On the Complexity of Resolution-based Proof Systems

by

Sergi Oliva

PhD Thesis submitted to the
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Directed by

Albert Atserias

March 2013

Abstract

Propositional Proof Complexity is the area of Computational Complexity that studies the length of proofs in propositional logic. One of its main questions is to determine which particular propositional formulas have short proofs in a given propositional proof system. In this thesis we present several results related to this question, all on proof systems that are extensions of the well-known resolution proof system.

The first result of this thesis is that TQBF, the problem of determining if a fully-quantified propositional CNF-formula is true, is PSPACE-complete even when restricted to instances of bounded tree-width, i.e. a parameter of structures that measures their similarity to a tree. Instances of bounded tree-width of many NP-complete problems are tractable, e.g. SAT, the boolean satisfiability problem. We show that this does not scale up to TQBF. We also consider Q-resolution, a quantifier-aware version of resolution. On the negative side, our first result implies that, unless NP = PSPACE, the class of fully-quantified CNF-formulas of bounded tree-width does not have short proofs in any proof system (and in particular in Q-resolution). On the positive side, we show that instances with bounded respectful tree-width, a more restrictive condition, do have short proofs in Q-resolution. We also give a natural family of formulas with this property that have real-world applications.

The second result concerns interpretability. Informally, we say that a first-order formula can be interpreted in another if the first one can be expressed using the vocabulary of the second, plus some extra features. We show that first-order formulas whose propositional translations have short R(const)-proofs, i.e. a generalized version of resolution with DNF-formulas of constant-size terms, are closed under a weaker form of interpretability (that with no extra features), called definability. Our main result is a similar result on interpretability. Also, we show some examples of interpretations and show a systematic technique to transform some Σ_1-definitions into quantifier-free interpretations.

The third and final result is about a relativized weak pigeonhole principle. This says that if at least $2n$ out of n^2 pigeons decide to fly into n holes, then some hole must be doubly occupied. We prove that the CNF encoding of this principle does not have polynomial-size DNF-refutations, i.e. refutations in the generalized version of resolution with unbounded DNF-formulas. For this proof we discuss the existence of unbalanced low-degree bipartite expanders satisfying a certain robustness condition.

Acknowledgements

First and foremost, I would like to thank my advisor, Albert, for his guidance and support at all times. Working with him has been, no doubt, the most challenging experience of my life. Even though he has tirelessly introduced me into lots of different topics and taught me hundreds of interesting things, the most profound influence he has had on me has come from seeing him work day by day. The way I think and act now, not only as a researcher, but as a person, has undoubtely changed due to my experience with him: the rigorous thinking, the attention to detail, and the will to do things the right way, are all proofs of his imprint. Sitting together in front of a blackboard, we have shared some moments of inspiration and enthusiasm, some moments of laughs and fun, but also some really tough moments of failure and disappointment. In those, he has always been a steady hand pointing to the light at the end of the tunnel when I was not sure there was such a thing. Even in those difficult times he has not stopped believing in me, and I am thankful for that.

I would also like to thank Moritz Müller for being so patient and kind in our work together. It has been a pleasure getting to know him. I thank my office mates at UPC for their unwavering support and their always fun but mostly off-topic conversation. Specially, I would like to thank Marc Galceran: starting at undergraduate, we have made the whole journey together, and it has been a good one. Furthermore, I would like to thank Jordi Cortadella for encouraging me to get into research, and Rafel Cases, whose Theory course in my second year at the university opened a new and fascinating world for me. Probably, without his initial spark, none of this would have been a reality.

Finally, I would like to thank my friends, my family and Cristina. They've been supportive at all times: encouraging me in the toughest moments, sharing my joy in the good ones, and patiently listening to my long and boring explanations way too many times. Their support has been key.

Contents

Chapter 1

Introduction

Since ancient times, one of the main philosophical struggles of humanity has been to distinguish truth from falsehood. In 1685, in an effort to approach this problem from a mathematical point of view, Gottfried Leibniz suggested to develop a universal language, or *characteristica universalis*, able to express every sentence in some sort of formal, universal way, together with a universal calculation framework, or *calculus ratiocinator*, a formal inference system that would allow to reason about general statements with the formality with which matemathicians reason about their own. Thus, a group of philosophers disputing the truth of a statement could simply say *calculemus!* (let's calculate) and use the system to determine who is right.

Later developments by Hilbert, Boole, Frege and others seemed to point to mathematical logic as the prime candidate for this role. However, in 1931, Gödel's incompleteness theorem showed that the task at hand was, indeed, impossible. Nonetheless, the spirit of Leibniz' dream did not die and, even with its known limitations, logic has been at the heart of the technological revolution of the last decades as the universal language and reasoning framework that Leibniz envisioned.

1.1 Logic expressions

One of the frameworks provided by logic to express and reason about statements is *first-order logic*. In first-order logic, we have a universe with relations and functions on it, along with variables to represent elements of the universe, logical connectives for negation ($\neg$), conjunction ($\wedge$) and disjunction ($\vee$), and existential and universal quantification ($\exists, \forall$) over these variables.

