Sapienza University of Rome
Faculty of Engineering, Computer Science and Statistics

Space in weak propositional proof systems

by

Ilario Bonacina

Ph.D. Thesis
Rome, December 2015

Sapienza University of Rome
Faculty of Engineering, Computer Science and Statistics

Space in weak propositional proof systems

by

Ilario Bonacina

Ph.D. Thesis
Rome, December 2015

Thesis Committee
Prof. Nicola Galesi
Prof. Giuseppe Ateniese
Prof. Lorenzo Carlucci

Reviewers
Prof. Yuval Filmus
Prof. Jacobo Torán

to my family,

Abstract

In this thesis we consider logical proof systems from the point of view of their *space complexity*, in particular we focus on the following two:

- *Resolution*, a well studied proof system that is at the core of state-of-the-art algorithms to solve SAT instances;
- *Polynomial Calculus*, a proof system that uses polynomials to refute contradictions.

Informally speaking, the space of a proof measures the size of an auxiliary memory that a verifier needs to check the correctness of the proof. For Polynomial Calculus the space measure counts the number of distinct monomials to be kept in memory (*monomial space*). For Resolution the measure refers to the number of clauses to be kept in memory (*clause space*) or to the total number of symbols (*total space*).

We introduce an abstract framework to prove monomial space lower bounds and we apply it to prove asymptotically optimal lower bounds for the monomial space for random k-CNF formulas in n variables and a linear number of clauses. This was an open problem mentioned for the first time in [4, 22] and since then reported many times in the literature. The same framework also applies to the *graph pigeonhole principle*; to all the previously known monomial space lower bounds from [4, 70]; to *Tseitin formulas*, cf. [68].

While the clause space in Resolution is a well studied measure, cf. for instance [4, 8, 24, 66, 109], regarding total space much less was known, cf. [4]. For instance it was an open problem to prove any super-linear (in n) lower bound for formulas with n variables and $poly(n)$ clauses, cf. [4]. We introduce a general framework to prove total space lower bounds in Resolution, we show such super-linear lower bound and asymptotically optimal lower bounds for the total space needed to refute random k-CNF formulas.

In the last chapter we analyze the *size* of Resolution proofs in connection with the Strong Exponential Time Hypothesis. The strong lower bounds for a sub-proof system of Resolution, we called δ-regular Resolution, are based on *game-characterizations* of proof size and width in Resolution, cf. [7, 118].

The introductory chapter is a presentation of general proof complexity and a summary of the results and techniques used in this thesis.

Acknowledgments

First of all, I want to thank my advisor Nicola Galesi. He made me interest into proof complexity, he supported the good ideas and shot down the (many) buggy ones that, along the years, I produced. This thesis without him simply wouldn't have existed.

I want to thank all the co-authors of the papers that led to this thesis, in particular Neil Thapen and Navid Talebanfard. Respectively the chapter on total space and the chapter on strong resolution size would not have existed without our works together.

More in general, I am grateful to the many people from the proof complexity community that I had the privilege to meet in the last years. In some sense my view of proof complexity and theoretical computer science was shaped by all of the discussions we had. Jakob Nordström and Massimo Lauria, for the many discussions we had about space at the Royal Institute of Technology (KTH) and for the kind hospitality. Pavel Pudlák, Jan Krajíček and Neil Thapen for insightful discussions around proof complexity in general. Olaf Beyersdorff for showing me the QBF proof systems landscape. Rahul Santhanam and Andrew Drucker for interesting discussions around proof complexity and the Strong Exponential Time Hypothesis. This thesis is the result of the support (in many cases also economical) from the people mentioned here.

Moreover, I am grateful to Albert Atserias for sharing insightful ideas in proof complexity and for his detailed questions about this work which definitely improved it. Last, but just in order of time, I want to thank the reviewers of this thesis: their careful reading and their questions really improved the overall quality.

Contents

List of Figures

List of Tables

1

Introduction

1.1 What is proof complexity?

Proof complexity is a research area that studies the concept of complexity from the point of view of logic. In particular, in proof complexity we are interested in questions such as: *how difficult is it to prove a theorem?* Or, given a formal system, we are interested in measuring the *complexity* of a theorem, that is answering questions such as *what is the shortest proof of the theorem?* This corresponds to questions in computational complexity about, for example, the number of steps of Turing machines, or the size of circuits needed to compute a function. On the other hand, we could also measure the complexity of a theorem as the strength of a theory needed to prove the theorem. This also has a counterpart in computational complexity, it is linked with questions about the smallest complexity class to which a given function belongs.

Propositional proof complexity, that is the complexity of propositional proofs, plays a role in the context of feasible proofs as important as the role of Boolean circuits in the context of efficient computations. Although the original motivations to study the complexity of propositional proofs came from proof-theoretical questions about first-order theories, it turns out that, essentially, the complexity of propositional proofs deals with the following question: *what can be proved by a prover with bounded computational abilities?* For example if its computational abilities are limited to small circuits from some circuit class. Hence, propositional proof complexity mirrors to non-uniform computational complexity and indeed there is a very productive cross-fertilization of techniques between the two fields, cf. [14, 123]. Simple propositional proof systems are non-uniform analogues of natural fragments of Peano Arithmetic, the various Bounded

Arithmetic systems, which capture in some sense '*polynomial time reasoning*'. Hence, lower bounds in the former yield independence results in the latter and then such lower bounds may clarify the limits of our (human or automated) proof techniques. This is important also from the practical point of view since Automated Theorem Provers are essential in various aspects of computer science. Since Theorem Provers are implemented with simple propositional proof systems, e.g. Resolution, then the study of such propositional systems helps in clarifying the limits of actual Theorem Provers, cf. [108].

Our understanding of propositional proof systems is similar to the general situation in complexity theory, in the sense that in both fields we can prove lower bounds in very special cases and indeed there are many very basic and important open problems, such as the very famous $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$. In propositional proof complexity the situation is similar in the sense that we can prove super-polynomial lower bounds on the length of proofs only for restricted proof systems. Indeed, by a result of Cook and Reckhow [56], proving super-polynomial lower bounds on the length of proofs in *every* propositional proof system is equivalent to showing that $\mathsf{NP} \neq \mathsf{coNP}$, which in turn is one of the open and very important problems in computational complexity.

In this thesis we investigate space complexity in propositional proof systems, so what is the *space* of a proof? We saw that proof systems have some analogies with computational complexity and in that context space notions have been investigated: for example the size of a working-tape needed by a Turing machine to compute a given function. The analogue of this question was asked by Armin Haken in 1998 in the context of proof complexity. Pictorially, we could state the space question in proof complexity as *what is the smallest blackboard a teacher needs to present the proof of a theorem to a class of students?*[1] We postpone additional high level considerations on space in proof complexity to Section 1.5.

It is time to make things more precise and we start recapping some basic definitions and notations from propositional proof complexity.

1.2 Proof systems

proof system

We consider *proofs* and *theorems* as strings over some alphabet, say strings in $\{0,1\}^*$. According to Cook and Reckhow [56] a *proof system* for a language $\mathcal{L}$ is a polynomial-time onto function $P : \{0,1\}^* \to \mathcal{L}$. Each string $\varphi \in \mathcal{L}$ is a *theorem* and if $P(\pi) = \varphi$, π is a *proof* of φ in P, or a *P-proof* of φ. Given a

[1] We suppose here that the students can understand just proofs written on the blackboard in some given formal system and they do not have any additional memory except the minimal one to understand the content of the blackboard. Moreover the teacher has to write with fonts of a fixed size.

polynomial-time function $P : \{0,1\}^* \to \{0,1\}^*$ the fact that $P(\{0,1\}^*) \subseteq \mathcal{L}$ is the *soundness* property and the fact that $P(\{0,1\}^*) \supseteq \mathcal{L}$ is the *completeness* property.

soundness completeness

The computational complexity of a proof system for a language $\mathcal{L}$ varies a lot depending on the language $\mathcal{L}$ itself. It can vary from very easy, say P, for a language in NP; to PSPACE for the language of True Quantified Boolean Formulas (TQBF); to be completely intractable for First Order Logic (FO), due to the recursive undecidability of the existence of a proof in FO.

In this thesis we focus on proof systems for languages that are coNP complete, such as TAUT and UNSAT. Proof systems for the language TAUT of propositional tautologies are called *propositional proof systems*. Equivalently, propositional proof systems can be defined for the language UNSAT of unsatisfiable propositional formulas, in this second case we call them *refutational*.

propositional proof systems

Given two proof systems P and Q for the same language $\mathcal{L}$, P *p-simulates* Q if there exists a polynomial-time function t such that for each $\pi \in \{0,1\}^*$, $P(t(\pi)) = Q(\pi)$. Two systems are called p-equivalent if they p-simulate each other. If Q p-simulates P and there exists some formulas requiring exponentially long proof in P but polynomially long proofs in Q we say that P *is exponentially weaker than* Q.

p-simulates

exponentially weaker

1.2.1 Propositional proof systems

Propositional proof systems operate with Boolean formulas, the simplest of which are *clauses*, that is $\vee$s (OR) of *literals*, where each literal is either a variable x_j or a negation of a variable $\neg x_i$. A conjunction of clauses is a *CNF formula* and it is in the language UNSAT if it is unsatisfiable that is if no truth assignment of the variables satisfies it.

CNF formula

The main open problem in propositional proof complexity is about the length of proofs and, in particular, it concerns proving (or more likely disproving) the existence of a propositional proof system where all proofs are polynomially bounded. More precisely, we say that a proof system P for $\mathcal{L}$ is *polynomially bounded* (p-bounded) if there exists a polynomial p such that every $\varphi \in \mathcal{L}$ has a P-proof of size $\leqslant p(|\varphi|)$, where $|\varphi|$ is the length of φ.

p-bounded

Cook and Reckhow [56] showed that the existence of a p-bounded propositional proof system is equivalent to NP = coNP. Since this equality is conjectured to be false, the main goal of propositional proof complexity is to show that p-bounded propositional proof systems do not exist. One approach to this problem is to show that particular proof systems are not p-bounded. This approach is known as *Cook's program* in proof complexity. Quoting [95]:

> “*Proving that* NP ≠ coNP *showing incrementally that examples of proof systems are not polynomially bounded seems unlikely. Rarely a universal statement is proved by proving all its instances. Nevertheless proving these lower bounds we may hope to uncover hidden computational hardness assumptions and then try to reduce the conjecture to some more approachable problem.*” ”

To show that a particular propositional proof system P is not p-bounded it is sufficient to exhibit a family of formulas $(\mathsf{F}_n)_{n\in\mathbb{N}}$ such that the minimal length of a proof of F_n in P grows super-polynomially with respect to $|\mathsf{F}_n|$. Examples of such families of formulas arising from interesting combinatorial principles can be found in Chapter 4: in particular we will consider the pigeonhole principle (PHP^m_n, cf. Section 4.4.1), Tseitin formulas ($\mathsf{Tseitin}(G,\sigma)$, cf. Section 4.5), the random k-CNF formulas (cf. Section 4.8) and the matching principles over graphs (G-PHP, cf. Section 4.9).

Before going more into details on the particular propositional proof systems we consider (Resolution and Polynomial Calculus) we want to give an idea of the richness of the landscape of proof systems studied in propositional proof complexity. Among all the propositional proof systems, by far the first ones that *everybody* encounters are Frege systems hence we start we those.

Frege systems Those are the common ‘textbook’ proof systems for propositional logic based on axioms and rules, cf. [56]. A Frege proof consists of lines that are propositional formulas built from propositional variables x_i and Boolean connectives $\neg$ (NOT), $\wedge$ (AND), and $\vee$ (OR). A Frege system comprises a finite set of axiom schemes and rules, for example, $\varphi \vee \neg\varphi$ is a possible axiom scheme. A Frege *proof* is a sequence of formulas where each formula is either a substitution instance of an axiom, or can be inferred from previous formulas by a valid inference rule, for example the *modus ponens*

Frege

$$\frac{\varphi \qquad \varphi \to \psi}{\psi}\,.$$

Frege systems are required to be *sound* and *complete*, that is each tautology has to have a Frege proof and no formula that is not a tautology should have a Frege proof. The exact choice of the axiom schemes and rules does not matter as any two Frege systems are p-equivalent, even when changing the basis of Boolean connectives, cf. [56, 128] and [99, Theorem 4.4.13]. Therefore, we can assume w.l.o.g. that modus ponens is the only rule of inference. Usually Frege systems are defined such as proof systems where the last formula is the proven

tautology. To include also weak systems as Resolution in this picture we use the equivalent setting of refutation Frege systems where we start with the negation of the tautology that we want to prove and derive the contradiction $\bot$.

There are several common restrictions that can be imposed on Frege; for example *bounded-depth* Frege systems are Frege systems where lines are formulas with negations only on variables and with a bounded number of alternations between $\wedge$'s and $\vee$'s. More in general, given a circuit class $\mathscr{C}$, $\mathscr{C}$-Frege is a restriction of Frege where lines are circuits from the class $\mathscr{C}$, for a formal definition cf. [88]. *Resolution*, a proof system we see thorough this thesis and on an high level on Section 1.3, is a particular kind of bounded-depth Frege system that refutes CNF formulas, which are formulas of depth 2.

bounded depth Frege

$\mathscr{C}$-Frege

Other propositional proof systems are based on algebraic reasoning (for instance *Polynomial Calculus*), geometric reasoning (*Cutting Planes*) or some graph theoretic constructions (*Hajoś Calculus*, cf. [115]). The idea of using propositional proof systems simulating the most basic algebraic facts and constructions dates back to Beame et al. [16] who introduced a propositional proof system motivated by Hilbert's Nullstellensatz. Then Clegg et al. [55] introduced an even more natural algebraic proof system that directly simulates the process of generating an ideal from a set of generators. This proof system, *Polynomial Calculus*, is the other proof system object of our study. An informal introduction to it is contained in Section 1.4.

Cutting Planes The method of *Cutting Planes* for integer linear programming was introduced by Gomory [77] and Chvátal [52]. The so-called *Gomory-Chvátal cuts* transform a polytope defined by a system of linear inequalities into its integral hull. If the system of linear inequalities has no integral solution then the inequalities define a polytope with empty integral hull and the sequence of cuts can be taken as a witness of the fact that there are no integral solutions. W. Cook et al. [57] used this idea to define the *Cutting Planes* propositional proof system. Boolean formulas can be translated into a set of linear inequalities[2] such that the original formula is satisfied if and only if the defined polytope has a $\{0,1\}$ point. Then a cutting plane refutation of an unsatisfiable CNF formula is a sequence of linear inequalities and there are inference rules to take linear combinations of inequalities and to perform (a version of) Gomory-Chvátal cuts. Cutting Planes is exponentially stronger than Resolution and we know

Cutting Planes

[2] E.g. the CNF formula $\varphi = (x \vee \neg y \vee z) \wedge (\neg x \vee z)$ is translated into the following set of linear inequalities:

$$\Big\{x + (1-y) + z \geqslant 1,\ (1-x) + z \geqslant 1,\ 0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1,\ 0 \leqslant z \leqslant 1\Big\}.$$

some exponential lower bounds on size, cf. [82, 117]. Recently Cutting Planes has started to be investigated also from the point of view of space, cf. [75] and Section 1.5 for more details on space.

The strongest proof system for which we know that there are contradictions requiring exponentially long proofs is bounded-depth Frege, cf. [96, 98, 116]. Such lower bounds on bounded-depth Frege rely on a *Switching Lemma*, cf. [12]. We will not see directly such proofs but we will see an application of a version of the Switching Lemma in Chapter 5 for a sub-system of Resolution, cf. Lemma 5.1 on page 108.

One of the major open problems in proof complexity is to prove exponential lower bounds for a subsystem of Frege, $AC_0[p]$-Frege, handling bounded depth formulas with the usual logical connective plus MOD_p gates, cf. [119, Problem 10]. Indeed, algebraic proof systems were introduced by Clegg et al. [55] as a possible way to attack this problem.

1.3 Resolution

Res

Resolution, Res, was introduced by Blake [35] and proposed by Robinson [130] for automated theorem proving. Since [130] and the introduction of the DPLL algorithm by Davis and Putnam [62], Davis et al. [63], Resolution is at the core of most of automated theorem provers and it is by far the most studied propositional proof system.

Resolution is a refutational proof system manipulating unsatisfiable CNF formulas as sets of clauses, that is unordered disjunctions. The only inference rule is the following:

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D} \text{ (Res rule)},$$

where C, D denote clauses and x is a variable that we say is *resolved*. A Res refutation of a CNF formula φ derives the empty clause $\bot$ by repeatedly applying the Res rule.

k-CNF formula

More formally, a *k-CNF formula* is a formula $\varphi = C_1 \wedge \cdots \wedge C_m$ where C_i are clauses, that is disjunctions, of at most k literals and each literal is either a Boolean variable x or its negation $\neg x$. A Resolution derivation of a clause C from a CNF formula φ is a sequence of clauses $\pi = (C_1, \ldots, C_\ell)$ such that $C_\ell = C$ and each C_i is either a clause from the ones in φ or $\frac{C_j \quad C_{j'}}{C_i}$ for some $j, j' < i$ is an instance of the Res inference rule. A CNF formula φ is unsatisfiable if and only if the empty clause, $\bot$, can be inferred from φ using the Res rule.

The number of clauses in a Resolution refutation π is its *size*, $\mathsf{size}(\pi)$, and the minimum over $\mathsf{size}(\pi)$ over all sequences of clauses π that are Resolution refutations of φ is $\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot)$. $\mathsf{size}(\pi)$ $\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot)$

Complexity of Resolution To understand the complexity of Resolution proofs various hardness measures were defined and investigated. Historically, the first and most studied is the *size*. The very first lower bounds on the size of Resolution proofs were proved by Tseitin [138] and Haken [81] and since then the complexity of Resolution proofs was investigated in depth, cf. the surveys [14, 108, 114, 119, 125, 134]. In particular, two techniques that turned out to be very useful in proving lower bounds are the *feasible interpolation* by Krajíček [97] which apply to many further proof systems, for instance Cutting Planes, and the *size-width relationship* by Ben-Sasson and Wigderson [30]. Where the *width* of a proof is the length of the biggest clause appearing in a proof. The machinery by Ben-Sasson and Wigderson [30] allows to prove size lower bounds in an elegant and uniform way. Moreover, it shows that Resolution is automatizable in sub-exponential time by an extremely simple dynamic programming algorithm, cf. [22].

More formally, given a sequence of clauses π, $\mathsf{width}(\pi)$ is the size of the largest clause appearing in the sequence π. Then given an unsatisfiable CNF formula φ, $\mathsf{width}(\varphi \vdash \bot)$ is the minimum of $\mathsf{width}(\pi)$ over all sequence of clauses π that are valid Resolution refutations of φ. It is immediate to see that if $\mathsf{width}(\varphi \vdash \bot) \leqslant w$ and φ is a formula in n variables then $\mathsf{width}(\varphi \vdash \bot)$

$$\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot) \leqslant n^{O(w)}.$$

This trivial upper bound turns out to be tight: Atserias et al. [10] showed that there are k-CNF formulas φ_n in n variables refutable in width w but each Resolution refutation of φ_n must have size at least $n^{\Omega(w)}$. More interestingly, width lower bounds imply size lower bounds: Ben-Sasson and Wigderson [30] prove that for each unsatisfiable k-CNF formula φ in n variables

$$\log_2\left(\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot)\right) \geqslant \frac{(\mathsf{width}(\varphi \vdash \bot) - k)^2}{16n}, \tag{1.1}$$

so if $\mathsf{width}(\varphi \vdash \bot) \geqslant \omega(\sqrt{n \log n})$ then immediately φ must require Resolution refutations of super-polynomial size. The relation in equation (1.1) was further investigated by Bonet and Galesi [42] proving that such result is essentially optimal. Recently Thapen [137] proved that there are CNF formulas in n variables having polynomial size Resolution proofs and hence, by equation (1.1), width $O(\sqrt{n \log n})$ but where this decrease in width comes at the expense of an increase in size.

In Chapter 5 we will prove some results on Resolution size stronger than the size lower bound we could get by the technique presented above. Those results will rely on game characterizations of size and width from [118] and [8]. Section 2.6 contains more details on width and a related measure, the *asymmetric width*.

Subsystems of Resolution We recall that Resolution refutations can be associated to branching programs, cf. [99], and to (labeled) Directed Acyclic Graphs (DAG). An example of Resolution refutation, due to Huang and Yu [84], presented directly as a DAG is in Figure 1.1 on the next page. In [84] the author noticed that each minimal size Resolution proof of the formula

$$\varphi = (\neg x \vee a \vee b) \wedge (x \vee a \vee b) \wedge (\neg b \vee z) \wedge (\neg a \vee c) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg w) \wedge (\neg c \vee y \vee z) \wedge (\neg c \vee \neg x \vee w) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee w \vee \neg z)$$

corresponds to a DAG where there is a path with a variable resolved twice. In Figure 1.1 there is one of such minimal size Resolution refutations.

The structure of the DAGs associated to Resolution proofs are used to define subsystems of Resolution. A Resolution derivation is *tree-like* if the associated DAG is a tree. In *tree-like Resolution*, tree-Res, only tree-like derivations are allowed. Similarly, a Resolution derivation is *regular* if there is an associated DAG such that each directed path has no variable resolved more the once. In *regular Resolution*, reg-Res, only regular derivations are allowed. We will see more about tree-like and regular Resolution proofs in Chapter 5. For the moment we just recall that tree-Res is exponentially weaker than reg-Res, that, in turn is exponential weaker than Res, cf. [5, 41, 136, 142].

tree-Res

reg-Res

Connection with SAT solvers From the theoretical point of view, Resolution was viewed as the very first step in proving Frege proofs lower bounds, and indeed proof length lower bounds are known for sub-systems of Frege such as bounded-depth Frege. Nowadays Resolution is mostly studied due to its importance in applied contexts such as SAT solvers, in particular due to a connection to the *DPLL algorithm* and the *CDCL solvers*. The *Davis-Putnam-Logemann-Loveland* (DPLL) algorithm is a backtracking method introduced by Davis and Putnam [62], Davis et al. [63] to search for assignments satisfying a CNF formula. It is a well known result that the track of runs of the DPLL algorithm on unsatisfiable CNF formulas is equivalent to tree-Res, the sub-system of Resolution where only proofs having a tree structure are allowed.

DPLL

A strengthening of the DPLL algorithm was defined a series of works by Bayardo Jr. and Schrag [11], Moskewicz et al. [105], Silva and Sakallah

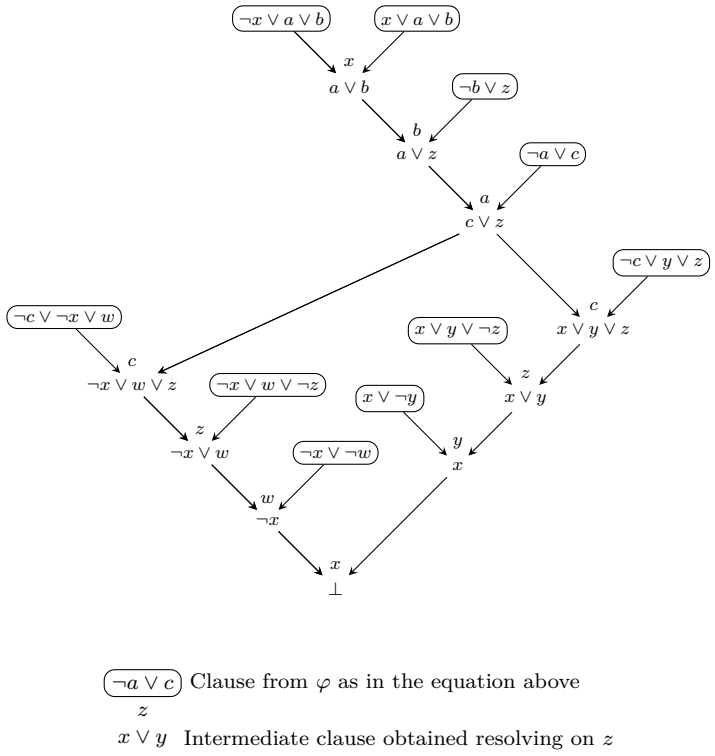


Figure 1.1: An example of Resolution refutation by Huang and Yu [84].

CDCL

[135] where the authors introduced the idea of *Conflict Driven Clause Learning* (CDCL) as a way for DPLL based SAT solvers to cut the search space and avoid duplicated work. This is done by performing a *conflict analysis* when the search for an assignment leads to a contradiction and then *learning* a clause encoding a reason for that failure. By definition Resolution p-simulates the CDCL solvers *viewed as proof systems*. Pipatsrisawat and Darwiche [112] and Atserias et al. [9] showed that the converse also holds *under certain assumptions* on the behaviour of the CDCL solver. In particular in both works a crucial hypothesis is that the CDCL solver *never* delete a learned clause. We stress out that this is not a realistic hypothesis and, at the moment, no extension of [9, 112] is known for CDCL solvers under more realistic models of the memory usage.

The running time and the memory consumption of SAT solvers relate to Resolution size and Resolution *space*. The latter is the complexity measure we will focus on in this thesis, cf. Section 1.5. However, size and space are not the only measures that are interesting with respect to applications to SAT solvers.

The question of what constitutes a good hardness measure for practical SAT solving is essentially open and a very important one, cf. [6, 32, 87]. For more details on the connection between Resolution, proof complexity and SAT solvers we refer to [108].

Game theoretic methods In proof complexity game theoretic methods and combinatorial characterizations of hardness measures have a long history. This started from the very first exponential size lower bound for Resolution by Haken [81]. Then, the Pudlák games characterized the size of Resolution proofs as a game, cf. [118], and we will use such games in Chapter 5. Interestingly, such games are meaningful for much stronger proof systems such as bounded-depth Frege, cf. [25]. Games have been used also to characterize other hardness measures, cf. [8] and [32], and the most notable example is the game and combinatorial characterization of Resolution width by Atserias and Dalmau [8]. In [8] the authors connected the width hardness measure to a combinatorial family of assignments and then to a game derived by the existential Ehrenfeucht-Fraïssé k-pebble game as used by Kolaitis and Vardi [92, 93] in the context of finite model theory and DATALOG. Games have proven to be useful also in the context of tree-like Resolution as shown by the optimal bounds obtained by Beyersdorff et al. [33, 34].

In this thesis we introduce some further combinatorial families and we relate them with hardness measures in Resolution and Polynomial Calculus. To each of the families we introduce we could have also associated a two player game such that the given family of assignments corresponds to a winning strategy for one of the two players. This could be done in a similar way as done explicitly in [118] for the size of proofs or in [8] for the width. In this thesis we will use explicitly such game characterisations in Chapter 5 to prove strong size lower bounds and implicitly in Chapter 4 to prove space lower bounds for random k-CNF formulas in Polynomial Calculus.

1.4 Polynomial Calculus

Our motivation to study algebraic proof systems is that they are not at all as well understood as Resolution and this lack of knowledge from the theoretical point of view is one of the reasons for not having efficient SAT solvers properly exploiting the potential of algebraic manipulations. Moreover, the study of algebraic proof systems could shed light on major open problems in propositional proof complexity such as super-polynomial size lower bounds for $AC_0[p]$-Frege.

The algebraic proof system introduced by Clegg et al. [55] is nowadays called *Polynomial Calculus*, PC, and it was improved by Alekhnovich et al. [4] to a system they call *Polynomial Calculus with Resolution*, PCR, that is a minimal extension of both Resolution and PC. Both in PC and PCR clauses are translated into polynomials as a sum of multilinear monomials having coefficients in a fixed field $\mathbb{F}$. Then, an unsatisfiable CNF formula φ is shown to be unsatisfiable by translating it into a set of polynomials $tr(\varphi)$ and then showing that 1 is in the ideal generated by $tr(\varphi)$, cf. Section 3.2 for a formal definition. This is done through the following two inference rules

PC

PCR

$$\frac{p \quad q}{\alpha p + \beta q}\ \alpha, \beta \in \mathbb{F} \qquad \frac{p}{xp}\ x \text{ variable.}$$

That is we can perform arbitrary linear combinations of already inferred polynomials and we can multiply an inferred polynomial by a variable. These rules model the fact that ideals are closed under the previous operations.

Both in PC and PCR the polynomials are manipulated in their expanded form as a sum of monomials, and the size of a proof is measured as the total number of monomials appearing in it. There are algebraic proof systems that allow manipulations on polynomials in an implicit form and this results in stronger, not so well understood, proof systems, cf. [122].

The difference between PC and PCR is that the latter one has separate formal variables to encode positive and negative literals over the same Boolean variable. Then, clauses with many literals are encoded more efficiently regardless of the polarity of the literals, which allows PCR to simulate Resolution efficiently. The first example of formulas requiring exponentially long proofs in Polynomial Calculus was given by [55], and since then many other size lower bounds were proved, cf. for instance [74, 103, 124].

Each PC refutation is also a valid PCR refutation and PCR refutations can be converted into PC refutations without increasing the degree of the polynomials involved, so, such systems, from the point of view of *degree lower bounds*, are exactly equivalent. Indeed, Impagliazzo et al. [86] showed that degree lower bounds imply PC size lower bounds. More precisely, given a k-CNF formula φ,

$$\log_2 \mathsf{size}_{\mathsf{PCR}}(\varphi \vdash \bot) \geqslant \Omega\left(\frac{(\mathsf{degree}_{\mathsf{PCR}}(\varphi \vdash \bot) - k)^2}{n}\right), \tag{1.2}$$

where $\mathsf{size}_{\mathsf{PCR}}(\varphi \vdash \bot)$ is the size needed in PC to refute φ and $\mathsf{degree}_{\mathsf{PCR}}(\varphi \vdash \bot)$ is the degree needed to refute φ. As observed by Mikša and Nordström [103], we have actually that the lower bound in equation (1.2) carries on also for PCR. It is interesting to notice the similarity between this lower bound and the, later, lower bound between width and size in Resolution by Ben-Sasson and Wigderson

[30], cf. equation (1.1). A lot of results on the complexity of Resolution proofs are indeed qualitatively similar to results on the complexity of PCR proofs. For instance a width upper bound of w implies a Resolution size upper bound of $n^{O(w)}$, for CNF formulas in n variables. Similarly a degree upper bound of d implies a size upper bound of $n^{O(d)}$ for formulas over n variables, cf. [55]. This result is qualitatively similar to the one for Resolution but the proof is a bit more involved. Both the upper bound in Resolution and the one in PCR are tight as recently shown by Atserias et al. [10]. As for Resolution, the degree-size lower bound is essentially optimal as shown by Galesi and Lauria [74], and the automatizability of PCR in sub-exponential time follows, cf. [73]. The formulas used in [74], the *ordering principles*, are the same used for Resolution. Similarly as for Resolution, most of the size lower bounds in PCR are obtained through degree lower bounds but the machinery to prove degree lower bounds in PCR is more involved and depends on the characteristic of the field $\mathbb{F}$ chosen.

In particular, if $\operatorname{char}\mathbb{F} \neq 2$ then some Fourier-like transformation can be used to reduce degree lower bounds to *Gaussian calculus* lower bounds and ultimately to Resolution, as in [26]. Another interesting result that depends on the characteristic of the ground field $\operatorname{char}\mathbb{F} \neq 2$ is the one by Razborov [127] about the hardness of *pseudorandom generators* for the Polynomial Calculus over the ground field $\mathbb{F}$. A more general technique to prove degree lower bounds, working also if $\operatorname{char}\mathbb{F} = 2$, was introduced in [3] and generalized in [73, 103].

Other algebraic proof systems have been considered, for example in [48, 49, 78–80, 113, 122]. In this thesis we focus on the proof system PCR and actually on some stronger *semantic* super-system of it (cf. Section 3.2) but we will be a bit sloppy when actually naming it since we will call it *Polynomial Calculus*, instead of the more precise *Polynomial Calculus with Resolution*.

Connection with SAT solvers The original name for Polynomial Calculus in [55] was *Gröbner proof system* due to its tight connection with the Gröbner basis algorithm and the system was intended to be a potential candidate for efficient new SAT solvers. Indeed, there are SAT solvers based on the Gröbner basis algorithm such as PolyBoRi [43, 44] but they are not competitive from the point of view of the pure practical performances with state of the art CDCL solvers based on Resolution. Some, very limited, form of algebraic reasoning is starting to be integrated into CDCL solvers but, at the moment of the writing of this thesis, this consist mostly of some form of Gaussian elimination. For more details on the interplay of algebraic proof systems and SAT solvers we refer to [108].

1.5 Space

The problem of the *space* taken by propositional proofs was posed for the first time by Armin Haken during the workshop "*Complexity Lower Bounds*" held at Fields Institute in Toronto 1998. Before that, apparently, the only paper devoted to the space of proofs was [94] but the author dealt only with equational theories involving no propositional connectives.

Intuitively, the space required by a refutation is the amount of information we need to keep simultaneously in memory as we work through the proof and convince ourselves that the original propositional formula is unsatisfiable. This model is inspired by the definition of space complexity for Turing machines, where a machine is given a read-only input tape from which it can download parts of the input to the working memory as needed. This model is sometimes called in the literature *blackboard model*. The name comes from the image of a teacher in front of a class of students. The goal of the teacher is to show that a particular CNF formula is contradictory writing down clauses and performing inferences on a blackboard. In this analogy students understand inferences based on the rules of some particular proof system, for example Frege or Res or PCR among others.

blackboard model

The formal definition of the space taken by Resolution proofs was given by Esteban and Torán [66] building on [91] and such definition was generalized by Alekhnovich et al. [4] to other proof systems. Esteban and Torán [66] proposed to measure the space of a Resolution proof as the number of clauses to be kept simultaneously in memory while refuting a contradiction[3], the *clause space*. As Alekhnovich et al. [4] point out, the very first question, when starting the investigation of space, is how to measure the memory content/blackboard size at any given moment in time for a specified propositional proof system. Recalling Krajíček [99], the most customary measures for the size complexity of propositional proofs are the bit size and the number of lines. Among the two the bit size is the most important and can be defined analogously also for space complexity. Similarly as what is done for size, usually we do not measure directly the bit size, but a logarithmically related measure that, in the case of space, is the total number of literals in memory, the *total space*[4]. The formal definitions of clause space and total space for Resolution are in Section 2.3. The line complexity is not an adequate space measure as long as the language of

clause space

total space

[3] As already noticed by [66], the clause space in Resolution is connected to the pebbling game on the DAGs associated to Resolution derivations but we do not exploit this analogy.

[4] Alekhnovich et al. [4] called this measure *variable space* but we prefer to call it *total space* following [27–29, 106, 107, 141]. The reason to do this is to distinguish this measure from another one, the *variable space*, where different occurrences of the same variable are not counted, cf. [141].

the proof system is strong enough to handle unbounded fan-in $\wedge$ gates: in this case just $O(1)$ memory cells are sufficient as one of them can contain a big-$\wedge$ of all the formulas derived in previous steps.

We already saw a proof system, Resolution, that is not closed under $\wedge$. For this system the lines are just clauses and the clause space makes prefect sense. An analogue of clause space makes sense also for stronger proof systems, such as Polynomial Calculus, where we consider the number of distinct *monomials* appearing in memory configuration, or Cutting Planes where the number of linear inequalities is considered, cf. [75].

monomial space

Lower bounds for the *monomial space* are one of the main topics of this thesis. The formal definition of monomial space is in Section 3.2.

Regarding the upper bounds, as shown by Esteban and Torán [66], all contradictions can be refuted within polynomial space for any 'reasonable' space measure. More precisely Esteban and Torán [66] showed that every contradictory CNF formula in n variables can be refuted by a (tree-like) Resolution proof of clause space $(n+1)$. Hence, the total space in Resolution is at most $n(n+1)$. Since the Resolution inference rule can be simulated efficiently in PCR, from the point of view of space, we have that the upper bounds in Resolution carry on for PCR. Hence, given an unsatisfiable CNF formula φ in n variables, the monomial space in PCR to refute it is at most linear in n and the total space is at most quadratic in n. Total space in PCR is not yet well understood and the only total space lower bound for PCR are the ones by Alekhnovich et al. [4] where this measure was originally introduced. Those total space lower bounds are for the complete tree formulas, CT_n, and for the pigeonhole principle, PHP^m_n, cf. Chapter 4, and rely on corresponding monomial space lower bounds.

An interesting property of clause space is that, informally, '*clause space is lower bounded by width*', cf. Proposition 2.2. This result was shown by Atserias and Dalmau [8] using a combinatorial characterisation of the width measure by some families of assignments, w-AD families in this work, cf. Theorem 2.3. Interestingly a more direct proof of this fact was recently shown in [69].

Regarding the monomial space the situation is quite different and recently there were some work widely improving over the results contained in [4]. In particular Filmus et al. [70] proved the first monomial space lower bounds for formulas of bounded width that are different encodings on the pigeonhole principle and we introduced a framework to prove monomial space lower bounds, cf. [36, 37]. That framework generalizes both the techniques used in [4] and the ones from [70] and self contained proofs of all the monomial space lower bounds from such papers are in Chapter 4. More importantly, such framework allows to prove the first monomial space lower bound for random k-CNF formulas, for

$k \geqslant 3$, and for the graph pigeonhole principle over a graph of (left) degree at least 3. Moreover, Filmus et al. [68] applied the framework to *Tseitin formulas* over random 4-regular graphs, some more information on such result are collected in cf. Section 4.5.

All the monomial space lower bounds we obtain are not dependent on the characteristic of the field, $\operatorname{char} \mathbb{F}$, where $\mathbb{F}$ is used as ground field in PCR. So the result from [68] is particularly interesting over $\mathbb{F}_2$ since over that field *Tseitin formulas* have polynomial size PCR refutations and those refutations, for the space lower bound shown, must require large monomial space.

Going back to space measures in general, interestingly two phenomenons happen: the first one is that for some space measures the actual inference rules of the proof systems do not matter, that is the space lower bound holds for some *semantic* version of the proof systems. What matters in such cases are the objects manipulated by the system, for example clauses or polynomials. This phenomenon was first observed by Alekhnovich et al. [4] for the clause space, for monomial space for some restricted class of formulas and for Frege in general, cf. [4, Corollary 6.6].

In this thesis we show that asymptotically optimal monomial space lower bounds hold for more general class of formulas than the ones in [4] and that for total space in Resolution the actual inference rule *does* matter, cf. Section 1.6 for more information.

The second interesting property of space is that this measure is actually non-trivial for not too strong proof systems, indeed Alekhnovich et al. [4, Theorem 6.3] showed that any tautology in n variables has a proof in Frege with "*formula space*" $O(1)$ and total space linear in the number of variables. This fact justifies the study of space for proof systems where super-linear lower bounds on space could be achieved, although total space in Frege is still a meaningful complexity measure. In this thesis we show some of such optimal space lower bounds for total space in Resolution and monomial space in Polynomial Calculus, cf. Section 1.6 for more information.

From the practical point of view, we already saw that Resolution is tightly connected to some class of SAT solvers, the CDCL solvers, cf. Section 1.3. Indeed, lower bounds for the space complexity measures for Resolution translate to lower bounds on the size of some auxiliary memory used by the CDCL solvers. Again from the practical point of view, there is a big difference between memory requirements that scale linearly or quadratically since that could be the difference between a feasible or a totally unfeasible problem. Linear lower bounds on space were implied by the lower bounds on clause space in Resolution. Quadratic lower bounds on space are implied by the new total space lower

bounds in Resolution we show in this thesis. On the other hand, it is not known if a generic CDCL solver p-simulates Resolution[5] on unsatisfiable CNF formulas and, from the space complexity point of view, it is perfectly possible that if the space usage of CDCL solvers is bounded, then they run, for example, in exponential time on instances easy for Resolution or worse. For more details on the connection between SAT solvers and proof complexity we refer to the survey [108].

We end this very introductory part on space in proof complexity citing some works that studied space related issues in Resolution and in some stronger proof systems:

- *Resolution*: [4, 8, 24, 66] and in particular concerning trade-offs [19, 20, 27, 29, 107, 109];
- *Resolution over k-DNFs*, a variation of Resolution handling k-DNF formulas instead of clauses: [28, 67];
- *Polynomial Calculus*: [4, 31, 37, 68, 70] and for tradeoffs for example [20, 20, 107];
- *Cutting Planes*: [75].

1.6 Main results + credits

From a very high level point of view, the backbone of the whole thesis is the use of combinatorial families of assignments (and games) to prove lower bounds[6]. This informal idea was applied many times in proof complexity and in general in complexity theory and here we apply it to prove:

- Lower bounds for monomial space in Polynomial Calculus, cf. Chapters 3 and 4.
- Lower bounds for total space in Resolution, cf. Chapters 2 and 4.
- *Strong* size lower bounds in (a sub-system of) Resolution, cf. Chapter 5.

[5] Interestingly the CDCL solvers that at the moment are known to be p-equivalent to Resolution behave very poorly from the point of view of the size of an auxiliary memory: they never cancel from the memory clauses learned, cf. [9, 112].

[6] We follow the convention to name families of assignments according to the initials of the authors that first introduced them, for instance the families from [8, Definition 2] are called w-AD families; the families from [40, Definition 2.3] are called r-BGT families; the families from [37, Definition 3.4] r-BG families and so on.

Space in Polynomial Calculus Regarding space in Polynomial Calculus the main results, in short, are the following:

- a combinatorial framework to prove space lower bounds in Polynomial Calculus, cf. Theorem 3.6;
- asymptotically optimal lower bounds on the space needed to refute random k-CNF formulas (and the graph pigeonhole principle) in Polynomial Calculus, cf. Theorem 4.36 and Theorem 4.38. This result was conjectured to be true and posed as an open problem in many works, for instance [4, 22, 70].

The space lower bound in Polynomial Calculus (Theorem 3.6) is one of the main contributions of this thesis and builds on the definition of r-BG *family* (Definition 3.4). This definition is one of the main innovations of this work, since it reduces space lower bounds in algebraic proof systems to a combinatorial property on families of Boolean assignments. Our definition resembles the definition of *k-dynamical satisfiability* in [67] which was used to prove space lower bounds for Resolution. Likewise, the definition of r-BG family is analogous to the definition of winning strategies for the Duplicator in the k-existential Spoiler-Duplicator game which led to the proof that in Resolution '*clause space is lower bounded by width*', cf. [8]. Informally, Theorem 3.6 states the following:

> *Given an unsatisfiable CNF formula φ, if there exists an r-BG family of partial assignments for φ then the monomial space in Polynomial Calculus to refute φ is at least $\frac{r}{4}$.*

We recall that Polynomial Calculus manipulates polynomials with coefficients in a field $\mathbb{F}$ but this result is independent from the characteristic of $\mathbb{F}$ and is valid over any field. The actual statement of Theorem 3.6 is more general and holds for a semantic version of Polynomial Calculus, cf. Section 3.1.

The space lower bound in Polynomial Calculus to refute random k-CNF formulas, Theorem 4.36, relies on an explicit construction of an r-BG family for such formulas. This is done in Section 4.8 and such construction relies on some general games, the *Cover Games*, defined in Section 4.6. Such games are an extension of the *Matching Game* devised in [24]. Unlike previous works that deal with classical matchings in bipartite graphs, here the game is generalized to *$\mathscr{C}$-matchings*: while a classical matching is a collection of vertex disjoint edges, a $\mathscr{C}$-matching is a collection of vertex disjoint graphs from the one in some sample space $\mathscr{C}$. For example a V-matching in a graph G is a collection of vertex disjoint subgraphs of G that looks like a V and similarly in VW-matchings

subgraphs that look like V or W are allowed. The formal definition needs some more technical details and it is in Section 4.6.

Then informally, given a bipartite graph G, the *Matching Game* guarantees that there is a family of matchings $\mathscr{F}$ such that each matching in $\mathscr{F}$ can be enlarged to cover new vertexes in G or shrunk while remaining in $\mathscr{F}$ and the family $\mathscr{F}$ has large matchings in it. The same kind of game is addressed for $\mathscr{C}$-matchings: the *Cover Game* guarantees that there is a family of $\mathscr{C}$-matchings $\mathscr{L}$ such that each to $\mathscr{C}$-matching in $\mathscr{L}$ can be added new connected components to cover new vertexes in G or removed connected components while remaining in $\mathscr{L}$ and the family $\mathscr{L}$ contains $\mathscr{C}$-matchings with many connected components.

Part of our contribution deals with extending classical results for matchings to V-matchings and VW-matchings. In particular we prove an analogue of Hall's Theorem, cf. Theorem 3.12, for VW-matchings; we prove an analogue of the Matching Game for V-matchings, cf. Theorem 4.15; we prove an analogue of the Matching Game for VW-matchings, cf. Theorem 4.22.

The construction we gave equally applies to random k-CNF formulas[7] and to the matching principle over graphs, G-PHP, cf. respectively Section 4.8 and Section 4.9.

The results on monomial space in Polynomial Calculus in this thesis rely on the following works:

(Bonacina and Galesi [36]) Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: a combinatorial characterization for the space measure in algebraic proof systems. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 455–472. ACM, 2013. doi: 10.1145/2422436.2422486. URL `http://doi.acm.org/10.1145/2422436.2422486`

(Bonacina and Galesi [37]) Ilario Bonacina and Nicola Galesi. A framework for space complexity in algebraic proof systems. *J. ACM*, 62(3): 23, 2015. doi: 10.1145/2699438. URL `http://doi.acm.org/10.1145/2699438`

(Bennett et al. [31]) Patrick Bennett, Ilario Bonacina, Nicola Galesi, Tony Huynh, Mike Molloy, and Paul Wollan. Space proof complexity for random 3-CNFs. *CoRR*, abs/1503.01613, 2015. URL `http://arxiv.org/abs/1503.01613`

[7]The case $k \geqslant 4$ was proved in [36, 37], while the case $k = 3$ was proved in [31] and required the introduction of the VW-matchings.

Total space in Resolution Regarding total space in Resolution the main result is a general technique to prove *total space* lower bounds in Resolution, cf. Theorem 2.5. In particular in Resolution we prove that '*total space is lower bounded by the square of width*', cf. Corollary 2.11. Then, as corollaries, we have the following:

1. An asymptotically optimal total space lower bound in Resolution for *Tseitin formulas* over d-regular expander graphs, cf. Theorem 4.7. This result completely answers an open problem from [4, Open question 2].

