$. ∎

Putting the pieces together, it turns out that if we first select δ as a function of ϵ, then select T as a function of ϵ and δ and finally select u as a function of ϵ, δ and T, we can prove that all the terms above sum up to something having magnitude strictly less than ϵ under the assumption that the verifier in the PCP accepts with probability $d^{-1} + d^{-2} + \epsilon$.

Lemma 10.12. For any integer $d \geq 2$ and any constant $\epsilon > 0$, there are choices of the parameters δ, T and u such that if ϕ is at most μ-satisfiable, the verifier in Fig. 10.1 with these parameter choices accepts with probability most $d^{-1} + d^{-2} + \epsilon$.

Proof. Given ϵ, first select $\delta \leq \epsilon^2/52$ and then select $T \geq \delta^{-3}d^3 \ln \delta^{-1}$ and $u > (\log \delta^{-1} + \log T + T \log d)/\log c_\mu^{-1}$. Now suppose that the verifier in Fig. 10.1 with these parameter choices accepts an incorrect input with probability $d^{-1} + d^{-2} + \epsilon$. Then

$$\epsilon = \operatorname{E}[L] + \operatorname{E}[Q] + \operatorname{E}[C_1] + \operatorname{E}[C_2] = \operatorname{E}[Q] + \operatorname{E}[C_1] + \operatorname{E}[C_2],$$

where L, Q, C_1, and C_2 are defined as in (10.1)–(10.4) and the last equality follows since $\operatorname{E}[L] = 0$ by Lemma 10.2. Jensen's inequality now implies that

$$\epsilon^2 \leq 3|\operatorname{E}[Q]|^2 + 3|\operatorname{E}[C_1]|^2 + 3|\operatorname{E}[C_2]|^2$$

Since $|\operatorname{E}[Q]|^2 \leq \delta$ by Corollary 10.5 and $|\operatorname{E}[C_1]|^2 + |\operatorname{E}[C_2]|^2 \leq 16\delta$ by Corollary 10.11, we obtain a contradiction. ∎

Combing Lemma 10.1 and Lemma 10.12, we get

Theorem 10.13. For any $d \geq 2$, for any constant $\epsilon > 0$,

$$\mathbf{NP} = \mathbf{naPCP}_{1,d^{-1}+d^{-2}+\epsilon}[O(\log n), 3, d].$$

AdaptiveTest .
Input: An E3-CNF(5) formula ϕ. The proof is a Generalized Standard Written d-Proof with parameters u and T:

1. Let W be a random (T,u)-block selected from Φ.
2. Let U be a random u-projection of W.
3. Select $f \in Z_d^{\mathrm{SAT}^U}$ and $g \in Z_d^{\mathrm{SAT}^W}$ uniformly at random.
4. Select $e_f \in Z_d^{\mathrm{SAT}^W}$ by selecting, independently for every $y \in \mathrm{SAT}^W$, $e_f(y)$ such that: $f(\pi(y)) \neq 1 \implies e_f(y) = 1$; $f(\pi(y)) = 1 \implies (\Pr[e_f(y) = 1] = 1 - \delta) \wedge (\Pr[e_f(y) = \omega] = \delta)$.
5. Select $e_{\omega f} \in Z_d^{\mathrm{SAT}^W}$ by selecting, independently for every $y \in \mathrm{SAT}^W$, $e_{\omega f}(y)$ such that: $\omega f(\pi^{U,W}(y)) \neq 1 \implies e_{\omega f}(y) = 1$; $\omega f(\pi^{U,W}(y)) = 1 \implies (\Pr[e_{\omega f}(y) = 1] = 1 - \delta) \wedge (\Pr[e_{\omega f}(y) = \omega] = \delta)$.
6. If $A_{U,\boldsymbol{Z}_d}(f) \neq 1$, accept if $A_{U,\boldsymbol{Z}_d}(f) A_{W,\boldsymbol{Z}_d}(h) A_{W,\boldsymbol{Z}_d}(h^{-1}(f \circ \pi^{U,W})^{-1} e_f) = 1$;
   If $A_{U,\boldsymbol{Z}_d}(f) = 1$, accept if $A_{W,\boldsymbol{Z}_d}(h) A_{W,\boldsymbol{Z}_d}(h^{-1}(\omega f \circ \pi^{U,W})^{-1} e_{\omega f}) = \omega^{-1}$; reject otherwise.

Figure 10.2. The above PCP is parameterized by the positive integers d, u and T and the positive real δ and tests if a E3-CNF(5) formula Φ is satisfiable by adaptively querying three positions in a Generalized Standard Written d-Proof with parameters u and T. With suitable choices of the parameters u, T and δ as functions of ϵ and d, the above PCP has perfect completeness and soundness $d^{-1} + \epsilon$ for any constant $\epsilon > 0$.

Proof. Choose the parameters of NonadaptiveTest as in Lemma 10.12.

By Theorem 4.1, deciding any language in **NP**can be reduced to distinguishing between the cases that an E3-CNF(5) formula ϕ is satisfiable of at most μ satisfiable.

By Lemma 10.1, the verifier has perfect completeness. By Lemma 10.12, if ϕ is at most μ-satisfiable, no proof is accepted with probability greater than $d^{-1} + d^{-2} + \epsilon$. The lemma follows. ∎

10.3 Adaptive PCP construction

In this section we give an adaptive version of the PCP construction in the previous section. The PCP is shown in Figure 10.2. The verifier proceeds as in the nonadaptive case when the first queried value, $A_{U,\boldsymbol{Z}_d}(f)$, is not 1. When $A_{U,\boldsymbol{Z}_d}(f) = 1$, the verifier essentially makes the same test as it would have if ωf would have taken the place of f. Since $A_{U,\boldsymbol{Z}_d}(\omega f) \neq 1$, the verifier may thus accept for only

one value of $A_{W,\boldsymbol{Z}_d}(\cdot)A_{W,\boldsymbol{Z}_d}(\cdot)$, as in the case $A_{U,\boldsymbol{Z}_d}(f) \neq 1$. Due to folding, $A_{U,\boldsymbol{Z}_d}(\omega f) = \omega A_{U,\boldsymbol{Z}_d}(f)$, so the test can be performed without reading any extra values. Again we have perfect completeness:

Lemma 10.14. If ϕ is satisfiable there is a proof which AdaptiveTest accepts with probability 1.

Turning to the soundness, we note that the acceptance probability of the verifier can be written as $\mathrm{E}[I_1 + I_2]$ where

$$I_1 = \left(1 - \frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)\Big)^\gamma\right) \times \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^\gamma\right)$$

and

$$\begin{aligned} I_2 &= \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(A_{U,\boldsymbol{Z}_d}(f)\Big)^\gamma\right) \times \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(\omega A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(\omega f\circ\pi^{U,W})^{-1}e_{\omega f})\Big)^\gamma\right) \\ &= \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(\omega^{-1}A_{U,\boldsymbol{Z}_d}(\omega f)\Big)^\gamma\right) \times \left(\frac{1}{d}\sum_{\gamma=0}^{d-1}\Big(\omega A_{W,\boldsymbol{Z}_d}(h^{-1})A_{W,\boldsymbol{Z}_d}(h^{-1}(\omega f\circ\pi^{U,W})^{-1}e_{\omega f})\Big)^\gamma\right), \end{aligned}$$

where the second equality follows since $A_{U,\boldsymbol{Z}_d}(\omega^{-1}f) = \omega^{-1}A_{U,\boldsymbol{Z}_d}(f)$. We have that

$$\mathrm{E}[I_1] = d^{-1} - d^{-2} - \mathrm{E}[L] - \mathrm{E}[Q] + \mathrm{E}[C_1] - \mathrm{E}[C_2]$$

where L, Q, C_1 and C_2 are defined as in (10.1)–(10.4). We use that, since f is selected uniformly at random in the protocol,

$$\mathop{\mathrm{E}}_{f}[\omega^{-\gamma}(A_{U,\boldsymbol{Z}_d}(\omega f))^\gamma] = \mathop{\mathrm{E}}_{f}[\omega^{-\gamma}(A_{U,\boldsymbol{Z}_d}(f))^\gamma],$$

and

$$\mathop{\mathrm{E}}_{f,e_f}\Big[\Big(A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^\gamma\Big] = \mathop{\mathrm{E}}_{f,e_{\omega f}}\Big[\Big(A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(\omega f\circ\pi^{U,W})^{-1}e_{\omega f})\Big)^\gamma\Big].$$

to write

$$\mathrm{E}[I_2] = d^{-2} + \mathrm{E}[L'] + \mathrm{E}[Q'] + \mathrm{E}[C']$$

where

$$\begin{aligned} L' &= \frac{1}{d}\sum_{\gamma=1}^{d-1} \omega^{-\gamma}(A_{U,\boldsymbol{Z}_d}(f))^\gamma, \\ Q' &= \frac{1}{d^2}\sum_{\gamma=1}^{d-1} \omega^{\gamma}\Big(A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^\gamma, \\ C' &= \frac{1}{d^2}\sum_{\gamma_1=1}^{d-1}\sum_{\gamma_2=1}^{d-1} \\ &\qquad \omega^{\gamma_2-\gamma_1}\Big(A_{U,\boldsymbol{Z}_d}(f)\Big)^{\gamma_1}\Big(A_{W,\boldsymbol{Z}_d}(h)A_{W,\boldsymbol{Z}_d}(h^{-1}(f\circ\pi^{U,W})^{-1}e_f)\Big)^{\gamma_2}. \end{aligned}$$

To sum up, $\mathrm{E}[I_1 + I_2] = d^{-1} - \mathrm{E}[L] - \mathrm{E}[Q] + \mathrm{E}[C_1] - \mathrm{E}[C_2] + \mathrm{E}[L'] + \mathrm{E}[Q'] + \mathrm{E}[C']$, and from the previous section, we already know how to bound all these terms.

Lemma 10.15. Let $\delta \in (0,1)$ be arbitrary and suppose that $T \geq \delta^{-3}d^3 \ln \delta^{-1}$ and $u > (\log T)/\log c_\mu^{-1}$. Then $\mathrm{E}[L'] = 0$, $|\mathrm{E}[Q']|^2 \leq \delta$, and $|\mathrm{E}[C']|^2 \leq 8\delta$

Proof. $E[L'] = 0$ since the tables in the proof are folded. Since

$$\mathrm{E}[Q'] = \mathop{\mathrm{E}}_{U,W}\left[\frac{1}{d^2}\sum_{\gamma=1}^{d-1}\omega^\gamma Q_\gamma\right]$$

and

$$\mathrm{E}[C'] = \mathop{\mathrm{E}}_{U,W}\left[\frac{1}{d^2}\sum_{\gamma_1=1}^{d-1}\sum_{\gamma_2=1}^{d-1}\omega^{\gamma_2-\gamma_1}C_{\gamma_1,\gamma_2}\right],$$

the two inequalities follow from Lemmas 10.4 and 10.10 in the same way as Corollaries 10.5 and 10.11. ∎

To prove that the adaptive protocol has soundness $d^{-1} + \epsilon$ for any constant $\epsilon > 0$ is now straightforward.

Lemma 10.16. For any integer $d \geq 2$ and any constant $\epsilon > 0$, there are choices of the parameters δ, T and u such that, if ϕ is at most μ-satisfiable, AdaptiveTest with these parameter choices accepts with probability most $d^{-1} + \epsilon$.