For instance, given a graph $H = (V, E)$ where V is a set of vertices and E is a binary relation indicating whether two vertices are an edge in H, we can express that H is 3-colorable as the fact that the first-order formula

$$\begin{aligned}
&\forall x\ (R(x) \vee G(x) \vee B(x)) \wedge \\
&\forall x \forall y\ (E(x,y) \implies \neg R(x) \vee \neg R(y)) \wedge \\
&\forall x \forall y\ (E(x,y) \implies \neg G(x) \vee \neg G(y)) \wedge \\
&\forall x \forall y\ (E(x,y) \implies \neg B(x) \vee \neg B(y))
\end{aligned}$$

is *satisfiable* in H; i.e. there are sets of vertices R, G and B that make the formula true when it is evaluated on H.

The same statement can also be expressed using *propositional logic*, that is, by means of (boolean) propositional variables and logical connectives only. To do that, we can *translate* the previous first-order formula into propositional logic using variables R_u, G_u and B_u for every vertex u. The statement is now expressed as the fact that the formula for H

$$\begin{aligned}
&R_u \vee G_u \vee B_u && \text{for every vertex } u, \\
&\neg R_u \vee \neg R_v && \text{for every edge } (u,v), \\
&\neg G_u \vee \neg G_v && \text{for every edge } (u,v), \\
&\neg B_u \vee \neg B_v && \text{for every edge } (u,v)
\end{aligned}$$

is satisfiable; i.e. there is a boolean assignment to the propositional variables that makes the formula true. This translation step is of great significance in this thesis, as expressing our statements in propositional logic allows to reason about them syntactically by using *propositional proof systems*, as we will introduce later.

As witnessed by 3-colorability, combinatorics is a good source of examples of mathematical statements that can be logically expressed. Another of these examples is the (non-)existence of a perfect matching in a bipartite graph. The particular case in which the graph is the complete bipartite graph with vertex sets of cardinalities $n+1$ and n is, precisely, the so-called *pigeonhole principle*. More graphically, the pigeonhole principle states that if $n+1$ pigeons fly into n holes, at least one hole will be doubly occupied. We can express this principle using propositional logic. To do that, we use boolean variables $P_{u,v}$ that indicate whether pigeon u flies to hole v, for all $u \in \{0, \ldots, n\}$ and $v \in \{1, \ldots, n\}$. The

principle is expressed by the fact that the following set of clauses is contradictory:

$$\begin{array}{ll} P_{u,1} \vee P_{u,2} \vee \ldots \vee P_{u,n} & \text{for every pigeon } u, \\ \neg P_{u,v} \vee \neg P_{u',v} & \text{for all different pigeons } u, u' \text{ and hole } v. \end{array}$$

Note that the first set of clauses forces every pigeon to fly to some hole and the second ensures that no hole will be doubly occupied. Therefore, refuting these clauses, that is, proving that they are contradictory, would prove the principle true. Observe that the principle is not expressed by the fact that the first-order formula

$$\begin{array}{l} \forall x \exists y \; (y \neq 0 \wedge P(x, y)) \wedge \\ \forall x \forall y \forall z \; (x \neq y \implies \neg P(x, z) \vee \neg P(y, z)). \end{array}$$

is contradictory, since it is not: it is satisfiable when the universe is infinite. The principle is expressed by the fact that the formula has no finite models of cardinality $n + 1$.

Other well-known examples of combinatorial principles that can be expressed in propositional and first-order logic include the Ramsey principle, that states that every graph with 2^{2n} vertices contains either a complete subgraph or an independent set of size n. Also, the least number principle, which states that every finite linear order has a minimum, or the dense linear order principle, which states that no finite linear order is dense, i.e. every pair of elements has another inbetween.

The ability to express and reason about logical statements is also relevant in multiple areas of computer science. Logic is at the heart of programming languages, artificial intelligence and security protocols; even queries on relational databases are no more than first-order formulas. Constraint satisfaction problems coming from a broad range of scientific and industrial applications are expressed as propositional formulas (or extensions of those) to take advantage of the powerful technology for determining satisfiability (SAT solvers). Also, the computational complexity of determining a property is intimately related to the difficulty of formally expressing it in logic [32]. Furthermore, the verification of complex systems, be they software or hardware, is based on model checking by verifying logic statements about the structure or the behaviour of those systems. For example, these statements can be invariants that guarantee that a given system does not reach an undesired state.

A particular sort of systems that we may want to verify are those in which there is an interaction between two or more parties, for example between the system itself and an external user. In this case, our invariants are required to take into account this interaction. To do

so, it is useful to consider an extension of propositional logic to state propositional formulas compactly by using existential and universal quantification of propositional variables. These formulas are called *quantified boolean formulas* (QBFs). A QBF takes the form

$$\forall x_1 \exists x_2 \exists x_3 \forall x_4 \ldots \forall x_k(\phi)$$

where ϕ is a propositional formula on the variables $x_1, \ldots, x_k$. Note that, even if they look like first order formulas, QBFs are no more than propositional formulas stated in abbreviated, compact form. Given a QBF, one can easily obtain an equivalent propositional formula, but its size may be exponential in the number of quantifiers. QBFs are a good fit to model adversarial games like the user-system interaction mentioned earlier, as well as other examples from Game Theory.