2. An asymptotically optimal total space lower bound in Resolution for random k-CNF formulas, cf. Theorem 4.36. This result completely answers an open problem from [4, 22, 68] among others.

3. An optimal separation of Resolution and *semantic* Resolution from the point of view of the total space measure, cf. the discussion at the end of Section 2.5 on page 33. This result completely answers [4, Open question 4] for Resolution.

Informally, our main theorem for total space in Resolution, Theorem 2.5, states the following

> *Given an unsatisfiable CNF formula φ, if there exists a r-BK family of assignments for φ then the total space in Resolution to refute φ is at least $\frac{r^2}{4}$. More precisely any refutation of φ must pass through a memory configuration of at least $r/2$ clauses each of width at least $r/2$.*

In [40] we proved an analogue of Theorem 2.5 using r-BGT families of assignments instead of the r-BK families from [32, Definition 21]. Although the extreme similarity among r-BGT and r-BK families, cf. Section 2.1 for more details, in this thesis we prefer to use r-BK families since those families allow us to prove that *'total space is lower bounded by the square of the width'*, cf. Corollary 2.11. This corollary is an original contribution of this thesis.

Regarding the corollaries listed above we have that all of them could be obtained from known width lower bounds and Corollary 2.11. On the other hand some particular r-BG families imply the existence of $(r-1)$-BK families[8], cf. Proposition 3.5, and this is the case for the families we will construct to prove monomial space lower bounds in PCR. Hence, typically, in this thesis we will prove *at the same time* monomial space lower bounds in PCR and total

[8]This implication is analogous to the implication between some r-BG families and $(r-1)$-BGT families showed in [31].

space lower bounds in resolution. Indeed the result on total space was inspired by the r-BG families from [36].

In Section 2.5 we show that Theorem 2.5 carries on for a bounded version of semantic Resolution, cf. Theorem 2.6. For total space in semantic Resolution we show some total space lower bounds but those are lower bunds for restricted class of formulas, the so called *n-semiwide formulas*, cf. Definition 2.7 and Theorem 2.8.

The results on total space in Resolution rely on the following works:

(Bonacina et al. [40]) Ilario Bonacina, Nicola Galesi, and Neil Thapen. Total space in resolution. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 641–650. IEEE Computer Society, 2014. doi: 10.1109/FOCS.2014.74. URL `http://dx.doi.org/10.1109/FOCS.2014.74`

(Bennett et al. [31]) Patrick Bennett, Ilario Bonacina, Nicola Galesi, Tony Huynh, Mike Molloy, and Paul Wollan. Space proof complexity for random 3-CNFs. *CoRR*, abs/1503.01613, 2015. URL `http://arxiv.org/abs/1503.01613`

Strong size lower bounds in Resolution The last chapter of this thesis, Chapter 5, contains the following results:

- A *strong* width lower bound for Resolution, cf. Thorem 5.6.
- A *strong* size lower bound for a generalisation of *regular* Resolution, cf. Corollary 5.8.

A *strong* size lower bound is a lower bound on the length of refutations of unsatisfiable k-CNF formulas φ in n variables of the form

$$2^{(1-\epsilon_k)n}, \tag{1.3}$$

where $\epsilon_k \to 0$ as $k \to \infty$. We show a strong size lower bound for δ-regular Resolution, a sub-system of Resolution where at most δn variables can be resolved multiple times while refuting an unsatisfiable CNF formula in n variables, cf. Corollary 5.8. In order to prove the result in equation (1.3) we further develop the game characterization of Resolution size by Pudlák [118], we show a general hardness amplification result lifting width lower bounds to size lower bound in δ-regular Resolution and we improve and simplify the strong width lower bound by Beck and Impagliazzo [21], cf. Theorem 5.6. The results on strong width and strong size lower bounds are based on:

(Bonacina and Talebanfard [39]) Ilario Bonacina and Navid Talebanfard. Strong ETH and Resolution via Games and the Multiplicity of Strategies. In Thore Husfeldt and Iyad Kanj, editors, *10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*, volume 43 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 248–257, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-92-7. doi: http://dx.doi.org/10.4230/LIPIcs.IPEC.2015.248. URL `http://drops.dagstuhl.de/opus/volltexte/2015/5587`

(Bonacina and Talebanfard [38]) Ilario Bonacina and Navid Talebanfard. Improving resolution width lower bounds for k-CNFs with applications to the Strong Exponential Time Hypothesis. *Information Processing Letters*, 116(2):120 – 124, 2015. ISSN 0020-0190. doi: http://dx.doi.org/10.1016/j.ipl.2015.09.013. URL `http://www.sciencedirect.com/science/article/pii/S0020019015001684`

1.7 Organization of this thesis

The results on space in Polynomial Calculus are technically more involved than the constructions for the total space lower bounds in Resolution, hence we chose to follow a gradual approach starting with Resolution and then moving to Polynomial Calculus. Then we end this thesis with some results not related with space but with size in Resolution, cf. Chapter 5.

Each chapter starts with some more detailed overview of the results/techniques introduced and ends with a list of related open problems.

Chapter 2 In this chapter we focus on the theoretical results we have on total space in Resolution, on semantic total space and on the connection with *width* and a similar measure called *asymmetric width*. Section 2.7 contains a recap of some interesting applications, more details will be given in Chapter 4.

Chapter 3 In this chapter we construct the framework to prove monomial space lower bounds in Polynomial Calculus. Then the applications of this framework are collected in Chapter 4.

This chapter is largely independent from Chapter 2. Only Section 2.2, containing some notations on partial assignments, is needed to understand Chapter 3. The rest of Chapter 2 is intended to be also helpful allowing the reader to familiarise with some notations and proof techniques.

Chapter 4 This chapter consists of several sections, each focusing on some particular class of formulas of interest in proof complexity. For each of them we will prove some space-related results and we will give some context and history. In particular each section contains the monomial and total space (in Resolution) lower bounds we can get as application of the main results of the previous chapters, cf. respectively Theorem 3.6 and Theorem 2.5.

The rationale behind the organization of this chapter is to start with less complicated applications, for instance CT_n formulas or PHP_n^m formulas, and then move to more involved applications, for instance random k-CNF formulas (cf. Section 4.8) and matching principles for graphs, cf. Section 4.9.

Chapter 5 This last chapter switch the focus from space to size: we show a strong size lower bound for *δ-regular Resolution*, a sub-system of Resolution intermediate between regular Resolution and Resolution; we further develop the game characterization of Resolution size; we show a general hardness amplification result lifting width lower bounds to size lower bounds in δ-regular Resolution; and we improve and simplify the strong width lower bound from [21].

Appendix For the convenience of the reader we collect in an appendix the definition of the r-BGT families from [40] and some additional proofs about the asymmetric width.

2

Total space in Resolution

In this chapter show a technique to prove total space lower bounds in Resolution and we connect it to other well studied complexity measures. Then in Chapter 4 we apply this construction to some families of contradictions well-studied in proof complexity. In Section 2.7 we recap some of the applications we can get from the abstract results we have in this chapter, in particular concerning total space for Tseitin formulas, cf. [4, Open question 2], and asymptotically optimal total space lower bounds for random k-CNF formulas.

2.1 Main results + credits

Informally our main theorem for total space in Resolution, Theorem 2.5, states the following

> *Given an unsatisfiable CNF formula φ, if there exists a r-BK family of assignments for φ then the total space in Resolution to refute φ is at least $\frac{r^2}{4}$. More precisely every resolution refutation of φ must pass through a memory configuration containing at least $r/2$ clauses of width at least $r/2$.*

The techniques we use in this chapter to prove the total space lower bounds in Resolution are deeply based on [40] but are different in the type of combinatorial objects used. In [40] we used r-BGT families of *piecewise* assignments here we use instead the r-BK families of assignments from [32, Definition 21], cf. Definition 2.4. Although the similarity among the r-BK families and the r-BGT families, the reason we prefer the k-BK families against the r-BGT families is that the k-BK families characterize the *asymmetric* width, cf. Beyersdorff

and Kullmann [32, Theorem 22] and Theorem 2.9. This characterization is analogous to the characterization of the Resolution width using the w-AD families, cf. [8] and Theorem 2.3. We refer to Appendix A.1 for the definition of r-BGT families and the main theoretical result of [40]. The use of r-BK families allows us to prove that *'total space is lower bounded by the square of the width'*, cf. Corollary 2.11, which is an original contribution of this thesis. From this result, and the fact that in the literature we have a lot of width lower bounds in Resolution, we will have immediately a lot of total space lower bounds. Hence, we narrowed the gap between the understanding we have of clause space and the understanding we had of total space, that is very limited before [4, 40] and this work. In Section 2.5 we show indeed that our total space lower bound argument carry on for a bounded version of semantic Resolution, cf. Theorem 2.6. For total space in semantic Resolution we show that there are some total space lower bounds but those are lower bunds for a restricted class of formulas, the so called n-semiwide formulas, cf. Definition 2.7 and Theorem 2.8. An optimal separation of Resolution and *semantic* Resolution from the point of view of the total space measure appears at the end of Section 2.5 on page 33. This result completely answers the question [4, Open question 4] for Resolution, that is asking whether total space in Resolution and semantic total space are asymptotically equivalent.

2.2 Partial assignments

We recall now some very basic definitions and notations about partial assignments. Given a set of variables X, a *partial assignment over* X (or just *assignment*) is a map $\alpha : X \to \{0, 1, \star\}$, where X is a set of variables.

dom
λ
$|\alpha|$
domain-disjoint

In the context of Resolution '1' has the meaning of *true* and '0' the meaning of *false*. The *domain* of α is $\mathrm{dom}(\alpha) = \alpha^{-1}(\{0, 1\})$ and we say that α is *assigning* a value to x if and only if $x \in \mathrm{dom}(\alpha)$. With λ we denote the partial assignment with empty domain. With $|\alpha|$ we denote $|\,\mathrm{dom}(\alpha)|$. Given a family of partial assignments F, let $\mathrm{dom}(F) = \bigcup_{\alpha\in F} \mathrm{dom}(\alpha)$. Two families of partial assignments F and F' are *domain-disjoint* if the sets $\mathrm{dom}(F)$ and $\mathrm{dom}(F')$ are disjoint.

$\alpha \cup \beta$

Given two partial assignments α and β, their *union* $\alpha \cup \beta$ is the partial

assignment

$$\alpha \cup \beta(x) = \begin{cases} \alpha(x) & \text{if } x \in \mathrm{dom}(\alpha) \setminus \mathrm{dom}(\beta), \\ \beta(x) & \text{if } x \in \mathrm{dom}(\beta) \setminus \mathrm{dom}(\alpha), \\ \alpha(x) & \text{if } x \in \mathrm{dom}(\alpha) \cap \mathrm{dom}(\beta) \text{ and } \alpha(x) = \beta(x), \\ \star & \text{otherwise.} \end{cases}$$

Given a partial assignment α over X and $Y \subseteq X$, the *restriction*, $\alpha \restriction_Y$, is the partial assignment $\alpha \restriction_Y$

$$\alpha \restriction_Y (x) = \begin{cases} \alpha(x) & \text{if } x \in Y, \\ \star & \text{otherwise.} \end{cases}$$

We say that β *extends* α, $\alpha \subseteq \beta$, if $\beta \restriction_{\mathrm{dom}(\alpha)} = \alpha$. Given a family F of partial assignments over X and given $Y \subseteq X$ we define $F \restriction_Y = \{\alpha \restriction_Y : \alpha \in F\}$. $\alpha \subseteq \beta$

Given a CNF formula φ over the variables X and a partial assignment α over X, we can apply α to φ obtaining a new formula $\alpha(\varphi)$, or denoted as $\varphi|_\alpha$, in this way: substitute each variable x in φ with the value $\alpha(x)$ if $x \in \mathrm{dom}(\alpha)$, or otherwise leave x untouched. Then simplify the result with the usual rules: $0 \vee A \equiv A$, $1 \vee A \equiv 1$, $0 \wedge A \equiv 0$, $1 \wedge A \equiv A$. We say that α *satisfies* φ, $\alpha \vDash \varphi$, if $\alpha(\varphi) = 1$. Similarly, for a family F of partial assignments we write $F \vDash \varphi$ if for each $\alpha \in F$, $\alpha \vDash \varphi$. $\alpha(\varphi)$ $\varphi|_\alpha$ $\alpha \vDash \varphi$

2.3 Space - preliminaries

The definition of the space measures was made formal for Resolution by Esteban and Torán [66] and Alekhnovich et al. [4] as follows. A *memory configuration*, or just *configuration*, is a set of clauses. We assume that a Resolution refutation of φ is given in the form of a sequence $\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell$ of memory configurations, where $\mathfrak{M}_0$ is empty, $\mathfrak{M}_\ell$ contains the empty clause, and each $\mathfrak{M}_{i+1}$ is derived from $\mathfrak{M}_i$ in one of the following three ways: memory configuration

(Axiom download) $\mathfrak{M}_{i+1} = \mathfrak{M}_i \cup \{C\}$, where C is a clause from φ;

(Erasure) $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i$;

(Inference) $\mathfrak{M}_{i+1} = \mathfrak{M}_i \cup \{D\}$ where D follows from two clauses in $\mathfrak{M}_i$ by the Resolution rule.

Following Esteban and Torán [66] and Alekhnovich et al. [4], given a sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$, the *clause space* used by π,

$\mathsf{CSpace}(\pi)$ $\mathsf{CSpace}(\pi)$, is the maximum number of clauses in any configuration $\mathfrak{M}_i$ in the sequence π. If we take the minimum of $\mathsf{CSpace}(\pi)$ over all the possible sequences of memory configurations π that are Resolution refutations of φ we have $\mathsf{CSpace}(\varphi \vdash \bot)$.

$\mathsf{CSpace}(\varphi \vdash \bot)$

$\mathsf{TSpace}(\pi)$ Given a sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$, the *total space* used by π, $\mathsf{TSpace}(\pi)$, is the maximum over i of the total number of instances of variables occurring in $\mathfrak{M}_i$ (we ignore punctuation and logical connectives). If we take the minimum of $\mathsf{TSpace}(\pi)$ over all the possible sequences of memory configurations π that are Resolution refutations of φ we have $\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot)$.

$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot)$

In some of the space related results in Resolution, for instance Propostion 2.2, the actual inference rule is not important, as long as we substitute that rule with a sound rule. This leads to the definition of *semantic Resolution*.

semantic Res A *semantic* Res *derivation* of a clause D from a CNF formula φ is sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ where $\mathfrak{M}_0 = \emptyset$, D is in $\mathfrak{M}_\ell$ and from a memory configuration $\mathfrak{M}_i$ we can move to a configuration $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i \cup C$, where either

(Semantic inference) C is implied by $\mathfrak{M}_i$, that is for each partial assignment α, if $\alpha \vDash \mathfrak{M}_i$ then $\alpha \vDash C$; or

(Axiom Download) $C \in \varphi$.

$\mathsf{CSpace}^{sem}(\varphi \vdash \bot)$ The *semantic* clause space, $\mathsf{CSpace}^{sem}(\varphi \vdash \bot)$, is defined in [4] analogously to $\mathsf{CSpace}(\varphi \vdash \bot)$: it is the minimum of $\mathsf{CSpace}(\pi)$ over all the possible sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ that are *semantic* Resolution refutations of φ.

$\mathsf{TSpace}^{sem}_{\mathsf{Res}}(\varphi \vdash \bot)$ The *semantic* total space, $\mathsf{TSpace}^{sem}_{\mathsf{Res}}(\varphi \vdash \bot)$, is defined in [4] analogously to $\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot)$: it is the minimum of $\mathsf{TSpace}(\pi)$ over all the possible sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ that are *semantic* Resolution refutations of φ.

Some results on space Esteban and Torán [66] showed that for any unsatisfiable CNF formula φ in n variables then

$$\mathsf{CSpace}(\varphi \vdash \bot) \leqslant n + 1. \tag{2.1}$$

The upper bound they show is indeed stronger since it holds for *tree-like* Resolution. An immediate consequence of this clause space upper bound is the following upper bound on total space in Resolution:

$$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) \leqslant n(n+1). \tag{2.2}$$

A first interesting property of the clause space in Resolution is that it is asymptotically equivalent to the clause space in *semantic* Resolution.

Proposition 2.1 (Alekhnovich et al. [4])**.** *For any unsatisfiable CNF formula* φ, $\mathsf{CSpace}^{sem}(\varphi \vdash \bot) \leqslant \mathsf{CSpace}(\varphi \vdash \bot) \leqslant 2 \cdot \mathsf{CSpace}^{sem}(\varphi \vdash \bot)$.

Alekhnovich et al. [4] asked if the analogue of this result could hold for total space in Resolution. A negative answer is shown in Section 2.5.

Regarding lower bounds on semantic clause space we have the following general result.

Proposition 2.2 (Atserias and Dalmau [8])**.** *Let* φ *be an unsatisfiable* k*-CNF formula, then* $\mathsf{CSpace}^{sem}(\varphi \vdash \bot) \geqslant \mathsf{width}(\varphi \vdash \bot) - k + 2$.

We are quoting this result, and the proof by Atserias and Dalmau [8], since, in Section 2.4, we will prove an analogous relation between total space in Resolution and *asymmetric* width, cf. Corollary 2.11. Moreover the proof is instructive since it has some structural analogies with the (more involved) proofs for total space in Resolution, cf. Section 2.4, and monomial space in algebraic proof systems, cf. Section 3.4.

In order to prove Proposition 2.2, Atserias and Dalmau [8] used the following characterization of width, which in turn is really helpful in proving width lower bounds and hence also size lower bounds, due to the Pudlák games [118] and the size-width relation by Ben-Sasson and Wigderson [30], cf. equation (1.1).

Theorem 2.3 (Atserias and Dalmau [8])**.** *Let* φ *be an unsatisfiable CNF formula, then* $\mathsf{width}(\varphi \vdash \bot) \geqslant w$ *if and only if there exists a non-empty family of partial assignments* $\mathscr{F}$ *with the following properties:*

(Consistency) *for every* $\alpha \in F$ *and every clause* C *in* φ, $\alpha(C) \neq 0$*;*

(Extension) *If* $\alpha \in \mathscr{F}$ *and* $\beta \subseteq \alpha$ *such that* $|\beta| < w$*, then for every variable* $x \notin \mathrm{dom}(\alpha)$*, there exist* $\beta' \in \mathscr{F}$ *with* $\beta \subseteq \beta'$ *such that* $x \in \mathrm{dom}(\beta')$.

We call families of partial assignments with the properties above w-AD families[1]. w-AD

We will use this result also in Chapter 5 but for the moment we use it to prove Proposition 2.2, as in Atserias and Dalmau [8][2].

[1] In [8] it is required that a w-AD family is closed under restrictions. This is not needed in order to obtain this characterization.

[2] An alternative proof of Proposition 2.2 for $\mathsf{CSpace}(\varphi \vdash \bot)$, not relying on w-AD families, was given by Filmus et al. [69].

Proof of Proposition 2.2. Let $\mathscr{F}$ be a non-empty w-AD family of assignments for φ. Suppose, by contradiction, that there exists sequence of memory configurations $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ that is a Resolution refutation of φ such that $\mathsf{CSpace}(\pi) \leqslant w - k + 1$. We show, by induction on $i = 0, \ldots, \ell$, that there exists $\beta_i \in \mathscr{F}$ such that $\beta_i \vDash \mathfrak{M}_i$. This leads to an immediate contradiction since $\mathfrak{M}_\ell = \{\bot\}$ is unsatisfiable. The base case $i = 0$ follows trivially: since $\mathscr{F}$ is non-empty then there is some α in $\mathscr{F}$. We take $\beta_0 = \alpha$ and clearly $\beta_0 \vDash \mathfrak{M}_0$ since $\mathfrak{M}_0 = \emptyset$. Suppose now that we are in the case where $1 \leqslant i \leqslant \ell$ and $\mathfrak{M}_i \subseteq \mathfrak{M}_{i-1} \cup \{C\}$, then we have to distinguish two cases:

(Sem. Inference case) $\mathfrak{M}_{i-1} \vDash C$. Hence $\beta_i = \beta_{i-1}$ is such that $\beta_i \vDash \mathfrak{M}_i$.

(Axiom Download case) $C \in \varphi$. It is immediate to see that if $\beta_{i-1} \vDash \mathfrak{M}_{i-1}$ then we can restrict it to some β'_{i-1} such that $\beta'_{i-1} \vDash \mathfrak{M}_{i-1}$ and $|\beta'_{i-1}| \leqslant |\mathfrak{M}_{i-1}|$ [3]. By hypothesis, $|\mathfrak{M}_i| \leqslant w - k + 1$, hence $|\mathfrak{M}_{i-1}| \leqslant w - k$ and $|\beta'_{i-1}| \leqslant w - k$. By the extension property of $\mathscr{F}$, it is possible to extend β'_{i-1} k times to a β_i with domain including all the variables appearing in C, since $|C| \leqslant k$. Now $\beta_i \in \mathscr{F}$ hence, by the consistency property of $\mathscr{F}$, β_i cannot falsify C. Since each variable of C is assigned by β_i, it must be that $\beta_i \vDash C$. Hence $\beta_i \vDash \mathfrak{M}_i$. □

2.4 Total space lower bounds

We are now going to show the main results we have on total space[4] and hence we start from the definition of r-BK families. Notice that this definition is very similar to the one of w-AD families, the difference is in the extension property: in the w-AD families we just require that we can extend to either setting the new variable to 0 or to 1, in the r-BK families we require to be able to extend to both values. More precisely we have the following definition.

r-BK **Definition 2.4** (r-BK, Beyersdorff and Kullmann [32])**.** *A family $\mathscr{F}$ of assignments is r-BK for a CNF formula φ if it has the following properties:*

(Consistency) *for every $\alpha \in F$ and every clause C in φ, $\alpha(C) \neq 0$;*

[3] For completeness we show how β'_{i-1} is constructed. For each clause C there is at least a literal ℓ_C in C such that $\beta_{i-1}(C) = 1$. Take one of such literals ℓ_C for each clause, clearly $|\{\ell_C : C \in \mathfrak{M}_{i-1}\}| \leqslant |\mathfrak{M}_{i-1}|$ and hence to satisfy $\mathfrak{M}_{i-1}$ it is sufficient to restrict β_{i-1} to the set of variables appearing in the set of literals $\{\ell_C : C \in \mathfrak{M}_{i-1}\}$. This restriction of β_{i-1} is our β'_{i-1}.

[4] This exposition follows closely [40], the only difference is that we use r-BK families of assignments instead of the r-BGT families from [40]. We recall the definition of r-BGT families in Appendix A.1.

(EXTENSION) *If $\alpha \in \mathscr{F}$ and $\beta \subseteq \alpha$ is such that $|\beta| < r$, then for every variable $x \notin \mathrm{dom}(\alpha)$ there exist $\beta_0, \beta_1 \in \mathscr{F}$ with $\beta \subseteq \beta_0, \beta_1$ such that $\beta_0(x) = 0$ and $\beta_1(x) = 1$.*

We show now that the existence of a non-empty r-BK family for a formula φ imply a $\frac{r^2}{4}$ total space lower bound for Resolution refutations of φ[5].

Theorem 2.5. *Let φ be an unsatisfiable CNF formula. If there is a family of assignments which is r-BK for φ, then*

$$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant \frac{r^2}{4}.$$

More precisely, we show that any refutation of φ in Res *must pass through a memory configuration containing at least $r/2$ clauses each of width at least $r/2$.*

Proof. Let $\mathscr{F}$ be a non-empty family of assignments which is a r-BK family for φ and let $\pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ be a Res refutation of φ, given as a sequence of memory configurations. Consider the set

$$S = \{C \text{ clause } : \exists \alpha \in \mathscr{F}\ \alpha(C) = 0\}.$$

Observe that, since $\mathscr{F}$ is non-empty, then $\bot \in S$ and, by the consistency property of $\mathscr{F}$, no clause from φ is in S. Hence, let

$$A = \{i \in [\ell] \ : \ \exists C \in \mathfrak{M}_i \cap S\ |C| < r/2\}.$$

Clearly A is non-empty, hence let $t = \min A$ and let $C \in \mathfrak{M}_t \cap S$ be a clause of width less than $r/2$. Let $\alpha \in \mathscr{F}$ be a partial assignment that falsifies C and let α_C be the minimal partial assignment contained in α falsifying C. We have that $|\alpha_C| = |C|$ and notice that α_C may not be in $\mathscr{F}$. Our goal now is to show that there is some $i < t$ such that $|\mathfrak{M}_i \cap S| \geqslant r/2$. Since for every $i < t$ every clause in $\mathfrak{M}_i \cap S$ has width at least $r/2$, this will give the desired result.

Suppose, for sake of contradiction, that for each $i < t$, $|\mathfrak{M}_i \cap S| < r/2$. We inductively construct a sequence of assignments $\beta_0, \ldots, \beta_t$ in $\mathscr{F}$ such that for each $i \leqslant t$ we have that $\alpha_C \subseteq \beta_i$ and that $\beta_i \vDash \mathfrak{M}_i \cap S$. This immediately give a contradiction when we reach β_t, since α_C falsifies the clause $C \in \mathfrak{M}_t \cap S$ and $\beta_t \supseteq \alpha_C$.

The first configuration $\mathfrak{M}_0$ is empty, so we can put $\beta_0 = \alpha_C$. Supposing that $0 \leqslant i < t$ and that we already have a suitable β_i, we construct β_{i+1} distinguishing between two cases.

[5]The proof of this result is analogous to the proof of [40, Theorem 2.4].

(Axiom download case) $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i \cup \{D\}$, where D is a clause from φ. By the consistency property of $\mathscr{F}$, D is not in S and we can simply put $\beta_{i+1} = \beta_i$.

(Inference case) $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i \cup \{D \vee E\}$ where $D \vee E$ follows by Resolution on some variable x from two clauses $D \vee x$ and $E \vee \neg x$ in $\mathfrak{M}_i$. Then, by the inductive hypothesis, there exists $\beta_i \in \mathscr{F}$ such that $\beta_i \vDash \mathfrak{M}_i \cap S$ and let β be a minimal size assignment contained in β_i such that $\alpha_C \subseteq \beta$ and $\beta \vDash \mathfrak{M}_i \cap S$. We have that[6]

$$|\beta| \leqslant |\alpha_C| + |\mathfrak{M}_i \cap S| < r/2 + r/2 = r.$$

Hence we can use the extension property of $\mathscr{F}$ on β_i. If $D \vee E$ contains a variable outside $\mathrm{dom}(\beta_i)$, then by the extension property we can extend β to some $\beta_{i+1} \in \mathscr{F}$ which satisfies $D \vee E$, as required. Suppose that all variables in $D \vee E$ are set by β. If $x \in \mathrm{dom}(\beta_i)$ let $\beta_{i+1} = \beta_i$, and otherwise let $\beta_{i+1} \in \mathscr{F}$ be any extension of β which assigns a value to x. Then β_{i+1} sets all variables in both $D \vee x$ and $E \vee \neg x$. It cannot falsify either clause, since that would imply that that clause is in S and thus is already satisfied by β_i. Therefore it must satisfy both clauses and thus also satisfy $D \vee E$. □

This proof deserves a bit of explanation. Informally, we can think of each element C of S as identified with a minimal assignment β_C in $\mathscr{F}$ which falsifies it. Then, since $\mathscr{F}$ is non-empty, S contains some assignment and, by the extension property of $\mathscr{F}$, has a rich structure. In particular, if a clause C in $\pi \cap S$ has width less than r and was derived by Resolution on a variable outside $\mathrm{dom}(\beta_C)$, then *both* parents of C in π are in S. The proof of Theorem 2.5 then uses an idea from [4], taking the first clause C in S with small width and applying the usual clause space lower-bound argument (cf. proof of Proposition 2.2) to the substructure of S which derives C.

We want to stress the fact that this proof is intrinsically different from the proof of Proposition 2.2. In particular in both proofs we have an axiom download case and an inference case. In this proof the easy case is the axiom download, while in the proof of Proposition 2.2 the easy case is the inference. If we, intuitively, believe that total space and *semantic* total space in Resolution are separated then the fact that inference case in the above proof is the '*hard*' case is to be expected. Otherwise (informally) the proof would have been valid for semantic total space too, while we (informally) believe this is not the case.

[6] The easy observation to prove $|\beta| \leqslant |\alpha_C| + |\mathfrak{M}_i \cap S|$ is virtually identical to the argument we gave in the footnote 3 on page 28.

2.5 Semantic total space

In the previous section we argued that the proof of Theorem 2.5 cannot be generalized to semantic total space but, indeed, it could be generalized to a *bounded version* of semantic Resolution. This is essentially due to the fact that the proof of Theorem 2.5 does not depend '*very much*' on the syntax of the inference rule but only on the number of premises.

In this section, first we see how to generalize Theorem 2.5 to a system that we called *d-semantic* Resolution, cf. Theorem 2.6; then we see a class of formulas for which we have semantic total space lower bounds; eventually we show that total space and semantic total space could be separated in the strongest way possible.

A *d-semantic* Resolution derivation of a clause D from a CNF formula φ is a sequence of memory configurations $\pi = (\mathfrak{M}_0, \dots, \mathfrak{M}_\ell)$ where $\mathfrak{M}_0 = \emptyset$, D is in $\mathfrak{M}_\ell$ and from a memory configuration $\mathfrak{M}_i$ we can move to a configuration $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i \cup \{C\}$, where either

d-semantic Res

(d-SEMANTIC INFERENCE) the clause C is implied by some set of at most d clauses in $\mathfrak{M}_i$, for a fixed integer d; or

(AXIOM DOWNLOAD) $C \in \varphi$.

Similarly as what we have seen before, we can easily adapt the space measures definitions to d-semantic Resolution: that is $\mathsf{TSpace}^{d\text{-}sem}_{\mathsf{Res}}(\varphi \vdash \bot)$ is the minimum of $\mathsf{TSpace}(\pi)$ over all sequences of memory configurations π that are a *d-semantic* Resolution refutation of φ. Then easily we have the following generalisation of Theorem 2.5 to total space in d-semantic Resolution[7].

$\mathsf{TSpace}^{d\text{-}sem}_{\mathsf{Res}}(\varphi \vdash \bot)$

Theorem 2.6. *Let φ be an unsatisfiable CNF formula and suppose $d \leqslant r$. If there is a non-empty family of assignments which is r-*BK *for φ, then*

$$\mathsf{TSpace}^{d\text{-}sem}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant (r-d)^2/4.$$

More precisely any d-semantic Resolution refutation of φ must pass through a configuration containing at least $(r-d)/2$ clauses each of width at least $(r-d)/2$.

Proof. The proof is the same as for Theorem 2.5, except that we replace the bound $r/2$ with $(r-d)/2$ and use a different argument for the inference case, as follows. Suppose $\mathfrak{M}_{i+1} \subseteq \mathfrak{M}_i \cup \{E\}$ where E is implied by clauses $D_1, \dots, D_d \in \mathfrak{M}_i$. We may assume that we have some $\beta_i \in \mathscr{F}$ such that

[7] The following result and proof is virtually identical to [40, Theorem 7.1].

$\beta_i \vDash \mathfrak{M}_i \cap S$ and let $\beta \subseteq \beta_i$ of minimal size such that $\beta \vDash \mathfrak{M}_i \cap S$. Since $|\alpha_C| < (r-d)/2$ and $|\mathfrak{M}_i \cap S| < (r-d)/2$, then $|\beta| \leqslant |\alpha_C| + |\mathfrak{M}_i \cap S| < r - d$.

Either D_1 is satisfied by β_i or it is not. If it is, let $\gamma_1 = \beta_i$. If not, then D_1 cannot be in S, since β_i satisfies all members of $\mathfrak{M}_i \cap S$. It follows that D_1 is not falsified by β_i either, otherwise D_1 would be in S. Then, by the inductive hypothesis, D_1 will be satisfied by β_i. So D_1 thus must contain some literal not set by β_i. In this case let $\gamma_1 \in \mathscr{F}$ be an extension of β which satisfies this literal.

We have found $\gamma_1 \in \mathscr{F}$ which satisfies D_1 with $\beta \subseteq \gamma_1$. We then take a minimal partial assignment γ_1' contained in γ_1 such that $\gamma_1' \vDash D_1$ and $\gamma \supseteq \beta$. We have that $|\gamma_1'| \leqslant |\beta| + 1 < r - d + 1$, so we can repeat the previous reasoning to γ_1' and D_2 instead of β and D_1 and again up to D_d. In this way we build a sequence of extensions $\gamma_1 \subseteq \gamma_2 \subseteq \cdots \subseteq \gamma_d$ in $\mathscr{F}$, finishing with γ_d which satisfies each of $D_1, \ldots, D_d$ and thus also satisfies the inferred clause E. We put $\beta_{i+1} = \gamma_d$. □

We show now that some particular CNF formulas have total space lower bounds in *semantic* Resolution. Those formulas are the *r-semiwide* formulas.

σ-consistent
r-semiwide

Definition 2.7 (semiwide formula, Alekhnovich et al. [4])**.** *For a CNF formula σ and a partial assignment α, we say that α is* σ-consistent *if α can be extended to satisfy σ. A CNF formula φ is* r-semiwide *if $\varphi = \sigma \wedge \omega$, where σ is a satisfiable CNF formula, and for each σ-consistent partial assignment α and each clause C from ω, if $|\alpha| < r$ then α can be extended to an ω-consistent assignment which satisfies C.*

Alekhnovich et al. [4] showed that if φ is r-semiwide then

$$\mathsf{CSpace}^{sem}(\varphi \vdash \bot) \geqslant r.$$

We now strengthen this resultand show that $\mathsf{TSpace}^{sem}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant \frac{r^2}{4}$.

The argument we present[8] is a straightforward generalisation of the total space lower bounds by Alekhnovich et al. [4] for two particular n-semiwide formulas: PHP^{n+1}_n and CT_n, the *only* total space lower bound known before [40]. We refer to Section 4.4 for the definition of PHP^m_n and to Section 4.3 for the definition of CT_n.

Theorem 2.8. *Let φ be an unsatisfiable r-semiwide CNF formula. Then,*

$$\mathsf{TSpace}^{sem}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant \frac{r^2}{4}.$$

[8] The same from [40, Theorem 7.3].

More precisely, every semantic Resolution refutation of φ must pass through a memory configuration containing $r/2$ clauses each of width at least $r/2$.

Proof. Let $\varphi = \sigma \wedge \omega$ as in Definition 2.7 and let $(\mathfrak{M}_1, \ldots, \mathfrak{M}_s)$ be a refutation of φ. Let $\mathfrak{M}_i^* = \{C \in \mathfrak{M}_i \ : \ \sigma \not\models C\}$. Take the first t such that there exists a clause $C \in \mathfrak{M}_t^*$ of width strictly less than $r/2$. Fix such a clause C and let α be the minimal partial assignment falsifying C. Then α is σ-consistent and $|\operatorname{dom}(\alpha)| = |C| < r/2$.

It is now enough to show that $|\mathfrak{M}_i^*| \geqslant r/2$ for some $i < t$, since for $i < t$ every clause in $|\mathfrak{M}_i^*|$ has width at least $r/2$. So suppose for a contradiction that $|\mathfrak{M}_i^*| < r/2$ for all $i < t$. We prove by induction that for each $i = 1, \ldots, t$ there exists some σ-consistent $\beta_i \supseteq \alpha$ such that $\beta_i \models \mathfrak{M}_i^*$. This leads immediately to a contradiction when $i = t$.

For the erasure case we put $\beta_{i+1} = \beta_i$. For the semantic inference case, that is $\mathfrak{M}_i \models \mathfrak{M}_{i+1}$, we let β_{i+1} be any extension of β_i which satisfies σ. Then from the fact that $\beta_{i+1} \models \mathfrak{M}_i^* \wedge \sigma$ it follows that $\beta_{i+1} \models \mathfrak{M}_i$ and hence $\beta_{i+1} \models \mathfrak{M}_{i+1}$. For the axiom download case, suppose that $\mathfrak{M}_{i+1} = \mathfrak{M}_i \cup \{D\}$ with D a clause from ω. We may assume without loss of generality that $|\operatorname{dom}(\beta_i)| \leqslant |\operatorname{dom}(\alpha)| + |\mathfrak{M}_i^*| < r$. Hence by the r-semiwideness of φ there is a σ-consistent $\beta_{i+1} \supseteq \beta_i$ such that $\beta_{i+1} \models D$. □

Alekhnovich et al. [4] ask whether the total space in Resolution and the total space in semantic Resolution are linearly related, that is if the same of Proposition 2.1 is true for total space.

This is not the case. Indeed for almost every k-CNF formula in n variables and $\Theta(n)$ many clauses

$$\mathsf{TSpace}_{\mathsf{Res}}^{sem}(\varphi \vdash \bot) = \Theta(n),$$

while

$$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) = \Theta(n^2).$$

In particular, in Section 4.8, we will see that given φ an unsatisfiable random k-CNF formula with n variables and Δn clause. We can refute φ in semantic Resolution by simply writing down all the clauses of φ and then deriving the empty clause in one step. This uses total space Δkn, the size of φ. So, if Δ and k are constants then $\mathsf{TSpace}_{\mathsf{Res}}^{sem}(\varphi \vdash \bot) = \Theta(n)$. On the other hand, when n is large, $\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) = \Theta(n^2)$, by Theorem 4.36.

2.6 From width to total space

In this section we exploit the connection between r-BK families and *asymmetric* width to obtain an analogue of the *'clause space is lower bounded by width'*, cf. Proposition 2.2 by Atserias and Dalmau [8]. That is we will prove that in Resolution *'total space is lower bounded by the square of the width'*, cf. Corollary 2.12. In order to prove this relation we define precisely the asymmetric width[9] and to do that we need to be more precise on the DAG structure we associate to Resolution proofs.

Let φ be an unsatisfiable CNF formula and $\pi = (C_1, \ldots, C_\ell)$ be a sequence of clauses such that $C_\ell = \bot$. We say that a function $\sigma : [\ell] \to \binom{[\ell]}{2} \cup \{\star\}$ is *witnessing* the fact that π is a refutation of φ if and only if

witness function

1. $\sigma(i) = \{j, k\}$ imply that $j, k < i$ and $\frac{C_j \quad C_k}{C_i}$ is a valid instance of the Res rule and
2. $\sigma(i) = \star$ imply that C_i is a clause in φ.

Then given a sequence of clauses $\pi = (C_1, \ldots, C_\ell)$ such that $C_\ell = \bot$, a witness function σ for the sequence π and a clause C_i in π, the *asymmetric width of C_i with respect to π and σ*, $aw_{\pi,\sigma}(C_i)$, is defined as follows

$aw_{\pi,\sigma}(C_i)$

$$aw_{\pi,\sigma}(C_i) = \begin{cases} 0 & \text{if } \sigma(i) = \star, \text{ that is } C_i \in \varphi, \\ \min_{j \in \sigma(i)} |C_j| & \text{otherwise.} \end{cases}$$

Then $\mathsf{awidth}(\pi)$ is the minimum over all the possible functions σ witnessing the validity of π of the maximum over i of $aw_{\pi,\sigma}(C_i)$, that is

$\mathsf{awidth}(\pi)$

$$\mathsf{awidth}(\pi) = \min_{\sigma} \max_{C_i \in \pi} aw_{\pi,\sigma}(C_i).$$

Finally, the *asymmetric width* needed to refute φ, $\mathsf{awidth}(\varphi \vdash \bot)$, is the minimum of $\mathsf{awidth}(\pi)$ over all possible sequence of clauses $\pi = (C_1, \ldots, C_\ell)$ that are Resolution refutations of φ.

$\mathsf{awidth}(\varphi \vdash \bot)$

The notion of asymmetric width was introduced by Kullmann [101, 102] and, although its definition is quite different from the definition of width, many properties of the width carry over. For instance an analogue of the size-width relation by Ben-Sasson and Wigderson [30]: given an unsatisfiable CNF formula φ in n variables

$$\ln\left(\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot)\right) \geqslant \frac{\mathsf{awidth}(\varphi \vdash \bot)^2}{8n},$$

[9] The original definition, as given by Beyersdorff and Kullmann [32], uses a different notation but we preferred to present the notion avoiding the introduction of too many notations.

cf. [102, Theorem 6.12]. For more information and history on the asymmetric width we refer to [32]. The main property we will use of the asymmetric width is the following characterisation using r-BK families of assignments.

Theorem 2.9 (Beyersdorff and Kullmann [32, Theorem 22])**.** *Let φ be an unsatisfiable CNF formula, then* $\mathsf{awidth}(\varphi \vdash_{\mathsf{Res}} \bot) > r$ *if and only if there exists a non-empty r-*BK *family of assignments for φ.*

The proof of this result, for completeness in Appendix A.2, is essentially the one in [32] and can be seen as a modification of the characterisation of Resolution width by Atserias and Dalmau [8]. Indeed, width and asymmetric width are complexity measures tightly connected, as the next result shows.

Theorem 2.10 (Kullmann [100, Lemma 8.5])**.** *Let φ be an unsatisfiable k-CNF formula, then*

$$\mathsf{awidth}(\varphi \vdash \bot) \leqslant \mathsf{width}(\varphi \vdash \bot) \leqslant \mathsf{awidth}(\varphi \vdash \bot) + \max\{\mathsf{awidth}(\varphi \vdash \bot), k\}.$$

For completeness, a proof of this result is in Appendix A.2. Such proof is essentially a self-contained exposition of the one in the underlying report of [32].

Putting all the pieces together then we have a relation between total space in Resolution and width.

Corollary 2.11. *Let φ be an unsatisfiable k-CNF formula, then*

$$\begin{aligned}\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) &\geqslant \frac{1}{4}\left(\mathsf{awidth}(\varphi \vdash_{\mathsf{Res}} \bot) - 1\right)^2 \\ &\geqslant \frac{1}{16}\left(\mathsf{width}(\varphi \vdash_{\mathsf{Res}} \bot) - k - 2\right)^2.\end{aligned}$$

Proof. Let $\mathsf{awidth}(\varphi \vdash_{\mathsf{Res}} \bot) = r$. By Theorem 2.9, there exists a non-empty $(r-1)$-BK family $\mathscr{F}$ for φ. By Theorem 2.5,

$$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant \frac{1}{4}\left(r - 1\right)^2.$$

Moreover, from Theorem 2.10 we immediately have that

$$\mathsf{width}(\varphi \vdash \bot) \leqslant 2 \cdot \mathsf{awidth}(\varphi \vdash \bot) + k,$$

hence the last inequality to prove follows. □

Notice that, with the exact same proof of Corollary 2.11, we have indeed the following stronger result, if instead of Theorem 2.5 we use Theorem 2.6.

Corollary 2.12. *Let φ be an unsatisfiable k-CNF formula, then*

$$\begin{aligned}\mathsf{TSpace}_{\mathsf{Res}}^{d\text{-}sem}(\varphi \vdash \bot) &\geqslant \frac{1}{4}\left(\mathsf{awidth}(\varphi \vdash \bot) - d - 1\right)^2 \\ &\geqslant \frac{1}{16}\left(\mathsf{width}(\varphi \vdash \bot) - k - 2d - 2\right)^2.\end{aligned}$$

We stress that an analogue of the previous corollaries cannot hold for total space in semantic Resolution. In fact we know that there are 3-CNF formulas[10] φ in n variables with a linear number of clauses such that

$$\mathsf{width}(\varphi \vdash \bot) = \Omega(n).$$

On the other hand, by the trivial semantic Resolution proof of φ, we have that

$$\mathsf{TSpace}_{\mathsf{Res}}^{sem}(\varphi \vdash \bot) \leqslant 3|\varphi| = O(n),$$

in memory, so general versions of Corollary 2.12 and Corollary 2.12 for semantic Resolution cannot hold.

2.7 Recap of applications

In Chapter 4 we collect some of the total space lower bounds we could obtain, for instance quadratic asymptotically optimal lower bounds for random k-CNF formulas among others. Although some total space lower bounds could be obtained through width lower bounds proved elsewhere, Chapter 4 contains also self contained proofs of some of the total space lower bounds we could get. This is due to the fact that the (more complicated) combinatorial objects we will use for space in Polynomial Calculus imply also total space lower bounds in Resolution, cf. Chapter 3.

One notable exception are *Tseitin formulas*: we are able, using Corollary 2.11 to prove quadratic total space lower bounds for such formulas built over random 3-regular graphs, hence completely answering to [4, Open question 2]. On the other hand, we do not know space lower bounds in Polynomial Calculus for such formulas, cf. Section 4.5 for more details.

The applications we chose to cover in Chapter 4 include: the *pigeonhole principles*, cf. Section 4.4, the *Tseitin formulas*, cf. Section 4.5, the *random k-CNF formulas*, cf. Section 4.8, and the *matching principles over graphs*, cf. Section 4.9.

[10] For instance random 3-CNF formulas with Δn clauses with Δ a constant, cf. Section 4.8.

2.8 Open problems

1. Can we prove non-trivial total space lower bounds for stronger proof systems such as bounded-depth Frege, Polynomial Calculus or Cutting Planes?

   We recall that for unrestricted Frege systems Alekhnovich et al. [4] showed a linear upper bound (in the size of the CNF formula being refuted) on total space. Regarding Cutting Planes some preliminary results on space are showed in [75].

2. Is there a family of k-CNF formulas in n variables and $\mathsf{poly}(n)$ clauses which have polynomial size Resolution refutations but which still require quadratic, or at least super-linear, total space in Resolution?

   Ben-Sasson and Wigderson [30] showed that if a k-CNF formula in n variables has a Resolution refutation of size S then it also has a refutation in which every clause has width at most $O(\sqrt{n \log S})$. Hence we cannot hope to use our arguments, which show large total space by finding many clauses of large width.

3. Is there any formula for which we can prove some trade-offs between total space and, for instance, size analogous to the ones we have for clause space in Resolution?

   Some of the size-related trade-offs in the literature are the following: [19, 20, 27, 29, 107, 109].