Proof. Given ϵ, first select $\delta \leq \epsilon^2/131$ and then select $T \geq \delta^{-3} d^3 \ln \delta^{-1}$ and $u > (\log T + T \log d)/\log c_\mu^{-1}$. Now suppose that the verifier in Fig. 10.1 with these parameter choices accepts an incorrect input with probability $d^{-1} + \epsilon$. Then

$$\begin{aligned} \epsilon &= -\operatorname{E}[L] - \operatorname{E}[Q] + \operatorname{E}[C_1] - \operatorname{E}[C_2] + \operatorname{E}[L'] + \operatorname{E}[Q'] + \operatorname{E}[C'] \\ &= -\operatorname{E}[Q] + \operatorname{E}[C_1] - \operatorname{E}[C_2] + \operatorname{E}[Q'] + \operatorname{E}[C'], \end{aligned}$$

where L, Q, C_1, and C_2 are defined as in (10.1)–(10.4), Q' and C' are defined above, and the last equality follows since $\operatorname{E}[L] = \operatorname{E}[L'] = 0$ by Lemma 10.2. Hence Jensen's inequality implies that

$$\epsilon^2 \leq 5|\operatorname{E}[Q]|^2 + 5|\operatorname{E}[C_1]|^2 + 5|\operatorname{E}[C_2]|^2 + 5|\operatorname{E}[Q']|^2 + 5|\operatorname{E}[C']|^2.$$

Since $|\operatorname{E}[Q]|^2 \leq \delta$ by Corollary 10.5, $|\operatorname{E}[C_1]|^2 + |\operatorname{E}[C_2]|^2 \leq 16\delta$ and $|\operatorname{E}[Q']|^2 + |\operatorname{E}[C']|^2 \leq 9\delta$ by Lemma 10.15, we obtain a contradiction. ∎

Combing Lemma 10.14 and Lemma 10.16, we get

Theorem 10.17. For any $d \geq 2$, for any constant $\epsilon > 0$,

$$\mathbf{NP} = \mathbf{PCP}_{1,d^{-1}+\epsilon}[O(\log n), 3, d].$$

10.4 Conclusions and open problems

We have established that there exists, for every constant $\epsilon > 0$, a non-adaptive PCP for **NP** that reads three values from a domain of size d, has perfect completeness and soundness $d^{-1} + d^{-2} + \epsilon$. Moreover, we have presented an adaptive version of that PCP where the soundness is improved to $d^{-1} + \epsilon$. As mentioned in the introduction, for $d > 2$ it is currently an open question whether these results are optimal. However, improvements would probably require new techniques for constructing PCPs. We believe that a search for better approximation algorithms for 3-ary constraint satisfaction over domains of size d could be very fruitful. In particular, an algorithm with approximation ratio better than $d^{-1} + d^{-2}$ for satisfiable instances would prove that adaptive verifiers are strictly more powerful than non-adaptive ones also for non-Boolean PCPs.

Chapter 11

Equations over finite groups

11.1 Introduction

Many fundamental computational problems can be naturally posed as questions concerning the simultaneous solvability of families of equations over finite groups. This connection has been exploited to achieve a variety of strong inapproximability results for problems such as Maximum Cut, Maximum Di-Cut, Exact Satisfiability, and Vertex Cover [10, 11, 13, 24, 28, 38, 45, 54]. A chief technical ingredient in these hardness results is a tight lower bound on the approximability of the problem of simultaneously satisfying equations over a finite *Abelian* group; in this chapter we extend these results to cover all finite groups. Recall that an *equation* in variables $x_1, \ldots, x_n$ over a group G is an expression of form $w_1 \ldots w_k = 1_G$, where each w_i is either a variable, an inverted variable, or a group constant and 1_G denotes the identity element. A collection of equations $\mathcal{E}$ over the same variables induces a natural optimization problem, the problem of determining the maximum number of simultaneously satisfiable equations in $\mathcal{E}$. Recall that by definitions 2.11–2.13, EQ_G denotes this optimization problem, and that the special case where a variable may only appear *once* in each equation is denoted EQ^1_G; furthermore when each equation has single occurrences of exactly k variables, the problem is denoted $\mathrm{EQ}^1_G[k]$. Our main theorem of this chapter asserts that for any finite group G it is **NP**-hard to approximate $\mathrm{EQ}^1_G[3]$ (and hence EQ^1_G and EQ_G) to within $|G| - \epsilon$ for any $\epsilon > 0$; this is tight.

If G is Abelian and $\mathcal{E}$ is a collection of equations over G, each of which can individually be satisfied, the trivial randomized approximation algorithm which independently assigns to each variable a uniformly selected value in G satisfies an expected fraction $|G|^{-1}$ of the equations. This algorithm can be efficiently derandomized by the method of conditional expectation and it in fact also applies to EQ^1_G for any finite group G.

In Chapter 4, we reviewed Håstad's proof that, for $G = \boldsymbol{Z}_2$, if $\mathbf{P} \neq \mathbf{NP}$, then

no polynomial time approximation algorithm can approximate $\mathrm{EQ}^1_G[3]$ to within $|G| - \epsilon$ for any $\epsilon > 0$. Håstad [28] also proves that the same bound holds for any finite *abelian* group.

The main theorem of this chapter shows that this same lower bound holds for all finite groups.

To adapt the methods to equations over arbitrary finite groups we need to overcome a couple of technical difficulties. These are surveyed briefly in Section 11.1.1, below.

11.1.1 Technical contributions

Our main technical contributions are from the part of the analysis where we establish the connection between the proof system that tests a group equation and the 2P1R game for E3-Sat. The first step in this analysis is to "arithmetize" the acceptance probability of the former protocol. In the case of $\boldsymbol{Z}_m$, this is fairly straightforward: The acceptance probability can be written as a sum of m terms. If the acceptance probability is large, there has to be at least one large term in the sum. We then proceeded by expanding this allegedly large term in its Fourier expansion and then uses the Fourier coefficients to devise a strategy for the provers in the 2P1R game.

For non-Abelian groups, the way to arithmetize the test turns out to require a sum of the traces of products of certain matrices given by the representation theory of the group in question. As in the case of $\boldsymbol{Z}_m$, we find that if the acceptance probability of the linear test is large, there has to be one product of matrices with a large trace. Our next step is to expand this matrix product in its Fourier series. Unfortunately, the Fourier expansion of each entry in those matrices contains matrices that could be very large; consequently, the Fourier expansion of the entire trace contains a product of matrices with potentially huge dimension. Thus, the fact that this trace is large does not necessarily mean that the individual entries in the matrices are large and directly using the entries in the matrices to construct the probabilistic strategy for the provers in the 2P1R game does not appear to work. Instead, and this is our first main technical contribution, we prove that the terms in the Fourier expansion corresponding to matrices with large dimension cannot contribute much to the value of the trace. Having done that, we know that the terms corresponding to matrices with reasonably small dimension actually sum up to a significantly large value and we use those terms to construct a strategy for the provers in the 2P1R game; this is our second main technical contribution.

11.2 Representation theory and the Fourier transform

In this section, we give a short account of the representation theory needed to state and prove our results. For more details, we refer the reader to the excellent accounts

by Serre [47] and Terras [49].

The traditional Fourier transform focuses on decomposing functions $f: G \to \boldsymbol{C}$ defined over an Abelian group G. This "decomposition" proceeds by writing f as a linear combination of characters of the group G. Unfortunately, this same procedure cannot work over a non-Abelian group since in this case there are not enough characters to span the space of all functions from G into $\boldsymbol{C}$; the theory of group representations fills this gap, being the natural framework for Fourier analysis over non-Abelian groups and shall be the primary tool utilized in the analysis the "non-Abelian PCP".

Group representation theory studies realizations of groups as collections of matrices: specifically, a *representation* of a group G associates a matrix with each group element so that the group multiplication operation corresponds to normal matrix multiplication. Such an association gives an embedding of the group into $\mathrm{GL}(V)$, the group of invertible linear operators on a finite dimensional $\boldsymbol{C}$-vector space V. (Note that if V is one-dimensional, then this is exactly the familiar notion of character used in the Fourier analysis over Abelian groups.)

Definition 11.1. Let G be a finite group and let V be a finite dimensional vector space over $\boldsymbol{C}$. A *representation* of G is a homomorphism $\gamma: G \to \mathrm{GL}(V)$; the dimension of V is denoted by d_γ and called the *dimension of the representation*.

Two representations are immediate: the *trivial representation* has dimension 1 and maps everything to 1. The permutation action of a group on itself gives rise to the *left regular representation*. Concretely, let V be a $|G|$-dimensional vector space with an orthogonal basis $B = \{e_g : g \in G\}$ indexed by elements of G. Then the *left regular representation* $\mathrm{reg}: G \to \mathrm{GL}(V)$ is given by $\mathrm{reg}(g): e_h \mapsto e_{gh}$; the matrix associated with $\mathrm{reg}(g)$ is simply the permutation matrix given by mapping each element h of G to gh.

If γ is a representation, then for each group element g, $\gamma(g)$ is a linear operator and, as mentioned above, can be identified with a matrix. We denote by $(\gamma(g)_{ij})$ the matrix corresponding to $\gamma(g)$. Two representations γ and θ of G are *isomorphic* if they have the same dimension and there is a change of basis U so that $U\gamma(g)U^{-1} = \theta(g)$ for all g. A representation non-isomorphic to the trivial representation is said to be *nontrivial*.

If $\gamma: G \to \mathrm{GL}(V)$ is a representation and $W \subseteq V$ is a subspace of V, we say that W is *invariant* if $\gamma(g)(W) \subseteq W$ for all g. If the only invariant subspaces are $\{0\}$ and V, we say that γ is *irreducible*. Otherwise, γ does have a nontrivial invariant subspace W_0 and notice that by restricting each $\gamma(g)$ to W_0 we obtain a new representation. When this happens, it turns out that there is always another invariant subspace W_1 so that $V = W_0 \oplus W_1$ and in this case we write $\gamma = \gamma_0 \oplus \gamma_1$, where γ_0 and γ_1 are the representations obtained by restricting to W_0 and W_1. This is equivalent to the existence of a basis in which the $\gamma(g)$ are all block diagonal, where the matrix of $\gamma(g)$ consists of $\gamma_0(g)$ on the first block and $\gamma_1(g)$ on the second block. In this way, any representation can be decomposed into a sum of irreducible representations.

Recall that a matrix A is *unitary* if $A^*A = I$, where A^* is the transpose conjugate of A, i.e., $A^* = \overline{A^t}$. If $\gamma(g)$ is unitary for all g, we say that γ is a unitary representation. It turns out that any irreducible representation is isomorphic to a unitary representation.

The matrix entries of unitary irreducible representations of a finite group G are orthogonal with respect to the inner product

$$\langle f_1 \mid f_2 \rangle_G = \frac{1}{|G|} \sum_{g \in G} f_1(g)\overline{f_2(g)} \tag{11.1}$$

for functions from G to $\boldsymbol{C}$ (See, for example Terras [49, Chapter 15, Theorem 1]).

Proposition 11.2. Let γ and θ be two unitary non-isomorphic irreducible representations of G. Suppose that they are represented by the matrices (γ_{ij}) and $(\theta_{k\ell})$, respectively. Then $\langle \gamma_{ij} \mid \theta_{k\ell} \rangle_G = 0$ for all i, j, k, l and $d_\gamma \langle \gamma_{ij} \mid \gamma_{k\ell} \rangle_G = \delta_{ik}\delta_{j\ell}$.

In particular, any nontrivial representation is orthogonal to the trivial representation:

Corollary 11.3. Let $\gamma\colon G \to \mathrm{GL}(V)$ be a nontrivial irreducible representation of G. Then $\sum_{g \in G} \gamma(g) = 0$.

For a finite group G, there are only a finite number of irreducible representations up to isomorphism; we let $\hat{G}$ denote the set of distinct unitary irreducible representations of G.

There is also a natural product of representations, the *tensor product*. We define the tensor product $A \otimes B$ of two matrices $A = (a_{ij})$ and $B = (b_{k\ell})$ to be the matrix indexed by pairs $(i,k);(j,\ell)$ so that $(A \otimes B)_{(i,k);(j,\ell)} = a_{ij}b_{k\ell}$. Note that $(A_1 \otimes A_2)(B_1 \otimes B_2) = (A_1B_1) \otimes (A_2B_2)$ and that the tensor product is bilinear. We will use the so called *inner trace* of a tensor product: For a matrix M indexed by pairs $(i,k);(j,\ell)$ the inner trace, denoted by $\operatorname{Tr} M$, is defined by $(\operatorname{Tr} M)_{ij} = \sum_k M_{(i,k),(j,k)}$. We let tr denote the normal trace. The inner trace is the "opposite" of the tensor product in the sense that $\operatorname{Tr}(A \otimes B) = (\operatorname{tr} B)A$. If γ and θ are representations of G and H, respectively, we define $\gamma \otimes \theta$ to be the representation of $G \times H$ given by $(\gamma \otimes \theta)(g,h) = \gamma(g) \otimes \theta(h)$.