1.2 Propositional Proof Complexity

The main reason for using propositional logic to express statements is to reason in a purely syntactical way by means of *propositional proof systems*. Specifically, we want to *refute* propositional formulas, i.e. proving that they are a contradiction. Note that refuting a formula is equivalent to proving that its negation is a *tautology*, i.e. true for all boolean assignments.

The best-known proof system to refute propositional formulas is *resolution*. This is a rule-based system with a single rule that reads

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D},$$

where C and D are clauses and x is a variable. Note that the rule is *sound*, in the sense that every assignment that satisfies both hypotheses, also satisfies the conclusion. Resolution is *complete* for sets of clauses (also known as CNF-formulas), i.e. every contradictory set of clauses has a resolution refutation. Note that the examples in the previous section are precisely sets of clauses.

Proof systems used in this thesis are extensions of resolution. The first one is DNF-resolution, that generalizes resolution by allowing x to be a term, that is, a conjunction of literals, and C and D to be DNF-formulas, that is, disjunctions of terms. Note that the rule is still sound in this case. Additional rules are introduced to make this proof system complete for all contradictory sets of DNF-formulas. We make special attention to the specific case

in which the size of the conjuctions is bounded by a natural k. This system is sometimes called k-DNF resolution, or R(k) for short [34]. The second resolution-based proof system used in this thesis is Q-resolution [13], a quantifier-aware version of resolution for refuting QBFs. This is a proof system with rules similar to those of resolution but that take into account the quantification of the variables, which is ignored in plain resolution.

Consider the satisfiability problem of propositional formulas (SAT). Given a formula, we can use a satisfying boolean assignment to its variables as a certificate of its satisfiability. On the other hand, a refutation in a particular proof system certifies that the formula is contradictory. Note that a satisfying assignment is not only a certificate, but a short one: its length is exactly the number of variables of the formula, and this puts SAT in NP. However, refutations of contradictory formulas need not be *short*, i.e. of length polynomial with respect to the size of the formula.

Propositional Proof Complexity is the area of Computational Complexity that studies the *length* of proofs. Its main question is to determine if there is a short proof of a given propositional formula in a given proof system. Results in this area have impact in areas that range from efficiency improvements of SAT solvers [37], up to fundamental questions in Computational Complexity, in particular the P vs. NP problem [19].

1.3 Our results

In this section we present the main contributions of this thesis and put them in the context of previous related work.

1.3.1 Bounded tree-width QBFs and Q-resolution

Tree-width is a well-known parameter that measures how close a structure is to being a tree. Many NP-complete problems have polynomial-time algorithms on inputs of bounded tree-width. In particular, SAT can be solved in polynomial time when the constraint graph of the input CNF-formula has bounded tree-width (cf. [25], [27]).

Tractability of TQBF A natural question suggested by this result is whether TQBF, the problem of determining if a QBF whose quantifier-free part is a CNF is true or false, can also be solved in polynomial time when restricted to formulas whose CNF-formula has bounded tree-width. In [16], Chen concludes that the problem stays tractable if the number of quantifier alternations, as well as the tree-width, is bounded. On the negative side,

Gottlob, Greco and Scarcello [30] proved that the problem stays PSPACE-complete when the number of alternations is unbounded even if the constraint graph of the CNF-formula has logarithmic tree-width (and indeed, its *incidence* graph is even a tree). By different methods, and improving upon [30], Pan and Vardi [40] show that, unless P = NP, the dependence of the running time of Chen's algorithm on the number of alternations must be non-elementary, and that the TQBF problem restricted to instances of tree-width $\log^*$ in the size of the input is PSPACE-complete. All these negative results hold also for path-width, which is a parameter that measures the similarity to a path and is in general smaller than tree-width. However, they leave open whether TQBF is tractable for instances whose constraint graph has constant path-width, or even constant tree-width.

Main result and proof techniques We resolve this question by showing that, even for inputs of constant path-width, TQBF is PSPACE-complete. Our construction builds on the techniques from [40] with two essential differences. The first difference is that instead of reducing from the so-called *tiling-game* and producing a quantified Boolean formula of $\log^*$-smaller path-width, our reduction starts at TQBF itself and produces a quantified Boolean formula whose path-width is *only* logarithmically smaller. Although this looks like backward progress, it leaves us in a position where iterating the reduction makes sense. However, in order to do so, we need to analyze which properties of the output of the reduction can be exploited by the next iteration. Here comes the second main difference: we observe that the output of the reduction has not only smaller path-width, but also smaller *window-size*, which means that any two occurences of the same variable appear close to each other in some ordering of the clauses. We call such formulas *n-leveled*, where n is a bound related to the window-size. Our main lemma exploits this structural restriction in a technical way to show that the TQBF problem for n-leveled formulas reduces to the TQBF problem for $O(\log n)$-leveled formulas. Iterating this reduction until we reach $O(1)$-leveled formulas yields the result.