3

Space in Polynomial Calculus

3.1 Introduction

In this chapter we focus on the construction of the framework to prove monomial space lower bounds in Polynomial Calculus[1]. Then, the applications of this framework are collected in Chapter 4. The monomial space lower bound (Theorem 3.6) is one of our main contributions and builds on the definition of r-BG *family* (Definition 3.4). This definition is one of the main innovations of this work, since it reduces space lower bounds in algebraic proof systems to a combinatorial property on families of Boolean assignments. Our definition resembles the definition of k-dynamical satisfiability in [67] which was used to prove clause space lower bounds for Resolution. Likewise, the definition of r-BG families is analogous to the definition of winning strategies for the Duplicator in the k-existential Spoiler-Duplicator game which led to prove that in Resolution '*clause space is lower bounded by width* ', cf. [8] and Proposition 2.2.

The main contribution of this chapter is Theorem 3.6. It states the existence of a precise relation between r-BG families and refutation space in Polynomial Calculus, informally:

> *If there exists a non-empty r-BG family for an unsatisfiable CNF formula φ, then the monomial space needed to refute φ in Polynomial Calculus is at least $r/4$.*

We recall that Polynomial Calculus is defined over a field $\mathbb{F}$ but this result is independent from the characteristic of $\mathbb{F}$ and is valid over any field.

[1] The content of this chapter is based on [36, 37].

Before introducing all the machinery we will need to show the monomial space lower bounds we spend few words on why some simpler naïve approach has no hope to work. A naïve approach could consist, for example, in trying to mimic the approach followed for the *'clause space is lower bounded by width'* (cf. proof of Proposition 2.2). That, in our context, will lead us to use the following property: if an assignment α satisfies some polynomial p, that is such that $\alpha(p) = 0$, then $|\operatorname{dom}(\alpha)|$ is at most the number of monomials in p. Unfortunately this property is *false*: for instance $p = 1 - \prod_{i=1}^{r} x_i$ has just two monomials but any α such that $\alpha(p) = 0$ must have $|\operatorname{dom}(\alpha)| \geqslant r$.

This phenomenon is not happening if we consider families of assignments, consisting of *many* assignments with a combinatorial structure we called *flippable products*, cf. Section 3.3. For such families we can define a notion of *size* that is roughly lower bounding the number of monomials. This notion of size turns out to be roughly the logarithm of the number of assignments in the family. Then the monomial space lower bound we show, Theorem 3.6, has a similar structure of the proof of the *'clause space is lower bounded by width'*, cf. Proposition 2.2, but instead of using partial assignments and the r-AD families we use *flippable products* and families of such flippable products we called r-BG families, cf. Definition 3.4.

The main technical difficulty of this chapter is Lemma 3.9, the *Locality Lemma*, that is a generalisation of [4, Lemma 4.14].

3.2 Polynomial Calculus and space - definitions

Following [4], given a set of variables X we define $\overline{X} = \{\bar{x} \ : \ x \in X\}$, which we regard as a set of new formal variables with the intended meaning of $\bar{x}$ as $\neg x$. Given a field $\mathbb{F}$, the ring $\mathbb{F}[X \cup \overline{X}]$ is the ring of polynomials in the variables $X \cup \overline{X}$ with coefficients in $\mathbb{F}$. In order to fix the semantical meaning of the X and $\overline{X}$ variables, when given a set of polynomials P in $\mathbb{F}[X \cup \overline{X}]$ we will always suppose that $x^2 - x$ and $x + \overline{x} - 1$ for $x \in X$ are in P. Thorough this section, I denotes a proper ideal in $\mathbb{F}[X \cup \overline{X}]$ and, given a set of polynomials P, ideal(P) is the ideal generated by P in $\mathbb{F}[X \cup \overline{X}]$.

ideal(P)

tr

We use the following *standard translation*[2] (tr) of CNF formulas over a set of Boolean variables X into a set of polynomials in $\mathbb{F}[X \cup \overline{X}]$:

$$tr(\varphi) = \{tr(C) \ : \ C \in \varphi\} \cup \{x^2 - x, x + \overline{x} - 1 : x \in X\},$$

[2]There are many possible ways to encode k-CNF formulas into polynomials. The one we choose here is the standard one used in proof complexity since it produces a set of polynomials with degree exactly k, that is low degree polynomials if k is a constant.

where

$$tr(x) = \overline{x}, \qquad tr(\neg x) = x, \qquad tr(\bigvee_{i=1}^{n} \ell_i) = \prod_{i=1}^{n} tr(\ell_i).$$

Notice that we use the convention that a variable is *true* (resp. *false*) if it is assigned the value 1 (resp. 0) and a polynomial is *true* if it vanishes[3].

A set of polynomials P in $\mathbb{F}[X]$ is *contradictory* if and only if 1 is in the ideal generated by P, $1 \in \text{ideal}(P)$. Notice that a CNF formula φ is unsatisfiable if and only if $tr(\varphi)$ is a contradictory set of polynomials, that is they do not have a common root.

Polynomial Calculus (PCR) is an algebraic proof system defined in [4] for polynomials in $\mathbb{F}[X, \overline{X}]$, using the compact representation of CNF formulas into polynomials provided by the translation tr above. Starting from a set of initial contradictory polynomials P in $\mathbb{F}[X, \overline{X}]$, PCR allows to derive the polynomial 1 using one of the following inference rules

PCR

$$\frac{p \qquad p'}{\alpha p + \beta p'}\ \alpha, \beta \in \mathbb{F}, \qquad\qquad \frac{p}{vp} v \in X \cup \overline{X}, \tag{3.1}$$

and the further *Boolean axioms* $\{x^2 - x, x + \overline{x} - 1\}_{x \in X}$ to respect the intended meaning of the $\overline{X}$ variables.

A *derivation* of q from P is a sequence of polynomials $p_0, \ldots, p_\ell$ such that $p_\ell = q$ and each p_i is either an initial polynomial (either in P or a Boolean axiom) or it is inferred by previous polynomials in the sequence by the inference rules in equation 3.1. We call *refutation* a derivation of the polynomial 1.

PCR is a propositional proof system: its *soundness* come from the fact that if P derives q in PCR then $q \in \text{ideal}(P)$ and obviously q vanish on the variety $V(P)$, that is the set of zeroes of P. The *completeness* of PCR follows since PCR simulates Res[4], cf. Figure 3.1.

Similarly as what done in Resolution, in order to study space of proofs we rephrase the definition of derivations in PCR following the model proposed in [4, 66]. Given a set of initial polynomials P, a PCR *derivation of a polynomial* q *from* P, $P \vdash q$, is a sequence of sets of polynomials $(\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$, called *memory configurations*, such that: $\mathfrak{M}_0 = \emptyset$, $q \in \mathfrak{M}_\ell$ and for all $i \leqslant \ell$,

$P \vdash q$

memory configuration $\mathfrak{M}$

$$\mathfrak{M}_i \subseteq \mathfrak{M}_{i-1} \cup \{p\},$$

where p is one of the following:

[3] Other authors sometimes use a different encoding of CNF formulas into polynomials and consider a variable *true* if its value is 0.

[4] Completeness of PCR comes also as a corollary of Hilbert's Nullstellensatz [58] or by the Gröbner basis algorithm [55]. We do not require $\mathbb{F}$ to be algebraically closed due to the fact that we always consider sets of polynomials that include the Boolean axioms.

$$\dfrac{\dfrac{\dfrac{tr(C)\cdot\bar{x}}{\vdots}}{tr(C)tr(D)\cdot\bar{x}} \quad \dfrac{\dfrac{tr(D)\cdot x}{\vdots}}{tr(C)tr(D)\cdot x}}{tr(C)tr(D)\cdot\bar{x}+tr(C)tr(D)\cdot x} \quad \dfrac{\dfrac{x+\bar{x}-1}{\vdots}}{tr(C)tr(D)\cdot(x+\bar{x}-1)}}{tr(C)tr(D)}$$

Figure 3.1: Simulation of the rule $\frac{C\vee x,\quad D\vee\neg x}{C\vee D}$ in Polynomial Calculus

(Axiom Download) $p \in P$ or p is a Boolean axiom;

(Inference) p is some polynomial inferred from polynomials occurring in $\mathfrak{M}_{i-1}$ using the inference rules of PCR, cf. equation (3.1).

The results we show hold for *semantical* PCR *derivations with respect to an ideal* I. Those are are a generalisation of *semantical* PCR *derivations* as defined in [4]. A *semantical* PCR *derivation* correspond to setting $I = \{0\}$ in our definition.

$P \vdash_I q$ Let I be an ideal, a *semantical* PCR derivation of q from P *with respect to* I, $P \vdash_I q$, is a sequence of memory configurations $(\mathfrak{M}_0, \ldots, \mathfrak{M}_\ell)$ such that: $\mathfrak{M}_0 = \emptyset$, $q \in \mathfrak{M}_\ell$ and for all $i \leqslant \ell$, $\mathfrak{M}_i$ is obtained by $\mathfrak{M}_{i-1}$ by the following inference rule:

(Semantical Inference w.r.t. I) $\mathfrak{M}_i \subseteq \text{ideal}(\mathfrak{M}_{i-1} \cup \{p\}) + I$, for some $p \in P$. Where $\text{ideal}(\mathfrak{M}_{i-1} \cup \{p\}) + I$ is just the sum among ideals[5].

$\mathsf{MSpace}(S)$ **Definition 3.1** (Monomial Space)**.** *The* monomial space*,* $\mathsf{MSpace}(S)$*, of a set of polynomials* S *is the number of distinct monomials occurring in* S[6]*. The monomial space* $\mathsf{MSpace}(\pi)$ *of a semantical* PCR *refutation* π *is the maximal* $\mathsf{MSpace}^{sem}(P \vdash_I 1)$ *monomial space of a memory configuration in* π*. We denote by*

$$\mathsf{MSpace}^{sem}(P \vdash_I 1)$$

the minimal $\mathsf{MSpace}(\pi)$ *over all semantical* PCR *refutations* π *of* P*. When considering the* 0 *ideal we will simply write* $\mathsf{MSpace}^{sem}(P \vdash 1)$ *instead of* $\mathsf{MSpace}^{sem}(P \vdash_0 1)$*.*

The clause space upper bound in Resolution, cf. equation (2.1), and the fact that the simulation of Resolution in PCR in Figure 3.1 is efficient from the point of view of the monomials involved, imply that given an unsatisfiable CNF formula φ in n variables

$$\mathsf{MSpace}^{sem}(tr(\varphi) \vdash 1) \leqslant O(n). \tag{3.2}$$

[5]Given two ideals I, J in $\mathbb{F}[X \cup \overline{X}]$, $I + J = \{a + b : a \in I \wedge b \in J\}$.

[6]With monomial we mean a product of variables in $X \cup \overline{X}$.

Analogously to the total space in Resolution, the total space in PCR is defined as follows.

Definition 3.2 (Total Space)**.** *The* total space, $\mathsf{TSpace}_{\mathsf{PCR}}(S)$*, of a set of polynomials S is the total number of occurrences of variables in S. The total space $\mathsf{TSpace}_{\mathsf{PCR}}(\pi)$ of a semantical* PCR *refutation π is the maximal total space of a memory configuration in π. We denote by*

$\mathsf{TSpace}_{\mathsf{PCR}}(S)$

$\mathsf{TSpace}_{\mathsf{PCR}}(P \vdash \bot)$

$$\mathsf{TSpace}_{\mathsf{PCR}}(P \vdash 1)$$

the minimal $\mathsf{TSpace}_{\mathsf{PCR}}(\pi)$ over all PCR *refutations π of set of polynomials P.*

Given a contradictory CNF formula φ in n variables we have that

$$\mathsf{TSpace}_{\mathsf{PCR}}(tr(\varphi) \vdash 1) \leqslant O(n^2),$$

due to the monomial space upper bound in equation (3.2) and the fact that each monomial has at most n variables in it.

In this thesis we do not deal in detail on total space in Polynomial Calculus, for more details on it we refer to [4] and Section 3.5 for some open problems.

Given two ideals I, J, if $I \subseteq J$ then

$$\mathsf{MSpace}^{sem}(P \vdash_I 1) \geqslant \mathsf{MSpace}^{sem}(P \vdash_J 1).$$

Hence the lower bounds for $\mathsf{MSpace}^{sem}(P \vdash_I 1)$ hold also for $\mathsf{MSpace}^{sem}(P \vdash 1)$.

The lower bounds we will give rely on the combinatorial objects we call r-BG families whose definition is given in the following section.

3.3 r-BG families

Let X be a set of variables, $\overline{X} = \{\bar{x} : x \in X\}$ a set of fresh new variables and the intended meaning of $\bar{x}$ is $\neg x$, $\mathbb{F}$ is a fixed arbitrary field and I a proper ideal in the ring $\mathbb{F}[X \cup \overline{X}]$.

Given a polynomial p in $\mathbb{F}[X \cup \overline{X}]$ and an assignment α[7] we define the *application of α to p*, $\alpha(p)$, or equivalently $p|_\alpha$, as follows: substitute each variable x in p with the value $\alpha(x)$ if $x \in \mathrm{dom}(\alpha)$ and each variable $\bar{x}$ with $\alpha(\bar{x})$, or otherwise leave the variable untouched. Then simplify the result with the usual simplification rules including: $0 \cdot m \equiv 0$, $1 \cdot m \equiv m$ and $m - m \equiv 0$ where m is a term[8] in p. The notation $\alpha \vDash_I p$ means that $\alpha(p) \in I$. If F is a

$\alpha(p)$ $p|_\alpha$

$\alpha \vDash_I p$

[7] To avoid confusion we recall that in this thesis α is a (partial) assignment if $\alpha : X \cup \overline{X} \to \{0, 1, \star\}$.

[8] A *term* is a monomial with a coefficient from $\mathbb{F}$ in front of it.

I-consistent

family of partial assignments and P a set of polynomials, we write $F \vDash_I P$ if $\alpha \vDash_I p$ for each $\alpha \in F$ and $p \in P$. We say that F is *I-consistent* if $F \vDash_I I$, that is for every $p \in I$ and $\alpha \in F$, $\alpha(p) \in I$.

We consider partial assignments over $X \cup \overline{X}$ that are consistent with the ideal generated by $\{x + \bar{x} - 1 : x \in X\}$, that is that are respecting the intended meaning of the variables $\overline{X}$: $\alpha(x + \bar{x} - 1) = 0$ or $\alpha(x + \bar{x} - 1) = x + \bar{x} - 1$. In particular, given an assignment β over X, it is always possible to extend it to $X \cup \overline{X}$ respecting the previous property.

Notice that if φ is a CNF formula and α is a partial assignment satisfying φ, then $\alpha(tr(\varphi)) = 0$ and vice-versa. Moreover, given a set of partial assignments F, a set of polynomials P and an ideal I, if $F \vDash_I P$ then $F \vDash_I \text{ideal}(P)$.

$H_1 \otimes \cdots \otimes H_t$

Definition 3.3 (product-families)**.** *Given non-empty sets of assignments*[9] *$H_1, \ldots, H_t$ pairwise domain-disjoint, the* product-family *$\mathcal{H} = H_1 \otimes \cdots \otimes H_t$ is the following set of assignments*

$$\mathcal{H} = H_1 \otimes \cdots \otimes H_t = \{\alpha_1 \cup \cdots \cup \alpha_t : \alpha_i \in H_i\},$$

$\|\mathcal{H}\|$

or, if $t = 0$, $\mathcal{H} = \{\lambda\}$, a set containing just the empty partial assignment λ. We call the H_is the factors *of $\mathcal{H}$ and the* rank *of $\mathcal{H}$, $\|\mathcal{H}\|$, is the number of factors of $\mathcal{H}$ different from $\{\lambda\}$. The domain of $\mathcal{H}$ is* $\text{dom}(\mathcal{H}) = \bigcup_i \text{dom}(H_i)$.

The same set of assignments could correspond to many product-families: in particular each family of assignments can be seen as a product of just one single factor. When we write $\mathcal{H} = H_1 \otimes \cdots \otimes H_t$ it means that we fixed a particular representation of the set of assignment as a product: the representation has $H_1, \ldots, H_t$ as factors. We do not count the $\{\lambda\}$ factors in the rank since they do not carry any additional information: the set of assignments corresponding to $\mathcal{H} \otimes \{\lambda\}$ always coincide with $\mathcal{H}$. Given two product-families $\mathcal{H}$ and $\mathcal{H}'$ we

$\mathcal{H}' \sqsubseteq \mathcal{H}$

write $\mathcal{H}' \sqsubseteq \mathcal{H}$ if and only if each factor of $\mathcal{H}'$ different from $\{\lambda\}$ is also a factor of $\mathcal{H}$. In particular $\{\lambda\} \sqsubseteq \mathcal{H}$ for any $\mathcal{H}$.

flippable

In what follow we are interested in particular product-families such that each factor is *flippable*. A set of partial assignments F is *flippable* if and only if for all $v \in \text{dom}(F)$ there exists α and β in F such that $\alpha(v) = 1$ and $\beta(v) = 0$. We call a product-family whose factors are flippable a *flippable product-family* or simply a *flippable product*.

The following definition is the central definition of this chapter[10].

[9] We always suppose that the partial assignments are respecting the intended meaning of the variables in $\overline{X}$, that is $\alpha(x + \bar{x} - 1) = 0$ or $\alpha(x + \bar{x} - 1) = x + \bar{x} - 1$, hence a variable x is in $\text{dom}(H_i)$ if and only if $\bar{x}$ is in $\text{dom}(H_i)$.

[10] It is the analogue of *winning strategies* in [37, Definition 3.4].

Definition 3.4 (r-BG)**.** *Let P be a set of polynomials in $\mathbb{F}[X \cup \overline{X}]$ and I a proper ideal in $\mathbb{F}[X \cup \overline{X}]$. A family of flippable products $\mathscr{F}$ is a* r-BG family for P with respect to I *if and only if for every $\mathcal{H} \in \mathscr{F}$ the following three conditions hold:* r-BG

(CONSISTENCY) *$\mathcal{H}$ is I-consistent;*

(RESTRICTION) *for each $\mathcal{H}' \sqsubseteq \mathcal{H}$, $\mathcal{H}' \in \mathscr{F}$;*

(EXTENSION) *if $\|\mathcal{H}\| < k$, then for each $p \in P$ there exists a I-consistent flippable product $\mathcal{H}_p$, domain-disjoint from $\mathcal{H}$, such that*

1. *$\mathcal{H} \otimes \mathcal{H}_p \in \mathscr{F}$ and*
2. *$\mathcal{H} \otimes \mathcal{H}_p \vDash_I p$.*

Before proving the main results on r-BG families and monomial space we show how those families are related with the r-BK families from the previous chapter[11], cf. Definition 2.4.

Proposition 3.5. *Let φ be an unsatisfiable CNF formula. If there exists a non-empty r-BG family for $tr(\varphi)$ with respect to the 0 ideal then there exists a non-empty $(r-1)$-BK family for φ.*

Proof. Let $\mathscr{F}$ be a non-empty r-BG family for $tr(\varphi)$ with respect to the 0 ideal and let $\mathscr{L}$ be the set of all the assignments α over X that appear in some flippable product $\mathcal{H}$ of $\mathscr{F}$ of rank at most $r-1$, that is

$$\mathscr{L} = \{\alpha : \exists \mathcal{H} \in \mathscr{F}\ \alpha \in \mathcal{H} \wedge \|\mathcal{H}\| \leqslant r-1\}.$$

We claim that $\mathscr{L}$ is a $(r-1)$-BK family for φ. To prove the *consistency* property of $\mathscr{L}$ assume, by contradiction, that there exists an $\alpha \in \mathscr{L}$ such that α falsifies some clause C in φ. By definition of $\mathscr{L}$, there exists $\mathcal{H} \in \mathscr{F}$ such that $\alpha \in \mathcal{H}$ and $\|\mathcal{H}\| \leqslant r-1$. By the extension property of $\mathscr{F}$, there exists an $\mathcal{H}' \sqsupseteq \mathcal{H}$ such that $\mathcal{H}' \vDash_0 tr(C)$. In particular there exists some partial assignment $\beta \supseteq \alpha$ such that $\beta \vDash_0 tr(C)$. Thus $tr(\varphi)|_\beta = 0$ and hence $\beta(C) = 1$, which is impossible since α falsifies C.

For the *extension* property of $\mathscr{L}$, let $\alpha \in \mathscr{L}$ and $\beta \subseteq \alpha$ with $|\beta| < r-1$ and let x be a variable of φ not in $\mathrm{dom}(\alpha)$. Since $\beta \subseteq \alpha$ and $\alpha \in \mathscr{L}$ there must exists some $\mathcal{H} \in \mathscr{L}$ such that $\beta \in \mathcal{H}$ and $\|\mathcal{H}\| \leqslant |\beta| < r-1$. By the extension property of $\mathscr{F}$, there exists some flippable product $\mathcal{H}' \in \mathscr{F}$ such that

[11] In [31] we showed that r-BG families and the r-BGT families from [40] are related. Here we do the same for r-BK families.

$\mathcal{H}' \sqsupseteq \mathcal{H}$ and $\mathcal{H}' \vDash_0 x + \bar{x} - 1$. By taking restrictions in $\mathscr{L}$,we can suppose that $\|\mathcal{H}'\| = \|\mathcal{H}\| + 1 \leqslant r - 1$. Hence there exist $\beta_0, \beta_1 \in \mathcal{H}'$ extending β, setting x respectively to 0 and 1. By construction those assignments belongs to $\mathscr{L}$. □

Notice that the previous proof is sound also if we substitute the 0 ideal with ideal($\{x^2 - x : x \in X\}$) or with ideal($\{x + \bar{x} - 1 : x \in X\}$) and querying for $x^2 - x$ in the extension step. On the other hand, in general, we cannot substitute the 0 ideal with any ideal I containing ideal($\{x + \bar{x} - 1, x^2 - x : x \in X\}$). The reason is that we want to use some polynomials from $tr(\varphi)$ to enforce, in the extension property, the assignment of some precise variable. If we have some ideal I containing ideal($\{x + \bar{x} - 1, x^2 - x : x \in X\}$) then, for example the empty assignment λ, is such that $\{\lambda\} \vDash_I x^2 - x$ or $\{\lambda\} \vDash_I x + \bar{x} - 1$. So, in the proof of the previous proposition, an ideal I of this form cannot really enforce the assignment of some precise variable from X.

3.4 Monomial space lower bounds

The main property of r-BG families is that they can be used to prove monomial space lower bounds in PCR and in the stronger system *semantic* PCR with respect to a proper ideal I. Indeed, we have the following result[12].

Theorem 3.6. *Let P be a contradictory set of polynomials in the ring of polynomials $\mathbb{F}[X \cup \overline{X}]$[13], I a proper ideal in $\mathbb{F}[X \cup \overline{X}]$ and $r \geqslant 1$ an integer. Suppose that there exists a non-empty r-BG family $\mathscr{F}$ for P with respect to the ideal I. Then*

$$\mathsf{MSpace}^{sem}(P \vdash_I 1) \geqslant r/4.$$

In the previous Theorem we do not make any assumption on the structure of the set of initial polynomials P. If we have some assumptions on P it is possible to have an analogous result requiring the existence of a non-empty r-BG family just for a subset of P. This in particular can be useful when in P we have some monomials of high initial degree[14].

Theorem 3.7. *Let $P = P_1 \cup P_2$ a contradictory set of polynomials in the ring $\mathbb{F}[X \cup \overline{X}]$. Suppose that the following conditions hold:*

1. *there exists a non-empty r-BG family $\mathscr{F}$ for P_1 with respect to the 0 ideal;*

[12] This result is analogous to [36, Theorem 1] and [37, Theorem 3.5].

[13] Recall that given a set of polynomials in $\mathbb{F}[X \cup \overline{X}]$ we always have that $\{x + \bar{x} - 1\}_{x \in X}$ in P to fix the semantic meaning of the variables in $\overline{X}$.

[14] This result is analogous to [36, Theorem 2] and [37, Theorem 3.6].

2. *every polynomial in* P_2 *is a* monomial *such that for each* $m \in P_2$ *and for each* $\mathcal{H} \in \mathscr{F}$ *with* $\|\mathcal{H}\| < r$ *then either there exists a variable in* m *not in* $\mathrm{dom}(\mathcal{H})$ *or* $\mathcal{H} \vDash_0 m$.

Then

$$\mathsf{MSpace}^{sem}(P \vdash 1) \geqslant r/4.$$

In order to prove Theorem 3.6 and Theorem 3.7 we introduce the definition of 2-*merge*. On a very high level a 2-merge on a product family $\mathcal{H}$ is a new product family $\mathcal{Z}$ whose factors are obtained combining, *'merging'*, disjoint pairs of factors from $\mathcal{H}$.

Definition 3.8 (2-merge)**.** *Let* $\mathcal{H} = H_1 \otimes \cdots \otimes H_t$ *be a product-family. A* 2-merge *on* $\mathcal{H}$ *is a product-family* $\mathcal{Z} = Z_{J_1} \otimes \cdots \otimes Z_{J_r}$*, where* $J_1, \ldots, J_r$ *are pairwise disjoint subsets of* $[t]$ *of size at most* 2*,* $Z_{J_i} \subseteq \bigotimes_{j \in J_i} H_j$ *and* $\mathcal{Z} \restriction_{\mathrm{dom}(H_j)} = H_j$ *for all* $j \in [t]$*. Notice that if* $\mathcal{H}$ *is flippable product family then* $\mathcal{Z}$ *is also a flippable product family.* 2-merge

As in [4] a key property in our monomial space lower bound proofs is a *Locality Lemma*[15], cf. Lemma 3.9. Informally, such lemma asserts that if a set S of polynomials is satisfiable by a 2-merge on a product family $\mathcal{H}$, then it is possible to build a new 2-merge $\mathcal{Z}'$ on a new product-family $\mathcal{H}'$ such that $\mathcal{Z}'$ still satisfies S and $\mathcal{H}' \sqsubseteq \mathcal{H}$ has rank bounded by the number of distinct monomials in S.

Lemma 3.9 (Locality Lemma)**.** *Let* I *be an ideal in* $\mathbb{F}[X \cup \overline{X}]$*,* S *a set of polynomials in* $\mathbb{F}[X \cup \overline{X}]$*,* $\mathcal{H}$ *a non-empty flippable product and* $\mathcal{Z}$ *a* 2*-merge on* $\mathcal{H}$ *such that* $\mathcal{Z} \vDash_I S$*. Then there exist a flippable product* $\mathcal{H}' \sqsubseteq \mathcal{H}$ *and a non-empty* 2*-merge* $\mathcal{Z}'$ *on* $\mathcal{H}'$ *such that:* $\mathcal{Z}' \vDash_I S$ *and* $\|\mathcal{H}'\| \leqslant 4 \cdot \mathsf{MSpace}(S)$*.*

We postpone the technical proof of this lemma to Section 3.4.2 and we prove now Theorem 3.6.

Proof of Theorem 3.6. Let $\Pi = (\mathfrak{M}_0, \ldots, \mathfrak{M}_s)$ be a refutation of P in semantical PCR with respect to the ideal I and assume, for sake of contradiction, that $\mathsf{MSpace}(\Pi) < r/4$. Suppose we have the following property.

> **Claim 3.10.** *For* $i = 0, \ldots, s$*, there exist a non-empty flippable product* $\mathcal{H}_i \in \mathscr{F}$ *and a non-empty* 2*-merge* $\mathcal{Z}_i$ *on* $\mathcal{H}_i$ *such that* $\mathcal{Z}_i \vDash_I \mathrm{ideal}(\mathfrak{M}_i) + I$*.*

[15] This Lemma is a generalization of analogue results in [4, 36, 70]. The way we present it is based on [37].

The previous claim immediately implies a contradiction: when $i = s$ it implies that there exists some assignment $\alpha \in \mathcal{Z}_s$ such that for every polynomial $p \in \text{ideal}(\mathfrak{M}_s) + I$, $\alpha(p) \in I$. However $1 \in \mathfrak{M}_s$, hence $1 = \alpha(1) \in I$ which instead, by assumption, is a proper ideal in $\mathbb{F}[X \cup \overline{X}]$.

By induction on $i = 0, \ldots, s$, we prove that Claim 3.10 follows from the existence of the r-BG family $\mathscr{F}$ and from the hypothesis that $\mathsf{MSpace}(\Pi) < r/4$. For the base case $i = 0$, set $\mathcal{H}_0 = \{\lambda\} \in \mathscr{F}$ and $\mathcal{Z}_0 = \mathcal{H}_0$. Then, trivially, $\mathcal{Z}_0 \vDash_I \text{ideal}(\mathfrak{M}_0) + I = I$.

For the inductive step, let $\mathfrak{M}_{i+1} \subseteq \text{ideal}(\mathfrak{M}_i \cup \{p\}) + I$ with $p \in P$. By the Locality Lemma (cf. Lemma 3.9), used with parameters $\mathcal{H} = \mathcal{H}_i$, $\mathcal{Z} = \mathcal{Z}_i$ and $S = \mathfrak{M}_i$, we obtain a $\mathcal{H}' \sqsubseteq \mathcal{H}_i$ and a non-empty 2-merge $\mathcal{Z}'$ on $\mathcal{H}'$ such that $\mathcal{Z}' \vDash_I \mathfrak{M}_i$ and $\|\mathcal{H}'\| \leqslant 4\mathsf{MSpace}(\mathfrak{M}_i)$. Observe that, as $\mathscr{F}$ is an r-BG family, then by the *restriction* property of $\mathscr{F}$, $\mathcal{H}' \in \mathscr{F}$, since $\mathcal{H}' \sqsubseteq \mathcal{H}_i$ and $\mathcal{H}_i \in \mathscr{F}$. Moreover, by the *$I$-consistency* property of $\mathscr{F}$, $\mathcal{H}'$ is I-consistent and then $\mathcal{Z}'$ is I-consistent, hence $\mathcal{Z}' \vDash_I \text{ideal}(\mathfrak{M}_i) + I$.

Since, by hypothesis, $\mathsf{MSpace}(\mathfrak{M}_i) < r/4$, then $\|\mathcal{H}'\| < r$ and, by the *extension* property applied to $\mathcal{H}'$ and p, there exists an I-consistent flippable product $\mathcal{H}_p$, domain-disjoint from $\mathcal{H}'$, such that $\mathcal{H}_{i+1} = \mathcal{H}' \otimes \mathcal{H}_p \vDash_I p$ and $\mathcal{H}_{i+1} \in \mathscr{F}$. Set $\mathcal{Z}_{i+1} = \mathcal{Z}' \otimes \mathcal{H}_p$. Since $\mathcal{Z}'$ is a 2-merge on $\mathcal{H}'$ and by the definitions of $\mathcal{Z}_{i+1}$ and $\mathcal{H}_{i+1}$, then $\mathcal{Z}_{i+1}$ is a 2-merge. Finally, the property that $\mathcal{Z}_{i+1} \vDash_I \text{ideal}(\mathfrak{M}_{i+1}) + I$, follows because:

- $\mathcal{Z}_{i+1} \vDash_I \text{ideal}(\mathfrak{M}_i) + I$, since $\mathcal{Z}' \vDash_I \text{ideal}(\mathfrak{M}_i) + I$ and $\mathcal{H}_p$ is I-consistent, and
- $\mathcal{Z}_{i+1} \vDash_I p$, since $\mathcal{H}_{i+1} = \mathcal{H}_i \otimes \mathcal{H}_p \vDash_I p$ and $\mathcal{Z}_{i+1} \subseteq \mathcal{H}_{i+1}$ as $\mathcal{Z}_{i+1}$ is a 2-merge on $\mathcal{H}_{i+1}$. □

We adapt the previous proof to prove Theorem 3.7 but, before doing that, we show an example of a 2-merge that will be useful both in the proof of Theorem 3.7 and in the proof of the Locality Lemma, cf. Lemma 3.9.

Example 3.11. *Let m be a monomial and $\mathcal{H} = H_1 \otimes H_2$ be a flippable product such that* $\text{var}(m) \cap \text{dom}(H_i) \neq \emptyset$ *for $i = 1, 2$. Let $O_{m,i} = \{\alpha \in H_i : \alpha \vDash_0 m\}$. We have that*

$$\mathcal{Z} = Z_{\{1,2\}} = (O_{m,1} \otimes H_2) \cup (H_1 \otimes O_{m,2}) = \{\alpha \in H_1 \otimes H_2 : \alpha \vDash_0 m\}$$

is a 2-merge on $\mathcal{H}$. $\mathcal{Z}$ is a product-family since it has only one factor: $Z_{\{1,2\}}$.

Proof of Theorem 3.7. The proof is essentially the same as in Theorem 3.6 hence we use the same notations and, for sake of contradiction, we assume that

$\mathsf{MSpace}(\Pi) < r/4$. To prove again Claim 3.10, we have to prove the inductive step $\mathfrak{M}_{i+1} \subseteq \text{ideal}(\mathfrak{M}_i \cup \{m\})$ only when $m \in P_2$, since in the other cases the proof is the same as in Theorem 3.6. By the Locality Lemma, cf. Lemma 3.9, used with parameters $\mathcal{H} = \mathcal{H}_i$, $\mathcal{Z} = \mathcal{Z}_i$ and $S = \mathfrak{M}_i$, we find a $\mathcal{H}' \in \mathscr{F}$ and a non-empty 2-merge $\mathcal{Z}'$ of $\mathcal{H}'$ such that $\mathcal{Z}' \vDash_0 \mathfrak{M}_i$ and

$$\|\mathcal{H}'\| \leqslant 4\mathsf{MSpace}(\mathfrak{M}_i) \leqslant 4(r/4 - 1) \leqslant r - 4.$$

Let $\mathfrak{M}' = \mathfrak{M}_i \cup \{m\}$, with $m \in P_2$. Then, by hypothesis (2) of the theorem, either $\mathcal{H}' \vDash_0 m$ or there exists a variable $x \in \text{var}(m) \setminus \text{dom}(\mathcal{H}')$. In the first case just set $\mathcal{H}_{i+1} = \mathcal{H}'$ and $\mathcal{Z}_{i+1} = \mathcal{Z}'$. Otherwise, by the *extension* property of $\mathscr{F}$ applied on $\mathcal{H}'$ and $x + \bar{x} - 1$, there exists a flippable product $\mathcal{H}_x$ domain-disjoint from $\mathcal{H}'$ such that $\mathcal{H}' \otimes \mathcal{H}_x \in \mathscr{F}$ and $\mathcal{H}' \otimes \mathcal{H}_x \vDash_0 x + \bar{x} - 1$. Since $x \notin \text{dom}(\mathcal{H}')$, then $x, \bar{x} \in \text{dom}(\mathcal{H}_x)$ and using the closure of $\mathscr{F}$ under $\sqsubseteq$, we can assume that $\mathcal{H}_x$ is just one factor containing x in its domain. Hence $\|\mathcal{H}' \otimes \mathcal{H}_x\| < r$ and then either $\mathcal{H}' \otimes \mathcal{H}_x \vDash_0 m$ or there exists another variable $y \in \text{var}(m)$ but not in $\text{dom}(\mathcal{H}' \otimes \mathcal{H}_x)$. In the first case set $\mathcal{H}_{i+1} = \mathcal{H}' \otimes \mathcal{H}_x$ and $\mathcal{Z}_{i+1} = \mathcal{Z}' \otimes \mathcal{H}_x$. In the second case, by the *extension* property of $\mathscr{F}$ applied to $\mathcal{H}' \otimes \mathcal{H}_x$ and $y + \bar{y} - 1$, we get a flippable product $\mathcal{H}_y$ domain-disjoint from $\mathcal{H}' \otimes \mathcal{H}_x$ such that $\mathcal{H}' \otimes \mathcal{H}_x \otimes \mathcal{H}_y \in \mathscr{F}$ and $\mathcal{H}' \otimes \mathcal{H}_x \otimes \mathcal{H}_y \vDash_0 y + \bar{y} - 1$. Exactly as above for $x, \bar{x}$, we have that $y, \bar{y} \in \text{dom}(\mathcal{H}_y)$. Set $\mathcal{H}_{i+1} = \mathcal{H}' \otimes \mathcal{H}_x \otimes \mathcal{H}_y$ and $\mathcal{Z}_{i+1} = \mathcal{Z}' \otimes \{\alpha \in H_x \otimes H_y : \alpha \vDash_0 m\}$. By what we observed in the Example 3.11, $\mathcal{Z}_{i+1}$ is a 2-merge on $\mathcal{H}_{i+1}$ and, since $\mathcal{Z}_{i+1} \vDash_0 \mathfrak{M}_i \cup \{m\}$, $\mathcal{Z}_{i+1} \vDash_0 \mathfrak{M}_{i+1}$. □

3.4.1 A Hall's theorem for ∨-matchings

In this section we give a first generalization of matchings in bipartite graphs. A further generalization will be defined in Section 4.6.

Let G be a bipartite graph with bipartition (L, U). From G we create an auxiliary bipartite graph G' with bipartition (L', U), where $L' = \{v^0, v^1\}_{v \in L}$ and there is an edge $\{v^b, w\}$ in $E(G')$ if and only if $\{v, w\} \in E(G)$.

matching

A collection of vertex-disjoint edges in $E(G)$ is a *matching* in G. A collection of edges M in G is a V-*matching* in G if and only if there exists a collection of edges M' that is a matching in G' such that

$\mathsf{V} - matching$

$$M = \{\{v, w\} : \exists b \in \{0, 1\}\ \{v^b, w\} \in M'\}.$$

This definition is essentially the same definition of V-matching we will see in Section 4.6[16].

[16] The difference is that a V-matching in Section 4.6 could include singleton vertices from U. Hence, from the point of view of the vertices covered in L, those notions are perfectly equivalent.

It is well known that, given a graph G with bipartition (L, U), the existence of a matching in G covering L is related with the expansion properties of G. In particular we have the following result.

Theorem 3.12 (Hall's Theorem)**.** *Let G be a bipartite graph with bipartition (L, U). The following are equivalent:*

1. *for each subset A of L, $|N_G(A)| \geqslant |A|$,*
2. *there exists a matching in G covering L.*

An easy corollary of the previous result is the following result on V-matchings proved by Alekhnovich et al. [4]. In Section 4.6 we prove analogue result for what we call VW-matchings, cf. Theorem 4.11.

Theorem 3.13 (Alekhnovich et al. [4])**.** *Let G be a bipartite graph with bipartition (L, U). The following are equivalent:*

1. *for each subset A of L, $|N_G(A)| \geqslant 2|A|$,*
2. *there exists a V-matching in G covering L.*

Proof. Clearly (2) imply (1). For the other implication, let G' be the auxiliary graph with bipartition (L', U) used to define the V-matchings. Now, G' is such that for each subset A of L', $|N_{G'}(A)| \geqslant |A|$. Hence, by Hall's Theorem, there exists a matching M' covering L'. Let

$$M = \{\{v, w\} : \exists b \in \{0, 1\}\ \{v^b, w\} \in M'\},$$

clearly M is a V-matching and it covers L. □

3.4.2 A Locality Lemma

This section entirely contains the proof of Lemma 3.9, restated below for convenience of the reader.

Restated Lemma 3.9 (Locality Lemma)**.** *Let I be an ideal in $\mathbb{F}[X \cup \overline{X}]$, S a set of polynomials in $\mathbb{F}[X \cup \overline{X}]$, $\mathcal{H}$ a non-empty flippable product and $\mathcal{Z}$ a 2-merge on $\mathcal{H}$ such that $\mathcal{Z} \vDash_I S$. Then there exist a flippable product $\mathcal{H}' \sqsubseteq \mathcal{H}$ and a non-empty 2-merge $\mathcal{Z}'$ on $\mathcal{H}'$ such that: $\mathcal{Z}' \vDash_I S$ and $\|\mathcal{H}'\| \leqslant 4 \cdot \mathsf{MSpace}(S)$.*

A visual hint for the notations used in this proof can be found in Figure 3.2 on the next page.

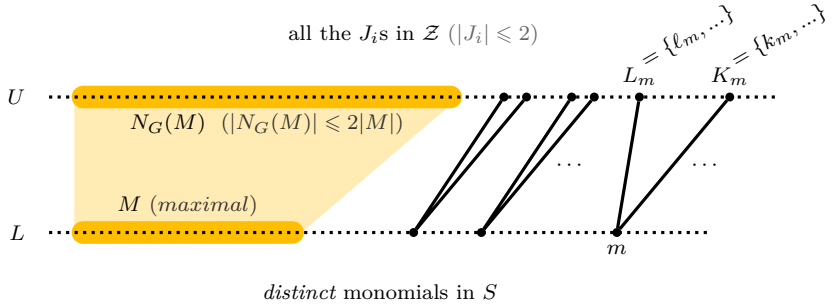


Figure 3.2: A locality lemma

Proof. Let $\mathcal{H} = H_1 \otimes \ldots \otimes H_t$ and $\mathcal{Z} = Z_{J_1} \otimes \ldots \otimes Z_{J_r}$. Let G be the following bipartite graph with bipartition (L, U): the *lower part* of G, L, is indexed by the set of all *distinct* monomials in S, the *upper part* of G, U, is indexed by the set $\{J_1, \ldots, J_r\}$ and there is an edge $(m, J_i) \in E(G)$ if and only if a variable of m appears in $\mathrm{dom}(Z_{J_i})$. For a set $M \subseteq L$ let $N(M)$ be the set of the neighbours of M in G and let $\mathcal{H}_M$ and $\mathcal{Z}_M$ be the following two flippable products:

$$\mathcal{Z}_M = \bigotimes_{J_i \in N(M)} Z_{J_i}, \qquad \mathcal{H}_M = \bigotimes_{J_i \in N(M)} \bigotimes_{j \in J_i} H_j.$$

Let M be a set of maximal size in L such that $|N(M)| \leqslant 2|M|$. Let $M^c = L \setminus M$. By maximality of M, for each $A \subseteq M^c$, $|N(A) \setminus N(M)| \geqslant 2|A|$. Hence, by Theorem 3.13, there is a V-matching F covering M^c and F is in the subgraph of G induced by $M^c \cup (U \setminus N(M))$.

For each monomial m in M^c, consider the upper part of the connected component F_m of F covering m and let $U(F_m) = \{L_m, K_m\}$ be such upper part, where $L_m, K_m \in \{J_1, \ldots, J_r\}$. By definition of G, there is a variable x both in $\mathrm{var}(m)$ and in $\mathrm{dom}(Z_{L_m})$. Let $\ell_m \in L_m$ such that x in $\mathrm{dom}(H_{\ell_m})$ (if there are more than one possible ℓ_m, we choose one). Analogously for $k_m \in K_m$. Define the product-family $\mathcal{H}'$ as

$$\mathcal{H}' = \mathcal{H}_M \otimes \bigotimes_{m \in M^c} H_{\ell_m} \otimes H_{k_m}.$$

Clearly $\mathcal{H}' \sqsubseteq \mathcal{H}$ and hence it is a flippable product. The rank of $\mathcal{H}'$ is $\|\mathcal{H}'\| = \|\mathcal{H}_M\| + 2|M^c|$. Since $|N(M)| \leqslant 2|M|$, and since the J_i's are of size at most 2, we have that $\|\mathcal{H}_M\| \leqslant 4|M|$. Hence, putting all together,

$$\|\mathcal{H}'\| \leqslant 4|L| = 4 \cdot \mathsf{MSpace}(S).$$

The construction of $\mathcal{Z}'$ goes as follows. Let $O_{m,i} = \{\alpha \in H_i : \alpha(m) = 0\}$. Observe that if a variable x of m is in $\mathrm{dom}(H_i)$ then $O_{m,i}$ is non-empty since

H_i is flippable and hence there is always an assignment in H_i setting x to satisfy m, that is setting m to 0. As in Example 3.11, let

$$Z_{\{\ell_m,k_m\}} = (O_{m,\ell_m} \otimes H_{k_m}) \cup (H_{\ell_m} \otimes O_{m,k_m}).$$

Let $\mathcal{Z}'$ be

$$\mathcal{Z}' = \mathcal{Z}_M \otimes \bigotimes_{m \in M^c} Z_{\{\ell_m,k_m\}}.$$

It is straightforward to see that $\mathcal{Z}'$ is a 2-merge on $\mathcal{H}'$ hence it is remaining to prove only that $\mathcal{Z}' \vDash_I S$. First we claim that $\mathcal{Z}' \subseteq \mathcal{Z} \restriction_{\mathrm{dom}(\mathcal{H}')}$.

Claim 3.14. $\mathcal{Z}' \subseteq \mathcal{Z} \restriction_{\mathrm{dom}(\mathcal{H}')}$.

Proof. By construction, $\mathcal{Z}_M = \mathcal{Z} \restriction_{\mathrm{dom}(\mathcal{H}_M)}$ hence we have to prove that for each $m \in M^c$,

$$\mathcal{Z}' \restriction_{\mathrm{dom}(H_{\ell_m}) \cup \mathrm{dom}(H_{k_m})} \subseteq \mathcal{Z} \restriction_{\mathrm{dom}(H_{\ell_m}) \cup \mathrm{dom}(H_{k_m})} .$$

This follows immediately from the following chain of inequalities

$$\begin{aligned} \mathcal{Z}' \restriction_{\mathrm{dom}(H_{\ell_m}) \cup \mathrm{dom}(H_{k_m})} &= Z_{\{\ell_m,k_m\}} && (3.3)\\ &\subseteq H_{\ell_m} \otimes H_{k_m} && (3.4)\\ &= \mathcal{Z} \restriction_{\mathrm{dom}(H_{\ell_m}) \cup \mathrm{dom}(H_{k_m})} . && (3.5) \end{aligned}$$

The equality (3.3) is by definition and the containment in (3.4) follows by construction. The equality (3.5) follow since, by definition of $\mathcal{Z}$, $\mathcal{Z} \restriction_{\mathrm{dom}(H_j)} = H_j$ and L_m and K_m are disjoint sets. □

To prove that $\mathcal{Z}' \vDash_I S$, let $\alpha \in \mathcal{Z}'$. As $\mathcal{Z}' \subseteq \mathcal{Z} \restriction_{\mathrm{dom}(\mathcal{H}')}$, there exists $\beta \in \mathcal{Z}$ extending α by setting variables not appearing in $\mathrm{dom}(\alpha)$ and hence not appearing in any $m \in M$. Hence, by construction, if $m \in M^c$, then $0 = \alpha(m) = \beta(m)$ and if $m \in M$ then $\alpha(m) = \beta(m)$. Then, α and β give the same value to the monomials in S and, by hypothesis, $\beta \vDash_I S$, hence $\alpha \vDash_I S$. □

3.5 Open problems

1. Given a set of contradictory polynomials P of bounded degree is it true that

$$\mathsf{MSpace}_{\mathsf{PCR}}(P \vdash 1) = \Omega(\mathsf{degree}_{\mathsf{PCR}}(P \vdash 1))?$$

2. Given an unsatisfiable set of polynomials P, is it true that

$$\mathsf{MSpace}^{sem}(P \vdash 1) = \Theta(\mathsf{MSpace}_{\mathsf{PCR}}(P \vdash 1))?$$

   This very same question was asked for PCR (and more general proof systems) already in [4, Open question n. 4].