Proposition 11.4. Let G and H be finite groups. Then the irreducible representations of $G \times H$ are precisely $\{\gamma \otimes \theta \mid \gamma \in \hat{G}, \theta \in \hat{H}\}$. Furthermore, any two of these representations are non-isomorphic.

For a representation γ, the function $g \mapsto \operatorname{tr} \gamma(g)$ is called the *character corresponding to* γ and is denoted by χ_γ. Note that χ_γ takes values in $\boldsymbol{C}$ even if γ has dimension larger than one. Our principal use of the character comes from the following fact:

Proposition 11.5. Let g be an element of G. Then $\sum_{\gamma \in \hat{G}} d_\gamma \chi_\gamma(g) = |G|$ if $g = 1_G$ and $\sum_{\gamma \in \hat{G}} d_\gamma \chi_\gamma(g) = 0$ otherwise.

Since $\gamma(1_G)$ is always an identity matrix, $\chi_\gamma(1_G) = d_\gamma$:

Corollary 11.6. $\sum_{\gamma \in \hat{G}} d_\gamma^2 = |G|$.

11.2.1 The Fourier transform

We now proceed to describe the Fourier transform of functions from an arbitrary finite group G to $\boldsymbol{C}$. Let f be a function from G to $\boldsymbol{C}$ and γ be an irreducible representation of G. Then

$$\hat{f}_\gamma = \frac{1}{|G|} \sum_{g \in G} f(g)\gamma(g) \tag{11.2}$$

is the *Fourier coefficient of f at γ*. Moreover, f can be written as a Fourier series

$$f(g) = \sum_{\gamma \in \hat{G}} d_\gamma \operatorname{tr}\bigl(\hat{f}_\gamma \gamma(g^{-1})\bigr) \tag{11.3}$$

In our analysis, we need the following version of Plancherel's equality:

Lemma 11.7. Suppose that f is a function from G to $\boldsymbol{C}$. Then

$$\sum_{\gamma \in \hat{G}} \sum_{1 \le i \le d_\gamma} \sum_{1 \le j \le d_\gamma} d_\gamma \bigl|\langle f \mid \gamma_{ij} \rangle_G\bigr|^2 = \frac{1}{|G|} \sum_{g \in G} |f(g)|^2 \tag{11.4}$$

if the representations $\gamma \in \hat{G}$ are represented in unitary bases.

Proof. We expand the expression above using the definition of $\langle \cdot \mid \cdot \rangle_G$:

$$\bigl|\langle f \mid \gamma_{ij} \rangle_G\bigr|^2 = \frac{1}{|G|^2} \sum_{g \in G} \sum_{h \in G} f(g) \bigl(\gamma_{ij}(g)\bigr)^* f^*(h) \gamma_{ij}(h)$$

Since γ is a unitary representation, $\gamma(h^{-1}) = \gamma^{-1}(h) = \gamma^*(h)$, and hence $\gamma_{ji}(h^{-1}) = \bigl(\gamma_{ij}(h)\bigr)^*$. Therefore,

$$\begin{aligned}
&\sum_{\gamma \in \hat{G}} \sum_{1 \le i \le d_\gamma} \sum_{1 \le j \le d_\gamma} d_\gamma \bigl|\langle f \mid \gamma_{ij} \rangle_G\bigr|^2 \\
&= \frac{1}{|G|^2} \sum_{\gamma \in \hat{G}} d_\gamma \sum_{1 \le i \le d_\gamma} \sum_{1 \le j \le d_\gamma} \sum_{g \in G} \sum_{h \in G} f(g) f^*(h) \gamma_{ji}(g^{-1}) \gamma_{ij}(h) \\
&= \frac{1}{|G|^2} \sum_{g \in G} \sum_{h \in G} f(g) f^*(h) \sum_{\gamma \in \hat{G}} d_\gamma \operatorname{tr}\bigl(\gamma(g^{-1})\gamma(h)\bigr) \\
&\qquad = \frac{1}{|G|^2} \sum_{g \in G} \sum_{h \in G} f(g) f^*(h) \sum_{\gamma \in \hat{G}} d_\gamma \operatorname{tr}\bigl(\gamma(g^{-1}h)\bigr) = \frac{1}{|G|} \sum_{g \in G} |f(g)|^2,
\end{aligned}$$

where the last equality follows from Proposition 11.5. (See also Serre's account [47, §6.2, Exercise 6.2], which discusses this in different language.) ∎

11.2.2 The Fourier transform of matrix-valued functions

We also need to use the Fourier transform on functions $f\colon G \to \mathrm{End}(V)$, where $\mathrm{End}(V)$ is the set of linear maps from the vector space V to itself; we here identify $\mathrm{End}(V)$ with the space of all $\dim V \times \dim V$ matrices over $\boldsymbol{C}$. Although we have not found any treatment of such transforms in the literature, it is straightforward to generalize the concepts from the previous section to matrix-valued functions. For a representation γ of G, we define

$$\hat{f}_\gamma = \frac{1}{|G|} \sum_{g \in G} f(g) \otimes \gamma(g). \tag{11.5}$$

Treating the $f(g)$ as matrices, this is nothing more than the component-wise Fourier transform of the function f. The reason for grouping them together into these tensor products is the following: Let $f, h\colon G \to \mathrm{End}(V)$ be two such functions, and define their convolution as

$$(f * h)(g) = \frac{1}{|G|} \sum_{t \in G} f(t) h(t^{-1}g),$$

this product being the ring product in $\mathrm{End}(V)$ (that is, function composition). Then it turns out that $\widehat{(f * h)}_\gamma = \hat{f}_\gamma \hat{h}_\gamma$, this product being matrix multiplication:

$$\begin{aligned}
\widehat{(f * h)}_\gamma &= \frac{1}{|G|} \sum_{g \in G} (f * h)(g) \otimes \gamma(g) \\
&= \frac{1}{|G|^2} \sum_{g \in G} \sum_{t \in G} \big(f(t) h(t^{-1}g)\big) \otimes \gamma(tt^{-1}g) \\
&= \frac{1}{|G|^2} \sum_{g \in G} \sum_{t \in G} \big(f(t) \otimes \gamma(t)\big)\big(h(t^{-1}g) \otimes \gamma(t^{-1}g)\big) \\
&= \hat{f}_\gamma \hat{h}_\gamma.
\end{aligned}$$

In this case, the Fourier series is

$$f(g) = \sum_{\gamma \in \hat{G}} d_\gamma \operatorname{Tr}\Big(\hat{f}_\gamma \big(\boldsymbol{I} \otimes \gamma(g^{-1})\big)\Big) \tag{11.6}$$

where $\operatorname{Tr} M$ is the inner trace. This also gives rise to a Plancherel equality: for two functions $f, h\colon G \to \mathrm{End}(V)$,

$$\frac{1}{|G|} \sum_{g \in G} f(g) h(g^{-1}) = (f * h)(1_G) = \sum_{\gamma \in \hat{G}} d_\gamma \operatorname{Tr}(\hat{f}_\gamma \hat{h}_\gamma). \tag{11.7}$$

As we noted above, the representations of a finite group can always be expressed in a unitary basis. Recall that a matrix A is *Hermitian* if $A^* = A$. When a function

from a finite group G to $\mathrm{End}(V)$ behaves like a unitary matrix in a certain sense and the representations of G are expressed in a unitary basis, the Fourier coefficients are Hermitian. This turns out to be important in our analysis.

Definition 11.8. Let G be a finite group, V be a finite-dimensional vector space, and f be a function from G to $\mathrm{End}(V)$. Then f is *skew-symmetric* if $f(g^{-1}) = f^*(g)$ for all $g \in G$.

Lemma 11.9. Let G be a finite group, V be a finite-dimensional vector space, and f be a skew-symmetric function from G to $\mathrm{End}(V)$. Then $\hat{f}_\gamma$ is Hermitian if γ is expressed in a unitary basis.

Proof. By equation (11.5),

$$\begin{aligned}\hat{f}_\gamma = \frac{1}{|G|}\sum_{g\in G} f(g)\otimes\gamma(g) &= \frac{1}{2|G|}\sum_{g\in G}\Big(f(g)\otimes\gamma(g) + f(g^{-1})\otimes\gamma(g^{-1})\Big)\\ &= \frac{1}{2|G|}\sum_{g\in G}\Big(f(g)\otimes\gamma(g) + f^*(g)\otimes\gamma^*(g)\Big),\end{aligned}$$

where the last equality follows since f is skew-symmetric and γ is expressed in a unitary basis. Now, $f(g)\otimes\gamma(g) + f^*(g)\otimes\gamma^*(g)$ is clearly Hermitian, and since a sum of Hermitian matrices is Hermitian, $\hat{f}_\gamma$ is Hermitian. ∎

11.3 The non-Abelian long code

Let K be a finite set and denote by G^K the set of all functions from K to G. The *long G-code* of some $x \in K$ is the function $\mathrm{LC}_G(x)$ from G^K to G such that $\mathrm{LC}_G(x)(f) = f(x)$.

In the analysis of the soundness of the verifier, we study the Fourier transform of such purported long codes composed with a representation of G, i.e., the Fourier transform of functions from G^K to $\mathrm{End}(V)$, where V is the underlying vector space of a representation ρ.

11.3.1 Folding

We first note that the concept of *folding* which we first introduced in Chapter 4 and which we generalized to long codes over $\boldsymbol{Z}_m$ in Chapter 5 further extends to the long G-code.

Definition 11.10. Let G be a finite group, $\gamma \in \hat{G}$ be arbitrary, V be the space corresponding to γ, K be a finite set, and A be a function from G^K to $\mathrm{End}(V)$. Then A is *γ-homogeneous* if $A(gf) = \gamma(g)A(f)$ for all $g \in G$ and all $f \in G^K$.

Lemma 11.11. Let G be a finite group, γ be an arbitrary nontrivial representation of G, V be the space corresponding to γ, K be a finite set, and A be a

γ-homogeneous function from G^K to End(V). Then $\hat{A}_\rho = 0$ when ρ is the trivial representation of G^K.

Proof. Since $\rho(f) = 1$ for all f when ρ is the trivial representation, (11.5) immediately yields

$$\hat{A}_\rho = \frac{1}{|G|^{|K|}} \sum_{f \in G^K} A(f) = \frac{1}{|G|^{|K|+1}} \sum_{g \in G} \sum_{f \in G^K} A(gf)$$
$$= \frac{1}{|G|^{|K|+1}} \sum_{f \in G^K} \left(\sum_{g \in G} \gamma(g) \right) A(f) = 0,$$

where the last equality follows from Corollary 11.3. ∎

By employing a certain access convention in the verifier, we can ensure that tables correspond to γ-homogeneous functions.

Definition 11.12. Let G be a finite group, K be a finite set, and A be a function from G^K to G. Partition G^K into equivalence classes by the relation $\equiv$, where $f \equiv h$ if there is $g \in G$ such that $f = gh$. Write $[f]$ for the equivalence class of f. Define A_G, *A left-folded over G* by choosing a representative for each equivalence class and defining $A_G(h) = gA(f)$, if $h = gf$ and f is the chosen representative for $[h]$.

Lemma 11.13. Let G be a finite group, K be a finite set, and A be a function from G^K to G. Then $\gamma \circ A_G$ is γ-homogeneous for every $\gamma \in \hat{G}$.