A few more words on the differences between our methods and those in [40] and [30] are in order. The technical tool from [40] that is used to achieve n-variable formulas of $O(\log^* n)$ path-width builds on the tools from [38] and [29] that were used for showing non-elementary lower-bounds for some problems related to second-order logic. These tools are based on an encoding of natural numbers that allows the comparison of two n-bit numbers by means of an extremely smaller formula; one of size $O(\log^* n)$. It is interesting that, by explicitely avoiding this technique, our iteration-based methods take us further: beyond $O(\log^* n)$ path-

width down to constant path-width. For the same reason our proof can stay purely at the level of propositional logic without the need to resort to second-order logic. Along the same lines, our method also shows that the TQBF problem for n-variable formulas of constant path-width and $O(\log^* n)$ quantifier alternations is NP-hard (and Σ_iP-hard for any $i \geq 1$), while the methods from [40] could only show this for $O(\log^* n)$ path-width and $O(\log^* n)$ alternations. It is worth noting that, in view of the results in [16], these hardness results are tight up to the hidden constants in the asymptotic notation.

QCSP and respectful tree-width Structural restrictions on the generalization of TQBF to unbounded domains, sometimes called QCSP, have also been studied. Gottlob et al. [30] proved that QCSP restricted to trees is already PSPACE-complete. Their hardness result for QBFs of logarithmic tree-width follows from this by *booleanization*. They also identify some new tractable fragments, and some other hardness conditions. Finally, Chen and Dalmau [18] introduced a general framework for studying structural restrictions on QCSP, and characterized the restrictions that make the problem tractable under complexity-theoretic assumptions.

One of the restrictions of QCSP that Chen and Dalmau showed tractable is that the constraint graph of the instance has bounded *respectful tree-width*. Note that the tree-width of the constraint graph is independent of the quantification of the instance. Respectful tree-width is precisely a quantifier-aware parameter, that considers only tree-decompositions that are respectful with the quantification, in the sense that bottom-up algorithms can be run on these tree-decompositions without violating precedence of quantifiers.

QBFs of bounded respectful tree-width In this thesis we observe that QBFs of bounded respectful tree-width are not only tractable but also have short Q-resolution proofs. We start by presenting different forms of quantifier-aware resolution introduced by Büning, Flögel and Karpinski [13] and Pan and Vardi [39] and show how they relate to each other. Next, we show that respectful tree-width is equivalent to *respectful induced width*. Here *induced width* refers to a measure equivalent to tree-width introduced in [24]. Finally, we show that false QBFs with bounded respectful induced width have short Q-resolution refutations, which yields our result.

As an application of this result, we show that a family of formulas inspired by one introduced by Dalmau, Kolaitis and Vardi [20], has bounded respectful tree-width. We give practical examples of how these formulas are useful.

1.3.2 Definability and interpretability

As mentioned earlier, Propositional Proof Complexity studies the length of refutations of propositional contradictions. We would like to compare contradictions in terms of how hard they are for a specific proof system, that is, determine if one has a (significantly) shorter refutation than the other.

One trivial way to determine if a formula is harder than another is to prove explicit bounds on the length of the refutations of the formulas that we want to compare. However, this requires solving a problem much bigger than the one we are interested in. A simpler, syntactical way to proceed would be the following: given two contradictions F and G and a proof system P, if there is a short proof of F from G in P, then it is clear that G is no harder than F in P, or equivalently, that F is at least as hard as G in P. This solution looks easier than the previous, but it still requires building a syntactical proof of one of the formulas from another, which may not be an easy task. We would like to compare formulas by a simpler, *semantical* argument. For a fixed proof system, this would be, in some sense, analogous to reducing one problem into another in the theory of NP-completeness in classic complexity. It is in the search of this semantical argument that we focus our interest on *interpretability*.

Interpretability Interpretability is a classic concept of model theory. Informally, interpreting a model M in a model N is expressing M in the vocabulary of N. Interpretability of formulas is as expected: given two first-order formulas ϕ and ψ, interpreting ϕ in ψ is to express the functions and relations of ϕ as formulas in the vocabulary of ψ ensuring that, for all models of ψ, the interpretation of ϕ is also satisfied, among other requirements. In this thesis, we also consider the concept of *definability*, a particular form of interpretability. The expressive power needed for the formulas that express functions and relations determines the complexity of a specific interpretation (or definition).

In [21], Dantchev and Martin showed that if the *relativization* of the aforementioned least number principle, that is, the sentence expressing the principle for all non-empty subsets of the universe, can be interpreted in a first-order formula without finite models, then there are short R(const)-refutations of the propositional translations of that formula. The interpretations allowed are those in which relations and functions are expressed as Σ_1-formulas except for the relativization relation, that is necessarily expressed as a quantifier-free formula.

Our result In this thesis we systematize Dantchev and Martin's idea for all formulas. First we show that the set of first-order formulas whose propositional unary translations have polynomial-size R(const)-refutations is closed under quantifier-free definitions. Here, unary translation corresponds to the translation from first-order to propositional formulas that has been used throughout this introduction. Our main theorem is a similar version of this result on interpretability. We also prove similar results for *binary translations*, that is, propositional translations that encode the first-order functions by their *bit-graph*, for quasipolynomial proofs in R(log).