3. All the questions about total space in [4] are still open. In particular: is there any CNF formula φ in n variables and with $\mathsf{poly}(n)$ clauses such that

$$\mathsf{TSpace}_{\mathsf{PCR}}(tr(\varphi) \vdash 1) = \omega(n)?$$

For more concrete open problems on explicit unsatisfiable CNF formulas see Section 4.10.

4

Space lower bounds: applications

This chapter consists of several sections, each one collecting some space-related results for a particular class of formulas, in particular the monomial and the total space lower bounds. We get such results as applications of the main results of the previous chapters, cf. respectively Theorem 3.6 and Theorem 2.5.

The rationale behind the organisation of this chapter is to start with the less complicated applications, CT_n formulas and PHP_n^m formulas, and then move to more involved applications, for instance random k-CNF formulas. The results after Section 4.6 and Section 4.7 depend on these sections. Otherwise the content of any two sections is largely independent. For instance the reader only interested in reading the full proof of the monomial space lower bound for random k-CNF formulas, that is in Section 4.8, has just to refer to Section 4.6, Section 4.7 and Section 4.8. The Table 4.1 recaps all the monomial and total space results we show in this chapter and in which section they are discussed in detail.

4.1 Some history + credits

All the monomial space lower bounds known before [36] were proven in two works: [4] and [70]. The first paper showed monomial space lower bounds for CT_n and PHP_n^m; the second paper improved it to different encodings of PHP_n^m of bounded initial width: the formulas bitPHP_n and xorPHP_n^m, cf. Section 4.4.2 and Section 4.4.3. In [36] we gave a general framework that allowed to prove

Table 4.1: Recap of formulas and space lower bounds

Formula φ	$N = \lvert \mathrm{var}(\varphi) \rvert$	$\mathsf{width}(\varphi)$	$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot)$	$\mathsf{MSpace}^{sem}(tr(\varphi) \vdash 1)$	§
CT_n	$N = n$	n	$\Theta(N^2) = \Theta(n^2)$	$\Theta(N) = \Theta(n)$	4.3
PHP_n^{n+1}	$N = n(n+1)$	n	$\Omega(N) = \Omega(n^2)$	$\Omega(\sqrt{N}) = \Omega(n)$	4.4.1
bitPHP_n	$N = n \cdot \log n$	$2 \log n$	$\tilde{\Omega}(N^2) = \Omega(n^2)$	$\tilde{\Omega}(N) = \Omega(n)$	4.4.2
xorPHP_n^{n+1}	$N = n(n+1)$	4	$\Omega(N) = \Omega(n^2)$	$\Omega(\sqrt{N}) = \Omega(n)$	4.4.3
$\mathsf{Tseitin}(G, \sigma)$ where G is a graph					
4-regular random	N	4	$\Omega(N^2)$	$\Omega(\sqrt{N})$	4.5
3-regular expander	N	3	$\Omega(N^2)$	?	4.5
(n, k, Δ)-random CNF (Δ const., $k \geqslant 3$)	$N = n$	k	$\Theta(N^2) = \Theta(n^2)$	$\Theta(N) = \Theta(n)$	4.8
G-PHP (G bipartite of degree $d \geqslant 3$)	$N = n$	d	$\Theta(N^2) = \Theta(n^2)$	$\Theta(N) = \Theta(n)$	4.9

more monomial space lower bounds, in particular for the (n, k, Δ)-random CNF formulas and for the graph pigeonhole principle, G, over an expander bipartite graph G, cf. Section 4.2 for more details. We proved the monomial space lower bound for (n, k, Δ)-random CNF formulas in two papers: for $k \geqslant 4$ in [36] and for $k = 3$ in [31].

The result for *Tseitin formulas* over 4-regular random graphs was obtained by Filmus et al. [68] as an application of [36, Theorem 1], which is a preliminary version of Theorem 3.6. More details are given in Section 4.5.

Regarding the total space lower bounds, before [40] the only total space lower bounds in Resolution known were the ones for CT_n and PHP_n^m from [4]. In [40] we introduced a framework to prove total space lower bounds in Resolution and, as an application, we proved the first superlinear total space lower bound for a formula with polynomially many clauses. That is for bitPHP cf. Section 4.4.2.

The quadratic total space lower bound in Resolution for (n, k, Δ)-random CNF fomulas was proven in two papers: for $k \geqslant 4$ in [40] and for $k = 3$ in [31]. Both results rely on constructions developed in [36], cf. Section 4.6. The quadratic total space lower bound for Tseitin formulas over 3-regular expander graphs is an original contribution of this thesis and rely on Theorem 2.5 that is a strengthening of [40, Theorem 2.4]. For more details on the relation between Theorem 2.5 and [40, Theorem 2.4] we refer to Section 2.1.

4.2 Main results and techniques

The main result of this chapter is the following monomial space and total space (in Resolution) lower bound for random CNF formulas.

Let k a positive integer and Δ a positive real number, an (n, k, Δ)-*random CNF formula* φ is a k-CNF formula with n variables and Δn clauses picked uniformly at random from the set of all CNF formulas in the variables $\{x_1, \ldots, x_n\}$ which consist of exactly Δn clauses, each clause containing exactly k literals and no variable appears twice in a clause. For large enough Δ (depending on k), an (n, k, Δ)-random CNF formula is unsatisfiable with high probability, cf. Section 4.8 for more details. We then have with high probability for large n that

(n, k, Δ)-random CNF

$$\mathsf{MSpace}^{sem}(tr(\varphi) \vdash 1) \geqslant \Omega(n) \tag{4.1}$$

and

$$\mathsf{TSpace}_{\mathsf{Res}}(tr(\varphi) \vdash \bot) \geqslant \Omega(n^2), \tag{4.2}$$

cf. Theorem 4.36. An analogue result holds for the *matching principle over a graph* G, G-PHP, where G is an expander bipartite graph with left degree at least 3, cf. Section 4.9. Both the lower bounds in equation (4.1) and (4.2), asymptotically match the trivial upper bounds for monomial space, cf. Section 3.2, and total space (in Resolution), cf. Section 2.3. For the results we have on *Tseitin formulas* we refer to Section 4.5.

In what follow we want to give an informal intuition on the constructions and techniques that we use to prove the lower bounds for (n, k, Δ)-random CNF formulas and G-PHP. The proof of the monomial space lower bound for (n, k, Δ)-random CNF formulas, cf. Theorem 4.36, consists on building suitable $\Omega(n)$-BG families and then apply the main tools to prove space lower bounds from the previous chapters, that is Theorem 3.6 and Theorem 2.5.

The path we choose to build the $\Omega(n)$-BG families for the (n, k, Δ)-random CNF formulas is not entirely direct and it is not tailored specifically to such formulas. The reason is that, to large extent, the same construction apply to the graph pigeonhole principle , cf. Section 4.9. Moreover, the way we choose to present the construction of the $\Omega(n)$-BG families is intended to highlight the structural properties of the (n, k, Δ)-random CNF we use to obtain a $\Omega(n)$-BG family.

$\mathscr{C}$-matchings We start generalizing the concept of *matchings* in bipartite graphs to what we called $\mathscr{C}$-matchings, cf. Section 4.6. Intuitively a *$\mathscr{C}$-matching* in a bipartite graph G is a collection of vertex-disjoint subgraphs of G isomorphic to some graph from the collection of graphs $\mathscr{C}$. We will construct such $\mathscr{C}$-matchings in the case of random k-CNF formula φ in the *clauses-variables adjacency graph* associated to φ, cf. Definition 4.34. Informally, it is a bipartite graph with on one side the clauses of φ and on the other side the variables of φ

$\mathscr{C}$ − *matching*

and edges if a variable appear in a clause. That is we have that the *semantics* of the vertices in the two elements of the bipartition of our bipartite graphs is different and our $\mathscr{C}$-matchings have to respect such semantical difference.

bipartite graphs

In order to do so, from Section 4.6, we assume that the *bipartite graphs* are subgraphs of the infinite bipartite graph B with vertex set $\mathbb{N} \times \{0,1\}$ and such that $\{(n,b),(m,b')\} \in E(B)$ if and only if $b \neq b'$. Given a bipartite graph G we call $V(G) \cap \{(n,0) \ : \ n \in \mathbb{N}\}$ the *lower part* of G, $L(G)$, and similarly $V(G) \cap \{(n,1) \ : \ n \in \mathbb{N}\}$ is the *upper part* of G, $U(G)$.

$L(G)$

$U(G)$

Then our main interest is for two particular cases of $\mathscr{C}$-matchings: the V-matchings and the VW-matchings. We already saw a version of the V-matchings in Section 3.4.1 and the VW-matchings are just particular $\mathscr{C}$-matchings in which each connected component looks like a 'V', a 'W' or a singleton from $U(G)$. In Section 3.4.1 we observed that a version of Hall's theorem holds for V-matchings. Here we prove a version of Hall's Theorem that holds for VW-matchings, cf. Theorem 4.11[1].

Cover Games In Section 4.6.2, we define a game over bipartite graphs using $\mathscr{C}$-matchings that is associated with r-BG families. The *Cover Game*, $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$, is a game between two players, Choose (he) and Cover (she), on a bipartite graph G. At each step i of the game the players maintain a $\mathscr{C}$-matching F_i in G. They start with the empty $\mathscr{C}$-matching and at step $i+1$ Choose can

$\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$

1. remove a connected component from F_i, or
2. if the number of connected components of F_i is strictly less than μ, pick a vertex (either in $L(G)$ or $U(G)$) and challenge Cover to find a $\mathscr{C}$-matching F_{i+1} in G such that
   a) each connected component of F_i is also a connected component of F_{i+1};
   b) F_{i+1} covers the vertex picked by Choose.

Cover loses the game $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$ if at some point she cannot answer a challenge by Choose. Otherwise, Cover wins.

Our main interest in such games are the winning strategies for Cover and the fact that, for some graphs G, similar to the clauses-variables adjacency graphs we just saw, the winning strategies for Cover in the game $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$ provide μ-BG families. This is, informally, the content of Lemma 4.13 and

[1] This theorem was originally proved in [31], the paper where VW-matchings were introduced.

Lemma 4.14[2]. This allow the application of such lemmas not only to the clauses-variables adjacency graph of a (n, k, Δ)-random CNF formula but also, for example, to some adjacency graphs of the graph pigeonhole principle.

We then show that, under some assumptions on the graph G, Cover has winning strategies for $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$ and for $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ for large μ, where μ is related to the expansion properties of the graph G. This, informally, is the content of Theorem 4.15 and Theorem 4.22[3]. These guarantee a winning strategy for Cover in the game $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$ and $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$. They rely on two main ingredients:

1. G is a $(\gamma n, \delta)$-*bipartite expander* graph, that is

$$\forall A \subseteq L(G),\ |A| \leqslant \gamma n \to |N_G(A)| \geqslant \delta |A|,$$

   where $\delta \geqslant 1.95$.

2. some (technical) upper bound on the number of high degree vertices in $U(G)$.

Random bipartite graphs In Section 4.7 we prove that random bipartite graphs satisfy the conditions (1.) and (2.) above in Theorem 4.15 and Theorem 4.22. Hence putting together all we had so far, we prove that Cover has a winning strategy for both the games $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$ and $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ when G is a random graph of constant left degree and $\mu = \Omega(L(G))$[4].

Since all we did so far is in common between the random k-CNF formulas (Section 4.8) and the graph pigeonhole principle (Section 4.9) then we easily prove the space lower bounds also for G-PHP for suitable bipartite graphs G.

We start now looking at some families of formulas and, in particular, we start with the *complete tree* formulas, CT_n.

4.3 Complete Trees

Let n be a natural number, the *complete tree formula*, CT_n, is the unsatisfiable CNF formula whose clauses are *all* the 2^n possible clauses with n distinct literals in the variables $X = \{x_1, \ldots, x_n\}$. For instance CT_n

$$\mathsf{CT}_2 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_2)$$

[2]The proof of such lemmas is based on the proof of [31, Lemma 5.1]. Here we generalize that proof to $\mathscr{C}$-matchings where $\mathscr{C}$ is a family of trees with no leaves in $L(G)$.

[3]Theorem 4.15 is essentially [40, Lemma 4.12] and Theorem 4.22 is from [31]. Both of them are also extensions of constructions that can be found in the literature for matchings for example in [7, 24].

[4]This result rely on probabilistic calculations from [31].

and its encoding as a family of polynomials in $\mathbb{F}[X \cup \overline{X}]$ is

$$tr(\mathsf{CT}_2) = \{x_1x_2,\ \bar{x}_1x_2,\ x_1\bar{x}_2,\ \bar{x}_1\bar{x}_2\} \cup \{x_i^2 - x_i, x_i + \bar{x}_i - 1\}_{i\in\{1,2\}},$$

where, $\{x_i^2 - x_i, x_i - \bar{x}_i + 1\}_{i\in\{1,2\}}$ are the Boolean axioms. Alekhnovich et al. [4, Theorem 3.13] showed that

$$\mathsf{CSpace}^{sem}(\mathsf{CT}_n \vdash \bot) = n + 1. \tag{4.3}$$

Using Theorem 2.8, since CT_n is n-semiwide, we immediately have that

$$\mathsf{TSpace}^{sem}_{\mathsf{Res}}(\mathsf{CT}_n \vdash \bot) = \Theta(n^2).$$

Alekhnovich et al. [4, Corollary 5.6] showed also the following stronger result:

$$\mathsf{TSpace}^{sem}_{\mathsf{PCR}}(tr(\mathsf{CT}_n) \vdash 1) \geqslant \Theta(n^2), \tag{4.4}$$

and also an upper bound on monomial space in Polynomial Calculus:

$$\mathsf{MSpace}^{sem}(tr(\mathsf{CT}_n) \vdash 1) \leqslant 2n/3 + 6,$$

cf. [4, Theorem 4.2]. Interestingly this upper bound is lower than the clause lower bound we have in resolution, cf. equation (4.3). Regarding the monomial space lower bounds in PCR they proved that

$$\mathsf{MSpace}^{sem}(tr(\mathsf{CT}_n) \vdash 1) \geqslant n/4.$$

This monomial space lower bound can be easily reproved as a consequence of Theorem 3.7. And this is what we are going to do now.

Theorem 4.1 (Alekhnovich et al. [4])**.**

$$\mathsf{MSpace}^{sem}(tr(\mathsf{CT}_n) \vdash 1) \geqslant n/4.$$

Proof. We use Theorem 3.7. Choose as P_1 the Boolean axioms and choose as P_2 the other axioms of $tr(\mathsf{CT}_n)$, those that have degree n. The set P_1 is the set for which we will have to build a n-BG family $\mathscr{F}$.

Given $i \in \{1, \ldots, n\}$, let α_i and α_i' be the following partial assignments of domain $\{x_i, \bar{x}_i\}$:

$$\alpha_i(x_i) = 1 - \alpha_i(\bar{x}_i) = \alpha_i'(\bar{x}_i) = 1 - \alpha_i'(x_i) = 0.$$

Let then $H_i = \{\alpha_i, \alpha_i'\}$. By construction H_i is clearly flippable and 0-consistent. Let $\mathscr{F}$ be the family of flippable products defined as follows: $\mathcal{H} \in \mathscr{F}$ if and only if there exists a set $A \subseteq \{1, \ldots, n\}$ such that

$$\mathcal{H} = \bigotimes_{i\in A} H_i.$$

First observe that each $\mathcal{H} \in \mathscr{F}$, with rank strictly less than n, leaves a variable unassigned in every monomial in P_2, hence P_2 satisfy the hypothesis of Theorem 3.7.

We prove now that $\mathscr{F}$ is a n-BG family for P_1, the Boolean axioms, with respect to the 0 ideal. For $A = \emptyset$ the construction implies that $\{\lambda\} \in \mathscr{F}$, hence $\mathscr{F}$ is non-empty. Moreover, by construction, $\mathcal{H} \in \mathscr{F}$ implies that $\mathcal{H}$ is 0-consistent. The *restriction* and *extension* properties of $\mathscr{F}$ are also clear. □

4.4 Pigeonhole principles

The *pigeonhole principle* asserts that there is no multi-valued total injective mapping from $[m]$ to $[n]$, if $m > n$. The elements of the set $[m]$ are traditionally called *pigeons* and the elements of the set $[n]$ are called *holes* and so the pigeonhole principle can be stated more pictorially saying that

pigeons

> *if $m > n$ pigeons fly to n holes then (at least) two of them must end to be in the same hole.*

Interestingly, the proof complexity of the pigeonhole principle essentially depends on the number of pigeons m (as a function of the number of holes n) and on some details of its encodings as an unsatisfiable CNF formula or as an unsatisfiable set of polynomials. We refer to the survey [125] for more details on the proof complexity of the pigeonhole principle.

The encodings of the pigeonhole principle as an unsatisfiable CNF formula that we consider are the following: PHP^m_n, fPHP^m_n, oPHP^m_n, cf. Section 4.4.1; bitPHP^m_n, cf. Section 4.4.2; xorPHP^m_n, cf. Section 4.4.3. The *graph pigeonhole principle* (or *matching principle over graphs*), G-PHP, is instead considered much later in this chapter, cf. Section 4.9.

4.4.1 The (standard) pigeonhole principles

Let $m, n \in \mathbb{N}$ be two integers such that $m > n$ and consider the set of mn variables $X = \{x_{ij} : i \in [m], j \in [n]\}$. The intended meaning of x_{ij} is the truth value of '*the i-th pigeon goes into the j-th hole*'. The standard encoding of the pigeonhole principle, PHP^m_n, assert that there is an injective multi-valued mapping from $[m]$ to $[n]$. It is the conjunction of the following clauses

PHP^m_n

1. $\neg x_{ij} \vee \neg x_{i'j}$ for all $i \neq i' \in [m]$ and for all $j \in [n]$ (*injectivity axioms*);
2. $x_{i1} \vee x_{i2} \vee \ldots \vee x_{in}$ for all $i \in [m]$.

Clearly PHP_n^m is unsatisfiable whenever $m > n$ and its proof complexity has been investigated in depth since Haken [81] used PHP_n^{n+1} to prove the first (sub-)exponential lower bounds on size for Resolution:

$$\mathsf{size}_{\mathsf{Res}}(\mathsf{PHP}_n^{n+1} \vdash \bot) \geqslant 2^{\Omega(n)}.$$

Regarding PHP_n^m, the larger m is with respect to n the 'more contradictory' PHP_n^m is and interestingly its proof complexity depends on the number of pigeons m with some *qualitative* changes occurring when

$$m = n + 1, 2n, n^2, \infty.$$

For example:

$$\mathsf{size}_{\mathsf{Res}}(\mathsf{PHP}_n^{n^2} \vdash \bot) \geqslant 2^{\Omega(n/\log n)},$$

cf. [121, 126], and, for every $m > n$,

$$\mathsf{size}_{\mathsf{Res}}(\mathsf{PHP}_n^m \vdash \bot) \geqslant 2^{\Omega(\sqrt[3]{n})},$$

cf. [126]. Regarding the upper bounds, in Resolution, Buss and Pitassi [47, Lemma 1] showed that PHP_n^{n+1} has Resolution refutations of size $O(n2^n)$. More in general, PHP_n^m has polynomial size refutations in proof systems such as *Cutting Planes*, cf. [57, 75], and Frege systems[5], cf. [46]. On the other hand constant-depth Frege proofs of the pigeonhole principle require exponential size, cf. [1, 15, 98, 116]. We refer to [125] for a survey on the proof complexity of the pigeonhole principle although we recall other results on it later in this section.

Regarding the space complexity of the pigeonhole principle, we have that it does not depend on the number of pigeons. Esteban and Torán [66] and Alekhnovich et al. [4] showed that

$$\mathsf{CSpace}(\mathsf{PHP}_n^m \vdash \bot) \geqslant n,$$

and since PHP_n^m is an n-semiwide formula, cf. Definition 2.7, by Theorem 2.8, we have the following total space lower bound:

$$\mathsf{TSpace}_{\mathsf{Res}}^{sem}(\mathsf{PHP}_n^m \vdash \bot) \geqslant \frac{n^2}{4}.$$

This result was proved previously by Alekhnovich et al. [4, Corollary 5.7] and indeed their proof can be seen as a special case of the proof of Theorem 2.8. Indeed Alekhnovich et al. [4, Corollary 5.7] showed also that

$$\mathsf{TSpace}_{\mathsf{PCR}}^{sem}(tr(\mathsf{PHP}_n^m) \vdash 1) \geqslant \Theta(n^2).$$

[5] We refer to Section 1.2.1 for the informal definitions of Cutting Planes and Frege systems.

The bound above together with the CT_n total space lower bound, cf. equation (4.4), at the moment of writing this thesis, are the only non-trivial total space lower bounds known for Polynomial Calculus and proof systems stronger than Resolution.

For convenience of the reader we recall that the encoding of PHP^m_n as a set of polynomials $tr(\mathsf{PHP}^m_n)$ in the ring $\mathbb{F}[X \cup \overline{X}]$ is the following:

$$tr(\mathsf{PHP}^m_n) = \Big\{x_{ij}\cdot x_{i'j}\Big\}_{\substack{i\neq i'\in[m]\\ j\in[n]}} \cup \Big\{\prod_{j\in[n]} \bar{x}_{ij}\Big\}_{i\in[m]} \cup \Big\{x^2_{ij}-x_{ij}, x_{ij}+\bar{x}_{ij}-1\Big\}_{\substack{i\in[m]\\ j\in[n]}}.$$

Notice that $tr(\mathsf{PHP}^m_n)$ contains polynomials of degree n. This make degree lower bounds trivial, hence sometimes in PCR an alternative encoding of PHP^m_n is used. It is an encoding of PHP^m_n as a set of small degree polynomials where the polynomials $\Big\{\prod_{j\in[n]} \bar{x}_{ij}\Big\}_{i\in[m]}$ from $tr(\mathsf{PHP}^m_n)$ are substituted by $\Big\{\sum_j x_{ij} - 1\Big\}_{i\in[n]}$. To avoid confusion we call linPHP^m_n the version of PHP^m_n described above, that is the version of PHP^m_n where large degree initial polynomials are substituted by linear polynomials. Considering linPHP^m_n instead of PHP^m_n makes sense when proving degree lower bounds but it trivially implies monomial space lower bounds, as already some of the polynomials in linPHP^m_n require a large number of monomials. Indeed the proof complexity of the pigeonhole principle in PCR was deeply studied, cf. e.g. [3, 55, 86, 124, 129]. In particular we have that for every $m > n$, linPHP^m_n require PCR refutations of degree $\Omega(n)$, cf. [124], and hence refutations of size $2^{\Omega(n)}$, due to equation (1.2), cf. [86].

On the other hand we trivially have that $\mathsf{MSpace}^{sem}(\mathsf{linPHP}^m_n \vdash 1) \geqslant n$, hence PHP^m_n is more interesting than linPHP^m_n from the point of view of monomial space lower bounds.

The monomial space lower bound we prove holds for the so called *onto* version of the pigeonhole principle, oPHP^m_n, that is the conjunction of PHP^m_n with the following clauses: oPHP^m_n

$$x_{1j} \vee x_{2j} \vee \ldots \vee x_{mj},$$

for all $j \in [n]$ (*onto axioms*). We recall that the encoding of oPHP^m_n as a set of polynomials $tr(\mathsf{oPHP}^m_n)$ in the ring $\mathbb{F}[X \cup \overline{X}]$ is the following

$$tr(\mathsf{PHP}^m_n) = tr(\mathsf{PHP}^m_n) \cup \Big\{\prod_{i\in[m]} \bar{x}_{ij}\Big\}_{j\in[n]}.$$

Clearly for any ideal I,

$$\mathsf{MSpace}^{sem}(tr(\mathsf{PHP}^m_n) \vdash_I 1) \geqslant \mathsf{MSpace}^{sem}(tr(\mathsf{oPHP}^m_n) \vdash_I 1),$$

so we are going to prove a monomial space lower bound for oPHP^m_n immediately obtaining a monomial space lower bound for PHP^m_n.

Theorem 4.2 (Alekhnovich et al. [4])**.**

$$\mathsf{MSpace}^{sem}(tr(\mathsf{oPHP}_n^m) \vdash 1) \geqslant n/4$$

Proof. We apply Theorem 3.7. Let P_1 be the set of all low degree polynomials in $tr(\mathsf{oPHP}_n^m)$, that is

$$P_1 = \Big\{x_{ij}x_{i'j}\Big\}_{\substack{i\neq i'\in[m]\\ j\in[n]}} \cup \Big\{x_{ij}^2 - x_{ij}, x_{ij} + \bar{x}_{ij} - 1\Big\}_{\substack{i\in[m]\\ j\in[n]}},$$

and let P_2 be the remaining polynomials in $tr(\mathsf{oPHP}_n^m)$, that is those with degree at least n:

$$P_2 = \Big\{\prod_{j\in[n]} \bar{x}_{ij}\Big\}_{i\in[m]} \cup \Big\{\prod_{i\in[m]} \bar{x}_{ij}\Big\}_{j\in[n]}.$$

Given $i \in [m]$ and $j \in [n]$, let α_{ij} be the partial assignment with domain $\{x_{i'j} : i' \in [m]\}$ defined as follows

$$\alpha_{ij}(x_{i'j}) = \begin{cases} 1 & \text{if } i' = i, \\ 0 & \text{if } i' \neq i. \end{cases}$$

Let $H_j = \{\alpha_{ij}\}_{i\in[m]}$ and let $\mathscr{F}$ be the following family of flippable products: $\mathcal{H} \in \mathscr{F}$ if and only if there exists a set of holes $A \subseteq \{1, \ldots, n\}$ such that

$$\mathcal{H} = \bigotimes_{i\in A} H_i.$$

By construction, each partial assignment in H_j has as domain the set $\{x_{ij}, \bar{x}_{ij}\}_{i\in[m]}$, H_j is flippable and 0-consistent, hence $\mathscr{F}$ is well defined and each $\mathcal{H} \in \mathscr{F}$ is 0-consistent.

We prove that $\mathscr{F}$ is a n-BG family for P_1 with respect to the 0 ideal. The family $\mathscr{F}$ is non-empty as for $A = \emptyset$ the definition implies that $\{\lambda\} \in \mathscr{F}$. The *restriction* property is immediate from the definition. For the *extension* property, let $p \in P_1$ and $\mathcal{H} \in \mathscr{F}$, with $\|\mathcal{H}\| < n$ such that $\mathcal{H} = \bigotimes_{j'\in A} H_{j'}$ for some $A \subseteq [n]$. There is exactly one $j \in [n]$ such that $\mathrm{var}(p) \subseteq \mathrm{dom}(H_j)$. If $j \in A$ then, by construction, $\mathcal{H} \vDash_I p$ hence we take $\mathcal{H}_p = \{\lambda\}$. If $j \notin A$ then H_j is domain-disjoint from $\mathcal{H}$, $\mathcal{H} \otimes H_j \in \mathscr{F}$ and by construction is such that $\mathcal{H} \otimes H_j \vDash_0 p$. Take $\mathcal{H}_p = H_j$ in this case.

The set P_2 satisfies the hypothesis of Theorem 3.7 since every $\mathcal{H} \in \mathscr{F}$ of rank strictly less than n leaves unset at least one variable in each each axiom of the form $\prod_{j\in[n]} \bar{x}_{ij}$ and each axiom of the form $\prod_{i\in[m]} \bar{x}_{ij}$ is either set to 0 or is left unset by elements in $\mathscr{F}$. □

Notice that the very same construction above can be used to prove monomial space lower bounds for $tr(\mathsf{oPHP}^m_n)$ with respect to the ideal generated by the injectivity axioms: let I be such ideal, then

$$\mathsf{MSpace}^{sem}(\mathsf{oPHP}^m_n \vdash_I 1) \geqslant \frac{n^2}{4}.$$

We recall that there is one more 'standard' pigeonhole principle considered in proof complexity, that is the *graph pigeonhole principle*, G-PHP, cf. Section 4.9.

We end this section observing that the technique above cannot work for the *functional pigeonhole principle*, fPHP^m_n, that is the conjunction of the PHP^m_n formula with the following clauses

$$\neg x_{ij} \vee \neg x_{ij'},$$

where $i \in [m]$ and $j, j' \in [n]$ distinct (*functionality axioms*). This observation is due to Filmus et al. [68], so to prove monomial space lower bounds in Polynomial Calculus for fPHP^m_n the constructions that we are using in this thesis will be probably not enough. In [68] it is moreover observed that monomial space lower bounds for fPHP^m_n are equivalent to monomial space lower bounds for a 3-CNF version of fPHP^m_n.

In the following two subsections we consider two less standard encoding of the pigeonhole principle: bitPHP_n and xorPHP^m_n.

4.4.2 The bit pigeonhole principle

Let $n = 2^k$ for $k \in \mathbb{N}$. The *bit pigeonhole principle on n holes*, bitPHP_n, is an unsatisfiable CNF formula over the variables $X = \{x_{ij} : i \in [n+1], j \in [k]\}$. It asserts that for all distinct $i, i' \in [n+1]$, the length-k binary strings $x_{i1} \dots x_{ik}$ and $x_{i'1} \dots x_{i'k}$ are distinct. We think of each element of $[n+1]$ as a pigeon and of the string $x_{i1} \dots x_{ik}$ as the address, in binary, of the hole in $[n]$ that pigeon i is flying to. Understood in this way, bitPHP_n asserts that there is an injective mapping of $n+1$ pigeons into n holes. Formally the principle consists of the clauses $B^{i,i'}_h$ bitPHP_n

$$B^{i,i'}_h = \bigvee_{j=1}^{k} (x_{ij} \not\equiv h_j) \vee (x_{i'j} \not\equiv h_j),$$

for each $i, i' \in [n+1]$ with $i < i'$ and each $h \in [n]$ such that its binary expansion is $h_1 \dots h_k \in \{0,1\}^k$. The expression $x_{ij} \not\equiv h_j$ is a shortcut for $\neg x_{ij}$ if $h_j = 1$ and for x_{ij} if $h_j = 0$.

Then the bitPHP_n is a formula over $(n+1)\log n$ variables consisting of $n^2(n+1)$ clauses each of width $2\log n$. Two motivations to study, and sometimes

prefer, bitPHP_n against PHP_n^{n+1} are that its encoding is more efficient from the point of view of the number of variables used and the its width is $O(\log n)$ instead of n.

Theorem 4.3. *There exists a non-empty $\frac{n}{2}$-BG family $\mathscr{F}$ for $tr(\mathsf{bitPHP}_n)$ with respect to the 0 ideal.*

Proof. Given a hole h with binary representation $h_1, \ldots, h_k$, let $\bar{h}$ be the hole with complementary binary representation $1 - h_1, \ldots, 1 - h_k$. Similarly, given a set of holes A, let $\overline{A} = \{\bar{h} : h \in A\}$. The notation $\left[i \mapsto h, i' \mapsto \bar{h}\right]$ where $i, i' \in [n+1]$ and $h \in [n]$ is a shortcut for the partial assignment with domain $\{x_{ij}, \bar{x}_{ij}, x_{i'j}, \bar{x}_{i'j} : j \in [k]\}$

$$\left[i \mapsto h, i' \mapsto \bar{h}\right](x_{i''j}) = 1 - \left[i \mapsto h, i' \mapsto \bar{h}\right](\bar{x}_{i''j}) = \begin{cases} h_j & \text{if } i'' = i, \\ 1 - h_j & \text{if } i'' = i'. \end{cases}$$

Given $h \in [n/2]$ and $\sigma : \{h, \bar{h}\} \to [n+1]$ an injective mapping[6], let H_h^σ be the set of partial assignments of domain $\{x_{\sigma(h)j}, x_{\sigma(\bar{h})j}, \bar{x}_{\sigma(h)j}, \bar{x}_{\sigma(\bar{h})j}\}_{j \in [k]}$:

$$H_h^\sigma = \left\{ \left[\sigma(h) \mapsto h, \sigma(\bar{h}) \mapsto \bar{h}\right], \quad \left[\sigma(h) \mapsto \bar{h}, \sigma(\bar{h}) \mapsto h\right] \right\}.$$

By construction all the assignments in H_h^σ have the same domain, H_h^σ is flippable and I-consistent. Consider then the following family $\mathscr{F}$ of flippable products: $\mathcal{H} \in \mathscr{F}$ if and only if there exists a set of holes $A \subseteq [n/2]$ and there exists an injective mapping $\sigma : A \cup \overline{A} \to [n+1]$ such that

$$\mathcal{H} = \bigotimes_{h \in A} H_h^\sigma.$$

We prove that $\mathscr{F}$ is a $\frac{n}{2}$-BG family for $tr(\mathsf{bitPHP}_n)$ with respect to the 0 ideal. For $A = \emptyset$ the definition implies that $\{\lambda\} \in \mathscr{F}$ so $\mathscr{F}$ is non-empty. By construction, $\mathcal{H} \in \mathscr{F}$ implies that $\mathcal{H}$ is 0-consistent and the *restriction* property of $\mathscr{F}$ is obvious, hence we focus on the *extension* property.

Let $\mathcal{H} = \bigotimes_{h \in A} H_h^\sigma \in \mathscr{F}$ such that $\|\mathcal{H}\| < n/2$ and consider $p = tr(B_{i,i'}^h)$. If both $i, i' \in \sigma(A \cup \bar{A})$ then, by construction, $\mathcal{H} \vDash_0 p$, hence we can take $\mathcal{H}_p = \{\lambda\}$.

Otherwise, without loss of generality, assume $i' \notin \sigma(A \cup \bar{A})$. Since we have that $\|\mathcal{H}\| = |A| < n/2$, there is some hole $h' \in [n/2] \setminus A$ and an injective σ' such that $\sigma' = \sigma \cup \{h' \mapsto i'\} \cup \{\bar{h}' \mapsto i''\}$ with i'' outside $\sigma(A \cup \bar{A}) \cup \{i'\}$. If $i \notin \sigma(A \cup \bar{A})$ take $i'' = i$. Let $\mathcal{H}_p = H_{h'}^{\sigma'}$: it is clearly 0-consistent and domain-disjoint from $\mathcal{H}$. Define $\mathcal{H}' = \mathcal{H} \otimes \mathcal{H}_p = \bigotimes_{h \in A'} H_h^{\sigma'} \in \mathscr{F}$, where $A' = A \cup \{h'\}$.

[6]Notice that as $h \in [n/2]$ then h and $\bar{h}$ are distinct.

Notice that $\mathcal{H}' \vDash_0 p$, as each assignment in $\mathcal{H}'$ set i and i' to go into two distinct holes. More precisely, if $i \in \sigma(A \cup \overline{A})$ then i goes somewhere inside $A \cup \overline{A}$ and i' goes either in h' or $\bar{h}'$. If $i \notin \sigma(A \cup \overline{A})$ then, by construction, i goes in $\bar{h}'$ and i' goes to h' or viceversa.

The case when p is a Boolean axiom, say $x_{ij}^2 - x_{ij}$ is completely analogous, either $i \in \sigma(A \cup \overline{A})$ and in this case we set $\mathcal{H}_p = \{\lambda\}$; or $i \notin \sigma(A \cup \overline{A})$ and in this case we can extend as done before. □

From Theorem 4.3 and Theorem 3.6 we immediately obtain the following corollary, proved by Filmus et al. [70].

Corollary 4.4. $\mathsf{MSpace}^{sem}(tr(\mathsf{bitPHP}_n) \vdash 1) \geqslant n/8$.

Moreover, by Proposition 3.5, the existence of a non-empty $\frac{n}{2}$-BG family immediately implies the existence of a non-empty $(\frac{n}{2} - 1)$-BK family and hence, by the characterization of the asymmetric width of Theorem 2.9, then

$$\mathsf{awidth}(\mathsf{bitPHP}_n \vdash \bot) = \Omega(n).$$

From this result and Corollary 2.11, a total space lower bound in Resolution follows immediately :

$$\mathsf{TSpace}_{\mathsf{Res}}(\mathsf{bitPHP}_n \vdash \bot) \geqslant \Omega(n^2). \tag{4.5}$$

Since bitPHP_n has only $(n+1)\log n$ variables, then the previous one is a total space lower bound in resolution that is *super-linear* in the number of variables. This result was proven in [40] and constitutes the very first super-linear total space lower bound for a formula with just polynomially many clauses.

A more direct proof of the monomial space lower showed in this section can be found in [37, 70] and a more direct proof of the total space lower bound can be found in [40][7].

4.4.3 The XOR pigeonhole principle

Searching for monomial space lower bounds for formulas of constant width, Filmus et al. [70] introduced the XOR-pigeonhole principle.

Let $m, n \in \mathbb{N}$ be two integers such that $m > n$ and let consider a set of Boolean variables $X = \{x_{i,j} : i \in [m],\ j \in [n] \cup \{0\}\}$. A pigeon $i \in [m]$ is considered assigned to a hole $j \in [n]$ when $x_{i,j-1} \not\equiv x_{i,j}$ is true. The *XOR-Pigeonhole Principle*, xorPHP_n^m, expresses the following weaker form of the xorPHP_n^m

[7] Due to Corollary 2.11 we could have obtained a total space lower bound in Resolution also proving a width lower bound for bitPHP_n.

pigeonhole principle: if each pigeon is assigned to an odd number of holes, then there exists a hole with at least 2 pigeons. The formula xorPHP_n^m is a contradictory 4-CNF formula encoding the negation of the principle as follows:

1. for each $i \in [m]$, $x_{i,0} \not\equiv x_{i,n}$, that is

$$(x_{i,0} \vee x_{i,n}) \wedge (\neg x_{i,0} \vee \neg x_{i,n});$$

2. for all distinct $i, i' \in [m]$ and all $j \in [n] \cup \{0\}$,

$$(x_{i,j-1} \equiv x_{i,j}) \vee (x_{i',j-1} \equiv x_{i',j}),$$

that is

$$(x_{i,j-1} \vee \neg x_{i,j} \vee x_{i',j-1} \vee \neg x_{i',j}) \wedge (\neg x_{i,j-1} \vee x_{i,j} \vee \neg x_{i',j-1} \vee x_{i',j}) \wedge$$
$$(x_{i,j-1} \vee \neg x_{i,j} \vee \neg x_{i',j-1} \vee x_{i',j}) \wedge (\neg x_{i,j-1} \vee x_{i,j} \vee x_{i',j-1} \vee \neg x_{i',j}).$$

Theorem 4.5. *There exists a non-empty $(n-1)$-BG family for $tr(\mathsf{xorPHP}_n^m)$ with respect to the ideal I generated by the Boolean axioms.*

Proof. Given $i \in [m]$ and $j \in [n]$, let $H_{i \mapsto j}$ be the following set of partial assignments of domain $\{x_{ij'}, \bar{x}_{ij'} : j' \in [n] \cup \{0\}\}$:

$$H_{i \mapsto j} = \{\alpha_{ij}, \alpha_{ij}^*\},$$

where

$$\alpha_{ij}(x_{ij'}) = 1 - \alpha_{ij}(\bar{x}_{ij'}) = 1 - \alpha_{ij}^*(x_{i'j}) = \alpha_{ij}^*(\bar{x}_{i'j}) = \begin{cases} 1 & \text{if } j' < j \\ 0 & \text{if } j' \geqslant j. \end{cases}$$

By construction $H_{i \mapsto j}$ is flippable and I-consistent.

Let $\mathscr{F}$ be defined as follows: $\mathcal{H} \in \mathscr{F}$ if and only if there exists a set $A \subseteq [m]$ of size at most $n-1$ and there exists an injective mapping $\mu : A \longrightarrow [n]$ such that

$$\mathcal{H} = \bigotimes_{i \in A} H_{i \mapsto \mu(i)}.$$

We prove that $\mathscr{F}$ is a $(n-1)$-BG family for $tr(\mathsf{xorPHP}_n^m)$ with respect to the ideal I. $\mathscr{F}$ is non-empty as for $A = \emptyset$ the definition implies that $\{\lambda\} \in \mathscr{F}$. By construction, $\mathcal{H} \in \mathscr{F}$ implies that $\mathcal{H}$ is I-consistent. The *restriction property* is immediate from the definition. To prove the *extension property*, let $\mathcal{H} = \bigotimes_{i \in A} H_{i \mapsto \mu(i)} \in \mathscr{F}$ with $\|\mathcal{H}\| < n-1$ and p the polynomial encoding of a initial clause C from xorPHP_n^m. Let us suppose first that C is a clause from some

$(x_{i,j-1} \equiv x_{i,j}) \vee (x_{i',j-1} \equiv x_{i',j})$. If both i and i' are in A, then, by construction, $\mathcal{H} \vDash_I p$ and we can take $\mathcal{H}_p = \{\lambda\}$. If $i \notin A$, then, as μ is an injective assignment of at most $n-2$ pigeons, we can find a hole h *different from* j which is not in $\mu(A)$. Then let $\mu' = \mu \cup \{i \mapsto h\}$ and $\mathcal{H}' = \bigotimes_{\ell \in A \cup \{i\}} H_{\ell \mapsto \mu'(\ell)} = \mathcal{H} \otimes H_{i \mapsto h}$. By construction $H_{i \mapsto h} \vDash_I p$, hence $\mathcal{H}' \vDash_I p$. In this case take $\mathcal{H}_p = H_{i \mapsto h}$. Similarly if $C = (x_{i,0} \not\equiv x_{i,n})$ we proceed as before extending μ to assign the pigeon i somewhere (if needed). □

From Theorem 4.5 and Theorem 3.6 we immediately obtain a monomial space lower bound for xorPHP^m_n.

Corollary 4.6. $\mathsf{MSpace}^{sem}(tr(\mathsf{xorPHP}^m_n) \vdash_I 1) \geqslant (n-1)/4$, *where I is the ideal generated by the Boolean axioms.*

In [70] the authors proved that $\mathsf{MSpace}^{sem}(tr(\mathsf{xorPHP}^m_n) \vdash 1) > n/4$. The result in Corollary 4.6 asymptotically matches the result from [70].

Notice that the proof of Theorem 4.5 could be adapted immediately for the 0 ideal and from that, by Proposition 3.5 and Theorem 2.5, it follows a total space lower bound in Resolution:

$$\mathsf{TSpace}_{\mathsf{Res}}(\mathsf{xorPHP}^m_n \vdash \bot) \geqslant n^2/4.$$

Differently from the bitPHP_n, this total space lower bound is just (at most) linear in the number of variables and not super-linear as for bitPHP_n, cf. equation (4.5).

4.5 Tseitin Formulas

Tseitin formulas are essentially based on a Boolean encodings of the fact that the total degree of any graph is an even number. Those formulas were originally used by Tseitin [138] to present the first super-polynomial lower bounds on refutation size for regular Resolution, a restricted form of the Resolution proof system. Then they were used to prove exponential lower bounds on the size of Resolution refutations, for example in [132, 139]. Since then the Tseitin formulas became one of the standard tools used in proof complexity to prove lower bounds and trade-offs, for example they have been investigated regarding the width, cf. [30], clause space, cf. [66] and recently Beck et al. [20] used Tseitin formulas over long skinny grids to prove size-space trade-offs in Polynomial Calculus.

Let $G = (V, E)$ be a connected graph of degree at most d over n vertices. An *odd-weight* function $\sigma : V \to \{0, 1\}$ is a function σ such that

$$\sum_{v \in V} \sigma(v) \equiv 1 \pmod 2.$$

Consider now the set of Boolean variables $X = \{x_e : e \in E\}$ and for each $v \in V$ let $\mathsf{PARITY}_{v,\sigma}$ be the CNF formula expressing the following:

$\mathsf{PARITY}_{v,\sigma}$

$$\sum_{e \ni v} x_e \equiv \sigma(v) \pmod 2.$$

The *Tseitin formula*, $\mathsf{Tseitin}(G, \sigma)$, is then the conjunction

$\mathsf{Tseitin}(G,\sigma)$

$$\mathsf{Tseitin}(G, \sigma) = \bigwedge_{v \in V} \mathsf{PARITY}_{v,\sigma}.$$

The formula $\mathsf{Tseitin}(G, \sigma)$ is then a d-CNF formula over at most $dn/2$ variables and $n2^{d-1}$ clauses. Moreover, as observed by Urquhart [140], if σ is odd-weight then $\mathsf{Tseitin}(G, \sigma)$ is unsatisfiable (and the other implication is also true).

It turns out that many properties of the proof complexity of Tseitin formulas $\mathsf{Tseitin}(G, \sigma)$ can be captured by the *connectivity expansion* of G.

Let $G = (V, E)$ be a finite connected graph, the *connectivity expansion* of G (or just *expansion*), $e(G)$, is

$e(G)$

$$e(G) = \min \left\{ |E \cap (V' \times (V \setminus V'))| : V' \subseteq V \wedge |V'| \in \left[\frac{|V|}{3}, \frac{2|V|}{3} \right] \right\}$$

Then G is an *expander graph* if $e(G) = \Omega(|V|)$, for instance random d-regular graphs with high probability are expanders.

As proven by Ben-Sasson and Wigderson [30, Theorem 4.4], given a connected graph $G = (V, E)$ and an odd-weight function on V, then

$$\mathsf{width}(\mathsf{Tseitin}(G, \sigma) \vdash \bot) \geqslant e(G). \tag{4.6}$$

Hence, for example, by the size-width relations by Ben-Sasson and Wigderson [30] a size lower bound for Tseitin formulas follows

$$\mathsf{size}_{\mathsf{Res}}(\mathsf{Tseitin}(G, \sigma) \vdash \bot) \geqslant 2^{\Omega(e(G))}.$$

Concerning total space lower bounds in Resolution, we have the following result that completely answer the open problem from [4, Open question 2].