Proof. Note that $A_G(gf) = gA_G(f)$ for all $g \in G$ and all $f \in G^K$. Hence $(\gamma \circ A_G)(gf) = \gamma(g)(\gamma \circ A_G)(f)$. ∎

It turns out that our analysis only requires some of the tables in the proof to be folded, while the other tables must be skew-symmetric functions as per Definition 11.8. Again, this can be accomplished by proper access conventions in the verifier. In this case, we achieve our goal by, when accessing $A(f)$, with probability $1/2$ using the value $A(f^{-1})^{-1}$ instead.

Lemma 11.14. Let G be a finite group, K be a finite set, and A be a function from G^K to G. Let $\gamma \in \hat{G}$ be unitary, and $B(f) = \mathrm{E}_{b \in \{-1,1\}}[(\gamma \circ A)(f^b)^b]$ Then B is skew-symmetric.

Proof. We have that

$$B(f) = \mathop{\mathrm{E}}_{b \in \{-1,1\}}[(\gamma \circ A)(f^b)^b]$$
$$= \frac{1}{2}(\gamma \circ A)(f) + \frac{1}{2}(\gamma \circ A)(f^{-1})^{-1}$$
$$= \frac{1}{2}(\gamma \circ A)(f) + \frac{1}{2}(\gamma \circ A)(f^{-1})^*$$

where the last equality holds since γ is unitary. Hence $B(f^{-1}) = \frac{1}{2}(\gamma \circ A)(f^{-1}) + \frac{1}{2}(\gamma \circ A)(f)^* = B(f)^*$, and B is skew-symmetric by Definition 11.8. ∎

We remark, that if A is a long G-code, A_G is identical to A, and $A(f^{-1})^{-1} = A(f)$.

11.3.2 Projection

To state the final long G-code property that we need, we have to develop a more precise and detailed description of the Fourier transform. Since we can represent a function $f\colon K \to G$ by a table containing $f(x)$ for every $x \in K$, we can identify G^K with $G^{|K|}$. In order to reason about the Fourier transform of a function from G^K to $\mathrm{End}(V)$, we need an understanding of the irreducible representations of powers of G. It follows from Proposition 11.4 that the irreducible representations of G^K are precisely those representations obtained by taking tensor products of $|K|$ irreducible representations of G: when $\rho_x \in \hat{G}$ for each $x \in K$ this is the representation given by

$$\rho = \bigotimes_{x\in K} \rho_x \qquad \text{where} \qquad \rho(f) = \bigotimes_{x\in K} \rho_x(f(x)).$$

We treat the tensor product of two matrices as a matrix indexed by pairs. Analogously, we treat the tensor product of $|K|$ matrices as a matrix indexed by $|K|$-tuples. In order to reason about single entries in the tensor product that forms a representation $\rho = \bigotimes_{x\in K} \rho_x$ we define the *set of indices* $\iota(\rho)$. An element $i \in \iota(\rho)$ is a vector indexed by elements of K so that for all $x \in K, 1 \le i_x \le d_{\rho_x}$; we refer to such an element i as an *index*. Then for two indices $i, j \in \iota(\rho)$ we define

$$\rho_{ij}(f) = \prod_{x\in K} \Big(\rho_x\big(f(x)\big)\Big)_{i_x,j_x}.$$

We also define the *weight* $|\rho|$ of an irreducible representation ρ of G^K to be the number of $x \in K$ such that ρ_x is nontrivial.

The verifier in our PCP checks positions in tables corresponding to two related long codes. The precise details of how these tables are related is described below; for now it is enough to know that the tables correspond to functions from $F = G^K$ to G and from $H = G^L$ to G, respectively, where there is an onto function $\pi\colon L \to K$. There is a natural way to transform an $f \in F$ into a function in H by composing it with π; we denote this function by $f \circ \pi$. The projection π can also be used to transform the components of a representation $\rho \in \hat{H}$ into a function on F. For $i, j \in \iota(\rho)$ the components ρ_{ij} are functions from H to $\boldsymbol{C}$. We denote the new, associated, function by $\rho^\pi_{ij}\colon F \to \boldsymbol{C}$; it is defined by the map $f \mapsto \rho_{ij}(f \circ \pi)$. Using our definition of the index sets,

$$\rho^\pi_{ij}(f) = \rho_{ij}(f \circ \pi) = \prod_{x\in K} \prod_{y\in\pi^{-1}(x)} \Big(\rho_y\big(f(x)\big)\Big)_{i_y,j_y}.$$

We are now ready to formulate the following projection lemma:

Lemma 11.15. Let K and L be finite sets and $\pi\colon L \to K$ be an onto function. Let $F = G^K$ and $H = G^L$. Define the relation $\sim$ on $\hat{F} \times \hat{H}$ so that for $\tau \in \hat{F}$ and $\rho \in \hat{H}$, $\tau \sim \rho$ if for all $x \in K$ such that τ_x is nontrivial, there is some $y \in \pi^{-1}(x)$ such that ρ_y is nontrivial. Then

1. $\tau \sim \rho \implies |\tau| \leq |\rho|$.
2. $\tau \not\sim \rho \implies \forall i,j \in \iota(\rho), \forall k,\ell \in \iota(\tau), \big(\langle \rho^\pi_{ij} \mid \tau_{k\ell}\rangle_F = 0\big)$.

Proof. The first implication follows directly from the definition of the relation $\sim$. To prove the second implication, assume that $\tau \not\sim \rho$; then there is some $x' \in K$ such that $\tau_{x'}$ is nontrivial but ρ_y is trivial for all $y \in \pi^{-1}(x')$. Recall that we can write

$$\rho^\pi_{ij}(f) = \prod_{x\in K} \prod_{y\in\pi^{-1}(x)} \Big(\rho_y\big(f(x)\big)\Big)_{i_y,j_y} \qquad \text{and} \qquad \tau_{k\ell}(f) = \prod_{x\in K} \Big(\tau_x\big(f(x)\big)\Big)_{k_x,\ell_x}$$

by our definition of the index sets; hence

$$\begin{aligned}
\langle \rho^\pi_{ij} \mid \tau_{k\ell}\rangle_F &= \frac{1}{|F|} \sum_{f\in F} \Big(\prod_{x\in K} \prod_{y\in\pi^{-1}(x)} \Big(\rho_y\big(f(x)\big)\Big)_{i_y,j_y} \Big) \Big(\prod_{x\in K} \Big(\tau_x\big(f(x)\big)\Big)^*_{k_x,\ell_x} \Big) \\
&= \frac{1}{|F|} \sum_{f\in F} \Big(\prod_{x\in K} \prod_{y\in\pi^{-1}(x)} \Big(\rho_y\big(f(x)\big)\Big)_{i_y,j_y} \Big) \Big(\prod_{x\in K} \Big(\tau_x\big(f(x)\big)\Big)^{-1}_{\ell_x,k_x} \Big) \\
&= \frac{1}{|F|} \sum_{f\in F} \prod_{x\in K} \Big(\tau_x\big(f^{-1}(x)\big)\Big)_{\ell_x,k_x} \prod_{y\in\pi^{-1}(x)} \Big(\rho_y\big(f(x)\big)\Big)_{i_y,j_y} \\
&= \prod_{x\in K} \Big(\frac{1}{|G|} \sum_{g\in G} \big(\tau_x(g^{-1})\big)_{\ell_x,k_x} \prod_{y\in\pi^{-1}(x)} \big(\rho_y(g)\big)_{i_y,j_y} \Big)
\end{aligned}$$

where the last equality holds since a sum over all functions $f \in F$ can be viewed as $|K|$ nested sums over the possible values of $f(x)$ for $x \in K$. We can then change the sums of a product into a product of $|K|$ sums, i.e., a product of sums over all $g \in G$.

Since ρ_y is trivial for all $y \in \pi^{-1}(x')$, the factor corresponding to x' in the above product is

$$\frac{1}{|G|} \sum_{g\in G} \big(\tau_{x'}(g^{-1})\big)_{\ell_{x'},k_{x'}} = 0,$$

where the equality follows from Corollary 11.3. ∎

11.4 Identities from linear and multilinear algebra

This section contains some identities and bounds that are needed in the next section. They all follow in a straightforward manner from standard linear and multilinear algebra.

Lemma 11.16. Let A be a unitary $n \times n$ matrix. Then $\sum_j |a_{ij}|^2 = 1$ for all $1 \le i \le n$.

Corollary 11.17. The elements of a unitary matrix have at most unit magnitude.

Proof of Lemma 11.16. Let A be unitary. Then AA^* is the identity matrix. Since for all $1 \le i \le n$, $(AA^*)_{ii} = \sum_j a_{ij} a_{ij}^* = \sum_j |a_{ij}|^2$, the lemma follows. ∎

The next lemma shows that a convex combination of unitary matrices has elements of at most unit magnitude.

Lemma 11.18. Let A_k be a family of unitary $n \times n$ matrices and let λ_k be non-negative real numbers such that $\sum_k \lambda_k = 1$. Let $B = \sum_k \lambda_k A_k$. Then for any $1 \le i, j \le n$, $|b_{ij}| \le 1$.

Proof.

$$|b_{ij}| = \left| \sum_k \lambda_k (A_k)_{ij} \right| \le \sum_k \lambda_k |(A_k)_{ij}| \le 1$$

where the first inequality follows by Jensen's inequality, and the second follows by Corollary 11.17 and $\sum_k \lambda_k = 1$. ∎

Lemma 11.19. Let $\epsilon \in [0,1]$, S_ρ be a family of positive semidefinite matrices and d_ρ and n_ρ be positive integers. Then

$$\sum_{\rho : n_\rho > c} d_\rho (1-\epsilon)^{n_\rho} \operatorname{tr} S_\rho \le (1-\epsilon)^c \sum_\rho d_\rho \operatorname{tr} S_\rho.$$

Proof. Since the S_ρ is positive semidefinite $\sum_{\rho : n_\rho > c} d_\rho (1-\epsilon)^{n_\rho} \operatorname{tr} S_\rho$ is a sum of non-negative numbers and thus non-negative. Therefore

$$\sum_{\rho : n_\rho > c} d_\rho (1-\epsilon)^{n_\rho} \operatorname{tr} S_\rho \le (1-\epsilon)^c \sum_{\rho : n_\rho > c} d_\rho \operatorname{tr} S_\rho \le (1-\epsilon)^c \sum_\rho d_\rho \operatorname{tr} S_\rho.$$

∎

Lemma 11.20. Let U be an $n \times n$ unitary matrix, $\{S_\rho\}$ be a family of positive semidefinite matrices that are tensor products of $n \times n$ matrices and $k \times k$ matrices, $\{V_\rho\}$ be a family of unitary matrices that are tensor products of $n \times n$ matrices and $k \times k$ matrices, and $\{a_\rho\}$ be a family of non-negative real numbers. Then

$$\left| \operatorname{tr} \left(U \sum_\rho a_\rho \operatorname{Tr}(S_\rho V_\rho) \right) \right| \le \sum_\rho a_\rho \operatorname{tr} S_\rho,$$

where the inner trace is with respect to the tensor products forming S_ρ and V_ρ.

Proof. Since $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for any matrices A and B,

$$\begin{aligned}\left|\operatorname{tr}\Big(U\sum_\rho a_\rho \operatorname{Tr}(S_\rho V_\rho)\Big)\right| &= \left|\operatorname{tr}\Big(\sum_\rho a_\rho \operatorname{Tr}(S_\rho V_\rho)U\Big)\right| \\ &= \left|\operatorname{tr}\Big(\sum_\rho a_\rho \operatorname{Tr}\big(S_\rho V_\rho(U\otimes I_k)\big)\Big)\right|\end{aligned}$$

where the last equality follows from Lemma 11.21 below. Since the a_ρ are non-negative,

$$\begin{aligned}\left|\operatorname{tr}\Big(\sum_\rho a_\rho \operatorname{Tr}\big(S_\rho V_\rho(U\otimes I_k)\big)\Big)\right| &\le \sum_\rho a_\rho \left|\operatorname{tr}\operatorname{Tr}\big(S_\rho V_\rho(U\otimes I_k)\big)\right| \\ &= \sum_\rho a_\rho \left|\operatorname{tr}\big(S_\rho V_\rho(U\otimes I_k)\big)\right| \\ &\le \sum_\rho a_\rho \operatorname{tr} S_\rho,\end{aligned}$$

where the last inequality follows from Lemma 11.22 below. ∎

Lemma 11.21. Let A and C be $n\times n$ matrices and B be a tensor product of an $n\times n$ matrix and a $k\times k$ matrix. Then $A(\operatorname{Tr} B)C = \operatorname{Tr}((A\otimes \boldsymbol{I}_k)B(C\otimes \boldsymbol{I}_k))$.