Finally, as an application of our results, we give some examples of definitions and interpretations among those principles that have been introduced, specially among different forms of the pigeonhole principle.

Proof techniques To prove these results, we first show a generalization of the upper bound of Riis' Gap Theorem [47] for tree-like resolution. Riis' result is on purely universal formulas, and we generalize it to Σ_2-formulas, as required by our application. Furthermore, we use a distributivity lemma that allows us to convert depth-3-refutations of the form "disjunction-conjunction-disjunction" into DNF-refutations without considerably increasing its size, provided that the innermost disjunctions are of bounded size. Finally, in the examples, we show a systematic technique to convert some Σ_1-definitions into quantifier-free interpretations.

1.3.3 Lower bounds for DNF-refutations

A generalized version of the classical pigeonhole principle, namely PHP_n^m, expresses the fact that there is no injection from m pigeons into n holes whenever m is bigger than n. As usual, we formulate PHP_n^m as a contradictory CNF in the propositional variables $P_{u,v}$ with u ranging over an m-element set $[m]$ of pigeons and v ranging over an n-element set $[n]$ of holes. The formula has clauses $\neg P_{u,v} \vee \neg P_{u',v}$ for $u, u' \in [m]$ with $u \neq u'$ and $v \in [n]$ forcing different pigeons to fly to different holes, and $\bigvee_{v\in[n]} P_{u,v}$ for $u \in [m]$ forcing every pigeon to fly to some hole. Estimating the refutation-complexity of this set of clauses in various proof systems has a long history in proof complexity dating back to Cook and Reckhow's seminal article [19].

Weak pigeonhole principles One of the most quoted results of Propositional Proof Complexity is that PHP_n^{n+1}, the particular case shown in section 1.1, does not have small

proofs in the standard propositional proof systems that "lack the ability to count". This is confirmed by the seminal results of Haken [31] for resolution, and Ajtai [1] for standard proof systems manipulating formulas of bounded depth (i.e. AC^0-Frege), followed by the great quantitative improvements by Beame, Impagliazzo and Pitassi [9] and Krajíček, Pudlák and Woods [35] on Ajtai's result. In contrast, short polynomial-size proofs exist as soon as the proof system is allowed formulas that express counting properties, such as arbitrary propositional formulas [14] (i.e. NC^1-Frege), or even threshold formulas of bounded depth (i.e. TC^0-Frege).

From the above, the ability to count looks like an essential ingredient for proving PHP_n^{n+1}. On the other hand, since *approximate counting* is available in AC^0 via explicit polynomial-size formulas [2], one may speculate that *weaker* pigeonhole principles with a much bigger gap between the number of pigeons and the number of holes, such as $PHP_n^{n^2}$ or PHP_n^{2n}, may have polynomial-size bounded-depth proofs. However, this is a notorious 25-year old open problem [42], the main obstacle being that although the known AC^0-formulas for approximate counting are explicit, their *correctness* seems hard to prove. The only known superpolynomial lower bounds are for resolution in the case of $PHP_n^{n^2}$ [44, 46], and for proofs manipulating k-DNFs with $k \leq \epsilon \log n / \log\log n$ for some $\epsilon > 0$ in the case of PHP_n^{2n} [8, 50, 45].

Indeed for those weaker pigeonhole principles, some positive results are known: Paris, Wilkie and Woods [42] proved that $PHP_n^{n^2}$ and PHP_n^{2n} do have quasipolynomial-size bounded-depth proofs. Their proof does not rely on approximate counting and is basically an amplification argument that reduces proving PHP_n^{2n} to proving an instance of $PHP_n^{n^n}$ which is then proved by a diagonalization argument. This was later improved by Maciel, Pitassi and Woods [36] who gave direct $n^{O(\log n)}$-size proofs of $PHP_n^{n^2}$ and PHP_n^{2n} by depth-2 formulas (indeed, by k-DNF formulas with $k \leq (\log n)^c$ for a constant c).

The question whether $PHP_n^{n^2}$ or PHP_n^{2n} have polynomial-size bounded-depth proofs remains open. A positive answer could have consequences for bounded arithmetic [42], and a negative answer could have consequences for our understanding of approximate counting as a computational problem.

The result Consider the following modified weak pigeonhole principle: if at least $2n$ out of n^2 pigeons fly into n holes, then some hole must be doubly occupied. We call $PHP_n^{n^2,2n}$ a propositional formula whose contradiction expresses this relativized weak pigeonhole principle in the mold of those shown in section 1.1.

The main result of this chapter is that every DNF-refutation of $PHP_n^{n^2,2n}$ has superpolyno-

mial size. By a DNF-refutation we mean a refutation in the aforementioned DNF-resolution proof system.

Proof outline and comparison to previous work Our proof follows the random restriction method, so successfully used in previous works in Propositional Proof Complexity, with some additional ideas. The typical skeleton of a proof by the random restriction method goes as follows: Assume a short proof of F is given. Apply a random restriction from a suitable distribution in such a way that, with high probability, every formula in the proof simplifies significantly, but the proved formula F remains hard. Finally argue directly that the restricted F cannot have a short proof with such simple formulas.