Theorem 4.7. *Let $G = (V, E)$ be a connected d-regular graph and σ an odd-weight function over V, then*

$$\mathsf{TSpace}_{\mathsf{Res}}(\mathsf{Tseitin}(G, \sigma) \vdash \bot) \geqslant \frac{1}{16} \left(e(G) - d - 2 \right)^2.$$

In particular if G is a 3-regular expander graph over n vertices then

$$\mathsf{TSpace}_{\mathsf{Res}}(\mathsf{Tseitin}(G,\sigma) \vdash \bot) \geqslant \Omega(n^2).$$

Proof. It follows immediately from equation (4.6) and Corollary 2.11. □

Regarding the monomial space in Polynomial Calculus the picture is more complex. We do not know non-trivial monomial space lower bounds for Tseitin formulas over 3-regular expander graphs. Yet we have some monomial space lower bounds for some Tseitin formulas. In particular we have the following results. If $G = (V, E)$ is a d-regular graph with double edges[8] and σ any odd-weight function over V, then

$$\mathsf{MSpace}^{sem}(tr(\mathsf{Tseitin}(G,\sigma)) \vdash 1) \geqslant \Omega(e(G) - d). \tag{4.7}$$

Filmus et al. [68] showed this result relying on a preliminary version of Theorem 3.6 as appeared in [36]. More in general they showed that given a k-CNF formula φ and its *xorification* $\varphi[\oplus]$, then

$$\mathsf{MSpace}^{sem}(tr(\varphi[\oplus]) \vdash 1) \geqslant \frac{1}{4}(\mathsf{width}(\varphi \vdash \bot) - k + 1), \tag{4.8}$$

The *xorification* of a CNF formula φ is a new CNF formula $\varphi[\oplus]$ obtained by replacing each occurrence of a variable x_i in φ with the XOR of two new variables $x'_i \oplus x''_i$ and then expanding everything as a CNF formula using the definition of the XOR and the De Morgan rules[9]. Regarding the proof of equation (4.8) it is done essentially showing that from a r-AD family for φ we can construct a suitable r'-BG family for $tr(\varphi[\oplus])$. From this observation equation (4.7) follows since $\mathsf{Tseitin}(G,\sigma)[\oplus]$ is equivalent to $\mathsf{Tseitin}(G',\sigma)$ where G' is a multigraph over the same vertex set of G obtained by doubling the multiplicity of each edge of G.

Filmus et al. [68] showed also that if $G = (V, E)$ is a random d-regular graph on n vertices, where $d \geqslant 4$, then, with high probability, for every odd-weight function σ on V

$$\mathsf{MSpace}^{sem}(tr(\mathsf{Tseitin}(G,\sigma)) \vdash 1) \geqslant \Omega(\sqrt{n}). \tag{4.9}$$

The proof of this result again relies the preliminary version of Thorem 3.6 as appeared in [36]. Interestingly the proof of equation (4.9) in [68] do not rely just on $e(G)$ but on the fact that actually G is a random graph.

[8] That is each edge has multiplicity 2.

[9] We will see more in detail properties of xorifications in Chapter 5.

Over $\mathbb{F}_2$ it is known that Tseitin formulas have polynomial size refutations in Polynomial Calculus, essentially mimicking Gaussian elimination. On the other hand, the monomial space lower bounds we showed do not depend on the characteristic of the ground field, that is despite such formulas over $\mathbb{F}_2$ have short proofs, such refutations still require large monomial space.

4.6 From $\mathscr{C}$-matchings to r-BG families

bipartite graphs

In what follow it is useful to consider a non-standard definition of bipartite graphs. We will consider as *bipartite graphs* subgraphs of the (infinite) bipartite graph B with vertex set $\mathbb{N} \times \{0,1\}$ and such that $\{(n,b),(m,b')\} \in E(B)$ if and only if $b \neq b'$. Given a bipartite graph G we call $V(G) \cap \{(n,0) \ : \ n \in \mathbb{N}\}$ the

$L(G)$ $U(G)$

lower part of G, $L(G)$, and similarly $V(G) \cap \{(n,1) \ : \ n \in \mathbb{N}\}$ is the *upper part* of G, $U(G)$. We identify bipartite graphs in this way since we will be looking at isomorphisms preserving the lower and upper parts of bipartite graphs.

matching

In this section we give a generalisation of matchings and the V-matchings we saw in the previous chapter, cf. Section 3.4.1. Let G be a bipartite graph, a *matching* in G is a collection of vertex-disjoint edges in $E(G)$, alternatively, we can see a matching in G as a subgraph F where each connected component of F is isomorphic to the graph G_{I} with vertices $(0,0)$ and $(0,1)$ and a single edge $\{(0,0),(0,1)\}$, cf. Figure 4.1.

Figure 4.1: G_{I}

$\mathscr{C}$-matchings

Definition 4.8 ($\mathscr{C}-$*matchings*)**.** *Let $\mathscr{C}$ be a collection of bipartite graphs and G be a bipartite graph. A* $\mathscr{C}$-matching *in G is a subgraph F of G such that each connected component F_i of F is isomorphic to some graph G_j in $\mathscr{C}$ by an isomorphism that maps $L(G_j)$ into $L(F_i)$ (and the same for $U(G_j)$ and $U(F_i)$).*

With this notation the usual matchings are $\{G_{\mathsf{I}}\}$-matchings. In what follow we are interested in $\mathscr{C}$-matchings for particular collections of graphs $\{G_\bullet, G_{\mathsf{V}}\}$ and $\{G_\bullet, G_{\mathsf{V}}, G_{\mathsf{W}}\}$, where

1. $G_\bullet = (\{(0,1)\}, \emptyset)$.

Figure 4.2: $G_\bullet$

2. G_{V} has vertex-set $\{(0,0),(0,1),(1,1)\}$ and two edges $\{(0,0),(0,1)\}$ and $\{(0,0),(1,1)\}$: it has the shape of a V.

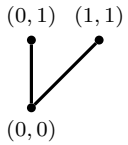


Figure 4.3: G_{V}

3. G_{W} has vertex-set $\{(0,0),(1,0),(0,1),(1,1),(2,1)\}$ and the four edges $\{(0,0),(0,1)\}$, $\{(0,0),(1,1)\}$, $\{(1,0),(1,1)\}$ and $\{(1,0),(2,1)\}\}$: it has the shape of a W.

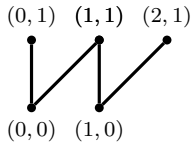


Figure 4.4: G_{W}

For simplicity we call the $\{G_{\mathsf{V}}, G_\bullet\}$-matchings simply V-*matchings* and the $\{G_{\mathsf{V}}, G_{\mathsf{W}}, G_\bullet\}$-matchings simply VW-*matchings*.

V-matchings

VW-matchings

Our main interest in V-matchings and VW-matchings is that those are among the simplest trees with no leaves in $\mathbb{N} \times \{0\}$ and, in some graphs associated to CNF formulas we use such trees to build r-BG families of assignments, cf. Section 4.6.2.

4.6.1 A Hall's Theorem for VW-matchings

Given a graph G, the existence of a matching in G covering $L(G)$ is related with the expansion properties of G. We recall now the definition of (s,δ)-*bipartite expander* graph.

(s,δ)-bipartite expander

Definition 4.9 ((s,δ)-bipartite expander)**.** *Let s be a positive integer and δ be a positive real number. A bipartite graph G is a* (s,δ)-bipartite expander *if and only if*

$$\forall A \subseteq L(G), |A| \leqslant s \to |N_G(A)| \geqslant \delta|A|.$$

If $|N_G(A)| \geqslant \delta|A|$ we say that A expands *in G at least δ.*

We are going to prove an analogue of the Hall's theorem, cf. Theorem 3.12 and its generalisation to V-matchings, cf. Theorem 3.13.

Before doing that, we argue why we can't get an exact analogue of such results for VW-matchings. Clearly we have the following implication.

Proposition 4.10. *Let G be a bipartite graph and let $|L(G)| = n$, if there exists a* VW*-matching in G covering $L(G)$ then G is a $(n, 3/2)$-bipartite expander.*

Proof. If G has as subgraph a VW-matching covering $L(G)$ then clearly G is a $(n, 3/2)$-bipartite expander since each subset of a VW-matching expands in G at least $3/2$. □

Unfortunately, the converse of Proposition 4.10 does not hold. An easy counterexample is the bipartite graph D_n with vertex-set

$$([2n] \times \{1\}) \cup (\{0, 2, 4, \ldots, 2n\} \times \{0\}),$$

and edge-set consisting of all

$$\{(0,0),(2i,1)\}, \{(2i,0),(2i,1)\}, \{(2i,0),(2i-1,1)\},$$

for $1 \leqslant i \leqslant n$, cf. Figure 4.5 for D_4. We have that D_n is $(n+1, \delta_n)$-bipartite expander where $\delta_n \to 2$ as $n \to \infty$. But on the other hand there is no VW-matching in D_n covering $L(D_n)$.

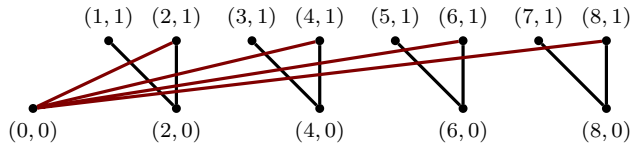


Figure 4.5: D_4

The next result behaves like a sort of converse of Proposition 4.10 and it is based on an analogous result in [31, Lemma 1.2]. The way we present it is somehow tailored to the applications we are interested in, that is bipartite graphs G such that each $v \in L(G)$ has degree at most 3.

Theorem 4.11. *Let G be a bipartite graph. Suppose that the the following three properties hold:*

1. *for each $v \in L(G)$, $\deg(v) \leqslant 3$ and no pair of degree 3 vertices in $L(G)$ have the same set of neighbors,*
2. *$|N(L(G))| \geqslant (2-\epsilon)|L(G)|$, for some $\epsilon < \frac{1}{5}$ and*
3. *each proper subset of $L(G)$ can be covered by a* VW*-matching,*

then $L(G)$ can be covered by a VW*-matching.*

The following proof is based on a simplification of the original proof from [31] due to Susanna Figueiredo de Rezenede (*pers. comm.*). Since it is shorter and gives a better bound ($\epsilon < \frac{1}{5}$ instead of $\epsilon < \frac{1}{23}$) we prefer to put that proof here instead of the original one.

Proof. By contradiction, let G be a bipartite graph witnessing the fact that the theorem is not true and for shortness let $L = L(G)$ and $U = U(G)$. Without loss of generality we can suppose that $U = N_G(L)$. By hypothesis, every proper subset of L can be covered by a VW-matching but the whole L cannot. This means that the configurations of edges in Figure 4.6 cannot appear in G.

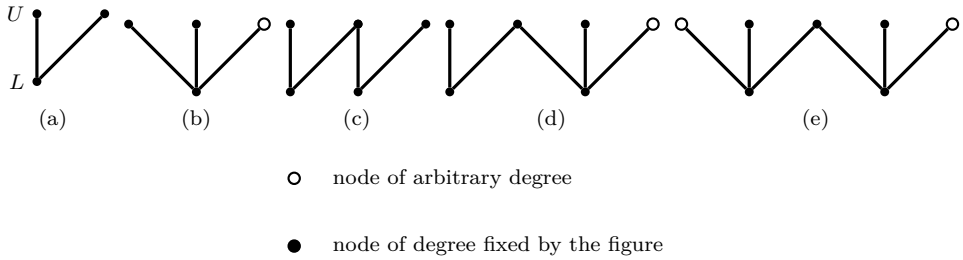


Figure 4.6: List of forbidden subgraphs

We say that two vertices v, v' in U are *close* if there exists a vertex $w \in L$ such that $v, v' \in N(w)$. We now weight each vertex in U by its degree and we redistribute the weight in the following way: each vertex in U of degree 1 gets weight $\frac{1}{3}$ from its close vertices. Let v be a vertex in U and let $w(v)$ be its weight at the end of the previous process. Then, since we are just redistributing the weight:

$$\sum_{v \in U} \deg(v) = \sum_{v \in U} w(v).$$

If $v \in U$ is such that $\deg(v) = 1$ then

$$w(v) = \begin{cases} 1 + 1/3 & \text{if } v \text{ has only one close vertex,} \\ 1 + 2/3 & \text{otherwise.} \end{cases}$$

In fact, since the edge configurations in Figures 4.6.(a) and (b) are not appearing in G, we have that two vertices of degree 1 in U cannot be close.

If $v \in U$ is such that $\deg(v) = 2$ then

$$w(v) \geqslant 2 - 1/3.$$

In fact, since the edge configurations in Figures 4.6. (c), (d) and (e) are not appearing in G, a vertex of degree 2 in U can be close to at most one vertex of degree 1 in U.

If $v \in U$ is such that $\deg(v) = d \geqslant 3$ then $w(v) \geqslant 2$, since it could be close to at most d vertices of degree 1 as (b) is forbidden, and hence $w(v) \geqslant d - \frac{d}{3} = \frac{2}{3}d \geqslant 2$.

Let $L = L_2 \cup L_3$, where L_i are the vertices of degree i in L and U' be the set of degree 1 vertices of U that have only one close vertex. This means that each $u \in U'$ is a neighbor of some vertex in L_2 and since no pair of vertices of degree 1 can be in the same neighborhood, then $|U'| \leqslant |L_2|$

Therefore we have

$$\begin{aligned} \frac{3|N(L)|}{(2-\epsilon)} &\geqslant 3|L| = 3|L_3| + 2|L_2| + |L_2| = \sum_{v \in U} \deg(v) + |L_2| = \\ &= \sum_{v \in U} w(v) + |L_2| \geqslant \frac{5}{3}|U| - \frac{1}{3}|U'| + |L_2| \geqslant \frac{5}{3}|U| = \frac{5}{3}|N(L)|, \end{aligned}$$

from which it follows the contradiction that $\epsilon \geqslant 1/5$. □

Notice that we are not really interested in optimising the constant ϵ in the previous result since, in the applications we show, it will be absorbed in some asymptotic notation.

Before proceeding to the applications we observe that the best possible value for ϵ in Theorem 4.11 would be $\epsilon = 1/3$. In fact in [31] we have the following proposition.

Proposition 4.12 (Bennett et al. [31])**.** *For all $\epsilon > \frac{1}{3}$ there exists a bipartite graph G_ϵ such that*

- *each vertex in $L(G_\epsilon)$ has degree at most 3 and no pair of degree 3 vertices in L have the same set of neighbors;*

- $|N(L(G_\epsilon))| \geqslant (2-\epsilon)|L(G_\epsilon)|$
- *each proper subset of* $L(G_\epsilon)$ *can be covered by a* VW*-matching;*
- $L(G_\epsilon)$ *cannot be covered by a* VW*-matching.*

4.6.2 Cover Games

Let G be a bipartite graph and $\mathscr{C}$ a collection of bipartite graphs. Given a subgraph F in G and a subset A of vertices of G, we recall that F *covers* A if $V(F) \supseteq A$.

The *Cover Game* $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$ is a game on the bipartite graph G between two players, Choose (he) and Cover (she). At each step i of the game the players maintain a $\mathscr{C}$-matching F_i in G. They start with the empty $\mathscr{C}$-matching and at step $i+1$ Choose can $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$

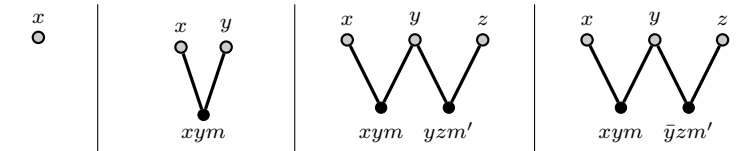
1. remove a connected component from F_i, or
2. if the number of connected components of F_i is strictly less than μ, pick a vertex (either in $L(G)$ or $U(G)$) and challenge Cover to find a $\mathscr{C}$-matching F_{i+1} in G such that
   a) each connected component of F_i is also a connected component of F_{i+1};
   b) F_{i+1} covers the vertex picked by Choose.

Cover loses the game $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$ if at some point she cannot answer a challenge by Choose. Otherwise, Cover wins.

We are interested in winning conditions for the player Cover for the cover games where V-matchings and VW-matchings are used, that is $\mathsf{CoverGame}_{\mathsf{V}}(G,\mu)$ and $\mathsf{CoverGame}_{\mathsf{VW}}(G,\mu)$. The reason for our interest is the fact that for suitable graphs G associated to sets of monomials then a winning strategy for the player Cover for the cover games where $\mathsf{CoverGame}_{\mathscr{C}}(G,\mu)$ imply the existence of a μ-BG family (under some assumption on $\mathscr{C}$) and hence ultimately some monomial space lower bound.

From winning strategies to r-BG *families*

Let $M = \{m_j\}_{j\in J}$ be an unsatisfiable set of monomials in the ring of polynomials $\mathbb{F}[X \cup \overline{X}]$, where $X = \{x_1,\ldots,x_n\}$ and $\overline{X} = \{\bar{x}_1,\ldots,\bar{x}_n\}$. Let G_M the *adjacency graph* of M, that is the bipartite graph with lower part $L(G_M) = \{(j,0) : j \in J\}$, upper part $U(G_M) = \{(\ell,1) : x_\ell \in X\}$ and there is an edge $\{(j,0),(\ell,1)\} \in E(G_M^{\mathscr{A}})$ if and only if $\mathrm{var}(m_j) \cap \{x_\ell, \bar{x}_\ell\} \neq \emptyset$. This definition generalizes G_M



$x \mapsto 0$ $x \mapsto 1$	$(x,y) \mapsto (0,1)$ $(x,y) \mapsto (1,0)$	$(x,y,z) \mapsto (0,1,0)$ $(x,y,z) \mapsto (1,0,1)$	$(x,y,z) \mapsto (0,1,1)$ $(x,y,z) \mapsto (1,0,0)$

Table 4.2: Flippable assignments from VW-matchings

immediately considering families of assignments instead of variables and this will be helpful in Section 4.9.

$G_M^{\mathscr{A}}$

Given a collection of families of assignments $\mathscr{A} = \{A_1, \ldots, A_s\}$, let $G_M^{\mathscr{A}}$ be the *$\mathscr{A}$-adjacency graph* of $M = \{m_j\}_{j \in J}$, that is the bipartite graph with lower part $L(G_M^{\mathscr{A}}) = \{(j,0) : j \in J\}$, upper part $U(G_M^{\mathscr{A}}) = \{(\ell,1) : A_\ell \in \mathscr{A}\}$ and there is an edge $\{(j,0),(\ell,1)\} \in E(G_M^{\mathscr{A}})$ if and only if $\mathrm{var}(m_j) \cap \mathrm{dom}(A_\ell) \neq \emptyset$.

The next lemma shows how from a $\mathscr{C}$-matching in $G_M^{\mathscr{A}}$ we can associate a flippable-product family, hence we are going to use the terminology and notations introduced in Section 3.3, that is flippable families of assignments and r-BG families. Since Lemma 4.13 is a bit general we provide an example showing how a VW-matching F in the adjacency graph G_M leads to flippable families of assignments with domain the variables corresponding to $U(F)$ and zeroing the monomials corresponding to $L(F)$, cf. Table 4.2. To simplify the picture in Table 4.2 we directly identify $U(G_M)$ with the Boolean variables and $L(G_M)$ with the monomials in M.

Lemma 4.13. *Let M be an unsatisfiable set of monomials in $\mathbb{F}[X \cup \overline{X}]$. Suppose we have a collection $\mathscr{A} = \{A_1, \ldots, A_s\}$ of flippable families of assignments in the Boolean variables $X \cup \overline{X}$ that are domain-disjoint. Let $G_M^{\mathscr{A}}$ be the $\mathscr{A}$-adjacency graph of M and F be a $\mathscr{C}$-matching in $G_M^{\mathscr{A}}$ where $\mathscr{C}$ is a collection of trees with all the leaves in $U(G_M^{\mathscr{A}})$[10]. Then there exists a flippable product-family of assignments H_F such that*

1. $H_F \vDash_0 \{m_j : (j,0) \in L(F)\}$,
2. $\mathrm{dom}(H_F) = \bigcup_{(\ell,1) \in U(F)} \mathrm{dom}(A_\ell)$, *and*
3. $\|H_F\|$ *is the number of connected components of F.*

Proof. We prove the result by induction on the number of connected components of F. If F is the union of two disjoint $\mathscr{C}$-matchings F', F'' then by the inductive

[10] In this context of unrooted trees a *leaf* is a vertex of degree 1.

hypothesis $H_{F'} \vDash_0 \{m_j : (j,0) \in L(F')\}$, $\|H_{F'}\|$ is the number of connected components of F' and $\mathrm{dom}(H_{F'}) = \bigcup_{(\ell,1)\in U(F')} \mathrm{dom}(A_\ell)$. As above we get the same for F'', then, since $U(F')$ and $U(F'')$ are disjoint, $H_F = H_{F'} \otimes H_{F''}$ is well-defined. We immediately see that $\|H_F\|$ is the number of connected components of F, $H_F \vDash_0 \{m_j : (j,0) \in L(F)\}$ and $\mathrm{dom}(H_F) = \bigcup_{(\ell,1)\in U(F)} \mathrm{dom}(A_\ell)$.

It remains to consider the case when the $\mathscr{C}$-matching F is just one connected component: a tree with all the leaves in $U(G^{\mathscr{A}}_M)$. In particular we have to show that $H_F \vDash_0 \{m_j : (j,0) \in L(F)\}$ and H_F is flippable. To prove the flippability we prove that for each $(\ell,1) \in U(F)$, each partial assignment in A_ℓ can be extended to an assignment $\beta \in H_F$.

We prove these properties by induction on the size of the tree F: if F is isomorphic to $G_\bullet$ (cf. Figure 4.2) then clearly the statement holds. So consider the minimal possible *non trivial* tree with all the leaves in $U(G^{\mathscr{A}}_M)$ is a tree isomorphic to $G_\vee$ (cf. Figure 4.3). Without loss of generality we can assume that a minimal F is as in Figure 4.7, then, analogously as what we did in Example 3.11, let

$$H_F = \{\alpha \in A_i \otimes A_\ell : \alpha \vDash_0 m_j\}.$$

Since both A_i and A_ℓ are flippable then H_F is non-empty, flippable and $\mathrm{dom}(H_F) = \mathrm{dom}(A_i) \cup \mathrm{dom}(A_\ell)$. Moreover, by construction, $H_F \vDash_0 m_j$ and $\|H_F\| = 1$ and clearly each $\alpha \in A_i$ or A_ℓ can be extended to an assignment in H_F.

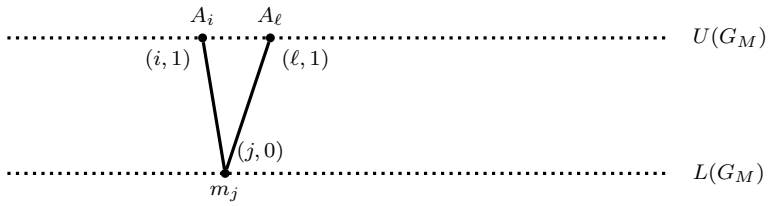


Figure 4.7: From $\mathscr{C}$-matchings to flippable products: a minimal example

Consider now a non-minimal tree F. If the tree F is a star, that is all vertices in $U(F)$ are leaves then we just take two of such leaves and reason as before. Otherwise there exists a vertex $(\ell,1)$ in $U(G_M)$ that is not a leaf of F and such that F is the union of two trees F' and F'' whose vertex-sets intersects only on $(\ell,1)$, cf. Figure 4.8.

By inductive hypothesis, there exists a flippable $H_{F'}$ such that $\|H_{F'}\| = 1$, $\mathrm{dom}(H_{F'}) = \bigcup_{(\ell,1)\in U(F')} \mathrm{dom}(A_\ell)$, $H_{F'} \vDash_0 \{m_j : (j,0) \in L(F')\}$. Similarly we have the same results for $H_{F''}$. Let H_F be the set of assignments obtained 'gluing' together compatible assignments from $H_{F'}$ and $H_{F''}$, more precisely

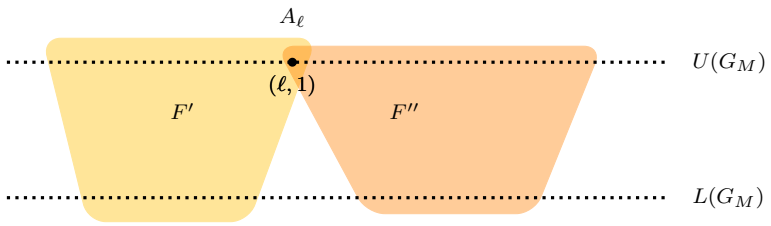


Figure 4.8: From $\mathscr{C}$-matchings to flippable products: inductive step

H_F is the following set of assignments:

$$H_F = \{\gamma : \exists \alpha \in A_\ell \; \exists \beta' \in H_{F'} \; \exists \beta'' \in H_{F''}, \beta' \supseteq \alpha \wedge \beta'' \supseteq \alpha \wedge \gamma = \beta' \cup \beta''\}.$$

Clearly $\|H_F\| = 1$, $\mathrm{dom}(H_F) = \mathrm{dom}(H_{F'}) \cup \mathrm{dom}(H_{F''})$ which in turn is $\bigcup_{(\ell,1)\in U(F)} \mathrm{dom}(A_\ell)$, and

$$H_F \vDash_0 \{m_j : (j,0) \in L(F') \cup L(F'')\} = \{m_j : (j,0) \in L(F)\}.$$

Let now be $i \in U(F) = U(F') \cup U(F'')$, we prove that each assignment in A_i can be extended to an assignment in H_F, hence the inductive hypothesis will be proved and in particular H_F is flippable. Without loss of generality let $i \in U(F')$ and let $\delta \in A_i$. By the inductive property on $H_{F'}$ there exists some $\beta' \in H_{F'}$ such that $\beta' \supseteq \delta$. Since $\mathrm{dom}(H_{F'}) \supseteq \mathrm{dom}(A_\ell)$, there exists $\alpha \in A_\ell$ such that $\alpha = \beta'|_{\mathrm{dom}(A_\ell)}$. By the inductive property on $H_{F''}$ there exists $\beta'' \in H_{F''}$ such that $\beta'' \supseteq \alpha$. Then, by construction, $\gamma = \beta' \cup \beta'' \in H_F$ and clearly $\gamma \supseteq \delta$. □

The next proposition shows that winning strategies for the cover game, played on the $\mathscr{A}$-adjacency graph of a set of monomials using $\mathscr{C}$-matchings, give raise to r-BG families.

Lemma 4.14. *Let J be a set of indices and $M = \{m_j\}_{j\in J}$ be a set of monomials and I a proper ideal in the ring $\mathbb{F}[X \cup \overline{X}]$ with $X = \{x_1, \ldots, x_n\}$. Suppose we have a collection $\mathscr{A}$ of flippable families of assignments $A_1, \ldots, A_s$ in the variables $X \cup \overline{X}$ that are I-consistent and domain-disjoint and let $G_M^{\mathscr{A}}$ the $\mathscr{A}$-adjacency graph of M. If* Cover *wins* $\mathsf{CoverGame}_{\mathscr{C}}(G_M^{\mathscr{A}}, \mu)$ *with $\mathscr{C}$ a collection of trees with no leaves in $L(G_M^{\mathscr{A}})$, then there is a non-empty μ-BG family $\mathscr{F}$ for $M \cup I$ with respect to the ideal I.*

Moreover if for each polynomial $p \in I$ there exists an $A_i \in \mathscr{A}$ such that $\mathrm{var}(p) \subseteq \mathrm{dom}(A_i)$ *then $\mathscr{F}$ is a μ-BG family for $M \cup I$ with respect to the* 0 *ideal.*

Proof. It is straightforward to check that a winning strategy for Cover in $\mathsf{CoverGame}_{\mathscr{C}}(G^{\mathscr{A}}_M, \mu)$ defines, by Lemma 4.13, a family $\mathscr{F}$ of flippable products such that for all flippable products $\mathcal{H} \in \mathscr{F}$

1. for each $\mathcal{H}' \sqsubseteq \mathcal{H}$, $\mathcal{H}' \in \mathscr{F}$;
2. if $\|\mathcal{H}\| < \mu$, then:
   (a) for each $m_j \in M$, there exists a flippable product $\mathcal{H}' \in \mathscr{F}$ such that $\mathcal{H}' \vDash_0 m_j$ and $\mathcal{H}' \sqsupseteq \mathcal{H}$; and
   (b) for each A_i in $\mathscr{A}$, there exists a flippable family $\mathcal{H}' \in \mathscr{F}$ such that $\mathcal{H}' \sqsupseteq \mathcal{H}$ and $\mathrm{dom}(A_i) \subseteq \mathrm{dom}(\mathcal{H}')$.

We claim that $\mathscr{F}$ is a μ-BG family for $M \cup I$ with respect to the ideal I. The *I-consistency* property follows immediately from the I-consistency of the families A_is. The *restriction* property is immediate. For the *extension* property we use the properties in (2) above: for all the monomials in M we just use property 2.(a).

If we have to extend to some polynomial p in I, we do nothing since $\mathcal{H}$ is I-consistent and hence by definition $\mathcal{H} \vDash_I p$. Moreover, if for each $p \in I$ there exists an $A_i \in \mathscr{A}$ such that $\mathrm{var}(p) \subseteq \mathrm{dom}(A_i)$ then $\mathscr{F}$ is a μ-BG family for $M \cup I$ with respect to the 0 ideal. We have just to check the *extension* property when $p \in I$. In this case, by hypothesis, there exists some $A_i \in \mathscr{A}$ such that $\mathrm{var}(p) \subseteq \mathrm{dom}(A_i)$. By property 2.(b) there exists some $\mathcal{H}'$ in $\mathscr{F}$ such that $\mathrm{dom}(A_i) \subseteq \mathrm{dom}(\mathcal{H}')$, hence, for each partial assignment $\alpha \in \mathcal{H}'$, $\alpha(p) \in \mathbb{F}$. But since $\mathcal{H}'$ is I-consistent and $p \in I$ then $\alpha(p) \in I$. Since I is a proper ideal $\alpha(p) \in I \cap \mathbb{F} = 0$ and hence $\alpha \vDash_0 p$. □

We now present two winning strategies, one for the game $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$, cf. Section 4.6.3, and the other for the game $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$, cf. Section 4.6.4.

The proofs of those results are modeled on the analogous results in [31, 40] but they are also similar to constructions that can be found in the literature for matchings for example in [7, 24]. The overall structures of the proofs of the winning strategies both for $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$ and for $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ are analogous, but the former one is simpler and hence is somehow preliminary to the latter one.

4.6.3 A winning strategy for $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$

The next theorem shows that Cover has a winning strategy for the game $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$ for expander graphs G with appropriately chosen parameters.

Theorem 4.15. *Let G be a bipartite graph, s a positive integer and $\epsilon > 0$ a real number. Moreover let d be the max degree of a vertex in $L(G)$ and $\mu = \frac{\epsilon s}{2d^2}$. Suppose that the following two properties hold*

1. *G is a $(s, 2+\epsilon)$-bipartite expander;*
2. *the max degree of a vertex in $U(G)$ is at most μ.*

Then Cover wins $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$.

Before giving the proof of Theorem 4.15 we need some preliminary lemmas. To simplify the exposition in this subsection we consider fixed a bipartite graph G, an integer s and a real number $\epsilon > 0$ such that G is an $(s, 2+\epsilon)$-bipartite expander. For shortness let $L = L(G)$, $U = U(G)$ and d be the maximum degree of a vertex in L. Given $A \subseteq L$ and $B \subseteq U$, we let $G_{A,B}$ be the subgraph of G induced by $(L \setminus A) \cup (U \setminus B)$.

$G_{A,B}$

Definition 4.16 (V-matching property)**.** *Given two sets $A \subseteq L$ and $B \subseteq U$, we say that the pair (A, B) has the* V-matching property*, if for every $C \subseteq L \setminus A$ with $|C| \leqslant s$, there exists a V-matching F in $G_{A,B}$ covering C.*

V-matching property

Lemma 4.17. *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) does not have the V-matching property. Then there exists a set $C \subseteq L \setminus A$ with $\epsilon|C| < |B|$, such that no V-matching in $G_{A,B}$ covers C.*

Proof. Take $C \subseteq L \setminus A$ of minimal size such that no V-matching in $G_{A,B}$ covers C. We have that $|C| \leqslant s$ and, by minimality of C and Theorem 3.13, it follows that

$$|N_{G_{A,B}}(C)| < 2|C|.$$

But, by hypothesis, G is an $(s, 2+\epsilon)$-bipartite expander; hence $(2+\epsilon)|C| \leqslant |N_G(C)|$. Therefore,

$$(2+\epsilon)|C| \leqslant |N_G(C)| \leqslant |N_{G_{A,B}}(C)| + |B| < 2|C| + |B|.$$

Hence $\epsilon|C| < |B|$, as required. □

Lemma 4.17 is the only place where we directly use the Hall's theorem for V-matchings, cf. Theorem 3.13 from the previous chapter. However, Lemma 4.17 itself plays a crucial role in proving the following lemmas.

Lemma 4.18. *The pair $(\emptyset, \emptyset)$ has the V-matching property.*

Proof. By contradiction suppose that $(\emptyset, \emptyset)$ has not the V-matching property, then, by Lemma 4.17, there exist a set $C \subseteq L \setminus A$ that has no V-matching in $G_{A,B}$ covering C and has *negative* size. Which is clearly not possible. □

Lemma 4.19 (component removal)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* V*-matching property and*

$$|B| \leqslant \epsilon s. \tag{4.10}$$

Then for each V*-matching F contained in the subgraph of G induced by $A \cup B$, we have that $(A \setminus L(F), B \setminus U(F))$ has the* V*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.9.

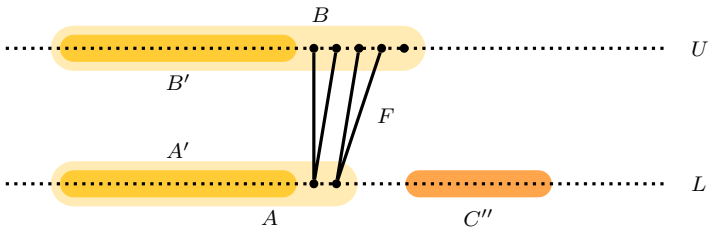


Figure 4.9: Component removal for V-matchings

Proof. Let $A' = A \setminus L(F)$ and $B' = B \setminus U(F)$ and suppose, by contradiction, that (A', B') does not have the V-matching property. By Lemma 4.17, it is sufficient to prove that for each set $C \subseteq L \setminus A'$ with $\epsilon|C| < |B'|$, there is a V-matching in $G_{A',B'}$ covering C. Let $C' = C \cap L(F)$ and $C'' = C \setminus C'$. By construction, F is a V-matching such that $L(F) \subseteq A$, $U(F) \subseteq B$ and F covers C'. Moreover, we have that

$$|C''| \leqslant |C| < \frac{1}{\epsilon}|B'| < \frac{1}{\epsilon}|B| \overset{\text{eq. } (4.10)}{\leqslant} s.$$

Hence there exists a V-matching F'' of C'' in $G_{A,B}$, and since F and F'' are vertex-disjoint, then $F \cup F''$ is a V-matching in $G_{A',B'}$. By construction $F \cup F''$ covers C. □

Lemma 4.20 (covering a vertex in L)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* V*-matching property. If*

$$d^2(|B| + 2) \leqslant 2\epsilon s, \tag{4.11}$$

then for each vertex v in $L \setminus A$, there exists a V*-matching F in $G_{A,B}$ covering v and such that $(A \cup L(F), B \cup U(F))$ has the* V*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.10 on the next page.

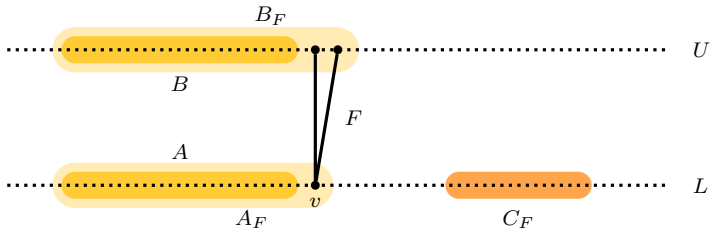


Figure 4.10: Covering a vertex in L *via* V-matchings

Proof. Fix $v \in L \setminus A$ and let S be the set of all V-matchings F in $G_{A,B}$, covering v and such that F has a single connected component, that is F is isomorphic to the graph G_{V} from Figure 4.3.

Since $1 \leqslant s$ and (A, B) has the V-matching property, we know that S is non-empty. For every $F \in S$, let (A_F, B_F) be the pair $(A \cup L(F), B \cup U(F))$, and suppose, for sake of contradiction, that for every $F \in S$, the pair (A_F, B_F) doesn't have the V-matching property. By Lemma 4.17, for every $F \in S$ there exists a set $C_F \subseteq L \setminus A_F$ with $|C_F| < \frac{1}{\epsilon}|B_F|$ and such that there is no V-matching of C_F in G_{A_F,B_F}.

Let $C = \bigcup_{F \in S} C_F$. Then

$$|C| \leqslant \sum_{F \in S} |C_F| < \frac{1}{\epsilon} \sum_{F \in S} |B_F| \leqslant \frac{1}{\epsilon}|S|(|B| + 2) \leqslant \frac{d^2}{2\epsilon}(|B| + 2) \overset{\text{eq. (4.11)}}{\leqslant} s,$$

since $|S| \leqslant \binom{d}{2} \leqslant \frac{d^2}{2}$ and $|B_F| = |B| + 2$. Hence $|C \cup \{v\}| \leqslant s$. Furthermore, $C \cup \{v\} \subseteq L \setminus A$, so by the fact that (A, B) has the V-matching property, there exists a V-matching F' covering $C \cup \{v\}$ in $G_{A,B}$.

There must be some $F \in S$ such that F is a connected component of F'. Let F'' be F' with the component F removed. Then F'' is a V-matching in G_{A_F,B_F} and F'' covers C_F, contradicting the choice of C_F. □

Lemma 4.21 (covering a vertex in U)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* V*-matching property and let v a vertex in $U \setminus B$ with degree e. If*

$$d^2(|B| + 2e) \leqslant 2\epsilon s, \tag{4.12}$$

then there is a V*-matching F in $G_{A,B}$ covering v and such that the pair $(A \cup L(F), B \cup U(F))$ has the* V*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.11 on the facing page.

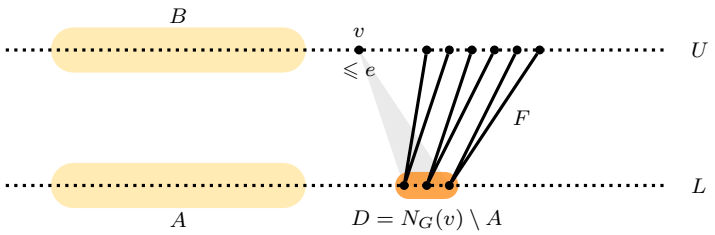


Figure 4.11: Covering a vertex in U *via* V-matchings

Proof. Fix $v \in U \setminus B$ and let D be $N_G(v) \setminus A$. If $|D| = 0$, then $N_G(v) \subseteq A$, and so we can cover v by taking F to be the V-matching consisting only of the vertex v. Since $v \in U$, this is a valid V-matching covering v and clearly $(A \cup L(F), B \cup U(F))$ has the V-matching property.

If $|D| > 0$, by hypothesis $|D| \leqslant e$ and hence, by the cardinality condition on B, cf. equation (4.12), we can apply Lemma 4.20 $|D|$ times obtaining a V-matching F in $G_{A,B}$ covering D and such that $(A \cup L(F), B \cup U(F))$ has the V-matching property.

Now, since $N_G(v) \subseteq A \cup L(F)$, it follows that $(A \cup L(F), B \cup U(F) \cup \{v\})$ has the V-matching property. Either v is covered by F, or it is possible to add $\{v\}$ as a new connected component to F while still maintaining the property of being a V-matching in $G_{A,B}$. □

We now have all the preliminary lemmas needed to prove Theorem 4.15[11], restated below for convenience of the reader.

Restated Theorem 4.15. *Let G be a bipartite graph, s a positive integer and $\epsilon > 0$ a real number. Moreover let d be the max degree of a vertex in $L(G)$ and $\mu = \frac{\epsilon s}{2d^2}$. Suppose that the following two properties hold*

1. *G is a $(s, 2+\epsilon)$-bipartite expander;*
2. *the max degree of a vertex in $U(G)$ is at most μ.*

Then Cover *wins* $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$.

Proof of Theorem 4.15. Let $\mathscr{L}$ to be the set of all V-matchings F in G such that $(L(F), U(F))$ has the V-matching property, and $|U(F)| \leqslant \epsilon s$.

We claim that Cover can use the V-matchings in $\mathscr{L}$ to win $\mathsf{CoverGame}_{\mathsf{V}}(G, \mu)$. By Lemma 4.18 the empty V-matching is in $\mathscr{L}$ and hence $\mathscr{L}$ is non-empty. Moreover, $\mathscr{L}$ is closed under removing connected components by Lemma 4.19.

[11] This theorem is from [40].

Suppose now that at step $i+1$ of the game Choose picks a vertex v in $G_{L(F_i),U(F_i)}$ and that F_i has strictly less than $\mu = \frac{\epsilon s}{2d^2}$ connected components. Then, $(L(F_i), U(F_i))$ satisfies both the cardinality constraints of Lemma 4.20 and Lemma 4.21. Let r is the max degree of a vertex in $U(G)$:

$$d^2(|U(F_i)| + 2r) \leqslant d^2(2\mu + 2r) \tag{4.13}$$
$$\leqslant d^2(4\mu) \tag{4.14}$$
$$= 2\epsilon s. \tag{4.15}$$

The inequality (4.13) follows from the fact that $|U(F_i)| \leqslant 2\mu$, and the inequality (4.14) follows by the hypothesis that $r \leqslant \mu$. The last equality is just the hypothesis on μ.

If v is covered by F_i we take $F_{i+1} = F_i$. Otherwise, by Lemma 4.20 and Lemma 4.21 applied to $(L(F_i), U(F_i))$, there exists a V-matching F_{i+1} extending F_i by a new connected component covering v such that $(L(F_{i+1}), U(F_{i+1}))$ has the V-matching property. From the previous chain of inequalities, it follows easily that the pair $(L(F_{i+1}), U(F_{i+1}))$ satisfies the cardinality condition $|U(F_{i+1})| \leqslant \epsilon s$. □

4.6.4 A winning strategy for $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$

The next theorem shows that Cover has a winning strategy for the game $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ for expander graphs G with appropriately chosen parameters.

Theorem 4.22. *Let G be a bipartite graph, s, D be integers, and $\epsilon < \frac{1}{5}$ be a real number. For every integer $d \geqslant D$ let S_d be the set of vertices of $U(G)$ with degree bigger than d. Suppose that*

1. *each vertex in $L(G)$ has degree at most 3;*
2. *G is an $(s, 2 - \frac{\epsilon}{2})$-bipartite expander[12];*
3. *for every $D_{\max} \geqslant d \geqslant D$, $144d(|S_d|+d) \leqslant \epsilon s$, where $D_{\max}$ is the maximum degree of a vertex in $U(G)$.*

Then Cover wins the cover game $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ with $\mu = \frac{\epsilon s}{144D}$.

The proof of this theorem is analogous to the proof that we have seen for Theorem 4.15, but there are some non-trivial small changes in some crucial preliminary lemmas we need.

[12]This hypothesis implies that no pair of degree 3 vertices in $L(G)$ have the same set of neighbors, in fact if $A \subseteq L(G)$ has size 2 then $|N(G)| \geqslant (2 - \frac{1}{10})2 = 3.8 > 3$.

To simplify the exposition in this subsection we consider fixed a bipartite graph G, an integer s and a real number $\epsilon < 1/5$ such that G is an $(s, 2-\frac{\epsilon}{2})$-bipartite expander. For shortness let $L = L(G)$, $U = U(G)$ and each vertex in L has degree at most 3. As done in the previous section, given $A \subseteq L$ and $B \subseteq U$, we let $G_{A,B}$ be the subgraph of G induced by $(L \setminus A) \cup (U \setminus B)$. $G_{A,B}$

Definition 4.23 (VW-matching property)**.** *Given two sets $A \subseteq L$ and $B \subseteq U$, we say that the pair (A, B) has the* VW-matching property*, if for every $C \subseteq L \setminus A$ with $|C| \leqslant s$, there exists a* VW*-matching F in $G_{A,B}$ covering C.* VW-matching property

Lemma 4.24. *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) does not have the* VW*-matching property. Then there exists a set $C \subseteq L \setminus A$ with $|C| < \frac{2}{\epsilon}|B|$, such that no* VW*-matching in $G_{A,B}$ covers C.*

Proof. Take $C \subseteq L \setminus A$ of minimal size such that no VW-matching in $G_{A,B}$ covers C. We have that $|C| \leqslant s$ and by minimality of C and Theorem 4.11 it follows that

$$|N_{G_{A,B}}(C)| < (2-\epsilon)|C|.$$

But, by hypothesis, G is an $(s, 2-\frac{\epsilon}{2})$-bipartite expander, hence

$$(2-\frac{\epsilon}{2})|C| \leqslant |N_G(C)|.$$

Therefore we have the following chain of inequalities

$$(2-\frac{\epsilon}{2})|C| \leqslant |N_G(C)| \leqslant |N_{G_{A,B}}(C)| + |B| < (2-\epsilon)|C| + |B|.$$

Hence $|C| < \frac{2}{\epsilon}|B|$, as required. □

Lemma 4.24 is the only place where we directly use the version of Hall's Theorem for VW-matchings, cf. Theorem 4.11. However, Lemma 4.24 itself plays a crucial role in proving the following lemmas.