Proof. Suppose that $B = B_1\otimes B_2$. Using the identity $(A_1\otimes A_2)(B_1\otimes B_2) = (A_1B_1)\otimes(A_2B_2)$, we get that $(A\otimes \boldsymbol{I}_k)B(C\otimes \boldsymbol{I}_k) = (AB_1\otimes B_2)(C\otimes I) = AB_1C\otimes B_2$ and hence $\operatorname{Tr}((A\otimes \boldsymbol{I}_k)B(C\otimes \boldsymbol{I}_k)) = (\operatorname{tr} B_2)AB_1C = A(\operatorname{Tr} B)C$, where we used $\operatorname{Tr}(A\otimes B) = (\operatorname{tr} B)A$. ∎

Lemma 11.22. Let A be a positive semidefinite matrix and U be a unitary matrix. Then $|\operatorname{tr}(AU)| \le \operatorname{tr} A$.

Proof. As A is positive semidefinite, it may be written VDV^* where V is unitary and D is diagonal with non-negative entries on the diagonal. Since the trace is invariant under similarity, $\operatorname{tr} A = \sum_i D_{ii}$ and we may rewrite

$$\operatorname{tr}(AU) = \operatorname{tr}(VDV^*U) = \operatorname{tr}(VDV^*UVV^*) = \operatorname{tr}(DV^*UV) = \operatorname{tr}(DW),$$

where $W = V^*UV$ is a product of unitary matrices and therefore unitary. All entries of a unitary matrix have absolute value less than or equal to one; hence

$$|\operatorname{tr}(AU)| = \left|\sum_i D_{ii}W_{ii}\right| \le \sum_i |D_{ii}W_{ii}| \le \sum_i D_{ii} = \operatorname{tr} A,$$

as desired. ∎

GroupTest.
Input: An E3-CNF(5) formula ϕ. The proof is a Standard Written G-Proof with parameter u.

1. Select uniformly at random a sequence $W = (C_1, \ldots, C_u)$ of u clauses and a sequence U of u variables by for each clause C_k choosing uniformly at random a variable occurring in C_k.
2. Select uniformly at random $f \in G^{\{0,1\}^U}$ and $g \in G^{SAT^W}$
3. Choose $e \in G^{SAT^W}$, such that, independently for each $y \in \mathrm{SAT}^W$,
   (a) With probability $1 - \epsilon$, $e(y) = 1_G$.
   (b) With probability ϵ, $e(y)$ is chosen uniformly at random from G.
4. Choose b_1 and b_2 independently and uniformly at random in $\{-1, 1\}$.
5. If $A_{U,G}(f)A_W(h^{b_1})^{b_1}A_W((h^{-1}(f \circ \pi^{U,W})^{-1}e)^{b_2})^{b_2} = 1_G$ then accept, else reject.

Figure 11.1. The above PCP is parameterized by the positive integer u and the positive real ϵ and tests if a E3-CNF(5) formula ϕ is satisfiable by querying three positions in a Standard Written G-Proof with parameter u.

11.5 The PCP

Definition 11.23. A Standard Written G-Proof with parameter u for a formula ϕ consists of a table $A_U \colon G^{\{0,1\}^U} \to G$ for each sequence U of u variables from ϕ and a table $A_W \colon G^{\mathrm{SAT}^W} \to G$ for each sequence W of u clauses from ϕ.

The protocol itself is similar to that used by Håstad [28] to prove inapproximability of equations over finite Abelian groups, which is in turn a straightforward generalization of the protocol which we reviewed in Chapter 4. The tables corresponding to sequences of variables are left-folded over G. The verifier is given in Figure 11.1.

Lemma 11.24. If ϕ is satisfiable, the verifier in Figure 11.1 accepts a correct proof with probability at least $1 - (1 - |G|^{-1})\epsilon$.

Proof. Let x be the assignment corresponding to a correct Standard Written G-Proof with parameter u for a formula ϕ. Then, by the definition of the long G-code, $A_{U,G}(f) = f(x|_U)$ for all f and $A_W(h) = h(x|_W)$ for all h; hence

$$\begin{aligned} &A_{U,G}(f)A_W(h^{b_1})^{b_1}A_W((h^{-1}(f \circ \pi^{U,W})^{-1}e)^{b_2})^{b_2} \\ &\qquad = f(x|_U)h(x|_W)h^{-1}(x|_W)f^{-1}(x|_U)e(x|_W) = e(x|_W). \end{aligned}$$

Considering how e is selected by the verifier, $e(x|_W) = 1_G$ with probability $1 - (1 - |G|^{-1})\epsilon$ and hence the verifier accepts a correctly encoded proof of a satisfying assignment with probability $1 - (1 - |G|^{-1})\epsilon$. ∎

11.5.1 Analysis of the soundness

We proceed as usual. We assume that the verifier accepts a proof corresponding to an unsatisfiable formula with probability $|G|^{-1} + \delta$ and prove that it is then possible to construct strategies for the provers in the 2P1R game that make the verifier of that game accept with high probability. Since it is known that this cannot be the case, there cannot exist a proof corresponding to an unsatisfiable formula that the PCP verifier accepts with probability $|G|^{-1}+\delta$. To this end, we first apply Proposition 11.5 to arrive at an expression for the acceptance probability. Since

$$|G|^{-1} \sum_{\gamma \in \hat{G}} d_\gamma \chi_\gamma \Big(A_{U,G}(f)(A_W(h^{b_1}))^{b_1} (A_W((h^{-1}(f \circ \pi^{U,W})^{-1} e)^{b_2}))^{b_2} \Big)$$

is an indicator of the event that the verifier accepts, the acceptance probability can be written as:

$$|G|^{-1} \sum_{\gamma \in \hat{G}} d_\gamma \, \mathrm{E}\Big[\chi_\gamma \Big(A_{U,G}(f) A_W(h^{b_1})^{b_1} A_W((h^{-1}(f \circ \pi^{U,W})^{-1} e)^{b_2})^{b_2} \Big) \Big] =$$
$$|G|^{-1} + |G|^{-1} \sum_{\gamma \in \hat{G} \setminus \{\mathbf{1}\}} d_\gamma \, \mathrm{E}\Big[\chi_\gamma \Big(A_{U,G}(f) A_W(h^{b_1})^{b_1} A_W((h^{-1}(f \circ \pi^{U,W})^{-1} e)^{b_2})^{b_2} \Big) \Big]$$

where the expectations are over the choice of U, W, f, h, e, b_1 and b_2. With the aid of Corollary 11.6, we deduce that if the verifier in Figure 11.1 accepts with probability $|G|^{-1}+\delta$, there must be some nontrivial irreducible representation γ of G such that

$$\Big| \mathrm{E}\Big[\chi_\gamma \Big(A_{U,G}(f) A_W(h^{b_1})^{b_1} A_W\big((h^{-1}(f \circ \pi^{U,W})^{-1} e)^{b_2}\big)^{b_2} \Big) \Big] \Big| > d_\gamma \delta. \tag{11.8}$$

We now proceed by applying Fourier-inversion to $\gamma \circ A_{U,G}$ and $\gamma \circ A_W$. More precisely, we first apply Fourier-inversion to $\gamma \circ A_W$, resulting in:

Lemma 11.25. Suppose that the verifier in Figure 11.1 accepts with probability $|G|^{-1} + \delta$. Then there exists a nontrivial representation γ of G such that

$$\Big| \mathop{\mathrm{E}}_{f,U,W} \Big[\mathrm{tr}\Big(A(f) \sum_{\rho \in \hat{H}} d_\rho (1-\epsilon)^{|\rho|} \, \mathrm{Tr}\Big(\hat{B}^2_\rho \big(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \big) \Big) \Big) \Big] \Big| > d_\gamma \delta.$$

where $A = \gamma \circ A_{U,G}$, $H = G^{\mathrm{SAT}^W}$ and $B(h) = \mathrm{E}_{b \in \{-1,1\}}[(\gamma \circ A_W)(h^b)^b]$.

Proof. Since the verifier in Figure 11.1 accepts with probability $|G|^{-1} + \delta$ there exists a nontrivial representation γ of G such that the inequality (11.8) holds; we

now fix that γ and select a basis such that it is unitary. We proceed by expanding the expectation in (11.8) in a Fourier series. We have that the expectation in (11.8) is equal to

$$\operatorname{tr} \mathop{\mathrm{E}}_{f,h,e,U,W}\left[\gamma(A_{U,G}(f)) \mathop{\mathrm{E}}_{b_1}\left[\gamma(A_W(h^{b_1})^{b_1})\right] \mathop{\mathrm{E}}_{b_2}\left[\gamma(A_W\left((h^{-1}(f\circ\pi^{U,W})^{-1}e)^{b_2}\right)^{b_2})\right]\right]$$

To shorten the notation, we introduce the shorthands $A = \gamma \circ A_{U,G}$ and

$$B(h) = \mathop{\mathrm{E}}_{b\in\{-1,1\}}[\gamma(A_W(h^{b_1})^{b_1})] = \mathop{\mathrm{E}}_{b\in\{-1,1\}}[(\gamma\circ A_W)(h^b)^b]$$

where we used that $\gamma(g^{-1}) = \gamma(g)^{-1}$. With these shorthands the expectation above is equal to

$$\operatorname{tr} \mathop{\mathrm{E}}_{f,h,e,U,W}\left[A(f)B(h)B(h^{-1}(f\circ\pi^{U,W})^{-1}e)\right] =$$
$$\operatorname{tr} \mathop{\mathrm{E}}_{f,U,W}\left[A(f) \mathop{\mathrm{E}}_{h,e}\left[B(h)B(h^{-1}(f\circ\pi^{U,W})^{-1}e)\right]\right]$$

We expand the inner expectation in its Fourier series. Since $\mathrm{E}_{h,e}[B(h)B(h^{-1}(f\circ\pi^{U,W})^{-1}e)] = \mathrm{E}_e[(B*B)((f\circ\pi^{U,W})^{-1}e)]$, this is immediate:

$$\mathop{\mathrm{E}}_{e}[(B*B)((f\circ\pi^{U,W})^{-1}e)] = \sum_{\rho\in\hat{H}} d_\rho \operatorname{Tr}\left(\hat{B}_\rho^2\left(\boldsymbol{I}_{d_\gamma}\otimes\left(\rho(f\circ\pi^{U,W}) \mathop{\mathrm{E}}_{e}[\rho(e^{-1})]\right)\right)\right).$$

To compute $\mathrm{E}_e[\rho(e^{-1})]$, note that

$$\mathop{\mathrm{E}}_{e}[\rho(e^{-1})] = \mathop{\mathrm{E}}_{e}\left[\bigotimes_{y\in\mathrm{SAT}^W}\left(\rho_y\big(e(y)^{-1}\big)\right)\right] = \bigotimes_{y\in\mathrm{SAT}^W} \mathop{\mathrm{E}}_{e(y)}\left[\rho_y\big(e(y)^{-1}\big)\right],$$

where the second equality follows since $e(y)$ is selected independently for every y. Now, $\mathrm{E}_{e(y)}[\rho_y(e(y)^{-1})] = \boldsymbol{I}_{d_{\rho_y}}$ if ρ_y is trivial; otherwise

$$\mathop{\mathrm{E}}_{e(y)}\left[\rho_y\big(e(y)^{-1}\big)\right] = (1-\epsilon)\rho_y(1_G) + \epsilon \mathop{\mathrm{E}}_{g\in G}[\rho_y(g)] = (1-\epsilon)\boldsymbol{I}_{d_{\rho_y}},$$

where the last equality follows from Corollary 11.3. Hence $\mathrm{E}_e[\rho(e^{-1})] = (1-\epsilon)^{|\rho|}\boldsymbol{I}_{d_\rho}$; when this is substituted into the above expression, (11.9) becomes

$$\operatorname{tr}\mathrm{E}\left[A(f)B(h)B(h^{-1}(f\circ\pi^{U,W})^{-1}e)\right] =$$
$$\mathop{\mathrm{E}}_{f}\left[\operatorname{tr}\left(A(f)\sum_{\rho\in\hat{H}} d_\rho(1-\epsilon)^{|\rho|}\operatorname{Tr}\left(\hat{B}_\rho^2\left(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi^{U,W})\right)\right)\right)\right]. \quad (11.9)$$

■

We will now see that for any fixed f, the terms in the resulting sum corresponding to ρ with $|\rho| \geq c$ contribute very little.