For an example, suppose PHP_n^{2n} has polynomial-size resolution refutations. For the random restriction we choose an assignment that describes a 1-1 mapping from $n/2$ randomly chosen pigeons onto $n/2$ randomly chosen holes, and leaves all the other variables unset. With these parameters, the restricted PHP_n^{2n} becomes $\mathrm{PHP}_{0.5\cdot n}^{1.5\cdot n}$, and each *complex* clause of the proof has been made true with high probability. Now a direct prover-adversary argument shows that a proof of $\mathrm{PHP}_{0.5\cdot n}^{1.5\cdot n}$ with non-complex clauses only is impossible.

Trying to apply this argument to DNF-refutations hits several difficulties. First, a random *matching* restriction as above is not likely to simplify an arbitrary DNF formula, even if this formula is small. Indeed, the DNF could be the negation of PHP_n^{2n} itself, and the point of the argument above was precisely that this formula does not simplify much. Here is where our modified version $\mathrm{PHP}_n^{n^2,2n}$ enters the picture. By choosing $2n$ out of n^2 pigeons at random and setting all the variables about the other pigeons completely at random, it is very likely that each DNF in the proof simplifies into one all whose terms mention very few of the $2n$ chosen pigeons. This sort of restriction comes inspired by the so-called Dantchev-Riis restrictions [22], and its analysis for our case requires arguments of the type Furst, Saxe, and Sipser introduced in their seminal work on bounded-depth circuits [28].

Continuing with the sketch of the proof, the application of the Dantchev-Riis restriction to $\mathrm{PHP}_n^{n^2,2n}$ leaves an instance of PHP_n^{2n}. Unfortunately, a term mentioning very few pigeons need not be short itself, which means that we are not yet at a contradiction with the known lower bounds for PHP_n^{2n} in k-DNF resolution for $k \leq \sqrt{\log n / \log\log n}$ from [50] which were later improved to $k \leq \epsilon \log n / \log\log n$ for some $\epsilon > 0$ [45]. Following the ideas in [11], as adapted to k-DNF proofs in [8, 50], this suggests that we restrict the principle further to a low-degree bipartite expander G to get a short proof of $\mathrm{PHP}(G)$. The low-degree condition on G guarantees that whenever a term mentions very few pigeons we can also assume that

the term is short, resulting in a k-DNF refutation of PHP(G) for small k. This would seem to open the door to using the methods in [50].

Unfortunately, the sort of bipartite expanders that are needed for the rest of the argument require degree at least as large as $\log n$, leaving k well above the quantity that a direct application of the methods in [50] can afford. Here comes the second main idea in our proof: we use a logarithmic degree expander G, but reduce our problem to proving lower bounds for a related formula BPHP(G) in which the flights of the pigeons along the edges of the graph are encoded in *binary*. This takes us from $k = \Omega(\log n)$ in the unary encoding to $k = O(\log \log n)$ in the binary encoding (at least in the case that we start with polynomial-size proofs), well below the critical $\sqrt{\log n / \log \log n}$.

Chapter 2

Preliminaries and auxiliary lemmas

In this chapter we introduce the main definitions and terminology that will be used throughout this thesis. In the last section, we prove several auxiliary lemmas of technical nature. Some standard terminology from Computational Complexity is used throughout. For that, see [5] and [41].

2.1 Basic definitions

For a natural $n \in \mathbb{N}$, we write $[n] := \{0, \ldots, n-1\}$ and $|n| := \lceil \log(n+1) \rceil$. All our logarithms are base two. Note that, for $n > 0$, the natural $|n|$ is the length of the binary representation of n without leading zeros. We define $\log^{(0)} n := n$ and $\log^{(i)} n := \log(\log^{(i-1)} n)$ for $i > 0$. Also, we use $\log^* n$ as the least integer i such that $\log^{(i)} n \leq 1$. If $\bar{a}$ is a k-tuple, its i-th component is a_i. For $b \in \mathbb{N}$ we write $\mathrm{bit}(b, n)$ for the $(b+1)$-th least significant bit in the binary representation of n; formally, $\mathrm{bit}(b, n) := \lfloor n/2^b \rfloor \mod 2$. Note that if $b \geq |n|$, then $\mathrm{bit}(b, n) = 0$.

2.2 Graphs

A *graph* is a pair $G = (V, E)$ where V is a set of *vertices* and E is a set of *edges*, that is, pairs of vertices. All graphs in this thesis are *undirected*, and therefore we use both (u, v) and (v, u) to refer to an edge between vertices u and v. Even though *bipartite graphs* are graphs and can be fit in this definition, we use a slightly modified definition for them, that will be handy for their use throughout this thesis.