Lemma 4.25. *The pair $(\emptyset, \emptyset)$ has the* VW*-matching property.*

Proof. By contradiction suppose that $(\emptyset, \emptyset)$ has not the VW-matching property, then, by Lemma 4.24, there exists a set $C \subseteq L \setminus A$ that has no V-matching in $G_{A,B}$ covering C and has *negative* size. Which is clearly not possible. □

The proof of the following lemma is similar to the proof of Lemma 4.19, but, for making this subsection as self contained as possible, we re-prove it in the context of VW-matchings.

Lemma 4.26 (component removal)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* VW*-matching property and*

$$2|B| \leqslant \epsilon s. \tag{4.16}$$

Then for each VW*-matching F contained in the subgraph of G induced by $A \cup B$, $(A \setminus L(F), B \setminus U(F))$ has the* VW*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.12.

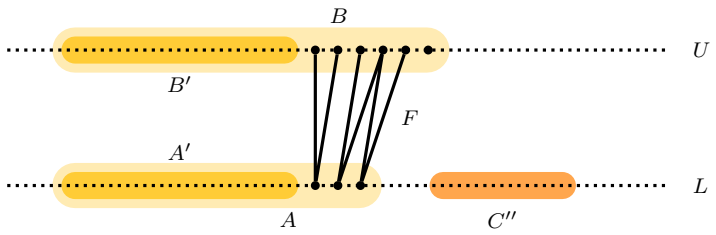


Figure 4.12: Component removal for VW-matchings

Proof. Let $A' = A \setminus L(F)$ and $B' = B \setminus U(F)$ and suppose, by contradiction, that (A', B') does not have the VW-matching property. By Lemma 4.24, it is sufficient to prove that for each set $C \subseteq L \setminus A'$ with $|C| < \frac{2}{\epsilon}|B'|$, there is a VW-matching in $G_{A',B'}$ covering C. Let $C' = C \cap L(F)$ and $C'' = C \setminus C'$. By construction, F is a VW-matching such that $L(F) \subseteq A$, $U(F) \subseteq B$ and F covers C'. Moreover, we have that

$$|C''| \leqslant |C| < \frac{2}{\epsilon}|B'| < \frac{2}{\epsilon}|B| \overset{\text{eq. (4.16)}}{\leqslant} s.$$

Hence there exists a VW-matching F'' of C'' in $G_{A,B}$, and so $F \cup F''$ is a VW-matching covering C in $G_{A',B'}$. □

Lemma 4.27 (covering a vertex in L)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* VW*-matching property and let d be the maximum degree of a vertex in $U \setminus B$. If*

$$24d(|B| + 3) \leqslant \epsilon s, \tag{4.17}$$

then for each vertex v in $L \setminus A$, there is a VW*-matching F in $G_{A,B}$ covering v and such that $(A \cup L(F), B \cup U(F))$ has the* VW*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.13 on the facing page.

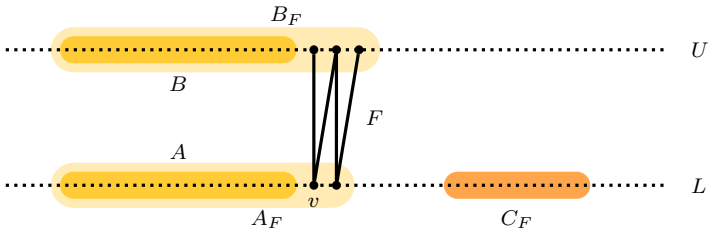


Figure 4.13: Covering a vertex in L via VW-matchings

Proof. Fix $v \in L \setminus A$ and let S be the set of all VW-matchings F in $G_{A,B}$, covering v and such that F is connected.

Since $1 \leqslant s$ and (A, B) has the VW-matching property, we know that S is non-empty. For every $F \in S$, let (A_F, B_F) be the pair $(A \cup L(F), B \cup U(F))$, and suppose for a contradiction that for every $F \in S$, (A_F, B_F) does not have the VW-matching property. By Lemma 4.24, for every $F \in S$ there is a set $C_F \subseteq L \setminus A_F$ with $|C_F| < \frac{2}{\epsilon}|B_F|$ and such that there is no VW-matching of C_F in G_{A_F,B_F}. Let $C = \bigcup_{F\in\Pi} C_F$. Then

$$|C| \leqslant \sum_{F\in S} |C_F| < \frac{2}{\epsilon} \sum_{F\in S} |B_F| \leqslant \frac{2}{\epsilon}|S|(|B| + 3) \leqslant \frac{2}{\epsilon}12d(|B| + 3),$$

since there at most three V-matchings covering v and at most $3 \cdot 2 \cdot (d-1) \cdot 2$ W-matchings covering v, we have that $|S| \leqslant 3 + 3 \cdot 2 \cdot (d-1) \cdot 2 \leqslant 12d$ and moreover $|B_F| \leqslant |B| + 3$. Hence, by equation (4.17), we have that $|C \cup \{v\}| \leqslant s$. Furthermore, $C \cup \{v\} \subseteq L \setminus A$, so by the fact that (A, B) has the VW-matching property, there is a VW-matching F' covering $C \cup \{v\}$ in $G_{A,B}$.

There must be some $F \in S$ such that F is a connected component of F'. Let F'' be F' with the component F removed. Then F'' is a VW-matching in G_{A_F,B_F} and F'' covers C_F, contradicting the choice of C_F. □

Lemma 4.28 (covering a vertex in U)**.** *Let $A \subseteq L$ and $B \subseteq U$ be such that the pair (A, B) has the* VW*-matching property and let d be the maximum degree of a vertex in $U \setminus B$. If*

$$24d(|B| + 3d) \leqslant \epsilon s, \tag{4.18}$$

then for each vertex v in $U \setminus B$, there is a VW*-matching F in $G_{A,B}$ covering v and such that $(A \cup L(F), B \cup U(F))$ has the* VW*-matching property.*

A visual hint for the notations used in this proof can be found in Figure 4.14 on the next page.

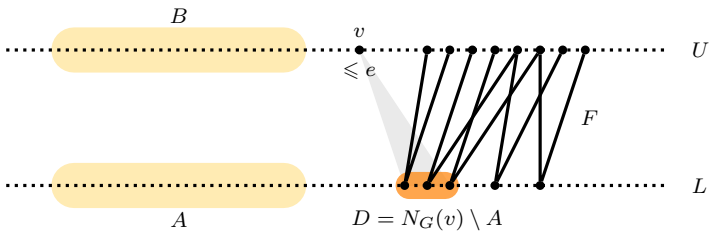


Figure 4.14: Covering a vertex in U *via* VW-matchings

Proof. Fix $v \in U \setminus B$ and let D be $N_G(v) \setminus A$. If $|D| = 0$, then $N_G(v) \subseteq A$, and so we can cover v by taking F to be the VW-matching consisting only of the vertex v. Since $v \in U$, this is a valid VW-matching covering v and clearly $(A \cup L(F), B \cup U(F))$ has the VW-matching property.

If $|D| > 0$, since by hypothesis $|D| \leqslant d$ and by (4.18), we can apply Lemma 4.27 $|D|$ times obtaining a VW-matching F in $G_{A,B}$ covering D and such that $(A \cup L(F), B \cup U(F))$ has the VW-matching property.

Now, since $N_G(v) \subseteq A \cup L(F)$, it follows that $(A \cup L(F), B \cup U(F) \cup \{v\})$ has the VW-matching property. Either v is covered by F, or it is possible to add $\{v\}$ as a new connected component to F while still maintaining the property of being a VW-matching in $G_{A,B}$. □

We now have all the preliminary lemmas needed to prove Theorem 4.22[13], restated below for the convenience of the reader.

Restated Theorem 4.22. *Let G be a bipartite graph, s, D be integers, and $\epsilon < \frac{1}{5}$ be a real number. For every integer $d \geqslant D$ let S_d be the set of vertices of $U(G)$ with degree bigger than d. Suppose that*

1. *each vertex in $L(G)$ has degree at most 3;*
2. *G is an $(s, 2 - \frac{\epsilon}{2})$-bipartite expander[14];*
3. *for every $D_{\max} \geqslant d \geqslant D$, $144d(|S_d|+d) \leqslant \epsilon s$, where $D_{\max}$ is the maximum degree of a vertex in $U(G)$.*

Then Cover wins the cover game $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$ with $\mu = \frac{\epsilon s}{144D}$.

Proof of Theorem 4.22. By the hypothesis on $|S_d|$, for each $D_{\max} \geqslant d \geqslant D$ we can repeatedly apply Lemma 4.28 starting from $(\emptyset, \emptyset)$ to cover vertices in U

[13]The proof of this theorem is from [31].

[14]This hypothesis implies that no pair of degree 3 vertices in $L(G)$ have the same set of neighbors, in fact if $A \subseteq L(G)$ has size 2 then $|N(G)| \geqslant (2 - \frac{1}{10})2 = 3.8 > 3$.

of degree larger than D. By starting from vertices of U of maximum degree and proceeding in decreasing order until reaching the vertices of degree D, we can build a VW-matching M covering S_D such that $(L(M), U(M))$ has the VW-matching property. Moreover, by the choice of S_D, $G_{L(M),U(M)}$ (the subgraph induced by $(L \cup U) \setminus (L(M) \cup U(M))$) has degree at most D. We say that a VW-matching F is *compatible* with M if each connected component of F is either a connected component of M or disjoint from all connected components of M.

Let $\mathscr{L}$ to be the set of all VW-matchings F in G compatible with M such that $(L(M) \cup L(F), U(M) \cup U(F))$ has the VW-matching property, and $|U(M) \cup U(F)| \leqslant \frac{\epsilon s}{2}$.

We show now that Cover can use the VW-matchings in $\mathscr{L}$ to win the game $\mathsf{CoverGame}_{\mathsf{VW}}(G, \mu)$. This family is non-empty since, by Lemma 4.25 the empty VW-matching is in $\mathscr{L}$. Moreover, $\mathscr{L}$ is closed under removing connected components by Lemma 4.26. Suppose now that at step $i+1$ of the game Choose picks a vertex v in $G_{L(M),U(M)}$ and that F_i has strictly less than $\mu = \frac{\epsilon s}{144D}$ components. Then, $(L(M) \cup L(F_i), U(M) \cup U(F_i))$ satisfies the hypotheses of Lemma 4.27 and Lemma 4.28:

$$\begin{aligned} 24D(|U(M) \cup U(F_i)| + 3D) &\leqslant 24D(|U(M)| + 3D) + 24D|R(F_i)| && (4.19)\\ &\leqslant 24D(3|S_D| + 3D) + 72D\mu && (4.20)\\ &\leqslant \frac{\epsilon s}{2} + 72D\mu = \frac{\epsilon s}{2} + 72D\frac{\epsilon s}{144D} = \epsilon s, && (4.21) \end{aligned}$$

where the inequality (4.20) follows from the fact that $|U(F_i)| \leqslant 3\mu$ and $|U(M)| \leqslant 3|S_D|$, where S_D is the set of vertices in U of degree bigger than D. The inequality (4.21) follows by the hypothesis on the size of S_D.

Hence, if v is covered by F_i we take $F_{i+1} = F_i$. If v is covered by M we take $F_{i+1} = F_i \cup M_v$, where M_v is the connected component of M covering v. Otherwise, by Lemma 4.27 and Lemma 4.28 applied to $(L(M) \cup L(F_i), U(M) \cup U(F_i))$, there exists a VW-matching F_{i+1} extending $F_i \cup M$ by a new connected component covering v such that $(L(F_{i+1}), U(F_{i+1}))$ has the VW-matching property. From the previous chain of inequalities, it follows easily that the pair $(L(F_{i+1}), U(F_{i+1}))$ satisfies the cardinality condition $|U(M) \cup U(F_{i+1})| = |U(F_{i+1})| \leqslant \frac{\epsilon s}{2}$. □

4.7 Random bipartite graphs

We say that a graph is a (n, d, Δ)-*random bipartite graph* if it is chosen according to the uniform distribution on the set of bipartite graphs G such that $U(G) = n$

(n, d, Δ)-random bipartite graph

and $L(G) = \Delta n$ and the maximum degree of a vertex in $L(G)$ is exactly d. The proof of the next theorem is standard and can be found for instance in [13, 24, 30, 54, 83].

Theorem 4.29 (Ben-Sasson and Galesi [24, Lemma 5.1])**.** *For any $d \geqslant 3$, $\Delta \geqslant 1$ and any real constant $\epsilon \in (0, d-2)$, there is a constant $\gamma = \gamma_{d,\epsilon,\Delta}$ such that, for large n if G is a (n, d, Δ)-random bipartite graph then, with high probability, G is a $(\gamma n, 1+\epsilon)$-bipartite expander.*

Lemma 4.30. *Let G be a (n, d, Δ)-random bipartite graph with Δ and d positive constants. Then, with high probability, there is no vertex in $U(G)$ of degree bigger than $\log n$.*

Proof. The expected number of vertices in $U(G)$ of degree at least $\log n$ is at most

$$n\binom{\Delta n}{\log n}\left(\frac{\binom{n-1}{d-1}}{\binom{n}{d}}\right)^{\log n} \leqslant n\left(\frac{e\Delta n}{\log n}\right)^{\log n}\left(\frac{d}{n}\right)^{\log n} = o(1).$$

So, with high probability, there are no such vertices. □

Theorem 4.31. *Let $d \geqslant 4$, $\Delta \geqslant 1$ and G a (n, d, Δ)-random bipartite graph then, for large n, with high probability there exists a constant γ such that* Cover *has a winning strategy for* $\mathsf{CoverGame}_{\mathsf{V}}(G, \gamma n)$.

Proof. Fix $\epsilon = 1.5$ then, by Theorem 4.29, with high probability for large n there exists a constant $\gamma = \gamma_{d,\epsilon,\Delta}$ such that G is a $(\gamma n, 2.5)$-bipartite expander. Moreover, by Lemma 4.30, no vertex in $U(G)$ has degree bigger than $\log n$ and henceforth, for large n, no vertex in $U(G)$ has degree bigger than γn. Hence, for large n, with high probability, G satisfies the hypotheses of Theorem 4.15, hence Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{V}}(G, \gamma n)$. □

Clearly the setting of $\epsilon = 1.5$ is not strictly necessary in the previous proof. Any $\epsilon > 1$ would have been equally good but the hypothesis $d \geqslant 4$ was crucial. Let $\Delta > 0$ a constant and G be a $(n, 3, \Delta)$-random bipartite graph: it is not true that Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{V}}(G, \gamma n)$ for some constant γ but, with high probability Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{VW}}(G, \gamma n)$ for some constant γ, cf. Theorem 4.33. In order to prove this result we have to show a preliminary lemma bounding the number of high degree vertices of $U(G)$. The next lemma is from [31].

Lemma 4.32. *Let Δ be a constant and G be a $(n, 3, \Delta)$-random bipartite graph. For every integer d, let $S_d = \{v \in U(G) : \deg_G(v) \geqslant d\}$. Then for every real constant $c > 0$, with high probability for sufficiently large n there exists a constant D such that for every $\log n \geqslant d \geqslant D$,*

$$d(|S_d| + d) \leqslant cn. \tag{4.22}$$

Proof. We claim that for every $\log n \geqslant d \geqslant 12e\Delta$, with high probability

$$|S_d| \leqslant \frac{en}{2^d}. \tag{4.23}$$

Before proving equation (4.23), we show how to conclude the desired bound on S_d. Fix a positive constant c and let $\log n \geqslant D \geqslant 12e\Delta$ big enough to have that $\frac{eD}{2^D} \leqslant c/2$. Moreover, for sufficiently large n, we have also that $\log^2 n \leqslant cn/2$. For d such that $\log n \geqslant d \geqslant D$ we have the following chain of inequalities:

$$d(|S_d| + d) \overset{\text{eq. (4.23)}}{\leqslant} \frac{end}{2^d} + d^2 \leqslant \frac{eDn}{2^D} + \log^2 n \leqslant \frac{cn}{2} + \frac{cn}{2} = cn.$$

It remains to show just equation (4.23). Consider $\log n \geqslant d \geqslant 24e\Delta$. The probability that there are at least $\frac{en}{2^d}$ many variable nodes of degree at least d is at most

$$\Pr\left[|S_d| \geqslant \frac{en}{2^d}\right] \leqslant \binom{n}{\frac{en}{2^d}} \left[\binom{\Delta n}{d}\left(\frac{3}{n}\right)^d\right]^{\frac{en}{2^d}}. \tag{4.24}$$

$$\leqslant \left[2^d \left(\frac{e\Delta n}{d}\right)^d \left(\frac{3}{n}\right)^d\right]^{\frac{en}{2^d}} \tag{4.25}$$

$$\leqslant \left(\frac{6e\Delta}{d}\right)^{\frac{edn}{2^d}} \tag{4.26}$$

$$\leqslant \left(\frac{1}{2}\right)^{\frac{edn}{2^d}} \tag{4.27}$$

$$= o(1), \tag{4.28}$$

where the equation (4.28) holds since $\log n \geqslant d \geqslant 12e\Delta$, and we used the standard estimation $\binom{n}{m} \leqslant \left(\frac{en}{m}\right)^m$. □

Theorem 4.33. *Let $\Delta \geqslant 1$ and G a $(n, 3, \Delta)$-random bipartite graph then, for large n, with high probability there exists a constant γ such that* Cover *has a winning strategy* $\mathsf{CoverGame}_{\mathsf{VW}}(G, \gamma n)$.

Proof. Fix $\epsilon = 0.95$ then, by Theorem 4.29, with high probability for large n there exists a constant $\gamma' = \gamma'_{\epsilon,\Delta}$ such that G is a $(\gamma' n, 1.95)$-bipartite expander.

Moreover,with high probability, by Lemma 4.30, the maximum degree of a vertex in $U(G)$ is $\log n$ and, by Lemma 4.32, for large enough n there exists a constant D such that for every $\log n \geqslant d \geqslant D$,

$$144d(|S_d| + d) \leqslant \epsilon\gamma' n,$$

where $S_d = \{v \in U(G) : \deg_G(v) \geqslant d\}$. Hence, for large n, with high probability G satisfies the hypotheses of Theorem 4.22, and then Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{VW}}(G, \gamma n)$ with $\gamma = \frac{\epsilon\gamma}{144D}$. □

Unlike Theorem 4.31, in the previous proof, the choice of ϵ is not completely arbitrary. To apply Theorem 4.11 and hence Theorem 4.22 we need to have $\epsilon > 0.9$, any $\epsilon \in (0.9, 1)$ would have been equally good in the previous proof.

4.8 Random *k*-CNF formulas

(n, k, Δ)-random CNF

We recall that given a positive integer k and a positive real number Δ, a (n, k, Δ)-*random CNF* is a k-CNF formula with n variables and Δn clauses picked uniformly at random from the set of all CNF formulas in the variables $\{x_1, \ldots, x_n\}$ which consist of exactly Δn clauses, each clause containing exactly k literals and no variable appearing twice in a clause.

A fundamental conjecture about the (n, k, Δ)-random CNF formula model, cf. [45, 50, 54, 72, 90], says that there exists a constant θ_k, the *satisfiability threshold*, such that if $\Delta > \theta_k$ then a (n, k, Δ)-random CNF formula is almost surely unsatisfiable, while if $\Delta < \theta_k$ then a (n, k, Δ)-random CNF formula is almost surely satisfiable. Friedgut [71] showed that for each n there exists a threshold $\theta_k(n)$ with such property. It is known that $\theta_2(n) = 1$, cf. [53, 64, 76], and that for each n, $\theta_k(n)$ is bounded between two constants that are independent of n, e.g. $3.003 \leqslant \theta_3(n) \leqslant 4.598$, cf. [72, 90]. For large k there exists an explicit constant γ_k not depending on n such that $\theta_k(n) = \gamma_k$, cf. [65].

Chvátal and Szemerédi [54] showed that every (n, k, Δ)-random CNF formula, with Δ a constant such that $\Delta > \theta_k$, is extremely hard for Resolution to refute, that is every Resolution refutation is $2^{\Omega(n)}$. On the other hand Beame et al. [17] showed that Resolution, for $k = 3$, produces refutations of size at most $2^{O(n/\Delta)}$, which is polynomial when $\Delta \geqslant n/\log n$. Indeed it is believed that (n, k, Δ)-random CNF formulas with Δ close to the satisfiability threshold θ_k are the ones that are most computationally hard, cf. e.g. [51].

The importance of [54] relies in showing that resolution is a very weak proof systems, in the sense that almost all 3-CNF formulas require exponential size Resolution refutations. Then the hardness of (n, k, Δ)-random CNF formulas

has been investigated in depth in proof complexity, in particular the lower bounds in [54] was improved and simplified by Beame and Pitassi [13], improved for a $\Delta = o(n^{1/4})$ by Beame et al. [18] and simplified using the size-width relation by Ben-Sasson and Wigderson [30]. All these results, as well the one we show in this section, hold for $k \geqslant 3$[15]. The (n, k, Δ)-random CNF formulas have been shown to be hard to refute also for the Polynomial Calculus, cf. [3, 26] and for $\mathsf{Res}(k)$, a version of Res manipulating k-DNF formulas, cf. [2].

With respect to space they have been shown to require large clause space in Resolution, more precisely given a (n, k, Δ)-random CNF formula φ,

$$\mathsf{CSpace}(\varphi \vdash \bot) \geqslant \Omega(n/\Delta^{1+\epsilon}),$$

cf. [24] while on the other hand Zito [143] showed an upper bound on clause space

$$\mathsf{CSpace}(\varphi \vdash \bot) \leqslant O(n\Delta^{-1/(k-2)}).$$

In this section we further deepen the understanding of the space complexity of random k-CNF formulas proving total space lower bounds in resolution and monomial space lower bounds in Polynomial Calculus, cf. Theorem 4.36. For simplicity we focus on the case where Δ is a constant.

If φ is a (n, k, Δ)-random CNF then the clauses-variables adjacency graph on φ, G_φ, is a (n, k, Δ)-random bipartite graph. The same holds if we consider the adjacency graph G_M on the set of monomials $\{tr(C) : C \in \varphi\}$. The formal definition of adjacency graph is the following.

Definition 4.34 (adjacency graph)**.** *Let $J \subseteq \mathbb{N}$ be a set of indices and let $\varphi = \bigvee_{j \in J} C_j$ be a CNF formula in the variables $\{x_1, \ldots, x_n\}$. The* adjacency graph of φ*, G_φ, is the bipartite graph with vertex-set $(J \times \{0\}) \cup ([n] \times \{1\})$ and $\{(j, 0), (i, 1)\} \in E(G_\varphi)$ if and only if $x_i \in \mathrm{var}(C_j)$. Hence the clauses of φ can be identified with the lower part of G_φ and variables with the upper part of G_φ.* G_φ

The following result was proven in [31, 37].

Theorem 4.35. *Let $k \geqslant 3$ and $\Delta > \theta_k$. If φ is a (n, k, Δ)-random CNF, then there is a constant $\gamma > 0$ and non-empty (γn)-BG family for $tr(\varphi)$ with respect to the 0 ideal and with respect to the ideal generated by the Boolean axioms.*

Proof. Let M be the translation of the clauses of φ into a set of polynomials (actually monomials) in the ring $\mathbb{F}[X \cup \overline{X}]$, that is $M = \{tr(C) : C \in \varphi\}$.

[15] For completeness we recall that $(n, 2, \Delta)$-random CNF formulas are easy for Resolution since 2-SAT is in P and the easy polynomial time algorithm to solve it can be formalized in Resolution.

Since φ is a (n, k, Δ)-random CNF, then the adjacency graph of G_M is a (n, k, Δ)-random bipartite graph. We distinguish now between the case of $k = 3$ and $k > 3$. If $k > 3$ then, by Theorem 4.31, for large n, with high probability there exists a constant γ such that Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{V}}(G_M, \gamma n)$.

If $k = 3$, by Theorem 4.33, for large n, with high probability there exists a constant γ such that Cover has a winning strategy $\mathsf{CoverGame}_{\mathsf{VW}}(G_M, \gamma n)$.

In both cases, we can apply Lemma 4.14 since the graph G_M is equivalent to the $\mathscr{A}$-adjacency graph $G_M^{\mathscr{A}}$, where $\mathscr{A} = \{A_1, \ldots, A_n\}$ with $A_i = \{\alpha_i, \alpha_i'\}$ where $\mathrm{dom}(\alpha_i) = \mathrm{dom}(\alpha_i') = \{x_i, \bar{x}_i\}$ and

$$\alpha_i(x_i) = 1 - \alpha_i(\bar{x}_i) = \alpha_i'(\bar{x}_i) = 1 - \alpha_i'(x_i) = 0.$$

Lemma 4.14 implies that there is a non-empty (γn)-BG family $\mathscr{F}$ for $M \cup B$, that is $tr(\varphi)$, where B is the set of boolean axioms, $B = \{x_i^2 - x_i, x_i + \bar{x}_i - 1\}_{i=1,\ldots,n}$. Moreover $\mathscr{F}$ is also a (γn)-BG family for $tr(\varphi)$ with respect to the 0 ideal, since clearly for each $p \in B$ we have that there exists some A_i such that $\mathrm{var}(p) \subseteq A_i$. □

Theorem 4.36. *Let $k \geqslant 3$ and $\Delta > 1$. If φ is a (n, k, Δ)-random CNF, then for large n, with high probability,*

1. $\mathsf{MSpace}^{sem}(tr(\varphi) \vdash_I 1) \geqslant \Omega(n)$ *with respect to the ideal I generated by the Boolean axioms.*

2. $\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant \Omega(n^2)$. *More precisely, for large n, with high probability, any* Res *refutation of φ passes through a memory configuration containing $\Omega(n)$ clauses of width at least $\Omega(n)$.*

Proof. By Theorem 4.35 for large n, with high probability, there exists a constant $\gamma > 0$ and a non-empty (γn)-BG family $\mathscr{F}$ for $tr(\varphi)$ with respect to the ideal I generated by the Boolean axioms. Hence to obtain (1.) we just apply Theorem 3.6. To obtain (2.) we use the fact that $\mathscr{F}$ is a also a (γn)-BG family for $tr(\varphi)$ with respect to the 0 ideal and we apply Proposition 3.5 obtaining a non-empty $(\gamma n - 1)$-BK family, and hence, by Theorem 2.5, the lower bound for the total space in resolution. □

Notice that an alternative proof of the total space lower bound could follows also from the fact that for large n, with high probability, there exists a constant $\gamma > 0$ such that $\mathsf{width}(\varphi \vdash \bot) \geqslant \gamma n$, cf. [30, 54], and hence a total space lower bound in Resolution follows also from Corollary 2.11.

4.9 Matching principles over graphs

Matching principles over graphs have been studied largely in the context of proof complexity, cf. for example [24, 30] or [89, Section 18.1].

Let G be a bipartite graph with $|L(G)| > |U(G)|$. We think of $L(G)$ as a set of *pigeons* and $U(G)$ as a set of *holes*. The *graph pigeonhole principle* over the graph G, G-PHP, is an unsatisfiable CNF formula in the variables G-PHP

$$X = \{x_{uv} : \{u, v\} \in E(G)\}.$$

It asserts that the variables describe a map, given by a subset of the edges of G, in which each pigeon gets mapped to at least one hole but no hole receives two pigeons or more. Formally, it is a conjunction of all the following clauses

(HOLE AXIOMS) for each distinct pair of variables $x_{uv}, x_{u'v} \in X$,

$$\neg x_{uv} \vee \neg x_{u'v};$$

(PIGEON AXIOMS) for each $u \in L(G)$,

$$\bigvee \{x_{uv} : x_{uv} \in X\}.$$

Notice that if $\max_{v \in L(G)} \deg_G(v) \leqslant d$ then G-PHP is a d-CNF formula and since $|L(G)| > |U(G)|$ it is unsatisfiable. The graph pigeonhole principle is a generalization of the standard pigeonhole principle: indeed PHP^m_n is the graph pigeonhole principle $K_{m,n}$-PHP, where $K_{m,n}$ is the complete bipartite graph between a set of vertices of size m and a (disjoint) set of vertices of size n.

We recall that the encoding of G-PHP as a set of polynomials $tr(\mathsf{PHP}^m_n)$ in $\mathbb{F}[X \cup \overline{X}]$ is the following

$$\begin{aligned} tr(G\text{-PHP}) = &\left\{x_{uv}x_{u'v}\right\}_{\substack{x_{uv}, x_{u'v} \in X \\ u \neq u'}} \cup \left\{x_{uv}^2 - x_{uv}, x_{uv} + \bar{x}_{uv} - 1\right\}_{\{u,v\} \in E(G)} \\ &\cup \left\{\prod_{v : x_{uv} \in X} \bar{x}_{uv}\right\}_{u \in L(G)}. \end{aligned}$$

Theorem 4.37. *Let $d \geqslant 3$ and $\Delta > 1$. If G is a (n, d, Δ)-random bipartite graph, then there is a constant $\gamma > 0$ and non-empty (γn)-BG family for $tr(G$-PHP$)$ with respect to the 0 ideal and the ideal I generated by the hole axioms and the Boolean axioms, that is the following set of polynomials*

$$\left\{x_{uv}x_{u'v} : x_{uv}, x_{u'v} \in X \wedge u \neq u'\right\} \cup \left\{x_{uv}^2 - x_{uv}, x_{uv} + \bar{x}_{uv} - 1 : x_{uv} \in X\right\}. \tag{4.29}$$

Proof. Given $v \in U(G)$ let X_v denote the set of variables representing the edges touching the hole v, that is

$$X_v = \{x_{uv} : \exists u \in L(G)\ \{u, v\} \in E(G)\}.$$

Given $u \in L(G)$ and $v \in U(G)$, let α_{uv} be the partial assignment with domain X_v defined as follows

$$\alpha_{uv}(x_{u'v}) = 1 - \alpha_{uv}(\bar{x}_{u'v}) = \begin{cases} 1 & \text{if } u' = u, \\ 0 & \text{if } u' \neq u. \end{cases}$$

Given $v \in U(G)$, let $A_v = \{\alpha_{uv} : \{u, v\} \in E(G)\}$ and let $\mathscr{A} = \{A_v : v \in U(G)\}$. Clearly we have that A_v is flippable; $\mathrm{dom}(A_v) = X_v$ and hence, if $v \neq v'$ then $\mathrm{dom}(A_v)$ and $\mathrm{dom}(A_{v'})$ are disjoint. Moreover A_v is I-consistent, where I is the ideal generated by the polynomials in equation (4.29).

Let M be the following set of monomials

$$M = \Big\{ \prod_{v : x_{uv} \in X} \bar{x}_{uv} : u \in L(G) \Big\}.$$

An edge $\{u, v\}$ in in $E(G)$ if and only if

$$\mathrm{var}\left(\prod_{v : x_{uv} \in X} \bar{x}_{uv} \right) \cap \mathrm{dom}(A_v) \neq \emptyset,$$

hence G and $G^{\mathscr{A}}_M$, the $\mathscr{A}$-adjacency graph of M, are isomorphic.

Since G is a (n, d, Δ)-random bipartite graph, then the $\mathscr{A}$-adjacency graph of M is a (n, d, Δ)-random bipartite graph. We distinguish now between the case of $d = 3$ and $d > 3$. If $d > 3$ then, by Theorem 4.31, for large n, with high probability there exists a constant γ such that Cover has a winning strategy for $\mathsf{CoverGame}_{\mathsf{V}}(G^{\mathscr{A}}_M, \gamma n)$.

If $d = 3$, by Theorem 4.33, for large n, with high probability there exists a constant γ such that Cover has a winning strategy $\mathsf{CoverGame}_{\mathsf{VW}}(G^{\mathscr{A}}_M, \gamma n)$.

In both cases, Lemma 4.14 implies that there is a non-empty (γn)-BG family $\mathscr{F}$ for $M \cup I = tr(G\text{-}\mathsf{PHP})$ with respect to the ideal I. Moreover $\mathscr{F}$ is also a (γn)-BG family for $tr(G\text{-}\mathsf{PHP})$ with respect to the 0 ideal, since clearly for each $p \in I$ we have that there exists some A_i such that $\mathrm{var}(p) \subseteq A_i$. □

Theorem 4.38. *Let $d \geqslant 3$ and $\Delta > 1$. If G is a (n, d, Δ)-random bipartite graph, then, for large n, with high probability,*

1. $\mathsf{MSpace}^{sem}(tr(G\text{-}\mathsf{PHP}) \vdash_I 1) \geqslant \Omega(n)$ *with respect to the ideal I generated by polynomial encodings of the hole axioms and the Boolean axioms of G-PHP.*

2. $\mathsf{TSpace}_{\mathsf{Res}}(G\text{-}\mathsf{PHP} \vdash \bot) \geqslant 1\Omega(n^2)$. *More precisely, for large n, with high probability, any* Res *refutation of* G-PHP *passes through a memory configuration containing $\Omega(n)$ clauses of width at least $\Omega(n)$.*

Proof. By Theorem 4.37 for large n, with high probability, there exists a constant $\gamma > 0$ and a non-empty (γn)-BG family $\mathscr{F}$ for $tr(G\text{-}\mathsf{PHP})$ with respect to the ideal I generated by the polynomial encodings of hole axioms and the Boolean axioms of G-PHP. Hence to obtain (1.) we just apply Theorem 3.6. To obtain (2.) we use the fact that $\mathscr{F}$ is a also a (γn)-BG family for $tr(G\text{-}\mathsf{PHP})$ with respect to the 0 ideal and we apply Proposition 3.5 obtaining a non-empty $(\gamma n - 1)$-BK family, and hence, by Theorem 2.5, the lower bound for the total space follows. □

4.10 Open problems

1. Give an explicit formula φ in n variables and $\mathsf{poly}(n)$ clauses such that

$$\mathsf{TSpace}^{sem}_{\mathsf{PCR}}(tr(\varphi) \vdash 1) \geqslant \omega(n).$$

   Moreover all the open problems on total space in Polynomial Calculus from [4] are still open.

2. There are plenty of examples of formulas that are well-studied in proof complexity but for which we do not know monomial space lower bounds, for instance the following.

   a) Given an arbitrary small $\epsilon > 0$, a constant γ and an unsatisfiable CNF formula φ in n variables such that the clauses-variables adjacency graph of φ is a $(\gamma n, 1+\epsilon)$-bipartite expander, is this expansion property enough to have that

$$\mathsf{MSpace}^{sem}(tr(\varphi) \vdash 1) \geqslant \Omega(\gamma n)?$$

   This is the case for the clause space in Resolution [24] and indeed we suspect that the same happens for the monomial space in Polynomial Calculus.

   b) We conjecture that, with high probability (for large n)

$$\mathsf{MSpace}^{sem}(tr(\mathsf{Tseitin}(G,\sigma)) \vdash 1) = \Omega(n),$$

   where G is a random 3-regular graph of n vertices and σ an odd weight function, cf. Section 4.5 for the definition of the formula $\mathsf{Tseitin}(G,\sigma)$.

c) Is it true that if φ is r-semiwide then

$$\mathsf{MSpace}^{sem}(tr(\varphi) \vdash 1) \geqslant \Omega(r)?$$

As particular cases of this question we have the question about monomial space lower bounds for the *functional pigeonhole principle*, fPHP^m_n, cf. Section 4.4.1, and the *Graph Tautologies*, GT_n, cf. [4, Definition 3.12].

5

A postlude: SETH and Resolution size

In this chapter we put the spotlight again on Resolution and in particular to its connection with conjectures about the exact complexity of the k-SAT problem, that is the conjectures known as the *Exponential Time Hypothesis* (ETH) and the *Strong Exponential Time Hypothesis* (SETH). For (a subsystem of) Resolution we show size lower bounds stronger than the one we get from the size-width relation of [30], cf. equation (5.3). Our technique use combinatorial characterizations of size and width, cf. respectively [118] and [8]. Then we show a general hardness amplification result. Before going into details we recap the state of the art of strong lower bounds known and their connection to Resolution and the complexity of k-SAT.

5.1 Introduction

We recall that the k-SAT problem is the decision problem for satisfiability of k-CNF formulas. There are several non-trivial algorithms known to solve k-SAT, cf. for instance [61, 110, 111, 133]. Despite this however, the exact complexity of k-SAT under suitable hardness assumptions remains unknown. Formalizing what this complexity could be, Impagliazzo and Paturi [85] formulated the following two hypotheses:

The *Exponential Time Hypothesis*, ETH, states that the k-SAT problem requires exponential time, for every $k \geqslant 3$. ETH

SETH The *Strong Exponential Time Hypothesis*, SETH, states that the complexity of k-SAT grows as k increases and it approaches that of exhaustive search. More precisely let $\sigma_k = \inf\{\delta : k\text{-SAT can be solved in time } O(2^{\delta n})\}$. SETH states that $\lim_{k\to\infty} \sigma_k = 1$.

Both ETH and SETH are stronger than NP $\neq$ P and hence any proof is far beyond reach at the moment but such hypotheses are important since they imply a plethora of fine grained complexity results in the realm of *parameterized complexity*. We refer to [59] for more details on how this hypotheses are useful in such context.

However one can ask whether SETH holds for specific algorithms, that is whether there are k-CNF formulas on which the algorithms run for at least $2^{(1-\epsilon_k)n}$ steps. This turns out to be the case for certain classes of algorithms, for instance for the PPSZ algorithm, cf. [111], such lower bound was proved by Scheder et al. [131]. Clearly, one may ask for such result for a *class of algorithms* rather than for a specific one. Since we can think on the run of a k-SAT algorithm on an unsatisfiable instance as a proof of its unsatisfiability. Then, if the algorithm is structured enough, we can employ tools from proof complexity and obtain lower bounds on the running time.

For natural proof systems, such as Resolution, exponential lower bounds consistent with ETH are known from a long time, cf. for instance [139]. These are $2^{\Omega(n)}$ lower bounds for k-CNF formulas on n variables and hence not strong enough to support SETH. For *tree-like Resolution* Pudlák and Impagliazzo [120] constructed unsatisfiable k-CNF formulas φ_n such that

$$\mathsf{size}_{\text{tree-Res}}(\varphi \vdash \bot) \geqslant 2^{(1-\epsilon_k)n}, \tag{5.1}$$

where $\epsilon_k = O(k^{-\frac{1}{8}})$. It is well known that a run of a DPLL algorithm on an unsatisfiable k-CNF formula gives a tree-like refutation (and viceversa), therefore a tree-like Resolution refutation lower bound implies a DPLL running time lower bound. So Pudlák and Impagliazzo [120] proved, in particular, a lower bound on the running time of the DPLL algorithm of the form

$$2^{(1-\epsilon_k)n},$$

where $\epsilon_k = O(k^{-\frac{1}{8}})$. If we ignore, for the moment, the precise asymptotic of ϵ_k, the result from [120] can be informally stated saying that

'SETH *is consistent with tree-like Resolution*',

that is no algorithm whose run result in a proof system that is p-equivalent to tree-like Resolution will be able to disprove SETH. So, proving that SETH is

consistent with a proof system P will be an indirect support to the truth of SETH or, at least, that no algorithm confuting SETH will be formalizable in P. For example, proving that SETH is consistent with Resolution will mean that no CDCL solver (with some hypothesis on its behavior) will be able to refute SETH, due to the fact that CDCL solvers are p-simulated by Resolution, cf. Section 1.3. Hence, from the proof complexity point of view, we could be interested in what is the strongest proof systems in which an inequality as in (5.1) holds.

strong size lower bound

Given a family of k-CNF formulas in n variables φ_n, we call *strong* size lower bound a lower bound of the form

$$\mathsf{size}(\varphi_n \vdash \bot) \geqslant 2^{(1-\epsilon_k)n},$$

strong width lower bound

where $\epsilon_k \to 0$ as $k \to \infty$. Similarly a *strong* width lower bound is a lower bound of the form

$$\mathsf{width}(\varphi_n \vdash \bot) \geqslant (1-\epsilon_k)n,$$

where $\epsilon_k \to 0$ as $k \to \infty$.

We recall that the size-width relationship by [30, Corollary 3.4] for tree-like Resolution has the following form

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \geqslant 2^{\mathsf{width}(\varphi \vdash \bot)-k}. \tag{5.2}$$

Hence a strong width lower bound in Resolution imply a strong size lower bound in tree-like Resolution. This is not the case for general Resolution, since the best known general relation between width and size is again from [30, Theorem 3.5]:

$$\mathsf{size}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant 2^{\frac{1}{16}\frac{(\mathsf{width}(\varphi \vdash \bot)-k)^2}{n}}. \tag{5.3}$$

Hence, in general, in Resolution is not true that a strong width lower bound imply a strong size lower bound: this is due to the constant in the exponent in equation (5.3). However, if the formula is structured in some sense, that is for instance it a *xorification*, we can avoid this loss.

Indeed it is not known if '**SETH** *is consistent with Resolution*' (although it is expected to be) but, in a recent construction, Beck and Impagliazzo [21] showed that

'**SETH** *is consistent with regular Resolution*',

reg-Res

where we recall that a *regular* Resolution derivation from a formula φ in n variables is a Resolution derivation in which along any path no variable is

resolved multiple times. This result was obtained relying on a strong width lower bound for Resolution of the form

$$\mathsf{width}(\varphi_n \vdash \bot) \geqslant (1-\epsilon_k)n,$$

where $\epsilon_k = \widetilde{O}(k^{-\frac{1}{4}})$ and the $\widetilde{O}$ notation is hiding log factors. Contextually their result improves also the asymptotic of the strong size lower bound in tree-like Resolution from the $\epsilon_k = O(k^{-\frac{1}{8}})$ of [120] to $\epsilon_k = \widetilde{O}(k^{-\frac{1}{4}})$.

In [38], we further improved the asymptotic of ϵ_k to $\widetilde{O}(k^{-\frac{1}{3}})$, hence an obvious question is how far this improvements on the asymptotic of ϵ_k can be pushed. It turns out that the best possible would be $\epsilon_k = O(k^{-1})$ since for every unsatisfiable k-CNF formula on n variables there exists a tree-like Resolution of size at most $2^{\left(1-\Omega(k^{-1})\right)n}$, cf. Theorem 5.2.

Moreover in [39] we improved the result in [21] proving that

'SETH *is consistent with* δ*-regular Resolution*',

δ-reg-Res for small $\delta \in [0,1]$, cf. Corollary 5.8, where a δ*-regular* Resolution refutation of a formula φ is a Resolution refutation in which along any path of its associated DAG a fraction of most δ variables is resolved multiple times. Hence a 0-regular Resolution refutation is just a regular refutation and a 1-regular Resolution refutation is one without any constraint. For $\delta = 0$ our result simplify [21] and induces a game theoretical proof of the fact that 'SETH *is consistent with regular Resolution*'.

5.2 Main results + credits

The main goal of this chapter is to prove that for any large enough natural numbers N and K there exists an unsatisfiable K-CNF formula ψ in N variables such that

$$\mathsf{size}_{\delta\text{-reg-Res}}(\psi \vdash \bot) \geqslant 2^{(1-\epsilon_K)N}, \tag{5.4}$$

where both ϵ_K and δ are $\widetilde{O}(K^{-1/4})$, cf. Corollary 5.8. In order to prove the result in equation (5.4) we further develop the game characterization of Resolution size by Pudlák [118]; we show a general hardness amplification result lifting width lower bounds to size lower bounds in δ-regular Resolution; and we improve (and simplify) the strong width lower bound by Beck and Impagliazzo [21].

Section 5.3 First of all we prove an upper bound on Resolution size of the form

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \leqslant 2^{\left(1-\Omega(1/k)\right)n},$$

where φ is a k-CNF formula in n variables, cf. Theorem 5.2. This result relies on *canonical decision trees* and a Switching Lemma, cf. Lemma 5.1. A similar argument was used by Miltersen et al. [104] to prove upper bounds on the size of decision trees for k-CNF formula's. However, we are not aware of any adaptation of this result in the proof complexity literature and hence we present a formal account of this observation.

Section 5.4 We give a common formalization of the *Pudlák games* characterizing Resolution size and the games characterizing Resolution width by Atserias and Dalmau [8]. Informally, we have two players Prover and Delayer that play on some formula φ. Prover has the objective of showing that the formula φ is unsatisfiable by querying variables. Delayer on the other hand wants to play as long as possible before the formula is falsified while answering to the queries Prover asks her. The size of Resolution proofs of φ is then characterized as the minimal number of *records*, i.e. partial assignments, Prover has to consider in a winning strategy. The w-AD families from Chapter 2 corresponds to the winning strategies of Delayer in such games, cf. Theorem 5.3.

We observe that the size in δ-regular Resolution is characterized by a *Pudlák game* where Prover is allowed to re-query in each run of the game at most δn variables, where n is the number of variables of the formula φ on which they are playing, cf. Theorem 5.4. Then to prove a Resolution size lower bound we show that, in order to win, Prover must keep a large number of records and we can do that by producing a lot of sufficiently different strategies for Delayer. Prover must win against each of them, hence in his winning strategy he must have a lot of distinct records, since the strategies of Delayer are sufficiently different. In the literature this is done essentially by making Prover play against a Delayer that plays according to a random strategy [60, 118]. Then the size lower bound, that is a lower bound on the number of records that Prover must have in a winning strategy, is obtained by probabilistic arguments. This may very likely lead to some loss in the constants and that is what we want to avoid to prove a SETH lower bound for Resolution size.

Section 5.5 We still use Pudlák's characterization of Resolution size as games but we will apply it to a structured formula, a *xorification* of some unsatisfiable CNF formula φ. This allows us to avoid the use of probabilistic arguments and it is the core of our main technical result, cf. Theorem 5.5. There we prove that if there is a width lower bound for refuting an unsatisfiable CNF formula φ in Resolution, then there exists a *'sufficiently strong'* exponential size lower bound for refuting a xorification of φ. For the Pudlák game, played

on the xorified instance of a formula φ, we give a series of strategies for Delayer to which Prover has to answer in order to win. To construct such strategies we use the characterization of Resolution width as a game [8], played on the original formula φ. At a very high level, the crucial idea here is to amplify a winning strategy for Delayer in the width game on φ. By this we mean that each strategy σ for the game corresponding to width gives rise to a multitude of strategies, each of them acting differently on the xorified formula, but in a sense they all act the same as σ on the original formula. This is done by exploiting the combinatorial properties of the xorified formula in such a way that the number of Delayer strategies, for the Pudlák game played on the xorified formula, does indeed hugely amplify. Then, the desired size lower bound follows from a counting argument.