Lemma 11.26. Let G be a finite group, V be a d_γ-dimensional vector space, $A(f) \in \mathrm{End}(V)$ be unitary, H be a power of G, and $B\colon H \to \mathrm{End}(V)$ be skew-symmetric. Furthermore, let B be such that for all $h \in H$, $B(h)$ is a convex combination of unitary matrices. Then

$$\left|\mathrm{tr}\left(A(f)\sum_{\substack{\rho\in\hat{H}\\|\rho|\geq c}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{Tr}\left(\hat{B}_\rho^2\left(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi^{U,W})\right)\right)\right)\right| \leq d_\gamma(1-\epsilon)^c \quad (11.10)$$

for any positive real ϵ and any positive integer $c > 0$.

Corollary 11.27. Suppose that the verifier in Figure 11.1 accepts with probability $|G|^{-1}+\delta$. Then, for any unitary $\gamma \in \hat{G}$ and any $c > \lceil(\log\delta - 1)/\log(1-\epsilon)\rceil$, where logs are taken base 2,

$$\left|\mathop{\mathrm{E}}_{f,U,W}\left[\mathrm{tr}\left(A(f)\sum_{\substack{\rho\in\hat{H}\\|\rho|\geq c}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{Tr}\left(\hat{B}_\rho^2\left(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi^{U,W})\right)\right)\right)\right]\right| < \frac{d_\gamma\delta}{2}$$

where $A = \gamma\circ A_{U,G}$, $H = G^{\mathrm{SAT}^W}$ and $B(h) = \mathrm{E}_{b\in\{-1,1\}}[((\gamma\circ A_W)(h^b))^b]$

Proof of Lemma 11.26. Note that, since by Lemma 11.14, B is skew-symmetric, $\hat{B}_\rho$ is Hermitian by Lemma 11.9 and thus $\hat{B}_\rho^2$ is a symmetric real matrix. Since the eigenvalues of A^2 are the squares of the eigenvalues of A, $\hat{B}_\rho^2$ has only non-negative real eigenvalues, i.e., $\hat{B}_\rho^2$ is positive semidefinite. Hence, a direct application of Lemma 11.20 bounds the left-hand side of (11.10) from above by

$$\sum_{\substack{\rho\in\hat{H}\\|\rho|\geq c}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{tr}\,\hat{B}_\rho^2.$$

By Lemma 11.19, the above expression can be bounded from above by

$$(1-\epsilon)^c\sum_{\rho\in\hat{H}} d_\rho\,\mathrm{tr}\,\hat{B}_\rho^2 = (1-\epsilon)^c\,\mathrm{tr}\left(\sum_{\rho\in\hat{H}} d_\rho\,\mathrm{Tr}\,\hat{B}_\rho^2\right).$$

By Plancherel's equality (11.7), $\sum_{\rho\in\hat{H}} d_\rho\,\mathrm{Tr}\,\hat{B}_\rho^2 = (B*B)(1_H)$. We have that

$$(B*B)(1_H) = \frac{1}{|H|}\sum_{h\in H} B(h)B(h^{-1}).$$

Since the product of unitary matrices is unitary, if follows that for any h, $B(h)B(h^{-1})$ is a convex combination of unitary matrices. By Lemma 11.18, $|(B(h)B(h^{-1}))_{ij}| \leq$

1. Hence $(B * B)(1_H)$ has entries of absolute value at most 1, and $\operatorname{tr}((B * B)(1_H))$ is at most d_γ. Thus

$$(1-\epsilon)^c \sum_{\rho \in \hat{H}} d_\rho \operatorname{tr} \hat{B}_\rho^2 \leq d_\gamma (1-\epsilon)^c,$$

which completes the proof. ∎

While we bound the terms corresponding to ρ with $|\rho| \geq c$ by a purely algebraic argument, we bound the terms corresponding to ρ with $|\rho| < c$ by using them to devise a strategy for the provers in the 2P1R game. Since this strategy has a success probability that is independent of u, the number of repetitions in the 2P1R game, we can then, in the standard fashion, select u in such a way that also the terms corresponding to ρ with $|\rho| < c$ have to be upper bounded by $d_\gamma \delta/2$.

Lemma 11.28. Suppose that for any nontrivial $\gamma \in \hat{G}$,

$$\left| \mathop{\mathrm{E}}_{f,U,W} \left[\operatorname{tr} \left(A(f) \sum_{\substack{\rho \in \hat{H} \\ |\rho| < c}} d_\rho (1-\epsilon)^{|\rho|} \operatorname{Tr} \left(\hat{B}_\rho^2 \left(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \right) \right) \right) \right] \right| \geq \eta \quad (11.11)$$

where $A \colon G^{\{0,1\}^U} \to \operatorname{End}(V)$ is unitary and $B \colon G^{\mathrm{SAT}^W} \to \operatorname{End}(V)$ is a convex combination of unitary matrices, V is the vector space corresponding to γ, and $H = G^{\mathrm{SAT}^W}$. Then there is a strategy for the provers in the 2P1R game with success probability at least $\eta^2 c^{-1} |G|^{-c} d_\gamma^{-6}$.

Corollary 11.29. Let $A_{U,G}$ and A_W be the tables in a Standard Written G-Proof with parameter u corresponding to an at most μ-satisfiable formula. Then, for any unitary $\gamma \in \hat{G}$,

$$\left| \mathop{\mathrm{E}}_{f,U,W} \left[\operatorname{tr} \left(A(f) \sum_{\substack{\rho \in \hat{H} \\ |\rho| < c}} d_\rho (1-\epsilon)^{|\rho|} \operatorname{Tr} \left(\hat{B}_\rho^2 \left(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \right) \right) \right) \right] \right| < \frac{d_\gamma \delta}{2} \quad (11.12)$$

where $A = \gamma \circ A_{U,G}$, $H = G^{\mathrm{SAT}^W}$ and $B(h) = \mathrm{E}_{b \in \{-1,1\}}[((\gamma \circ) A_W(h^b))^b]$ provided that $u > \lceil (2 \log \delta^{-1} + \log c + c \log |G| + 4 \log d_\gamma + 2) / \log c_\mu^{-1} \rceil$ where c_μ is the constant from Section 4.1.

Proof of Lemma 11.28. Expand $A(f)$ using Fourier inversion (11.6). Then the left hand side of (11.11) becomes

$$\left| \mathop{\mathrm{E}}_{U,W,f} \left[\operatorname{tr} \left(\sum_{\tau \in \hat{F}} \sum_{\substack{\rho \in \hat{H} \\ |\rho| < c}} d_\tau d_\rho (1-\epsilon)^{|\rho|} \operatorname{Tr} \left(\hat{A}_\tau \left(\boldsymbol{I}_{d_\gamma} \otimes \tau(f^{-1}) \right) \right) \operatorname{Tr} \left(\hat{B}_\rho^2 \left(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \right) \right) \right) \right] \right|$$

where $F = G^{2^U}$. If this expression is larger than η, then there must be some index t such that

$$\frac{\eta}{d_\gamma} \leq \left| \mathop{\mathrm{E}}_{U,W,f} \left[\left(\sum_{\tau \in \hat{F}} \sum_{\substack{\rho \in \dot{H} \\ |\rho| < c}} d_\tau d_\rho \operatorname{Tr}\Big(\hat{A}_\tau \big(\boldsymbol{I}_{d_\gamma} \otimes \tau(f^{-1}) \big) \Big) \operatorname{Tr}\Big(\hat{B}_\rho^2 \big(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \big) \Big) \right)_{tt} \right] \right| \tag{11.13}$$

We now fix this value of t. By our notation for the index sets $\iota(\tau)$ and $\iota(\rho)$ and the "projected" representation ρ^π_{op} from Section11.3.2,

$$\begin{aligned} \Big(\operatorname{Tr}\Big(\hat{A}_\tau \big(\boldsymbol{I}_{d_\gamma} \otimes \tau(f^{-1}) \big) \Big) \Big)_{tk} &= \sum_{m,n \in \iota(\tau)} \big(\hat{A}_\tau \big)_{tn,km} \tau_{mn}(f^{-1}), \text{ and} \\ \Big(\operatorname{Tr}\Big(\hat{B}_\rho^2 \big(\boldsymbol{I}_{d_\gamma} \otimes \rho(f \circ \pi^{U,W}) \big) \Big) \Big)_{kt} &= \sum_{o,p \in \iota(\rho)} \big(\hat{B}_\rho^2 \big)_{ko,tp} \rho^\pi_{op}(f) \\ &= \sum_{o,p \in \iota(\rho)} \sum_{\substack{q \in \iota(\rho) \\ 1 \leq r \leq d_\gamma}} (\hat{B}_\rho)_{ko,rq} (\hat{B}_\rho)_{rq,tp} \rho^\pi_{op}(f). \end{aligned}$$

Inserting these expressions into the right hand side of (11.13), we get

$$\begin{aligned} \frac{\eta}{d_\gamma} \leq \Bigg| \sum_{\substack{1 \leq k \leq d_\gamma \\ 1 \leq r \leq d_\gamma}} \mathop{\mathrm{E}}_{U,W} \Bigg[\sum_{\tau \in \hat{F}} \sum_{\substack{\rho \in \dot{H} \\ |\rho| < c}} \sum_{\substack{m,n \in \iota(\tau) \\ o,p,q \in \iota(\rho)}} \\ d_\tau d_\rho (\hat{A}_\tau)_{tn,km} (\hat{B}_\rho)_{ko,rq} (\hat{B}_\rho)_{rq,tp} \mathop{\mathrm{E}}_{f} \big[\tau_{mn}(f^{-1}) \rho^\pi_{po}(f) \bigm| U, W \big] \Bigg] \Bigg|. \end{aligned} \tag{11.14}$$

Focus now on the innermost expectation

$$\mathop{\mathrm{E}}_{f} \big[\tau_{mn}(f^{-1}) \rho^\pi_{po}(f) \big] = \mathop{\mathrm{E}}_{f} [\tau^*_{nm} \rho^\pi_{po}(f)] = \langle \rho^\pi_{po} \mid \tau_{nm} \rangle_F.$$

By Lemma 5.17, this is zero unless $\tau \sim \rho$, where $\sim$ is the relation defined in Lemma 5.17. Hence (11.14) becomes

$$\begin{aligned} \frac{\eta}{d_\gamma} \leq \Bigg| \sum_{\substack{1 \leq k \leq d_\gamma \\ 1 \leq r \leq d_\gamma}} \mathop{\mathrm{E}}_{U,W} \Bigg[\sum_{\substack{\rho \in \dot{H} \\ |\rho| < c}} \sum_{\substack{\tau \in \hat{F} \\ \tau \sim \rho}} \sum_{\substack{m,n \in \iota(\tau) \\ o,p,q \in \iota(\rho)}} \\ d_\tau d_\rho (\hat{A}_\tau)_{tn,km} (\hat{B}_\rho)_{ko,rq} (\hat{B}_\rho)_{rq,tp} \langle \rho^\pi_{po} \mid \tau_{nm} \rangle_F \Bigg] \Bigg|. \end{aligned} \tag{11.15}$$