Bipartite graphs Let $G = (U, V, E)$ with $E \subseteq U \times V$ be a bipartite graph. For a vertex $u \in U \cup V$ let $N_G(u)$ be the set of neighbors of u in G and for a set of vertices $A \subseteq U \cup V$, let $N_G(A) := \bigcup_{u \in A} N_G(u)$. A set $M \subseteq E$ is a *matching (in G)* if no two edges in M share an endpoint. Note that matchings M are bijections and thus have an image $\mathrm{Im}(M)$ and a domain $\mathrm{Dom}(M)$.

We say G is a (U, V, d_L, d_R)-graph if for every $u \in U$ we have that $|N_G(u)| \leq d_L$ and for every $v \in V$ we have that $|N_G(v)| \leq d_R$. With such a graph we associate a bijection ϕ_G with $\mathrm{Dom}(\phi_G) \subseteq U \times [d_L]$ such that for every $u \in U$ and every $v \in N_G(u)$ there is (exactly one) $i \in [d_L]$ such that $(u, i) \in \mathrm{Dom}(\phi_G)$ and $\phi_G(u, i) = v$. For a subset $C \subseteq U \cup V$ we let $G \cap C$ denote the subgraph of G induced by the vertices of C; if ϕ_G is associated to G, then $G \cap C$ is a $(U \cap C, V \cap C, d_L, d_R)$-graph and the map associated to $G \cap C$ is (as a set of pairs) $\phi_{G \cap C} := \phi_G \cap ((C \times [d_L]) \times C)$. We also write $G \setminus C$ for $G \cap ((U \cup V) \setminus C)$.

2.3 Propositional logic

Atoms, literals, formulas Propositional variables are also called *atoms*. A *literal* is an atom X or its *negation* $\neg X$. A *formula* is built from literals by means of $\vee$ and $\wedge$. Note that we allow the negation symbol only in front of atoms. The *negation* $\neg F$ of a formula F is defined as the formula obtained from F by interchanging $\wedge$ and $\vee$, and replacing every literal by its complementary literal (i.e. X by $\neg X$ and $\neg X$ by X). We use $\overline{X}$ for the negation of an atom X. Also, we use the notation $X^{(1)}$ and $X^{(0)}$ to denote X and $\overline{X}$, respectively. Note that the notation is chosen so that $X^{(a)}$ is made *true* by the assignment $X = a$. The *underlying variable* of $X^{(a)}$ is X, and its sign is a. We use $\mathrm{var}(\phi)$ to denote the set of variables occurring in a formula ϕ. If Γ is a set of formulas, we write $\bigwedge \Gamma$ for the iterated conjunction of the formulas in Γ; the elements in Γ are the *conjuncts*. Similarly, we write $\bigvee \Gamma$ for the iterated disjunction, and the elements of Γ are the *disjuncts*. We omit parenthesis in iterated conjuntions and disjunctions. We allow the empty disjunction 0 and the empty conjunction 1, and refer to them as *constants*. Note $\neg 1 = 0$ and $\neg 0 = 1$. Sometimes we use $\square$ for the empty disjunction. A *(k-)term* is a conjunction of (at most k many) literals; and a *(k-)clause* is a disjunction of (at most k many) literals. Both k-terms and k-clauses are said to have *width* k. A *(k-)CNF* is a conjunction of (k-)clauses, and, analogously, a *(k-)DNF* is a disjunction of (k-)terms. By CNF, sometimes we also refer to sets of clauses or even sets of conjunctions of clauses.

Proof system We define the proof system. A *structural inference* allows to pass from F to G whenever F is a disjunction (or a conjunction) and G has the same set of disjuncts (respectively, conjuncts) as F. Furthermore 0 (respectively, 1) may be freely added or deleted. The system has four further rules of inference, namely *axiom* (AXM), *weakening* (WKG), *introduction of conjunction* (IOC), and *cut* (CUT):

$$\frac{}{F \vee \neg F} \qquad \frac{H}{H \vee F} \qquad \frac{H \vee F \qquad H' \vee G}{H \vee H' \vee (F \wedge G)} \qquad \frac{H \vee F \qquad H' \vee \neg F}{H \vee H'},$$

where F, G, are formulas. Note that the common rules $\frac{}{\bar{1}}$ and $\frac{\Box}{F}$ (ex falso quodlibet) follow from (AXM) respectively (WKG) plus a structural inference.

A *proof (of G from $F_1, \ldots, F_m$)* takes assumptions $F_1, \ldots, F_m$ and produces a *conclusion* G through the application of these rules. A *refutation* of $F_1, \ldots, F_m$ is a proof of $\Box$ from $F_1, \ldots, F_m$. A *(k-)DNF-proof* is one where all formulas are (k-)DNFs. We specifically call the previous system R(k) when all formulas are k-DNFs. Consequently, we refer to a k-DNF-proof also as an R(k)-proof. Note that the resolution proof system is precisely R(1).

By $|F|$ we denote the *size* of the formula F: literals and constants have size 1, and $|(F \wedge G)| = |(F \vee G)| = 1 + |F| + |G|$. Note that $|F| = |\neg F|$. The size of a proof is the sum of the sizes of the formulas it contains.