Section 5.6 The fact that SETH is consistent with δ-regular Resolution follows then from the hardness amplification in Theorem 5.5, and a strong width lower bound: for any large n and k, there exist an unsatisfiable k-CNF formula φ on n variables such that

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1 - \zeta_k)n,$$

where $\zeta_k = \widetilde{O}(k^{-1/3})$, cf. Theorem 5.6. More precisely the result we show is that for any large enough N and $K \in \mathbb{N}$ there exists an unsatisfiable K-CNF formula ψ in N variables such that

$$\mathsf{size}_{\delta\text{-reg-Res}}(\psi \vdash \bot) \geqslant 2^{(1-\epsilon_K)N},$$

where $\epsilon_K = \delta = \widetilde{O}(K^{-1/4})$, cf. Corollary 5.8. Contextually, we also prove that for any large enough $k \in \mathbb{N}$ there exists an unsatisfiable k-CNF formula φ in N variables such that

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \geqslant 2^{(1-\epsilon_k)N},$$

where $\epsilon_k = \widetilde{O}(k^{-1/3})$, cf. Corollary 5.7. Both the strong lower bound for *tree-like* Resolution and the strong lower bound for *δ-regular* Resolution give better bounds on the asymptotic of ϵ_k with respect to the bounds given in [21].

Section 5.7 We end this chapter with the proof of the fact that for any large n and k, there exist an unsatisfiable k-CNF formula φ on n variables such that

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1 - \zeta_k)n,$$

where $\zeta_k = \widetilde{O}(k^{-1/3})$, cf. Theorem 5.6. As in [21], we prove this width lower bound for a family of CNF formulas encoding unsatisfiable linear systems of equations over a finite field with p element, $\mathbb{F}_p$, for large enough p. If the

coefficients in such system of linear equation are chosen uniformly at random then with high probability they satisfy a certain kind of expansion property, cf. Definition 5.9 and Proposition 5.10.

The main technical difference, between the our width lower bound and [21], is in the way such linear systems over $\mathbb{F}_p$ are encoded using Boolean variables. We encode such systems of linear equations in a more efficient way in the number variables used and in this way we improve the asymptotic of the width lower bound. The main technical improvement over [21] is Lemma 5.11. Our width lower bound, Theorem 5.6, is a modification of the analogue of [21, Theorem 5.5] and ultimately relies on the widely used idea in proof complexity that medium complexity clauses in a proof *should* have large width.

The results presented in this chapter rely on two joint works with Navid Talebanfard, cf. [38, 39].

5.3 An upper bound on Resolution size

Using the *Switching Lemma*, cf. Lemma 5.1, we show that if φ is a k-CNF formula then

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \leqslant 2^{\left(1-\Omega\left(\frac{1}{k}\right)\right)n}.$$

Before proving this result we need to recall some notations and terminology, in particular *decision trees* and *canonical decision trees*.

decision tree

Let φ be an unsatisfiable CNF formula. A *decision tree* for φ is a binary tree where the inner nodes are labeled with variables from the variables of φ and the leaves are labeled with clauses from φ. Each path in the decision tree corresponds to a partial assignment where a variable x gets the value 0 or 1 according whether the path branches left or right at the node labeled with x. The condition on the tree is that each clause on the leaves is falsified by the partial assignment given by the path reaching the clause. Decision trees for an unsatisfiable CNF formula φ are in a bijective correspondence with tree-like Resolution refutations of φ, cf. for example Beyersdorff et al. [34].

canonical decision tree

$T(\varphi)$

Following [12], a *canonical decision tree* is defined as follows. Given a CNF formula $\varphi = \bigwedge_i C_i$ consider fixed an ordering $\leqslant$ on the variables of φ and an ordering $\preceq$ on the clauses of φ. The *canonical decision tree* of φ, $T(\varphi)$, is inductively defined as follows: look at the first clause C of φ according to the ordering $\preceq$ and let $\varphi = C \wedge \varphi'$. Then do a full decision tree on the variables of C respecting the order $\leqslant$ of the variables, that is along each directed path from the root to leaves the sequence of variables encountered $x_{i_1}, \ldots, x_{i_\ell}$ is such that $x_{i_1} \leqslant \cdots \leqslant x_{i_\ell}$. Each path from the root to a leaf defines a partial assignment and there is exactly one path from the root to a leaf v that correspond to a

partial assignment that falsifies C. We label the leaf v with the clause C. For all the other other leaves w, let α_w be the partial assignment corresponding to the path from the root to the leaf w. We replace the leaf w with $T(\alpha_w(\varphi'))$, cf. Figure 5.1.

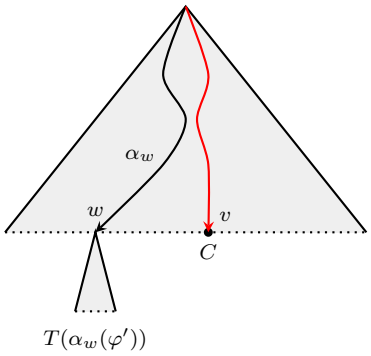


Figure 5.1: A canonical decision tree

random restriction

A *random restriction* leaving ℓ unassigned variables in a set of variables X can be obtained as follows: first pick a subset S of the variables of size $|X| - \ell$ uniformly at random, then set each $x \in S$ to either 0 or 1 with equal probability. The following variant of the Switching Lemma is due to [12].

Lemma 5.1 (Switching Lemma, Beame [12])**.** *Let φ be a k-CNF formula on n variables. Let ρ be a random restriction chosen uniformly at random from the set of all restrictions that leave exactly ℓ variables unset, with $\ell \leqslant \frac{n}{7}$. The probability that the canonical decision tree of $\rho(\varphi)$ has depth bigger than d is at most $\left(\frac{7k\ell}{n}\right)^d$.*

Theorem 5.2. *For any unsatisfiable k-CNF formula φ on n variables*

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \leqslant 2^{\left(1-\Omega\left(\frac{1}{k}\right)\right)n}.$$

The proof of this result is based on [39].

Proof. Miltersen et al. [104] showed that every k-CNF formula has a decision tree representation of size $2^{\left(1-\Omega\left(\frac{1}{k}\right)\right)n}$. We follow their argument and adapt it to the unsatisfiable setting.

Let $\ell = n/14k$ and let $d = \ell/2$. By the Switching Lemma, for a $1 - 2^{-d}$ fraction of partial assignments ρ with $|\rho| = n - \ell$, the depth of $T(\rho(\varphi))$ is at most d. Then, by an averaging argument, there exists a subset S of the variables

of φ with $|S| = n - \ell$ such for at least $1 - 2^{-d}$ of the partial assignments ρ with domain S, the depth of the canonical decision tree $T(\rho(\varphi))$ is at most d. Then we can construct a decision tree for φ as follows: first we do a full decision tree on variables in S; then for each leaf with the corresponding restriction σ, we append $T(\sigma(\varphi))$ to that leaf. The number of leaves of this tree is upper bounded by

$$2^d 2^{n-\ell} + 2^{-d} 2^{n-\ell} 2^{\ell}, \tag{5.5}$$

since at most a 2^{-d} fraction of the leaves of the full decision tree on S can have maximal depth ℓ, cf. Figure 5.2.

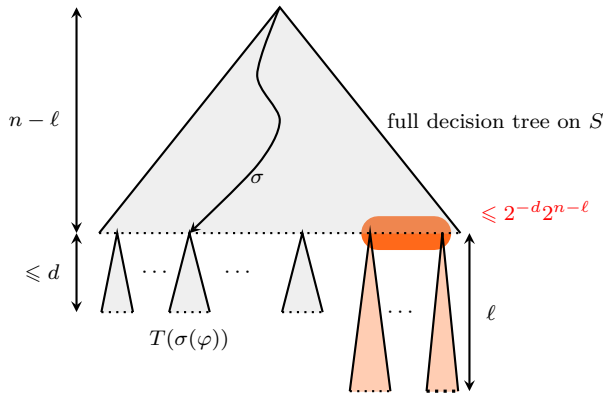


Figure 5.2: Size upper bound via canonical decision trees

Since $d = \ell/2$ then equation (5.5) is upper bounded by

$$2^{n-\frac{\ell}{2}+1} = 2^{\left(1-\Omega(\frac{1}{k})\right)n}.$$

Hence we constructed a decision tree for φ with size at most $2^{\left(1-\Omega(\frac{1}{k})\right)n}$. Since decision trees correspond to tree-like Resolution refutations we have the desired upper bound. □

5.4 Resolution size and width as games

We start introducing a common framework for the games described by Atserias and Dalmau [8] for Resolution width, and by Pudlák [118] for Resolution size.

Given an unsatisfiable CNF formula φ and a set of partial assignments $\mathcal{R}$ containing the empty assignment, we define a game, $\mathsf{Game}(\varphi, \mathcal{R})$, between two players Prover (he) and Delayer (she). At each step i of the game the players maintain a partial assignment $\alpha_i \in \mathcal{R}$, where α_0 is the empty partial assignment, then at step $i + 1$ the following moves take place:

$\mathsf{Game}(\varphi, \mathcal{R})$

1. Prover picks some variable $x \notin \mathrm{dom}(\alpha_i)$.
2. Delayer answers $x = b$ for some bit $b \in \{0, 1\}$.
3. Prover set $\alpha_{i+1} \in \mathcal{R}$ such that $\alpha_{i+1} \subseteq \alpha_i \cup \{x \mapsto b\}$.

If at any point in the game α_i falsify φ then Prover wins; otherwise Delayer wins. As usual, we say that Prover has a *winning strategy* for the game $\mathsf{Game}(\varphi, \mathcal{R})$ if for any strategy of Delayer, he can play so that he wins the game. Otherwise we say that Delayer has a *winning strategy*.

Let φ be a CNF formula in n variables and let δ be a parameter. If in each play of $\mathsf{Game}(\varphi, \mathcal{R})$, Prover is allowed to re-query at most δn variables, we call the corresponding game $\mathsf{Game}_\delta(\varphi, \mathcal{R})$.

$\mathsf{Game}_\delta(\varphi, \mathcal{R})$

For a suitable choice of $\mathcal{R}$ the $\mathsf{Game}(\varphi, \mathcal{R})$ is exactly the one used by [8] to characterize the minimal width of Resolution refutations of φ. In particular [8] show the following result (rephrased here with the notations we just set up).

Theorem 5.3 (Atserias and Dalmau [8])**.** *Given an unsatisfiable CNF formula φ and an integer w, the following are equivalent*

1. Delayer *has a winning strategy for* $\mathsf{Game}(\varphi, \mathcal{W})$*, where $\mathcal{W}$ is the set of all possible partial assignments with a domain of size strictly less than w;*
2. $\mathsf{width}(\varphi \vdash \bot) \geqslant w$*;*
3. *there exists a w-*AD *family for φ.*

Due to this equivalence, we will denote $\mathsf{Game}(\varphi, \mathcal{W})$ *by* $\mathsf{width\text{-}Game}(\varphi, w)$.

$\mathsf{width\text{-}Game}(\varphi, w)$

Pudlák [118] showed that one can characterize the minimal size of Resolution refutations of a CNF formula φ in terms of the games we just introduced. Essentially from a Resolution refutation π we can construct a winning strategy for Prover for the game $\mathsf{Game}(\varphi, \mathcal{R})$ with a set of assignments $\mathcal{R}$ with the same size of π and vice versa: each play of the $\mathsf{Game}(\varphi, \mathcal{R})$ correspond to a path in the DAG associated to π. If π is a δ-regular refutation, then, in each play of the game $\mathsf{Game}(\varphi, \mathcal{R})$, the set of variables Prover is going to re-query has size at most $\delta |\mathrm{var}(\varphi)|$, hence he is playing a $\mathsf{Game}_\delta(\varphi, \mathcal{R})$. The next result, contained in [39], is essentially based on [118], just adapted to δ-regular Resolution.

Theorem 5.4. *Let φ be an unsatisfiable CNF formula and let δ be any real in the interval $[0, 1]$. The following are equivalent*

1. Prover *has a winning strategy for* $\mathsf{Game}_\delta(\varphi, \mathcal{R})$*, for some set of partial assignments $\mathcal{R}$ such that $|\mathcal{R}| \leqslant s$;*

2. $\mathsf{size}_{\delta\text{-reg-Res}}(\varphi \vdash \bot) \leqslant s$.

Proof. (1) $\Leftarrow$ (2) Given a refutation π of φ of minimal size, for each clause C in π let θ_C denote the minimal partial assignment mapping C to false. We now define the set of $\mathcal{R}$ to be $\mathcal{R} = \{\theta_C \ : \ C \in \pi\}$. By minimality of π and by (2.) we have that $|\mathcal{R}| \leqslant s$. A winning strategy for Prover can be described simply taking the DAG associated to π and reversing the direction of all edges. Notice that each play of the $\mathsf{Game}_\delta(\varphi, \mathcal{R})$ correspond to a path in π. Then, if π is δ-regular then in each play the set of variables Prover is going to query many times is at most $\delta|\operatorname{var}(\varphi)|$.

(1) $\Rightarrow$ (2) (*sketch*) Take a minimal winning strategy for Prover. This can be described as a DAG G where each node v has a label α_v from $\mathcal{R}$ and the label of the leaves of G are falsifying some clause from φ. By minimality, there is a unique source, labeled with the empty partial assignment, and each internal node v has one or two children corresponding to the possible choices of Prover when Delayer is challenged to set some variable x_v to 0 or to 1. We now reverse all the edges of G obtaining a DAG G' over the same vertex set of G. Each internal node v in the DAG G', starting from the sink, can then be labeled with the clause C_v of minimal size such that that

(a) $\alpha_v(C_v) = 0$;

(b) if v has just one predecessor w then $C_w \vDash C_v$, that is every partial assignment satisfying C_w satisfies C_v;

(c) if v has two predecessors w_1 and w_2 then there exists a variable x_v such that C_{w_1} has the literal x_v, C_{w_2} has the literal $\neg x_v$ and C_v is the weakening of a resolvent of C_{w_1} and C_{w_2} with respect to x_v.

Since G correspond to a winning strategy of Prover then the source nodes of G' are labeled with clauses from φ that are falsified by the corresponding assignments in $\mathcal{R}$. By the properties (b) and (c) above G' corresponds to a Resolution (with weakening) derivation π from φ. By the property (a) above we have that π is a refutation of φ. Each run of the game corresponds to a path in G and the variable queried multiple times in each path are exactly those resolved multiple times. Hence if G was obtained from a winning strategy for Prover for $\mathsf{Game}_\delta(\varphi, \mathcal{R})$, then π is a δ-reg-Res refutation of φ. □

Notice that to prove δ-reg-Res size lower bounds we will use only the implication "(2) $\Rightarrow$ (1)" from the previous theorem. This, and the fact that the reverse implication was already proven in [118], are the main reason why we just sketched the proof of "(1) $\Rightarrow$ (2)".

5.5 Hardness amplification

We now prove our structural hardness amplification result that relies on xorifications, cf. Theorem 5.5. Given a CNF formula φ over the set of Boolean variables $X = \{x_1, \ldots, x_n\}$, the *ℓ-xorification* of φ, $\varphi[\oplus^\ell]$, is a formula over the set of new Boolean variables $Y = \{y_i^j : i \in [n], j \in [\ell]\}$, and it is obtained by replacing each occurrence of x_i in φ with $y_i^1 \oplus \cdots \oplus y_i^\ell$. Notice that if φ is a k-CNF formula, then $\varphi[\oplus^\ell]$ can be expanded to a $k\ell$-CNF formula.

$\varphi[\oplus^\ell]$

Xorifications have proved to be helpful in proof complexity, see for example [107, Section 2.4] and [23]. Here we use them to have many different strategies in the game $\mathsf{Game}(\varphi[\oplus^\ell], \mathcal{R})$. The next result is based on the analogous result in [39].

Theorem 5.5. *Let φ an unsatisfiable CNF formula in n variables and let w, δ and ℓ be parameters. If $\mathsf{width}(\varphi \vdash \bot) \geqslant w$ then*

$$\mathsf{size}_{\delta\text{-}\mathsf{reg}\text{-}\mathsf{Res}}(\varphi[\oplus^\ell] \vdash \bot) \geqslant 2^{(1-\epsilon)w\ell},$$

where $\epsilon = \frac{1}{\ell}\log\left(\frac{e^3\ell n}{w}\right) + \frac{\delta n}{w}\log\left(\frac{e^3\ell}{\delta}\right)$.

Proof. We start setting up some notations and terminology we use in the proof. Let X and Y be the set of variables defined above. We call the variables in Y, *y-variables*, the variables in X, *x-variables* and we say that all the y-variables $y_i^1, \ldots, y_i^\ell$ form a *block* of variables corresponding to the x-variable x_i.

y-variables

x-variables

For each partial assignment α over Y there is naturally associated a partial assignment α' over the variables X, defined as follows

α'

$$\alpha'(x_i) = \begin{cases} \alpha(y_i^1) \oplus \ldots \oplus \alpha_r(y_i^\ell) & \text{if } \forall j = 1, \ldots, \ell,\ y_i^j \in \mathrm{dom}(\alpha), \\ \star & \text{otherwise.} \end{cases}$$

By Theorem 5.4, it is enough to show that if Prover wins $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$ then

$$\log_2 |\mathcal{R}| \geqslant w(\ell - \log(\frac{e^3\ell n}{w}) - \frac{\delta\ell n}{w}\log\frac{e^3\ell}{\delta}).$$

So suppose Prover wins $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$ for some set of partial assignments $\mathcal{R}$. Since $\mathsf{width}(\varphi \vdash \bot) \geqslant w$, by Theorem 5.3, there is a winning strategy σ for $\mathsf{Delayer}$ in the game $\mathsf{width}\text{-}\mathsf{Game}(\varphi, w)$. We use such strategy σ to build many strategies for delayer in the game $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$. For each total assignment β over Y, consider a strategy σ_β for $\mathsf{Delayer}$ in the game $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$ as follows: let α_r be the partial assignment over Y at the stage r of the game $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$ and suppose that Prover at the stage $r+1$ of the game $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$ queries y_i^j. Then the strategy σ_β for $\mathsf{Delayer}$ goes as follows:

$\mathcal{R}$

σ_β

1. if there exists $j' \neq j$ such that $y_i^{j'} \notin \mathrm{dom}(\alpha_r)$, set y_i^j to $\beta(y_i^j)$;

2. otherwise, if for all $j' \neq j$, $y_i^{j'} \in \mathrm{dom}(\alpha_r)$, then look at the value $b \in \{0,1\}$ the strategy σ sets the variable x_i when given the partial assignment α'_r. Then set y_i^j to $q \in \{0,1\}$ such that

$$q \oplus \bigoplus_{j' \neq j} \alpha_r(y_i^j) = b. \tag{5.6}$$

By induction on the length of partial assignments α' we can see easily that α'_r must appear in the strategy σ, hence the value b we get as an answer for the variable x_i is well-defines. Moreover, the assignment of q in equation (5.6) can done since $x_i \equiv y_i^1 \oplus \ldots \oplus y_i^\ell$ and the value of x_i can be set freely to 0 or 1 appropriately even after all but one of $y_i^1, \ldots, y_i^\ell$ have been set.

Since we are assuming that Prover has a winning strategy for the game $\mathsf{Game}_\delta(\varphi[\oplus^\ell], \mathcal{R})$, this means, in particular, that for every total assignment β over Y, he wins against the Delayer's strategy σ_β. On the other hand, it is immediate to see that for each total assignment β over Y, σ_β is a winning strategy for Delayer in the game $\mathsf{width\text{-}Game}(\varphi[\oplus^\ell], w\ell)$. This means that for each total assignment β over Y, $\mathcal{R}$ must contain some partial assignment, denoted by ρ_β, with domain of size at least $w\ell$ and such that at least w blocks of y-variables are completely fixed by ρ_β. Without loss of generality we assume that each ρ_β fixes exactly w blocks of y-variables, that is if ρ_β is setting more y-variables we simply ignore some of the variables and only consider w blocks. Our goal is to show that we have 'many distinct' such partial assignments ρ_β. ρ_β

Let $B \subseteq [n]$ denote a generic set of size w and consider for each possible such B the set S_B of the total assignments βs over the y-variables such that ρ_β is fixing all the variables $y_i^1, \ldots, y_i^\ell$ corresponding to all i in B, that is B

$$S_B = \{\beta \text{ tot. ass. over } Y : \forall i \in B,\ x_i \in \mathrm{dom}(\rho_\beta)\}.$$

Clearly we have that for any possible $B \subseteq [n]$ of size w,

$$|\mathcal{R}| \geqslant |\{\rho_\beta : \beta \in S_B\}|. \tag{5.7}$$

This last part of the proof is just to show that there exists some B^* such that equation (5.7) for B^* provide the desired lower bound.

There are $2^{n\ell}$ possible total assignments β over Y and $\binom{n}{w}$ possible sets $B \subseteq [n]$ of size w, hence, by the pigeonhole principle, there is a set $B^* \subseteq [n]$ of size w such that B^*

$$|S_{B^*}| \geqslant \frac{2^{n\ell}}{\binom{n}{w}}. \tag{5.8}$$

S'_{B^*} Let S'_{B^*} be the set of all the restrictions of partial assignments in S_{B^*} to $\{y_i^j : i \in B^* \wedge 1 \leqslant j \leqslant \ell\}$. We clearly have that

$$|S_{B^*}| \leqslant |S'_{B^*}| \cdot 2^{n\ell - \ell|B^*|} = |S'_{B^*}| \cdot 2^{n\ell - w\ell},$$

and, by equation (5.8), we get that

$$|S'_{B^*}| \geqslant \frac{2^{w\ell}}{\binom{n}{w}}. \tag{5.9}$$

We have now that S'_{B^*} and $\{\rho_\beta : \beta \in S_{B^*}\}$ both consist of assignments of domain $\{y_i^j : i \in B^* \wedge 1 \leqslant j \leqslant \ell\}$. We show that $|\{\rho_\beta : \beta \in S_{B^*}\}|$ cannot be too small compared to $|S'_{B^*}|$, this will be, intuitively, due to the fact that the βs we start with are very different.

Z^β Let Z^β be the set of variables that Prover re-queried when playing against σ_β and for any $i = 1, \ldots, n$ let $Z_i^\beta = Z^\beta \cap \{y_i^1, \ldots, y_i^\ell\}$. By hypothesis $|Z^\beta| \leqslant \delta\ell n$.

When Delayer follows the strategy σ_β and fixes all y-variables in a block corresponding to x_i, this assignment is within Hamming distance $|Z_i^\beta| + 1$ from β in the block corresponding to x_i. This means that for each $\beta \in S_{B^*}$ and for each i, ρ_β restricted to the set $\{y_1^i, \ldots, y_\ell^i\}$ has Hamming distance at most

$\mathcal{Z}$ $|Z_i^\beta| + 1$ from some partial assignment in S'_{B^*} restricted to $\{y_1^i, \ldots, y_\ell^i\}$. Let $\mathcal{Z}$ be the set of all possible sets $Z \subseteq Y$ of size $\delta\ell n$ such that there exists $\beta \in S_{B^*}$ with $Z^\beta \subseteq Z$. For any $i = 1, \ldots, n$, let $Z_i = Z \cap \{y_i^1, \ldots, y_i^\ell\}$. Then, by counting the variables where ρ_β and an assignment in S'_{B^*} could differ, we have that

$$|S'_{B^*}| \leqslant |\{\rho_\beta : \beta \in S_{B^*}\}| \cdot \sum_{Z \in \mathcal{Z}} \prod_{i \in B^*} 2^{|Z_i|+1} \binom{\ell}{|Z_i| + 1}. \tag{5.10}$$

Hence we have the following chain of inequalities

$$|S'_{B^*}| \overset{(5.10)}{\leqslant} |\{\rho_\beta : \beta \in S_{B^*}\}| \cdot \sum_{Z \in \mathcal{Z}} \prod_{i \in B^*} 2^{|Z_i|+1} \binom{\ell}{|Z_i| + 1} \tag{5.11}$$

$$\leqslant |\{\rho_\beta : \beta \in S_{B^*}\}| \cdot \sum_{Z \in \mathcal{Z}} \prod_{i \in B^*} \left(\frac{e^2\ell}{|Z_i| + 1}\right)^{|Z_i|+1} \tag{5.12}$$

$$\leqslant |\{\rho_\beta : \beta \in S_{B^*}\}| \cdot \sum_{Z \in \mathcal{Z}} \left(\frac{\sum_{i \in B^*} e^2\ell}{\sum_{i \in B^*}(|Z_i| + 1)}\right)^{\sum_{i \in B^*}(|Z_i|+1)} \tag{5.13}$$

$$\leqslant |\{\rho_\beta : \beta \in S_{B^*}\}| \cdot \binom{\ell n}{\delta\ell n} \cdot \left(e^2\ell\right)^{\delta\ell n + w} \tag{5.14}$$

The inequality (5.13) follows from the weighted AM-GM inequality[1] and the inequality (5.14) follows from the fact that $w \leqslant \sum_{i \in B^*}(|Z_i|+1) \leqslant \delta\ell n + w$.

Putting all together we have that

$$\begin{aligned}|\mathcal{R}| \overset{(5.7)}{\geqslant} |\{\rho_\beta : \beta \in S_{B^*}\}| \geqslant \frac{|S'_{B^*}|}{\binom{n\ell}{\delta\ell n}\,(e^2\ell)^{\delta\ell n+w}} &\overset{(5.9)}{\geqslant} \frac{2^{w\ell}}{\binom{n}{w}\binom{\ell n}{\delta\ell n}\,(e^2\ell)^{\delta\ell n+w}} \\ &\geqslant \frac{2^{w\ell}}{\left(\frac{en}{w}\right)^w \left(\frac{e}{\delta}\right)^{\delta\ell n}\,(e^2\ell)^{\delta\ell n+w}} \\ &= 2^{w\left(\ell - \log\left(\frac{e^3\ell n}{w}\right) - \frac{\delta\ell n}{w}\log\left(\frac{e^3\ell}{\delta}\right)\right)}. \square\end{aligned}$$

5.6 SETH is consistent with δ-regular Resolution

The next step now is to obtain formulas which require very large Resolution width. Such a construction is in [21] and improved in [38], where the next theorem is from.

Theorem 5.6. *For any large n and k, there exist an unsatisfiable k-CNF formula φ on n variables such that*

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1-\zeta_k)n,$$

where $\zeta_k = \widetilde{O}(k^{-\frac{1}{3}})$.

The proof of this result is a bit long and hence it is postponed to the next section, cf. Section 5.7. We now prove how from Theorem 5.5 and Theorem 5.6 follow both a strong size lower bound in tree-like Resolution and in δ-regular Resolution.

Corollary 5.7. *For any large enough $k \in \mathbb{N}$ there exists an unsatisfiable k-CNF formula φ in n variables such that*

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \geqslant 2^{(1-\epsilon_k)n},$$

where $\epsilon_k = \widetilde{O}(k^{-\frac{1}{3}})$.

[1] The *weighted Arithmetic Mean - Geometric Mean inequality* says that given non-negative numbers $a_1, \ldots, a_n$ and non-negative weights $w_1, \ldots, w_n$ then

$$\prod_i a_i^{w_i} \leqslant \left(\frac{\sum_i w_i a_i}{w}\right)^w,$$

where $w = \sum_i w_i$. We applied this inequality with $a_i = \frac{e^2\ell}{|Z_i|+1}$ and $w_i = |Z_i|+1$.

Proof. Let φ be the unsatisfiable CNF formula coming from Theorem 5.6. By [30] we have that

$$\mathsf{size}_{\text{tree-Res}}(\varphi \vdash \bot) \geqslant 2^{\mathsf{width}(\varphi \vdash \bot)-k},$$

hence the strong size lower bound follows. □

Corollary 5.8. *For any large enough N and $K \in \mathbb{N}$ there exists an unsatisfiable K-CNF formula ψ in N variables such that*

$$\mathsf{size}_{\delta\text{-reg-Res}}(\psi \vdash \bot) \geqslant 2^{(1-\epsilon_K)N},$$

where $\epsilon_K = \delta = \widetilde{O}(K^{-\frac{1}{4}})$.

Proof. Let φ be the k-CNF formula in n variables given by Theorem 5.6, in particular

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1-\zeta_k)n,$$

where $\zeta_k = \widetilde{O}(k^{-\frac{1}{3}})$.

Then $\varphi[\oplus^\ell]$ is a K-CNF formula on $N = n\ell$ variables where $K = k\ell$. If we choose $\ell = \widetilde{\Theta}(k^{\frac{1}{3}})$, $\delta = \widetilde{O}(k^{-\frac{1}{3}})$ then, by Theorem 5.5, it follows that

$$\begin{aligned}
\mathsf{size}_{\delta\text{-reg-Res}}(\varphi[\oplus^\ell] \vdash \bot) &\geqslant 2^{(1-\zeta_k)n(\ell-\log(\frac{\epsilon^3\ell n}{w})-\frac{\delta\ell n}{w}\log\frac{\epsilon^3\ell}{\delta})} && (5.15)\\
&= 2^{(1-\zeta_k)n(\ell-O(\log k)-\ell\widetilde{O}(k^{-\frac{1}{3}}))} && (5.16)\\
&= 2^{(1-\widetilde{O}(k^{-\frac{1}{3}}))n\ell}. && (5.17)
\end{aligned}$$

In particular the equality (5.16) follows from the choice of $\ell = \widetilde{\Theta}(k^{\frac{1}{3}})$ and $\delta = \widetilde{O}(k^{-\frac{1}{3}})$. To obtain the asymptotic behavior of ϵ_K with respect to K, just observe that $K = k\ell = \widetilde{\Theta}(k^{\frac{4}{3}})$ and $\widetilde{O}(k^{-\frac{1}{3}}) = \widetilde{O}(K^{-\frac{1}{4}})$, hence $\epsilon_K = \widetilde{O}(K^{-\frac{1}{4}})$. Similarly we get the asymptotic behavior of δ as a function of K. □

5.7 Proof of Theorem 5.6

$\mathbf{v} = (v_1, v_2, \ldots)$ $\mathrm{supp}(\mathbf{v})$

Let p be a prime, $\mathbb{F}_p$ be the finite field with p elements and $\mathbf{v} = (v_1, v_2, \ldots)$ be a vector over $\mathbb{F}_p$, then by $\mathrm{supp}(\mathbf{v})$ we denote the indices of $\mathbf{v}$ with non-zero entries mod p, that is $\mathrm{supp}(\mathbf{v}) = \{i \ : \ v_i \not\equiv 0 \mod p\}$.

We now construct a system of linear equations over $\mathbb{F}_p$. In this section we use the letter E, with subscripts, to denote linear equations mod p, that is expressions of the form

$$\sum_j a_j z_j \equiv b \mod p,$$

$\mathrm{supp}(E)$

with $a_j, b \in \mathbb{F}_p$. We denote with $\mathrm{supp}(E)$ the set of indices j having non-zero a_js. Given two linear equalities E and E', respectively $\sum_j a_j z_j \equiv b \mod p$ and

$\sum_j a'_j z_j \equiv b' \mod p$, we denote the sum of E and E' with $E + E'$, that is the following linear equation: $\sum_j (a_j + a'_j) z_j \equiv (b + b') \mod p$. Similarly we define αE for $\alpha \in \mathbb{F}$ and $\sum_i \alpha_i E_i$ for a set of linear equations $\{E_i\}_i$ and $\alpha_i \in \mathbb{F}$.

Definition 5.9 ((α, β, γ)-expander, Beck and Impagliazzo [21])**.** *Let α, β, γ in $\mathbb{R}_{\geqslant 0}$, $m \in \mathbb{N}$ and $\mathscr{E} := \{E_1, \ldots, E_m\}$, that is $\mathscr{E}$ is a set of m linear equations over $\mathbb{F}_p$. We say that the set $\mathscr{E}$ is an* (α, β, γ)-expander *if and only if* (α, β, γ)-expander

$$\forall \mathbf{v} \in \mathbb{F}_p^m,\ \alpha \leqslant |\operatorname{supp}(\mathbf{v})| \leqslant \beta \to \Big|\operatorname{supp}\big(\sum_{i=1}^{m} v_i E_i\big)\Big| \geqslant \gamma.$$

Proposition 5.10 (Beck and Impagliazzo [21, Lemma 4.2])**.** *Let p a sufficiently large prime. There exists a set $\mathscr{E} := \{E_1, \ldots, E_{n+1}\}$ consisting of linear equations in n variables over $\mathbb{F}_p$ such that:*

1. *$\mathscr{E}$ is unsatisfiable,*
2. *for each $E_i \in \mathscr{E}$ $|\operatorname{supp}(E_i)| \leqslant p^2$,*
3. *$\mathscr{E}$ is $(\delta n, 3\delta n, (1 - c\theta)n)$-expander, where $\delta = O(1/p)$, $\theta = \widetilde{O}(1/p)$ and c is a constant[2],* $\delta = O(1/p)$ $\theta = \widetilde{O}(1/p)$
4. *no subset of at most $3\delta n$ equations from $\mathscr{E}$ is unsatisfiable.*

In [21] the authors encode each variable of the set of linear equations from Proposition 5.10 using a sum of roughly p^2 Boolean variables and show that with this encoding the linear system requires very large Resolution width.

The key property of this representation is the following: let $z = \sum_{i=0}^{p^2} x_i$, where x_i are boolean variables, then even setting a lot of variables (that is $p^2 - p$) we still can obtain all possible $\mathbb{F}_p$ values for z setting the remaining variables. In other words what is really needed in [21] is a function that can extract $\log p$ bits even after many bits in the input are fixed. We show that a random function satisfies this property, cf. Lemma 5.11, and we use this function instead of the sum of p^2 Boolean variables. Then the arguments of [21] still goes through Theorem 5.6 and following lemma is then the main technical improvement over the construction in [21].

Lemma 5.11. *Let p a sufficiently large prime and θ be the parameter* coming *from Proposition 5.10 and let $u = \theta^{-1} \log^2 p$. Then there exists a function $g : \{0,1\}^u \to \{0,1\}^{\log p}$ such that for any restriction σ with $|\sigma| \leqslant u - \log^2 p$ we have that the image of $g|_\sigma$, $\operatorname{Img}(g|_\sigma)$, is $\{0,1\}^{\log p}$.* $u = \theta^{-1} \log^2 p$ $g : \{0,1\}^u \to \{0,1\}^{\log p}$

[2]Of course the constant c in this statement is redundant, but it will be helpful in what follows to have it explicitly written.

Proof. Let g be random function that assigns to every $x \in \{0,1\}^u$ a value in $\{0,1\}^{\log p}$ independently and uniformly at random. We bound the probability that there exist a $y \in \{0,1\}^{\log p}$ and a restriction ρ with $|\rho| = u - \log^2 p$ such that $y \notin \mathrm{Img}(g|_\rho)$. Let A be such event. A bound on $\Pr[A]$ is easily given as follows:

$$\Pr[A] \leqslant 2^{\log p}\binom{u}{\log^2 p}2^{u-\log^2 p}\left(1-\frac{1}{p}\right)^{2^{\log^2 p}} \tag{5.18}$$

$$\leqslant p\theta^{-\log^2 p}2^{u-\log^2 p}e^{\log^2 p-\frac{1}{p}2^{\log^2 p}} \tag{5.19}$$

$$= o(1), \tag{5.20}$$

since $\theta = \widetilde{O}(1/p)$. The inequality in (5.18) follows by the union bound, since once we fixed $y \in \{0,1\}^{\log p}$ and a restriction ρ such that $|\rho| = u - \log^2 p$ then

$$\Pr[y \notin \mathrm{Img}(g|_\rho)] \leqslant (1-1/p)^{2^{\log^2 p}}.$$

Then there must exist at least one function g realizing the complementary event that we bounded. Such function works also for each σ such that $|\sigma| \leqslant u - \log^2 p$. □

We have now all the ingredients to define the family of CNF formulas for which we will prove the strong width lower bound.

Let $Z = \{z_1, \ldots, z_n\}$ a set of variables taking values over $\mathbb{F}_p$. The function $g : \{0,1\}^{\theta^{-1}\log^2 p} \to \{0,1\}^{\log p}$ obtained from Lemma 5.11 can be used to define each variable z_i over $\mathbb{F}_p$ using $u = \theta^{-1}\log^2 p$ new Boolean variables $X = \{x_{i1}, \ldots, x_{iu}\}$:

$$z_i = \sum_{j=1}^{\log p} 2^{j-1} g_j(x_{i1}, \ldots, x_{iu}), \tag{5.21}$$

where g_j represents the projection of g on the j-th coordinate. Hence a linear equation mod p in n variables, say

$$\sum_i a_i z_i \equiv b \mod p,$$

can be transformed into a Boolean function using equation (5.21) and $N = nu = n\theta^{-1}\log^2 p$ Boolean variables x_{ij}:

$$\sum_{j=1}^{n} a_{ij} \sum_{k=1}^{\log p} 2^{k-1} g_k(x_{i1}, \ldots, x_{iu}) \equiv b_i \mod p.$$

Moreover if $|\operatorname{supp}(a_1, \ldots, a_n)| \leqslant d$ then the Boolean encoding of this function as a CNF formula turns out to be a (du)-CNF formula. We will use this construction applied to the system of linear equations from Proposition 5.10 to prove Theorem 5.6 from previous section.

Restated Theorem 5.6. *For any large n and k, there exist an unsatisfiable k-CNF formula φ on n variables such that*

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1 - \zeta_k)n,$$

where $\zeta_k = \widetilde{O}(k^{-\frac{1}{3}})$.

Proof. Let p be a sufficiently large prime and let $\mathscr{E} := \{E_1, \ldots, E_m\}$ be the set of linear equations in n variables over $\mathbb{F}_p$ from Proposition 5.10. Let δ, θ and c as in Proposition 5.10, that is $\delta = O(1/p)$ and $\theta = \widetilde{O}(1/p)$. Let $u = \theta^{-1} \log^2 p$ and $g : \{0,1\}^u \to \{0,1\}^{\log p}$ as in Lemma 5.11. Let E_i be the linear equation $\sum_{j=1}^{n} a_{ij} z_j \equiv b_i \mod p$ with $a_{ij}, b_i \in \mathbb{F}_p$. Replacing each z_j with the expression given in (5.21), we obtain a Boolean function

$$E_i^b := \sum_{j=1}^{n} a_{ij} \sum_{k=1}^{\log p} 2^{k-1} g_k(x_{i1}, \ldots, x_{iu}) \equiv b_i \mod p.$$

The CNF formula φ we will consider is the encoding of the following Boolean function:

$$\varphi := \bigwedge_{i=1}^{m} E_i^b, \tag{5.22}$$

as a CNF formula. Notice that, since for each i we have $|\operatorname{supp}(E_i)| \leqslant p^2$, φ is a $(p^2\theta^{-1} \log^2 p)$-CNF formula in $N = nu = n\theta^{-1} \log^2 p$ variables. Let φ be the unsatisfiable CNF formula above, then we prove that

$$\mathsf{width}(\varphi \vdash \bot) \geqslant (1 - (c+1)\theta)N,$$

where c is as in Proposition 5.10. From this follows immediately the strong width lower bound we want to prove recalling that $\theta = \widetilde{O}(1/p)$.

Let $\mathscr{E}^b := \{E_i^b \ : \ E_i \in \mathscr{E}\}$ and for each clause C let let $\mu(C)$ be the following complexity measure: $\mu(C)$

$$\mu(C) := \min\{|S| \ : \ S \subseteq \mathscr{E}^b \wedge S \vDash C\}.$$

We say that a clause C has *medium complexity with respect to* μ if and only if medium complexity

$$\mu(C) \in \left(\frac{3}{2}\delta n, 3\delta n\right].$$

We have that if $C, D \vDash E$ then $\mu(E) \leqslant \mu(C) + \mu(D)$ and hence in each possible refutation of φ there will be a clause of medium complexity with respect to μ. We prove that for each medium complexity clause C it must hold that $\mathsf{width}(C) \geqslant (1 - (c+1)\theta)N$. By contradiction, fix a medium complexity clause C and suppose that

$$\mathsf{width}(C) < (1 - (c+1)\theta)N.$$

Z-variables
X-variables

To avoid confusion we call the variables in the set $Z = \{z_1, \ldots, z_n\}$, *Z-variables* and, similarly, the variables in the set $X = \{x_{ij} : i \in [n] \wedge j \in [u]\}$, *$X$-variables*. Take the minimal restriction ρ over the X-variables setting C to false, then

free

$|\rho| = \mathsf{width}(C)$. We say that a variable z_i is *free* if and only if

$$|\operatorname{dom}\rho \cap \{x_{i1}, \ldots, x_{iu}\}| \leqslant u - \log^2 p.$$

ξ

First we prove that there are at least $c\theta n$ free variables. Let ξ be the number of Z-variables that are free. We have both an upper and a lower bound for the $(N - \mathsf{width}(C))$ X-variables non-assigned by ρ:

$$(c+1)\theta N < N - \mathsf{width}(C) \leqslant (n - \xi)(u - (u - \log^2 p)) + u\xi.$$

Hence

$$c\theta N + \theta N < n \log^2 p - \xi \log^2 p + u\xi.$$

Now if $\xi \leqslant c\theta n$ a contradiction follows immediately recalling that $N = un$ and $\theta N = n \log^2 p$.

completion of ρ

We say that an assignment $\sigma : X \to \{0, 1, \star\}$ is a *completion of* ρ if it extends ρ and has domain $\{x_{ij} : z_i \text{ not-free}\}$. Let A be the set of all partial assignments over X that are completions of ρ. Recalling the definition of the X-variables in term of the Z-variables, cf. equation (5.21), we have that each $\sigma \in A$ naturally

σ'

define a partial assignment $\sigma' : Z \to \mathbb{F}_p \cup \{\star\}$ with domain $\{z_i : z_i \text{ non-free}\}$:

$$\sigma'(z_i) = \begin{cases} \sum_{j=1}^{\log p} 2^{j-1} g_j(\sigma(x_{i1}), \ldots, \sigma(x_{iu})) & \text{if } z_i \text{ non-free,} \\ \star & \text{otherwise.} \end{cases}$$

So, for each $\sigma \in A$, the Z-variables that are free are exactly, by construction, the ones not in the domain of σ' and for each $\sigma \in A$, σ set C to false. As observed we have that the number of free variables $\xi > c\theta n$ and hence

$$|\sigma'| < n - c\theta n = (1 - c\theta)n. \tag{5.23}$$

As C is of medium complexity with respect to μ, there exists some set of

S

equations $S \subseteq \mathscr{E}^b$ such that $S \vDash C$, $|S| \in (\frac{3}{2}\delta n, 3\delta n]$ and S is minimal with respect to inclusion. This implies that for each possible $\sigma \in A$ of the form

described above, $S|_\sigma$ is unsatisfiable and hence also $S' = \{E : E^b \in S\}$ is such that $S'|_{\sigma'}$ is unsatisfiable. Moreover, by the minimality of S, for each equation $E \in S'$ there exists some $\sigma \in A$ such that $E|_{\sigma'}$ is not a trivial constraint, that is a constraint that is always satisfied. S'

The fact that, for each $\sigma \in A$, $S'|_{\sigma'}$ is unsatisfiable means exactly that for all $\sigma \in A$ there exists some $\mathbf{v} = (v_1, \ldots, v_{n+1}) \in \mathbb{F}_p^{n+1}$ (dependent on σ) with $|\operatorname{supp}(\mathbf{v})| \leqslant |S| = |S'|$ and such that $\sum_{i=1}^{n+1} v_i E_i|_{\sigma'}$ is unsatisfiable. Hence for each $\sigma \in A$,

$$\operatorname{supp}(\sum_{i=1}^{n+1} v_i E_i) \subseteq \operatorname{dom}(\sigma'),$$

otherwise we could use the variables not fixed by σ' to satisfy the equality $\sum_i v_i E_i|_{\sigma'}$. Moreover by what observed before, for each $E \in S$ there exists some $\sigma \in A$ such that $E|_{\sigma'}$ does not trivialize and hence E_i will appear in the sum above for that particular σ.

Given $\sigma \in A$, let $E^\sigma = \sum_i v_i E_i$, where $\mathbf{v} = (v_1, \ldots, v_m)$ depends on σ as in the sum above. Then we take a random linear combination of all the E^σ for all the possible $\sigma \in A$: let $\sum_{\sigma \in A} \alpha_\sigma E^\sigma$ be such combination. Again we have that

$$\operatorname{supp}(\sum_{\sigma \in A} \alpha_\sigma E^\sigma) \subseteq \bigcup_{\sigma \in A} \operatorname{dom}(\sigma').$$

Each $E_i \in S'$ appears in the previous sum since, as already observed, for each E_i there exists some $\sigma \in A$ such that E_i appears in E^σ. Moreover, the coefficient of each $E_i \in S$ is uniformly random, and hence by averaging, there exists a linear combination such that at least $(1 - 1/p)\frac{3}{2}\delta n \geqslant \delta n$ of the E_i have non-zero coefficient. But this contradicts the expansion property of $\mathscr{E}$ as we have that

$$|\operatorname{supp}(\sum_{\sigma \in A} \alpha_\sigma E^\sigma)| \leqslant |\bigcup_{\sigma \in A} \operatorname{dom}(\sigma')| < (1 - c\theta)n,$$

where the last inequality follows from the inequality in (5.23) and the fact that all the $\sigma \in A$ have the same domain. □

5.8 Open problems

1. Prove any strong size lower bound for Resolution or any stronger system where we already have some exponential size lower bounds, for instance Polynomial Calculus.

2. Is there any unsatisfiable k-CNF formula φ in n variables such that

$$\mathsf{size}_{\mathsf{tree\text{-}Res}}(\varphi \vdash \bot) \geqslant 2^{(1-O(k^{-1}))n}?$$

That is formulas for which the upper bound of Theorem 5.2 is tight.

Bibliography

[1] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994. doi: 10.1007/BF01302964. URL `http://dx.doi.org/10.1007/BF01302964`.