We now apply Cauchy–Schwartz twice, first to the sum over k, r and then to the remaining sums, to simplify the above expression further:

$$\begin{aligned}\frac{\eta^2}{d_\gamma^4} &\le \sum_{\substack{1\le k\le d_\gamma\\ 1\le r\le d_\gamma}} \mathop{\mathrm{E}}_{U,W}\Bigg[\Bigg|\sum_{\substack{\rho\in\hat{H}\\ |\rho|<c}}\sum_{\substack{\tau\in\hat{F}\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o,p,q\in\iota(\rho)}} \\ &\qquad\qquad d_\tau d_\rho (\hat{A}_\tau)_{tn,km}(\hat{B}_\rho)_{ko,rq}(\hat{B}_\rho)_{rq,tp}\langle \rho^\pi_{po} \mid \tau_{nm}\rangle_F\Bigg|^2\Bigg] \\ &\le \sum_{\substack{1\le k\le d_\gamma\\ 1\le r\le d_\gamma}} \mathop{\mathrm{E}}_{U,W}\Bigg[\Bigg(\sum_{\substack{\rho\in\hat{H}\\ |\rho|<c}}\sum_{\substack{\tau\in\hat{F}\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o,p,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat{A}_\tau)_{tn,km}\big|^2\big|(\hat{B}_\rho)_{ko,rq}\big|^2\Bigg) \\ &\qquad\qquad \Bigg(\sum_{\substack{\rho\in\hat{H}\\ |\rho|<c}}\sum_{\substack{\tau\in\hat{F}\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o,p,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat{B}_\rho)_{rq,tp}\big|^2\big|\langle \rho^\pi_{po} \mid \tau_{nm}\rangle_F\big|^2\Bigg)\Bigg].\end{aligned} \tag{11.16}$$

We proceed by bounding the second factor above, i.e.,

$$\begin{aligned}&\sum_{\substack{\rho\in\hat{H}\\ |\rho|<c}}\sum_{\substack{\tau\in\hat{F}\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o,p,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat{B}_\rho)_{rq,tp}\big|^2\big|\langle \rho^\pi_{po} \mid \tau_{nm}\rangle_F\big|^2 \\ &\qquad = \sum_{\substack{\rho\in\hat{H}\\ |\rho|<c}}\sum_{p,q\in\iota(\rho)} d_\rho \big|(\hat{B}_\rho)_{rq,tp}\big|^2\Bigg(\sum_{\substack{\tau\in\hat{F}\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o\in\iota(\rho)}} d_\tau\big|\langle \rho^\pi_{po} \mid \tau_{nm}\rangle_F\big|^2\Bigg).\end{aligned} \tag{11.17}$$

By equation (11.4) in Lemma 11.7,

$$\sum_{\tau\in\hat{F}}\sum_{\substack{m,n\in\iota(\tau)\\ o\in\iota(\rho)}} d_\tau\big|\langle \rho^\pi_{po} \mid \tau_{nm}\rangle_F\big|^2 = \frac{1}{|F|}\sum_{f\in F}\sum_{o\in\iota(\rho)}|\rho_{po}(f\circ\pi^{U,W})|^2 = 1,$$

where the last equality follows since ρ is written in a unitary basis and the inner sum is therefore exactly one for every f. Regarding the rest of (11.17), note that $(\hat{B}_\rho)_{rq,tp} = |H|^{-1}\sum_{h\in H} B_{rt}(h)\rho_{qp}(h) = \langle B_{rt} \mid \rho^{-1}_{pq}\rangle_H$; another application of Lemma 11.7 therefore shows that

$$\sum_{\rho\in\hat{H}}\sum_{p,q\in\iota(\rho)} d_\rho\big|(\hat{B}_\rho)_{rq,tp}\big|^2 = \sum_{\rho\in\hat{H}}\sum_{p,q\in\iota(\rho)} d_\rho\big|\langle B_{rt} \mid \rho^{-1}_{pq}\rangle_H\big|^2 = \frac{1}{|H|}\sum_{h\in H}|B_{rt}(h)|^2 \le 1,$$

where the inequality follows since B is a convex combination of unitary matrices and therefore by Lemma 11.18 has entries with at most unit magnitude. Using the

above bounds in (11.16) transforms that bound into

$$\begin{aligned}\frac{\eta^2}{d_\gamma^4} &\le \sum_{\substack{1\le k\le d_\gamma\\ 1\le r\le d_\gamma}} \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\tau\in\hat F\\ \tau\sim\rho}}\sum_{\substack{\rho\in\hat H\\ |\rho|<c}}\sum_{\substack{m,n\in\iota(\tau)\\ o,p,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat A_\tau)_{tn,km}\big|^2\big|(\hat B_\rho)_{ko,rq}\big|^2\Bigg]\\ &\le |G|^c \sum_{\substack{1\le k\le d_\gamma\\ 1\le r\le d_\gamma}} \mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\tau\in\hat F\\ \tau\sim\rho}}\sum_{\substack{\rho\in\hat H\\ |\rho|<c}}\sum_{\substack{m,n\in\iota(\tau)\\ o,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat A_\tau)_{tn,km}\big|^2\big|(\hat B_\rho)_{ko,rq}\big|^2\Bigg]\end{aligned}$$

where the second inequality follows by summing over p in the innermost sum. To conclude, there must be some $k, r \in \{1, \dots, d_\gamma\}$ such that

$$\mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\rho\in\hat H\\ |\rho|<c}}\sum_{\substack{\tau\in\hat F\\ \tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\ o,q\in\iota(\rho)}} d_\tau d_\rho \big|(\hat A_\tau)_{tn,km}\big|^2\big|(\hat B_\rho)_{ko,rq}\big|^2\Bigg] \ge \frac{\eta^2}{|G|^c d_\gamma^6}. \tag{11.18}$$

We now describe the strategies for the provers in the 2P1R protocol. The index t is independent of U and W and can be calculated by the provers in advance. Also the values of k and r mentioned above can be calculated in advance.

Upon receiving W, P_1 first picks $\rho \in \hat H$ with probability

$$\sum_{o,q\in\iota(\rho)} d_\rho |(\hat B_\rho)_{ko,rq}|^2.$$

This is a well-defined procedure since

$$\sum_{\rho\in\hat H}\sum_{o,q\in\iota(\rho)} d_\rho \big|(\hat B_\rho)_{ko,rq}\big|^2 = \frac{1}{|H|}\sum_{h\in H}|B_{kr}(h)|^2 \le 1.$$

If the sum is < 1, P_1 takes some arbitrary action with the remaining probability. Having selected ρ, P_1 then returns a random y such that ρ_y is nontrivial. If no such y exists—this happens only if ρ is trivial—P_1 gives up.

Upon receiving U, P_2 picks $\tau \in \hat F$ with probability

$$\sum_{n,m\in\iota(\tau)} d_\tau |(\hat A_\tau)_{tn,km}|^2.$$

This is a well-defined procedure since

$$\sum_{\tau\in\hat F}\sum_{n,m\in\iota(\tau)} d_\tau \big|(\hat A_\tau)_{tn,km}\big|^2 = \frac{1}{|F|}\sum_{f\in F}|A_{tk}(f)|^2 \le 1.$$

Again, if the sum is < 1, P_2 takes some arbitrary action with the remaining probability. Then P_2 picks a random x such that τ_x is nontrivial and returns this x as

its answers. This is always possible since $\hat{A}_\tau$ is nonzero only for nontrivial τ by Lemma 4.10.

To give a lower bound on the success rate of this strategy, we argue that there are many choices of the provers that make the verifier accept: Specifically, suppose that P_1 picks ρ and P_2 picks τ such that $\tau \sim \rho$. If P_2 returns x', then there must be some $y' \in \pi^{-1}(x')$ such that $\rho_{y'}$ is nontrivial. The probability of P_1 picking this y' is at least $|\rho|^{-1}$. Hence the probability of success is at least

$$\mathop{\mathrm{E}}_{U,W}\Bigg[\sum_{\substack{\rho\in\hat{H}\\|\rho|<c}}\sum_{\substack{\tau\in\hat{F}\\\tau\sim\rho}}\sum_{\substack{m,n\in\iota(\tau)\\o,q\in\iota(\rho)}}\frac{d_\tau d_\rho\big|(\hat{A}_\tau)_{tn,km}\big|^2\big|(\hat{B}_\rho)_{ko,rq}\big|^2}{|\rho|}\Bigg] \geq \frac{\eta^2}{c|G|^c d_\gamma^6}$$

where the inequality follows from the bound (11.18). ∎

Finally, we put together these two parts and establish the soundness of the verifier.

Lemma 11.30. For any constants $\delta > 0$ and $0 < \epsilon < 1$, there is a choice of the parameters c and u such that the soundness of the PCP in Fig. 11.1 is at most $|G|^{-1} + \delta$.

Proof. Suppose for contradiction that ϕ is at most μ-satisfiable and there is a proof which the verifier accepts with probability $|G|^{-1} + \delta$. By Lemma 11.25, for this proof, there is a nontrivial irreducible representation γ of G such that

$$\left|\mathop{\mathrm{E}}_{f,U,W}\left[\mathrm{tr}\Big(A(f)\sum_{\rho\in\hat{H}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{Tr}\Big(\hat{B}_\rho^2\big(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi)\big)\Big)\Big)\right]\right| > d_\gamma\delta.$$

where $A = \gamma \circ A_{U,G}$, $H = G^{\mathrm{SAT}^W}$ and $B(h) = \mathrm{E}_{b\in\{-1,1\}}[(\gamma \circ A_W(h^b))^b]$. However, by selecting constants $c > \lceil(\log\delta - 1)/\log(1-\epsilon)\rceil$ and $u > \lceil(2\log\delta^{-1} + \log c + c\log|G| + 4\log d_\gamma + 2)/\log c_\mu^{-1}\rceil$, Corollaries 11.27 and 11.29 show that

$$\left|\mathop{\mathrm{E}}_{f,U,W}\left[\mathrm{tr}\Big(A(f)\sum_{\substack{\rho\in\hat{H}\\|\rho|\geq c}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{Tr}\Big(\hat{B}_\rho^2\big(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi)\big)\Big)\Big)\right]\right| < \frac{d_\gamma\delta}{2},$$

$$\left|\mathop{\mathrm{E}}_{f,U,W}\left[\mathrm{tr}\Big(A(f)\sum_{\substack{\rho\in\hat{H}\\|\rho|<c}} d_\rho(1-\epsilon)^{|\rho|}\,\mathrm{Tr}\Big(\hat{B}_\rho^2\big(\boldsymbol{I}_{d_\gamma}\otimes\rho(f\circ\pi)\big)\Big)\Big)\right]\right| < \frac{d_\gamma\delta}{2},$$

which is a contradiction. ∎

11.6 Hardness of approximating $\mathbf{EQ}_G^1[3]$

We now apply this PCP to obtain hardness results for approximating systems of equations over G.

Theorem 11.31. For any finite group G and any constant $\epsilon > 0$, it is **NP**-hard to approximate $\mathrm{EQ}^1_G[3]$ to within $|G| - \epsilon$.

Proof. By Lemma 11.24 and Lemma 11.30 for any constants $\epsilon > 0$ and $\delta > 0$ it is possible to choose the parameters of the verifier in Figure 11.1 such that it is **NP**-hard to distinguish between the case that there is a proof which the verifier accepts with probability $1 - \epsilon$, and the case that there is no proof which is accepted with probability more than $|G|^{-1} + \delta$.