[2] Michael Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. *Computational Complexity*, 20(4):597–614, 2011. doi: 10.1007/s00037-011-0026-0. URL `http://dx.doi.org/10.1007/s00037-011-0026-0`.

[3] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 190–199. IEEE Computer Society, 2001. doi: 10.1109/SFCS.2001.959893. URL `http://dx.doi.org/10.1109/SFCS.2001.959893`.

[4] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002. doi: 10.1137/S0097539700366735. URL `http://dx.doi.org/10.1137/S0097539700366735`.

[5] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory of Computing*, 3(1):81–102, 2007. doi: 10.4086/toc.2007.v003a005. URL `http://dx.doi.org/10.4086/toc.2007.v003a005`.

[6] Carlos Ansótegui, Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Measuring the hardness of SAT instances. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 222–228. AAAI Press, 2008. URL `http://www.aaai.org/Library/AAAI/2008/aaai08-035.php`.

[7] Albert Atserias. On sufficient conditions for unsatisfiability of random formulas. *J. ACM*, 51(2):281–311, 2004. doi: 10.1145/972639.972645. URL http://doi.acm.org/10.1145/972639.972645.

[8] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008. doi: 10.1016/j.jcss.2007.06.025. URL http://dx.doi.org/10.1016/j.jcss.2007.06.025.

[9] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res. (JAIR)*, 40:353–373, 2011. doi: 10.1613/jair.3152. URL http://dx.doi.org/10.1613/jair.3152.

[10] Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 286–297. IEEE, 2014. doi: 10.1109/CCC.2014.36. URL http://dx.doi.org/10.1109/CCC.2014.36.

[11] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In Benjamin Kuipers and Bonnie L. Webber, editors, *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island.*, pages 203–208. AAAI Press / The MIT Press, 1997. URL http://www.aaai.org/Library/AAAI/1997/aaai97-032.php.

[12] Paul Beame. A Switching Lemma Primer. Technical report, Department of Computer Science and Engineering, University of Washington. URL http://homes.cs.washington.edu/~beame/papers/primer.ps. UW-CSE-95-07-01.

[13] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 274–282. IEEE Computer Society, 1996. doi: 10.1109/SFCS.1996.548486. URL http://dx.doi.org/10.1109/SFCS.1996.548486.

[14] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science*, pages 42–70. 2001.

[15] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan R. Woods. Exponential lower bounds for the pigeonhole principle. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 200–220. ACM, 1992. doi: 10.1145/129712.129733. URL `http://doi.acm.org/10.1145/129712.129733`.

[16] Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, and Pavel Pudlák. Lower bound on Hilbert's Nullstellensatz and propositional proofs. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 794–806. IEEE Computer Society, 1994. doi: 10.1109/SFCS.1994.365714. URL `http://dx.doi.org/10.1109/SFCS.1994.365714`.

[17] Paul Beame, Richard M. Karp, Toniann Pitassi, and Michael E. Saks. On the complexity of unsatisfiability proofs for random k-CNF formulas. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 561–571. ACM, 1998. doi: 10.1145/276698.276870. URL `http://doi.acm.org/10.1145/276698.276870`.

[18] Paul Beame, Richard M. Karp, Toniann Pitassi, and Michael E. Saks. The efficiency of resolution and Davis–Putnam procedures. *SIAM J. Comput.*, 31(4):1048–1075, 2002. doi: 10.1137/S0097539700369156. URL `http://dx.doi.org/10.1137/S0097539700369156`.

[19] Paul Beame, Christopher Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: superpolynomial lower bounds for superlinear space. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 213–232. ACM, 2012. doi: 10.1145/2213977.2213999. URL `http://doi.acm.org/10.1145/2213977.2213999`.

[20] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus: extended abstract. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 813–822. ACM, 2013. doi: 10.1145/2488608.2488711. URL `http://doi.acm.org/10.1145/2488608.2488711`.

[21] Christopher Beck and Russell Impagliazzo. Strong ETH holds for regular resolution. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 487–494. ACM, 2013. doi: 10.1145/2488608.2488669. URL http://doi.acm.org/10.1145/2488608.2488669.

[22] Eli Ben-Sasson. *Expansion in Proof Complexity*. PhD thesis, 2001. Hebrew University.

[23] Eli Ben-Sasson. Size space tradeoffs for resolution. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 457–464. ACM, 2002. doi: 10.1145/509907.509975. URL http://doi.acm.org/10.1145/509907.509975.

[24] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Struct. Algorithms*, 23(1):92–109, 2003. doi: 10.1002/rsa.10089. URL http://dx.doi.org/10.1002/rsa.10089.

[25] Eli Ben-Sasson and Prahladh Harsha. Lower bounds for bounded depth Frege proofs via Pudlák-Buss games. *ACM Trans. Comput. Log.*, 11 (3), 2010. doi: 10.1145/1740582.1740587. URL http://doi.acm.org/10.1145/1740582.1740587.

[26] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. *Computational Complexity*, 19(4):501–519, 2010. doi: 10.1007/s00037-010-0293-1. URL http://dx.doi.org/10.1007/s00037-010-0293-1.

[27] Eli Ben-Sasson and Jakob Nordström. Understanding space in resolution: optimal lower bounds and exponential trade-offs. In Peter Bro Miltersen, Rüdiger Reischuk, Georg Schnitger, and Dieter van Melkebeek, editors, *Computational Complexity of Discrete Problems, 14.09. - 19.09.2008*, volume 08381 of *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2008. URL http://drops.dagstuhl.de/opus/volltexte/2008/1781/.

[28] Eli Ben-Sasson and Jakob Nordström. A space hierarchy for k-DNF resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:47, 2009. URL http://eccc.hpi-web.de/report/2009/047.

[29] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In Bernard Chazelle, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 401–416. Tsinghua University Press, 2011. URL `http://conference.itcs.tsinghua.edu.cn/ICS2011/content/papers/3.html`.

[30] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. doi: 10.1145/375827.375835. URL `http://doi.acm.org/10.1145/375827.375835`.

[31] Patrick Bennett, Ilario Bonacina, Nicola Galesi, Tony Huynh, Mike Molloy, and Paul Wollan. Space proof complexity for random 3-CNFs. *CoRR*, abs/1503.01613, 2015. URL `http://arxiv.org/abs/1503.01613`.

[32] Olaf Beyersdorff and Oliver Kullmann. Unified characterisations of resolution hardness measures. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8561 of *Lecture Notes in Computer Science*, pages 170–187. Springer, 2014. doi: 10.1007/978-3-319-09284-3_13. URL `http://dx.doi.org/10.1007/978-3-319-09284-3_13`.

[33] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games. *Inf. Process. Lett.*, 110(23):1074–1077, 2010. doi: 10.1016/j.ipl.2010.09.007. URL `http://dx.doi.org/10.1016/j.ipl.2010.09.007`.

[34] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A characterization of tree-like resolution size. *Inf. Process. Lett.*, 113(18):666–671, 2013. doi: 10.1016/j.ipl.2013.06.002. URL `http://dx.doi.org/10.1016/j.ipl.2013.06.002`.

[35] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, 1937. University of Chicago.

[36] Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: a combinatorial characterization for the space measure in algebraic proof systems. In Robert D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January*

9-12, 2013, pages 455–472. ACM, 2013. doi: 10.1145/2422436.2422486. URL http://doi.acm.org/10.1145/2422436.2422486.

[37] Ilario Bonacina and Nicola Galesi. A framework for space complexity in algebraic proof systems. *J. ACM*, 62(3):23, 2015. doi: 10.1145/2699438. URL http://doi.acm.org/10.1145/2699438.

[38] Ilario Bonacina and Navid Talebanfard. Improving resolution width lower bounds for k-CNFs with applications to the Strong Exponential Time Hypothesis. *Information Processing Letters*, 116(2):120 – 124, 2015. ISSN 0020-0190. doi: http://dx.doi.org/10.1016/j.ipl.2015.09.013. URL http://www.sciencedirect.com/science/article/pii/S0020019015001684.

[39] Ilario Bonacina and Navid Talebanfard. Strong ETH and Resolution via Games and the Multiplicity of Strategies. In Thore Husfeldt and Iyad Kanj, editors, *10th International Symposium on Parameterized and Exact Computation (IPEC 2015)*, volume 43 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 248–257, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-92-7. doi: http://dx.doi.org/10.4230/LIPIcs.IPEC.2015.248. URL http://drops.dagstuhl.de/opus/volltexte/2015/5587.

[40] Ilario Bonacina, Nicola Galesi, and Neil Thapen. Total space in resolution. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 641–650. IEEE Computer Society, 2014. doi: 10.1109/FOCS.2014.74. URL http://dx.doi.org/10.1109/FOCS.2014.74.

[41] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 422–432. IEEE Computer Society, 1999. doi: 10.1109/SFFCS.1999.814614. URL http://dx.doi.org/10.1109/SFFCS.1999.814614.

[42] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001. doi: 10.1007/s000370100000. URL http://dx.doi.org/10.1007/s000370100000.

[43] M. Brickenstein, A. Dreyer, G. Greuel, M. Wedler, and O. Wienand. New developments in the theory of Gröbner bases and applications to formal verification. *Journal of Pure and Applied Algebra*, 213(8):1612–1635, 2009.

[44] Michael Brickenstein and Alexander Dreyer. PolyBoRi: A framework for Gröbner-basis computations with boolean polynomials. *Journal of Symbolic Computation*, 44(9):1326 – 1345, 2009.

[45] Andrei Z. Broder, Alan M. Frieze, and Eli Upfal. On the satisfiability and maximum satisfiability of random 3-CNF formulas. In Vijaya Ramachandran, editor, *Proceedings of the Fourth Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 25-27 January 1993, Austin, Texas.*, pages 322–330. ACM/SIAM, 1993. URL `http://dl.acm.org/citation.cfm?id=313559.313794`.

[46] Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *J. Symb. Log.*, 52(4):916–927, 1987. doi: 10.2307/2273826. URL `http://dx.doi.org/10.2307/2273826`.

[47] Samuel R. Buss and Toniann Pitassi. Resolution and the weak pigeonhole principle. In Mogens Nielsen and Wolfgang Thomas, editors, *Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL, Aarhus, Denmark, August 23-29, 1997, Selected Papers*, volume 1414 of *Lecture Notes in Computer Science*, pages 149–156. Springer, 1997. doi: 10.1007/BFb0028012. URL `http://dx.doi.org/10.1007/BFb0028012`.

[48] Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiri Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1997. doi: 10.1007/BF01294258. URL `http://dx.doi.org/10.1007/BF01294258`.

[49] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In Jeffrey Scott Vitter, Lawrence L. Larmore, and Frank Thomson Leighton, editors, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 547–556. ACM, 1999. doi: 10.1145/301250.301399. URL `http://doi.acm.org/10.1145/301250.301399`.

[50] Ming-Te Chao and John V. Franco. Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k satisfiability problem. *Inf. Sci.*, 51(3):289–314, 1990. doi: 10.1016/0020-0255(90)90030-E. URL `http://dx.doi.org/10.1016/0020-0255(90)90030-E`.

[51] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 331–340. Morgan Kaufmann, 1991.

[52] Vašek Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305 – 337, 1973. ISSN 0012-365X. doi: http://dx.doi.org/10.1016/0012-365X(73)90167-2. URL `http://www.sciencedirect.com/science/article/pii/0012365X73901672`.

[53] Vasek Chvátal and Bruce A. Reed. Mick gets some (the odds are on his side). In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 620–627. IEEE Computer Society, 1992. doi: 10.1109/SFCS.1992.267789. URL `http://dx.doi.org/10.1109/SFCS.1992.267789`.

[54] Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988. doi: 10.1145/48014.48016. URL `http://doi.acm.org/10.1145/48014.48016`.

[55] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 174–183. ACM, 1996. doi: 10.1145/237814.237860. URL `http://doi.acm.org/10.1145/237814.237860`.

[56] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi: 10.2307/2273702. URL `http://dx.doi.org/10.2307/2273702`.

[57] William Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, 1987. doi: 10.1016/0166-218X(87)90039-4. URL `http://dx.doi.org/10.1016/0166-218X(87)90039-4`.

[58] David A. Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.)*. Undergraduate texts in mathematics. Springer, 1997. ISBN 978-0-387-94680-1.

[59] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. ISBN 978-3-319-21274-6. doi: 10.1007/978-3-319-21275-3. URL `http://dx.doi.org/10.1007/978-3-319-21275-3`.

[60] Stefan S. Dantchev. Relativisation provides natural separations for resolution-based proof systems. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science - Theory and Applications, First International Computer Science Symposium in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 147–158. Springer, 2006. doi: 10.1007/11753728_17. URL `http://dx.doi.org/10.1007/11753728_17`.

[61] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch, Ravi Kannan, Jon M. Kleinberg, Christos H. Papadimitriou, Prabhakar Raghavan, and Uwe Schöning. A deterministic $(2 - 2/(k + 1))^n$ algorithm for k-SAT based on local search. *Theor. Comput. Sci.*, 289(1):69–83, 2002. doi: 10.1016/S0304-3975(01)00174-8. URL `http://dx.doi.org/10.1016/S0304-3975(01)00174-8`.

[62] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960. doi: 10.1145/321033.321034. URL `http://doi.acm.org/10.1145/321033.321034`.

[63] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962. doi: 10.1145/368273.368557. URL `http://doi.acm.org/10.1145/368273.368557`.

[64] Wenceslas Fernandez de la Vega. Random 2-SAT: results and problems. *Theor. Comput. Sci.*, 265(1-2):131–146, 2001. doi: 10.1016/S0304-3975(01)00156-6. URL `http://dx.doi.org/10.1016/S0304-3975(01)00156-6`.

[65] Jian Ding, Allan Sly, and Nike Sun. Proof of the satisfiability conjecture for large k. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 59–68. ACM, 2015. doi: 10.1145/2746539.2746619. URL `http://doi.acm.org/10.1145/2746539.2746619`.

[66] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Inf. Comput.*, 171(1):84–97, 2001. doi: 10.1006/inco.2001.2921. URL http://dx.doi.org/10.1006/inco.2001.2921.

[67] Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. *Theor. Comput. Sci.*, 321(2-3): 347–370, 2004. doi: 10.1016/j.tcs.2004.04.004. URL http://dx.doi.org/10.1016/j.tcs.2004.04.004.

[68] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds - (extended abstract). In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, 2013. doi: 10.1007/978-3-642-39206-1_37. URL http://dx.doi.org/10.1007/978-3-642-39206-1_37.

[69] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. In Ernst W. Mayr and Natacha Portier, editors, *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, volume 25 of *LIPIcs*, pages 300–311. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014. doi: 10.4230/LIPIcs.STACS.2014.300. URL http://dx.doi.org/10.4230/LIPIcs.STACS.2014.300.

[70] Yuval Filmus, Massimo Lauria, Jakob Nordström, Noga Ron-Zewi, and Neil Thapen. Space complexity in polynomial calculus. *SIAM J. Comput.*, 44(4):1119–1153, 2015. doi: 10.1137/120895950. URL http://dx.doi.org/10.1137/120895950.

[71] Ehud Friedgut. Sharp thresholds of graph properties, and the k-SAT problem. *J. Amer. Math. Soc*, 12:1017–1054, 1998.

[72] Alan M. Frieze and Stephen Suen. Analysis of two simple heuristics on a random instance of k-SAT. *J. Algorithms*, 20(2):312–355, 1996. doi: 10.1006/jagm.1996.0016. URL http://dx.doi.org/10.1006/jagm.1996.0016.

[73] Nicola Galesi and Massimo Lauria. On the automatizability of polynomial calculus. *Theory Comput. Syst.*, 47(2):491–506, 2010.

[74] Nicola Galesi and Massimo Lauria. Optimality of size-degree tradeoffs for polynomial calculus. *ACM Trans. Comput. Log.*, 12(1):4, 2010. doi: 10.1145/1838552.1838556. URL `http://doi.acm.org/10.1145/1838552.1838556`.

[75] Nicola Galesi, Pavel Pudlák, and Neil Thapen. The space complexity of cutting planes refutations. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 433–447. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi: 10.4230/LIPIcs.CCC.2015.433. URL `http://dx.doi.org/10.4230/LIPIcs.CCC.2015.433`.

[76] Andreas Goerdt. A threshold for unsatisfiability. *J. Comput. Syst. Sci.*, 53(3):469–486, 1996. doi: 10.1006/jcss.1996.0081. URL `http://dx.doi.org/10.1006/jcss.1996.0081`.

[77] Ralph E. Gomory. An algorithm for integer solutions to linear programs. *Recent Advances in Mathematical Programming*, pages 269–302, 1963.

[78] Dima Grigoriev and Edward A. Hirsch. Algebraic proof systems over formulas. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(11), 2001. URL `http://eccc.hpi-web.de/eccc-reports/2001/TR01-011/index.html`.

[79] Dima Grigoriev, Edward A. Hirsch, and Dmitrii V. Pasechnik. Complexity of semi-algebraic proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, (103), 2001. URL `http://eccc.hpi-web.de/eccc-reports/2001/TR01-103/index.html`.

[80] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 110–119. IEEE Computer Society, 2014. ISBN 978-1-4799-6517-5. doi: 10.1109/FOCS.2014.20. URL `http://dx.doi.org/10.1109/FOCS.2014.20`.

[81] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985. doi: 10.1016/0304-3975(85)90144-6. URL `http://dx.doi.org/10.1016/0304-3975(85)90144-6`.

[82] Armin Haken and Stephen A. Cook. An exponential lower bound for the size of monotone real circuits. *J. Comput. Syst. Sci.*, 58(2):326–335, 1999.

doi: 10.1006/jcss.1998.1617. URL http://dx.doi.org/10.1006/jcss.1998.1617.

[83] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *BULL. AMER. MATH. SOC.*, 43(4):439–561, 2006.

[84] Wenqi Huang and Xiangdong Yu. A DNF without Regular Shortest Consensus Path. *SIAM Journal on Computing*, 16(5):836–840, 1987. ISSN 0097-5397. doi: 10.1137/0216054.

[85] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi: 10.1006/jcss.2000.1727. URL http://dx.doi.org/10.1006/jcss.2000.1727.

[86] Russell Impagliazzo, Pavel Pudlák, and Jiri Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999. doi: 10.1007/s000370050024. URL http://dx.doi.org/10.1007/s000370050024.

[87] Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Zivny. Relating proof complexity measures and practical hardness of SAT. In Michela Milano, editor, *Principles and Practice of Constraint Programming - 18th International Conference, CP 2012, Québec City, QC, Canada, October 8-12, 2012. Proceedings*, volume 7514 of *Lecture Notes in Computer Science*, pages 316–331. Springer, 2012. doi: 10.1007/978-3-642-33558-7_25. URL http://dx.doi.org/10.1007/978-3-642-33558-7_25.

[88] Emil Jeřábek. *Weak pigeonhole principle, and randomized computation.* PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2005.

[89] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. ISBN 978-3-642-24507-7. doi: 10.1007/978-3-642-24508-4. URL http://dx.doi.org/10.1007/978-3-642-24508-4.

[90] Lefteris M. Kirousis, Evangelos Kranakis, Danny Krizanc, and Yannis C. Stamatiou. Approximating the unsatisfiability threshold of random formulas. *Random Struct. Algorithms*, 12(3):253–269, 1998. doi: 10.1002/(SICI)1098-2418(199805)12:3<253::AID-RSA3>3.0.CO;2-U. URL http://dx.doi.org/10.1002/(SICI)1098-2418(199805)12:3<253::AID-RSA3>3.0.CO;2-U.

[91] Hans Kleine Büning and Theodor Lettmann. *Aussagenlogik - Deduktion und Algorithmen*. Leitfäden und Monographien der Informatik. Teubner, 1994. ISBN 978-3-519-02133-9.

[92] Phokion G. Kolaitis and Moshe Y. Vardi. On the expressive power of datalog: Tools and a case study. *J. Comput. Syst. Sci.*, 51(1):110–134, 1995. doi: 10.1006/jcss.1995.1055. URL `http://dx.doi.org/10.1006/jcss.1995.1055`.

[93] Phokion G. Kolaitis and Moshe Y. Vardi. Conjunctive-query containment and constraint satisfaction. *J. Comput. Syst. Sci.*, 61(2):302–332, 2000. doi: 10.1006/jcss.2000.1713. URL `http://dx.doi.org/10.1006/jcss.2000.1713`.

[94] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 254–266. IEEE Computer Society, 1977. doi: 10.1109/SFCS.1977.16. URL `http://dx.doi.org/10.1109/SFCS.1977.16`.

[95] Jan Krajíček. Propositional proof complexity I. URL `http://www.karlin.mff.cuni.cz/~krajicek/ds1.ps`.

[96] Jan Krajícek. Lower bounds to the size of constant-depth propositional proofs. *J. Symb. Log.*, 59(1):73–86, 1994. doi: 10.2307/2275250. URL `http://dx.doi.org/10.2307/2275250`.

[97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62 (2):457–486, 1997. doi: 10.2307/2275541. URL `http://dx.doi.org/10.2307/2275541`.

[98] Jan Krajíček, Pavel Pudlák, and Alan R. Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Struct. Algorithms*, 7(1):15–40, 1995. doi: 10.1002/rsa.3240070103. URL `http://dx.doi.org/10.1002/rsa.3240070103`.

[99] Jan Krajíček. *Bounded Arithmetic, Propositional Logic and Complexity Theory*. Cambridge University Press, 1995.

[100] Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, (41), 1999. URL `http://eccc.hpi-web.de/eccc-reports/1999/TR99-041/index.html`.

[101] Oliver Kullmann. An improved version of width restricted resolution. In *AMAI*, 2000. URL http://rutcor.rutgers.edu/~amai/aimath00/regular/kullmann.ps.

[102] Oliver Kullmann. Upper and lower bounds on the complexity of generalised resolution and generalised constraint satisfaction problems. *Ann. Math. Artif. Intell.*, 40(3-4):303–352, 2004. doi: 10.1023/B:AMAI.0000012871.08577.0b. URL http://dx.doi.org/10.1023/B:AMAI.0000012871.08577.0b.

[103] Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In David Zuckerman, editor, *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, volume 33 of *LIPIcs*, pages 467–487. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi: 10.4230/LIPIcs.CCC.2015.467. URL http://dx.doi.org/10.4230/LIPIcs.CCC.2015.467.

[104] Peter Bro Miltersen, Jaikumar Radhakrishnan, and Ingo Wegener. On converting CNF to k-DNF. *Theor. Comput. Sci.*, 347(1-2):325–335, 2005. doi: 10.1016/j.tcs.2005.07.029. URL http://dx.doi.org/10.1016/j.tcs.2005.07.029.

[105] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pages 530–535. ACM, 2001. doi: 10.1145/378239.379017. URL http://doi.acm.org/10.1145/378239.379017.

[106] Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution. *SIAM J. Comput.*, 39(1):59–121, 2009. doi: 10.1137/060668250. URL http://dx.doi.org/10.1137/060668250.

[107] Jakob Nordström. Pebble games, proof complexity, and time-space trade-offs. *Logical Methods in Computer Science*, 9(3), 2013. doi: 10.2168/LMCS-9(3:15)2013. URL http://dx.doi.org/10.2168/LMCS-9(3:15)2013.

[108] Jakob Nordström. On the interplay between proof complexity and SAT solving. *ACM SIGLOG News*, 2(3):19–44, August 2015. ISSN 2372-3491.

[109] Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, 2013.

doi: 10.4086/toc.2013.v009a014. URL http://dx.doi.org/10.4086/toc.2013.v009a014.

[110] Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. In *38th Annual Symposium on Foundations of Computer Science, FOCS*, pages 566–574, 1997. doi: 10.1109/SFCS.1997.646146. URL http://doi.ieeecomputersociety.org/10.1109/SFCS.1997.646146.

[111] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005. doi: 10.1145/1066100.1066101. URL http://doi.acm.org/10.1145/1066100.1066101.

[112] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011. doi: 10.1016/j.artint.2010.10.002. URL http://dx.doi.org/10.1016/j.artint.2010.10.002.

[113] Toniann Pitassi. Algebraic propositional proof systems. In Neil Immerman and Phokion G. Kolaitis, editors, *Descriptive Complexity and Finite Models, Proceedings of a DIMACS Workshop, January 14-17, 1996, Princeton University*, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 215–244. American Mathematical Society, 1996.

[114] Toniann Pitassi. Propositional proof complexity: A survey on the state of the art, including some recent results. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, page 119. IEEE Computer Society, 2011. doi: 10.1109/LICS.2011.42. URL http://dx.doi.org/10.1109/LICS.2011.42.

[115] Toniann Pitassi and Alasdair Urquhart. The complexity of the Hajoś calculus. *SIAM J. Discrete Math.*, 8(3):464–483, 1995. doi: 10.1137/S089548019224024X. URL http://dx.doi.org/10.1137/S089548019224024X.

[116] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993. doi: 10.1007/BF01200117. URL http://dx.doi.org/10.1007/BF01200117.

[117] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi: 10.2307/2275583. URL http://dx.doi.org/10.2307/2275583.

[118] Pavel Pudlák. Proofs as games. *The American Mathematical Monthly*, 107(6):541–550, 2000. URL http://www.jstor.org/stable/2589349.

[119] Pavel Pudlák. Twelve problems in proof complexity. In Edward A. Hirsch, Alexander A. Razborov, Alexei L. Semenov, and Anatol Slissenko, editors, *Computer Science - Theory and Applications, Third International Computer Science Symposium in Russia, CSR 2008, Moscow, Russia, June 7-12, 2008, Proceedings*, volume 5010 of *Lecture Notes in Computer Science*, pages 13–27. Springer, 2008. doi: 10.1007/978-3-540-79709-8_4. URL http://dx.doi.org/10.1007/978-3-540-79709-8_4.

[120] Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for k-SAT (preliminary version). In David B. Shmoys, editor, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA.*, pages 128–136. ACM/SIAM, 2000. URL http://dl.acm.org/citation.cfm?id=338219.338244.

[121] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. doi: 10.1145/972639.972640. URL http://doi.acm.org/10.1145/972639.972640.

[122] Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Computational Complexity*, 17(3):407–457, 2008. doi: 10.1007/s00037-008-0246-0. URL http://dx.doi.org/10.1007/s00037-008-0246-0.

[123] Alexander A. Razborov. Lower bounds for propositional proofs and independence results in bounded arithmetic. In Friedhelm Meyer auf der Heide and Burkhard Monien, editors, *Automata, Languages and Programming, 23rd International Colloquium, ICALP96, Paderborn, Germany, 8-12 July 1996, Proceedings*, volume 1099 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 1996. doi: 10.1007/3-540-61440-0_116. URL http://dx.doi.org/10.1007/3-540-61440-0_116.

[124] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, 1998. doi: 10.1007/s000370050013. URL http://dx.doi.org/10.1007/s000370050013.

[125] Alexander A. Razborov. Proof complexity of pigeonhole principles. In Werner Kuich, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory, 5th International Conference, DLT 2001, Vienna, Austria, July 16-21, 2001, Revised Papers*, volume 2295 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2001.

[126] Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci.*, 1(303):233–243, 2003. doi: 10.1016/S0304-3975(02)00453-X. URL `http://dx.doi.org/10.1016/S0304-3975(02)00453-X`.

[127] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181:415–472, 2015. doi: 10.4007/annals.2015.181.2.1.

[128] Robert A. Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, 1975.

[129] Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, 1993.

[130] John Alan Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965. doi: 10.1145/321250.321253. URL `http://doi.acm.org/10.1145/321250.321253`.

[131] Dominik Scheder, Bangsheng Tang, Shiteng Chen, and Navid Talebanfard. Exponential lower bounds for the PPSZ k-SAT algorithm. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1253–1263. SIAM, 2013. doi: 10.1137/1.9781611973105.91. URL `http://dx.doi.org/10.1137/1.9781611973105.91`.

[132] Uwe Schöning. Resolution proofs, exponential bounds, and Kolmogorov complexity. In Igor Prívara and Peter Ruzicka, editors, *Mathematical Foundations of Computer Science 1997, 22nd International Symposium, MFCS'97, Bratislava, Slovakia, August 25-29, 1997, Proceedings*, volume 1295 of *Lecture Notes in Computer Science*, pages 110–116. Springer, 1997. doi: 10.1007/BFb0029954. URL `http://dx.doi.org/10.1007/BFb0029954`.

[133] Uwe Schöning. A probabilistic algorithm for k-SAT based on limited local search and restart. *Algorithmica*, 32(4):615–623, 2002.

doi: 10.1007/s00453-001-0094-7. URL http://dx.doi.org/10.1007/s00453-001-0094-7.

[134] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007. URL http://www.math.ucla.edu/~asl/bsl/1304/1304-001.ps.

[135] João P. Marques Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Computers*, 48(5):506–521, 1999. doi: 10.1109/12.769433. URL http://doi.ieeecomputersociety.org/10.1109/12.769433.

[136] Gunnar Stålmarck. Short resolution proofs for a sequence of tricky formulas. *Acta Informatica*, 33(3):277–280, 1996. ISSN 0001-5903. doi: 10.1007/s002360050044. URL http://dx.doi.org/10.1007/s002360050044.

[137] Neil Thapen. A trade-off between length and width in resolution. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:137, 2014. URL http://eccc.hpi-web.de/report/2014/137.

[138] G.S. Tseitin. On the complexity of derivation in propositional calculus. In Jörg H. Siekmann and Graham Wrightson, editors, *Automation of Reasoning*, Symbolic Computation, pages 466–483. Springer Berlin Heidelberg, 1983. ISBN 978-3-642-81957-5. doi: 10.1007/978-3-642-81955-1_28. URL http://dx.doi.org/10.1007/978-3-642-81955-1_28.

[139] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987. doi: 10.1145/7531.8928. URL http://doi.acm.org/10.1145/7531.8928.

[140] Alasdair Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1(4):425–467, 1995. URL http://www.math.ucla.edu/~asl/bsl/0104/0104-003.ps.

[141] Alasdair Urquhart. The depth of resolution proofs. *Studia Logica*, 99(1-3):349–364, 2011. doi: 10.1007/s11225-011-9356-9. URL http://dx.doi.org/10.1007/s11225-011-9356-9.

[142] Alasdair Urquhart. A near-optimal separation of regular and general resolution. *SIAM J. Comput.*, 40(1):107–121, 2011. doi: 10.1137/090772897. URL http://dx.doi.org/10.1137/090772897.

[143] Michele Zito. An upper bound on the space complexity of random formulae in resolution. *ITA*, 36(4):329–339, 2002. doi: 10.1051/ita:2003003. URL http://dx.doi.org/10.1051/ita:2003003.

A

Appendix

A.1 r-BGT families

A *piecewise assignment* α to a set of variables X is a set of non-empty partial assignments to X, with pairwise disjoint domains.

A piecewise assignment α to X naturally gives rise to a partial assignment to X, namely $\bigcup\alpha$, the union of all the partial assignments in α. It also gives rise to a partition of the domain of $\bigcup\alpha$, into the set of domains of all the members of α. Therefore an alternative, but notationally less convenient, way to define a piecewise assignment would be as such a pair of a partial assignment and a partition of its domain, and we will often write α when our intended meaning is the partial assignment $\bigcup\alpha$. For example, we will write $\alpha(\varphi)$ for the evaluation of φ under $\bigcup\alpha$, and $\mathrm{dom}(\alpha)$ for the domain of $\bigcup\alpha$

We call the elements of α the *pieces* of α. For piecewise assignments α, β we will write $\alpha \sqsubseteq \beta$ to mean that every piece of α appears in β. We will write $\|\alpha\|$ to mean the number of pieces in α. Note that these are formally exactly the same as $\alpha \subseteq \beta$ and $|\alpha|$, using the definition of α and β as sets of partial assignments.

The following definition is the main combinatorial tool used in [40] to prove total space lower bounds. It was inspired by similar, but more complicated, combinatorial properties used in [36, 37]. The analysis of that combinatorial objects, the r-BG families, and their relation with monomial space in PCR, is in Chapter 3.

Definition A.1 (r-BGT)**.** *A non-empty family $\mathcal{H}$ of piecewise assignments is* r-free for a CNF φ *if it has the following properties.*

(Consistency) *No $\alpha \in \mathcal{H}$ falsifies any clause from φ.*

(Retraction) *If $\alpha \in \mathcal{H}$, β is a piecewise assignment and $\beta \sqsubseteq \alpha$ then $\beta \in \mathcal{H}$.*

(Extension) *If $\alpha \in \mathcal{H}$ and $\|\alpha\| < r$, then for every variable $x \notin \mathrm{dom}(\alpha)$ there exist $\beta_0, \beta_1 \in \mathcal{H}$ with $\alpha \sqsubseteq \beta_0, \beta_1$ such that $\beta_0(x) = 0$ and $\beta_1(x) = 1$.*

Theorem A.2 (Bonacina et al. [40])**.** *Let φ be an unsatisfiable CNF formula. If there is a non-empty family of piecewise assignments which is r-BGT for φ, then*

$$\mathsf{TSpace}_{\mathsf{Res}}(\varphi \vdash \bot) \geqslant r^2/4.$$

More precisely, any Resolution refutation of φ must pass through a memory configuration containing at least $r/2$ clauses each of width at least $r/2$.

Notice that, although independently introduced, the r-BGT families are similar to r-BK families and in both cases we do not require that the families are closed under generic restrictions. The r-BGT families are closed under *some* restrictions respecting the piecewise structure of the assignments. It turns out that this is an inessential feature in the proof of Theorem A.2. Hence we have the proof of Theorem A.2 in [40] is essentially the same of Theorem 2.5.

A.2 Asymmetric width, full proofs

In this section we collect some results on asymmetric width that are needed in Section 2.6.

Restated Theorem 2.9 (Beyersdorff and Kullmann [32, Theorem 22])**.** *Let φ be an unsatisfiable CNF formula, then $\mathsf{awidth}(\varphi \vdash_{\mathsf{Res}} \bot) > r$ if and only if there exists a non-empty r-BK family of assignments for φ.*

Proof. Suppose that the $\mathsf{awidth}(\varphi \vdash_{\mathsf{Res}} \bot) > r$, then let

$$S = \{C \text{ clause} : \mathsf{awidth}(\varphi \vdash C) \leqslant r\},$$

then clearly $\varphi \subseteq S$ and $\bot \notin S$. Let $\mathscr{F}$ be the family of all the partial assignments of maximal size that are not falsifying a clause in S. Then, since $\varphi \subseteq S$ we have that $\mathscr{F}$ is consistent and since $\bot \notin S$, then $\lambda \in \mathscr{F}$ so $\mathscr{F}$ is non-empty. We have to show the *extension* property of $\mathscr{F}$: let $\alpha \in \mathscr{F}$, $\beta \subseteq \alpha$ such that $|\beta| < r$ and $x \notin \mathrm{dom}(\alpha)$. For ease of notation, given $\epsilon \in \{0, 1\}$ let

$$x^{\epsilon} = \begin{cases} x & \text{if } \epsilon = 0, \\ \neg x & \text{if } \epsilon = 1. \end{cases}$$

By maximality of α we have that for each $\epsilon \in \{0,1\}$ there exists a clause C_ϵ in S such that $\alpha_\epsilon = \alpha \cup \{x \mapsto \epsilon\}$ falsify C_ϵ. Since $\alpha \in \mathscr{F}$ then $x \in \mathrm{var}(C_\epsilon)$, so it must be that $C_\epsilon = C'_\epsilon \vee x^\epsilon$ where C'_ϵ is a clause such that $\mathrm{var}(C'_\epsilon) \subseteq \mathrm{dom}(\alpha)$ and $\alpha(C'_\epsilon) = \bot$. Suppose, for sake of contradiction, that there exists $\epsilon \in \{0,1\}$ such that there is no $\beta' \in \mathscr{F}$ such that $\beta' \supseteq \beta$ and $\beta'(x) = \epsilon$. In particular $\beta_\epsilon = \beta \cup \{x \mapsto \epsilon\}$ is not in $\mathscr{F}$. Then, by construction, there exists a clause $D \in S$ such that $\beta_\epsilon(D) = \bot$. Since $|\beta_\epsilon| = |\beta| + 1 \leqslant r$ then $|D| \leqslant r$. Since $\beta \subseteq \alpha$ and $\alpha \in \mathscr{F}$ does not falsify any clause in S, then it must be that $D = D' \vee x^\epsilon$ and D' is a clause such that $\alpha(D') = \beta(D') = \bot$. But now

$$\frac{D \qquad C_{1-\epsilon}}{D' \vee C'_{1-\epsilon}}$$

is a valid instance of the Res rule. Hence, by definition of asymmetric width,

$$\mathsf{awidth}(\varphi \vdash D' \vee C'_{1-\epsilon}) \leqslant \max\{\mathsf{awidth}(\varphi \vdash D), \mathsf{awidth}(\varphi \vdash C_{1-\epsilon}), \mathsf{awidth}(D' \vee C'_{1-\epsilon})\},$$

that means that $\mathsf{awidth}(\varphi \vdash D' \vee C'_{1-\epsilon}) \leqslant r$ and hence $D' \vee C'_{1-\epsilon} \in S$. On the other hand $\alpha(D' \vee C'_{1-\epsilon}) = \bot$ contradicting the fact that $\alpha \in \mathscr{F}$.

Suppose now that we are given a non-empty r-BK family $\mathscr{F}$ and, by contradiction, suppose that there exists a sequence of clauses $\pi = (C_1, \ldots, C_\ell)$ such that $\mathsf{awidth}(\pi) \leqslant r$ and π is a resolution refutation of φ. We show, by induction on $i = 1, \ldots, \ell$, that no assignment for $\mathscr{F}$ falsify C_i. Since $\mathscr{F}$ is non-empty when $i = \ell$ this will be a contradiction. By the *consistency* property of r-BK families, no assignment in $\mathscr{F}$ can falsify clauses from φ. So the only case to consider is when C_i is inferred by some $C_j, C_{j'}$ in π such that $j, j' < i$. Let x be the variable resolved in the inference $\frac{C_j \quad C_{j'}}{C_i}$. Since $\mathsf{awidth}(\pi) \leqslant r$, without loss of generality suppose that $|C_j| \leqslant r$. Let $\alpha \in \mathscr{F}$. By the inductive hypothesis we have that α does not falsify C_j. If, by contradiction, $\alpha(C_i) = \bot$ then it cannot be that $x \in \mathrm{dom}(\alpha)$, otherwise α will falsify one among C_j and $C_{j'}$. Let $\beta \subseteq \alpha$ the restriction of α to $\mathrm{var}(C_j) \setminus \{x\}$. Since $|C_j| \leqslant r$ then $|\beta| < r$. By the extension property of $\mathscr{F}$ there are $\beta_0, \beta_1 \in \mathscr{F}$ such that $\beta_0 \supseteq \beta \cup \{x \mapsto 0\}$ and $\beta_1 \supseteq \beta \cup \{x \mapsto 1\}$. Either β_0 or β_1 falsify C_j. Contradicting the inductive hypothesis. □

Restated Theorem 2.10 (Kullmann [100, Lemma 8.5])**.** *Let φ be an unsatisfiable k-CNF formula, then*

$$\mathsf{awidth}(\varphi \vdash \bot) \leqslant \mathsf{width}(\varphi \vdash \bot) \leqslant \mathsf{awidth}(\varphi \vdash \bot) + \max\{\mathsf{awidth}(\varphi \vdash \bot), k\}.$$

Proof. Clearly $\mathsf{awidth}(\varphi \vdash \bot) \leqslant \mathsf{width}(\varphi \vdash \bot)$, hence we focus on proving the other inequality. Given a set of clauses A we call *A-input Resolution derivation* of a clause C a Resolution derivation of C from A such that each application of

A-input derivation

the inference rule has at least a premise from A. The main property of A-input Resolution derivations is the following: if there exists an A-input derivation of a clause C then

$$\mathsf{width}(A \vdash C) \leqslant \mathsf{width}(A) + |C|. \tag{A.1}$$

Notice that, to prove the property above, we can restrict to consider A-input *refutations*, that is A-input derivations of the empty clause $\bot$. Indeed, suppose we have π an A-input derivation of a clause C, and let ρ the minimal partial assignment mapping C to false. Clearly $|\rho| \leqslant |C|$ and $\pi|_\rho$ is a $A|_\rho$-input refutation, hence, if the property we want to prove holds for input refutations, then $\mathsf{width}(A|_\rho \vdash \bot) \leqslant \mathsf{width}(A|_\rho)$. So, by the fact that ρ is killing at most $|C|$ literals from each clause, then $\mathsf{width}(A \vdash C) \leqslant \mathsf{width}(A) + |C|$.

So now we focus on proving the property in equation (A.1) in the case when $C = \bot$ and there exists an A-input Resolution refutation. Let $\mathcal{A}$ be the set of all set of clauses A that have an A-input Resolution refutation but $\mathsf{width}(A \vdash \bot) > \mathsf{width}(A)$. By contradiction suppose that $\mathcal{A}$ is non-empty, so there will be some $\bar{A} \in \mathcal{A}$ with the minimum number of variables. Since $\bar{A} \in \mathcal{A}$ then it must be that $\bar{A}$ is non-trivial, e.g. $\bot$ cannot appear in $\bar{A}$.

By hypothesis there exists some $\bar{A}$-input refutation π and let ℓ be the last literal resolved in π. Since π is an $\bar{A}$-input refutation it must be that either $\ell \in \bar{A}$ or $\neg\ell \in \bar{A}$. Without loss of generality suppose that $\neg\ell \in \bar{A}$. Now consider $\pi|_{\ell=0}$, this is an $\bar{A}|_{\ell=0}$-input Resolution refutation and $\bar{A}|_{\ell=0}$ has strictly less variables than $\bar{A}$, hence, by the minimality of $\bar{A}$, it cannot be in $\mathcal{A}$. So $\mathsf{width}(\bar{A}|_{\ell=0} \vdash \bot) \leqslant \mathsf{width}(\bar{A}|_{\ell=0})$ and there must exist some π' which is a refutation of $\bar{A}|_{\ell=0}$ and such that $\mathsf{width}(\pi') \leqslant \mathsf{width}(\bar{A}|_{\ell=0})$.

Now we just construct π'' as follows: $\pi'' = (\bar{A}, \pi')$, that is we just write down before π' all the clauses in $\bar{A}$. Notice that π'' is not, in general, a valid $\bar{A}$-input Resolution refutation. Still π'' is a valid Resolution refutation of $\bar{A}$. This is because $\neg\ell \in \bar{A}$ and hence each clause in $\bar{A}|_{\ell=0}$ can be seen as the result of an inference step between some clause in $\bar{A}$ and $\neg\ell$. Since $\mathsf{width}(\pi') \leqslant \mathsf{width}(\bar{A}|_{\ell=0})$, we clearly have that $\mathsf{width}(\pi'') \leqslant \mathsf{width}(\bar{A})$, which implies that

$$\mathsf{width}(\bar{A} \vdash \bot) \leqslant \mathsf{width}(\pi'') \leqslant \mathsf{width}(\bar{A}). \tag{A.2}$$

On the other hand $\bar{A} \in \mathcal{A}$ implies that $\mathsf{width}(\bar{A} \vdash \bot) > \mathsf{width}(\bar{A})$ and this clearly contradicts equation (A.2).

S Let w be $\mathsf{awidth}(\varphi \vdash \bot)$ and consider the following set S defined as the

closure of φ under input derivations, that is:

$$\begin{cases} S_0 & = \varphi, \\ S_{i+1} & = S_i \cup \{C \text{ clause} : |C| \leqslant w \wedge C \text{ has an } S_i\text{-input Resolution derivation}\}, \\ S & = \bigcup_i S_i. \end{cases}$$

Notice that each clause in S has width at most $\max\{w, k\}$ and hence S is just a finite union as S_{i+1} can be strictly bigger than S_i at most $O(n^{\max\{w,k\}})$ many times, since this is the number of clauses in n variables of width at most $\max\{w, k\}$. Now we claim to have the two following properties:

1. $\bot$ has an S-input Resolution derivation;
2. if C has an S-input Resolution derivation then

$$\mathsf{width}(\varphi \vdash C) \leqslant w + \max\{w, k\}. \tag{A.3}$$

Then (1) and (2) immediately imply the inequality between $\mathsf{width}(\varphi \vdash \bot)$ and $\mathsf{awidth}(\varphi \vdash \bot)$ we want to prove.

To prove (1), consider a refutation π of φ such that $\mathsf{awidth}(\pi) = w$: we claim that π is an S-input Resolution derivation of $\bot$. Let, by contradiction, C be the first clause in π inferred from previous C', C'' in π with both $C', C'' \notin S$. Since $\mathsf{awidth}(\pi) = w$ we have that without loss of generality $|C'| \leqslant w$, hence it must be that for each i, C' does not have an S_i-input Resolution derivation, otherwise $C' \in S_{i+1}$ but we are supposing that $C' \notin S$. Hence, C' doesn't have a S-input Resolution derivation either, contradicting the minimality of C in π.

We now prove (2) by induction on S_i. That is, we prove that if C has an S_i-input Resolution derivation then $\mathsf{width}(\varphi \vdash C) \leqslant w + \max\{w, k\}$.

For S_0 this is clearly true. For the inductive step let C be a clause in $S_{i+1} \setminus S_i$, let $S_i = \{C_1, \ldots, C_m\}$ and let π be an S_i-input Resolution derivation of C. By what observed before, there exists some π' which is a Resolution derivation of C from S_i such that

$$\mathsf{width}(\pi') \leqslant |C| + \mathsf{width}(S_i) \leqslant w + \max\{w, k\}.$$

Finally, by induction, for each $j = 1, \ldots, m$, C_j has a Resolution derivation π_j from φ of width at most $w + \max\{w, k\}$, hence

$$\tilde{\pi} = (\pi_1, \ldots, \pi_m, \pi')$$

is a Resolution derivation of C from φ and

$$\begin{aligned} \mathsf{width}(\varphi \vdash C) &\leqslant \mathsf{width}(\tilde{\pi}) \\ &= \max\{\mathsf{width}(\pi_1), \ldots, \mathsf{width}(\pi_m), \mathsf{width}(\pi')\} \\ &\leqslant w + \max\{w, k\}. \end{aligned}$$

□