Now we create a system of equations in the obvious way; the variables correspond to the positions in the proofs, and we add an equation for each random string corresponding to the test made for this random string. It can be shown that the instance so obtained contains $4(5n)^u|G|^{2^u}(|G|^2/\epsilon)^{7^u}$ equations, which is polynomial in n if u, $|G|$ and ϵ are constants. One may think that the equations would always be of the form $xy^iz^j = 1_G$, where $i, j \in \{1, -1\}$, but this is not the case due to folding over G and in general an equation is of the form $gxy^iz^j = 1_G$, where g is a group constant. There is also a technicality in that there is a small probability that $h = h^{-1}(f \circ \pi)^{-1}e$ in the protocol, and thus a variable can occur more than once. But since this happens with very small probability, such equations may be omitted from the instance. Hence we may construct instances of $\mathrm{EQ}^1_G[3]$ in which it is hard to distinguish between the case that $1 - \epsilon$ of all equations are satisfiable, and the case that at most $|G|^{-1} + \delta$ of all equations are satisfiable, which completes the proof. ∎

11.7 Conclusions and open problem

An interesting question is that of *satisfiable instances*. Some problems, such as E3-Sat, retain their inapproximability properties even when restricted to satisfiable instances. This is not the case for $\mathrm{EQ}^1_G[k]$ when G is a finite Abelian group, since if such a system is satisfiable a solution may be found essentially by Gaussian elimination. However, when G is non-Abelian, deciding whether a system of equations over G is satisfiable is **NP**-complete [22], so it seems reasonable that the problem over non-Abelian groups retains some hardness of approximation for satisfiable instances. However, the following simple argument shows that we can not hope, even for the non-Abelian groups, for a lower bound of $|G|^{-1} + \delta$: Given an instance σ of $\mathrm{EQ}^1_G[k]$ over some non-Abelian group G, we construct an instance σ' over $\mathrm{EQ}^1_H[k]$, where $H = G/G'$ and G' is the commutator subgroup of G, i.e., the subgroup generated by the elements $\{g^{-1}h^{-1}gh : g, h \in G\}$. The instance σ' is the same as σ, except that all group constants are replaced by their equivalence class in G/G'. Now since H is an Abelian group, we can solve over H. The solution is an assignment of cosets to the variables. We then construct a random solution of x by for each variable choosing a random element in the corresponding coset. Now the value of the left hand side of each equation will be uniformly distributed in the

coset of the right hand side, and thus we will satisfy an expected fraction $|G'|^{-1}$ of all equations.

Chapter 12

Concluding remarks

We have studied several combinatorial optimization problems and proved lower bounds on their approximability by constructing special purpose probabilistic proof systems, further establishing the power of PCPs to get strong inapproximability results. We analyzed these PCPs using the discrete Fourier transform.

What are the limitations of these techniques? Can we come up with new general techniques for proving **NP**-hardness of approximation other than constructing special purpose PCPs? And can we come up with new methods of constructing and analyzing PCPs?

It turns out that for some problems one can actually prove that PCPs in some sense are necessary. More precisely, one can prove that an **NP**-hardness of approximation result is equivalent to the existence of a PCP with certain parameters. For constraint satisfaction problems such as Maximum E3-Lin-2 or E3-Sat such a reverse connection holds almost trivially. It was also established by Bellare et al. [5] that the reverse connection holds for the Maximum Clique problem as well.

For many problems it is not known whether PCPs are necessary, and even for the problems where an inapproximability result implies the existence of a PCP, there might be an easier route to the inapproximability result than explicitly constructing the PCP. Thus is makes sense to search for other general methods of proving hardness of approximation.

A possible step in this direction was recently suggested by Feige [17], who proved a connection between the average case complexity of refuting CNF-formulas and lower bounds for several problems for which there currently are no **NP**-hardness of approximation results, among them Minimum Bisection. To turn these results into **NP**-hardness of approximation results would require a strong connection between worst case complexity and average complexity to be proven, however, and this seems to be far beyond us at present.

The PCPs which we have constructed in this thesis were analyzed using the discrete Fourier transform, a technique introduced by Håstad [27, 28]. Up until recently, this was essentially the only available tool to analyze the kind of PCPs

which we construct. However, Dinur and Safra [10], proving an improved lower bound for Minimum Vertex Cover, introduced combinatorial techniques which appears to open up new possibilities. Extensions of their techniques have already been used to get improved lower bounds for Minimum Vertex Cover on k-uniform hypergraphs [8], and for proving **NP**-hardness of coloring two-colorable 3-uniform hypergraphs with a constant number of colors [9]. Hopefully the techniques of Dinur and Safra can be applied to many other problems as well.

Let us finish with some intriguing open problems, whose solution will probably require significant new ideas.

Prove or disprove:

- For all $k \geq 2$, Minimum Ek-Vertex Cover is **NP**-hard to approximate within a factor $k - \epsilon$.
- Minimum Bisection is **NP**-hard to approximate within any constant.
- It is **NP**-hard to color a three-colorable graph with any constant number of colors.

Our conjecture is that the three statements above are true.

References

[1] Noga Alon and Joel H. Spencer. *The Probabilistic Method*. John Wiley & Sons, New York, 2nd edition, 2000.

[2] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Márió Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.

[3] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, January 1998.

[4] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity 1*, pages 3–40, 1991.

[5] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs and non-approximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

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

[7] Stephen A. Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing*, pages 151–158. Shaker Heights, Ohio, 3–5 May 1971.

[8] Irit Dinur, Venkatesan Guruswami, and Subhash Khot. Vertex cover on k-uniform hypergraphs is hard to approximate within factor (k-3-ϵ). Technical Report TR02-027, Electronic Colloquium on Computational Complexity, April 2002.

[9] Irit Dinur, Oded Regev, and Clifford Smythe. On the hardness of coloring 3-uniform hyper-graphs, 2002. Manuscript.

[10] Irit Dinur and Shmuel Safra. The importance of being biased. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 33–42. Montréal, Québec, Canada, 19–21 May 2002.

[11] Lars Engebretsen. The non-approximability of non-Boolean predicates. In Michel Goemans, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Proceedings of 5th International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 2129 of *Lecture Notes in Computer Science*, pages 241–248. Springer-Verlag, Berkeley, California, USA, 18–20 August 2001.

[12] Lars Engebretsen and Venkatesan Guruswami. Is constraint satisfaction over two variables always easy? In *Proceedings of 6th International Workshop on Randomization and Approximation Techniques in Computer Science*, volume 2483 of *Lecture Notes in Computer Science*, pages 224–238. Springer-Verlag, Cambridge, Massachusetts, USA, 13–15 September 2002.

[13] Lars Engebretsen and Jonas Holmerin. Clique is hard to approximate within $n^{1-o(1)}$. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Proceedings of 27th International Colloquium on Automata, Languages and Programming*, volume 1853 of *Lecture Notes in Computer Science*, pages 2–12. Springer-Verlag, Geneva, 9–15 July 2000.

[14] Lars Engebretsen and Jonas Holmerin. Three-query PCPs with perfect completeness over non-Boolean domains. Technical Report TR02-040, Electronic Colloquium on Computational Complexity, July 2002.

[15] Lars Engebretsen, Jonas Holmerin, and Alexander Russell. Inapproximability results for equations over finite groups. In Peter Widmayer, Francisco Triguero, Rafael Morales, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *Proceedings of 29th International Colloquium on Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 73–84. Springer-Verlag, Málaga, Spain, 8–13 July 2002.

[16] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, July 1998.

[17] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 534–543. Montréal, Québec, Canada, 19–21 May 2002.

[18] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Márió Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, March 1996.

[19] Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. In *41st Annual Symposium on Foundations of Computer Science*. IEEE, Redondo Beach, California, 12–14 November 2000.

[20] Michael R. Garey and David S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W. H. Freeman and Company, San Fransisco, 1979.

[21] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[22] Mikael Goldmann and Alexander Russell. The computational complexity of solving systems of equations over finite groups. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, pages 80–86. Atlanta, Georgia, 4–6 May 1999.

[23] Oded Goldreich. Using the FGLSS-reduction to prove inapproximability results for minimum vertex cover in hypergraphs. Technical Report TR01-102, Electronic Colloquium on Computational Complexity, December 2001.

[24] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. In *41st Annual Symposium on Foundations of Computer Science*, pages 149–158. IEEE, Redondo Beach, California, 12–14 November 2000.

[25] Venkatesan Guruswami, Daniel Lewin, Madhu Sudan, and Luca Trevisan. A tight characterization of NP with 3-query PCPs. In *39th Annual Symposium on Foundations of Computer Science*, pages 8–17. IEEE, Palo Alto, California, 8–11 November 1998.

[26] Eran Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 329–337. San Francisco, California, 9–11 January 2000.

[27] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

[28] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001.

[29] Johan Håstad and Subhash Khot. Query efficient PCPs with perfect completeness. In *42nd Annual Symposium on Foundations of Computer Science*, pages 610–619. IEEE, Las Vegas, Nevada, 14–17 November 2001.

[30] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and pcp. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity*, pages 244–254. Chicago, Illinois, USA, 18–21June 2001.

[31] Jonas Holmerin. Improved inapproximability results for vertex cover on k-uniform hypergraphs. In Peter Widmayer, Francisco Triguero, Rafael Morales, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *Proceedings of 29th International Colloquium on Automata, Languages and Programming*, volume 2380 of *Lecture Notes in Computer Science*, pages 1005–1016. Springer-Verlag, Málaga, Spain, 8–13 July 2002.

[32] Jonas Holmerin. Vertex cover on 4-regular hyper-graphs is hard to approximate within $2 - \epsilon$. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 544–552. Montréal, Québec, Canada, 19–21 May 2002.

[33] Jonas Holmerin and Subhash Khot. A strong inapproximability result for a generalization of minimum bisection, October 2002. Manuscript.

[34] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, December 1974.

[35] David R. Karger, Rajeev Motwani, and Madhu Sudan. Approximate graph coloring by semidefinite programming. *Journal of the ACM*, 45(2):246–265, 1998.

[36] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[37] Marek Karpinski. On approximability of minimum bisection problem. Technical Report TR02-046, Electronic Colloquium on Computational Complexity, July 2002.

[38] Subhash Khot. Improved inapproximability results for maxclique, chromatic number and approximate graph coloring. In *42nd Annual Symposium on Foundations of Computer Science*, pages 600–609. IEEE, Las Vegas, Nevada, 14–17 November 2001.

[39] Subhash Khot. Hardness of coloring 3-uniform hypergraphs and hardness of the not-all-equal predicate, April 2002. Manuscript.

[40] Subhash Khot. Hardness results for approximate hypergraph coloring. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 351–359. Montréal, Québec, Canada, 19–21 May 2002.

[41] Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *29th Annual Symposium on Foundations of Computer Science*, pages 422–431. IEEE, White Plains, New York, 24–26 October 1988.

[42] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[43] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, June 1998.

[44] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 475–484. El Paso, Texas, 4–6 May 1997.

[45] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 191–199. Portland, Oregon, 21–23 May 2000.

[46] Maria Serna, Luca Trevisan, and Fatos Xhafa. The (parallel) approximability of non-Boolean satisfiability problems and restricted integer programming. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1373 of *Lecture Notes in Computer Science*, pages 488–498. Springer-Verlag, Paris, 25–27 February 1998.

[47] Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1977.

[48] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[49] Audrey Terras. *Fourier Analysis on Finite Groups and Applications*, volume 43 of *London Mathematical Society student texts*. Cambridge University Press, Cambridge, 1999.

[50] Luca Trevisan. Parallel approximation algorithms by positive linear programming. *Algorithmica*, 21(1):72–88, May 1998.

[51] Luca Trevisan. Approximating satisfiable satsifiabiliy problems. *Algorithmica*, 28(1):145–172, September 2000.

[52] Luca Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 453–461. Hersonissos, Crete, Grece, 6–8 July 2001.

[53] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM Journal on Computing*, 29(6):2074–2097, December 2000.

[54] Uri Zwick. Approximation algorithms for constraint satisfaction programs involving at most three variables per constraint. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 201–210. San Francisco, California, 25–27 January 1998.