We write $F_1, \ldots, F_m \vdash_k^s G$ if there is a R(k)-proof of size s that takes the assumptions $F_1, \ldots, F_m$ and produces G. We write $\vdash_{k,*}^s$ for tree-like such proofs. Observe that $F_1, \ldots, F_m \vdash_1^s G$ (resp. $\vdash_{1,*}^s$) if and only if there is a (resp. tree-like) resolution proof of G from $F_1, \ldots, F_m$ of size s. An R(log)*-proof* is one in which all formulas are ($\log s$)-DNFs, where s is the size of the proof.

2.4 First-order logic

First order vocabularies Fix a first-order vocabulary σ split into σ_R and σ_F, where σ_R is the set of relation symbols and σ_F is the set of function symbols. We view constant symbols as 0-ary function symbols. It will be convenient to assume that every vocabulary has at least one constant symbol that we denote by 0. For each symbol S in σ, let r_S denote its arity. If $\bar{r}$ and $\bar{s}$ are k-tuples of first-order terms, sometimes we write $\bar{r} = \bar{s}$ instead of $r_1 = s_1 \wedge \cdots \wedge r_k = s_k$. Similarly, we write $\forall \bar{x}$ and $\exists \bar{x}$ instead of $\forall x_1 \cdots \forall x_k$ and $\exists x_1 \cdots \exists x_k$, and $\psi[\bar{x}/\bar{a}]$ instead of $\psi[x_1/a_1, \ldots, x_k/a_k]$.

Universal, flattened, function-negative formulas Let ϕ be a first-order formula over σ, which by standard manipulation we may assume to have all atomic formulas of one of the following forms:

1. $x_i = x_j$ for $i, j \in \{1, \ldots, k\}$,
2. $R(x_{i_1}, \ldots, x_{i_{r_R}})$ for R in σ_R and $i_1, \ldots, i_{r_R} \in \{1, \ldots, k\}$,
3. $F(x_{i_1}, \ldots, x_{i_{r_F}}) = x_{i_0}$ for F in σ_F and $i_0, i_1, \ldots, i_{r_F} \in \{1, \ldots, k\}$.

In other words we do not allow nested terms. This is no loss of generality since nested terms can be *flattened*. To see this, note that $\psi(t)$ is logically equivalent to both $\forall z(t = z \to \psi(z))$ and $\exists z(t = z \wedge \psi(z))$. By repeatedly replacing $\psi(t)$ by one of these formulas, we flatten the formula. If we apply this transformation to an already flattened clause we end up with all the atoms of the form $F(\bar{x}) = y$ occurring negatively. If all terms in ϕ are flattened, we call it a *flattened formula*. If in addition atoms of the form $F(\bar{x}) = y$ occur only negatively, we call it a *flattened function-negative formula*. We want our formulas to be conjunctions of formulas of the form

$$\forall x_1 \cdots \forall x_k \exists y_1 \cdots \exists y_\ell (C) \tag{2.1}$$

where C is a clause with variables within $x_1, \ldots, x_k$ and $y_1, \ldots, y_\ell$. We use the term *standardized universal-existential formula* for flattened function-negative formulas of the form (2.1). We use *standardized universal formula* when there are no existentially quantified variables.

Note finally that, by standard Skolemization, every first-order sentence can be brought into a conjunction of standardized universal sentences while preserving the satisfiability at each cardinality. Formally:

Fact 1. *For every first-order sentence ϕ there is a conjunction of standardized universal sentences ϕ' that has the same spectrum; that is, for every finite or infinite cardinal κ the sentence ϕ has a model of cardinality κ if and only if ϕ' does.*

Propositional encodings Let ϕ be a flattened sentence and let $n \geq 1$ be a natural number. We will define two propositional formulas $\langle \phi \rangle_n^{\mathrm{u}}$ and $\langle \phi \rangle_n^{\mathrm{b}}$. In both cases the satisfying assigments of the propositional formula will be in one-to-one correspondence to the models of ϕ with universe $[n]$. We start with $\langle \phi \rangle_n^{\mathrm{u}}$ (the u stands for *unary*). The variables are the following:

1. $R_{\bar{a}}$ for each $R \in \sigma_R$ and each $\bar{a} \in [n]^{r_R}$,

2. $F_{\bar{a};a}$ for each $F \in \sigma_F$, each $\bar{a} \in [n]^{r_F}$, and $a \in [n]$.

These variables correspond in an obvious way to the ground first-order atoms of ϕ through the translation $\langle R(\bar{a})\rangle_n := R_{\bar{a}}$ and $\langle F(\bar{a}) = a\rangle_n := F_{\bar{a};a}$. First-order equality atoms $a = b$ are translated by their truth values.

Once the translation is defined for atoms it extends to arbitrary formulas through the usual recurrence:

1. $\langle \neg\psi\rangle_n := \neg\langle\psi\rangle_n$,
2. $\langle \psi \wedge \theta\rangle_n := \langle\psi\rangle_n \wedge \langle\theta\rangle_n$,
3. $\langle \psi \vee \theta\rangle_n := \langle\psi\rangle_n \vee \langle\theta\rangle_n$,
4. $\langle \forall x\psi\rangle_n := \bigwedge_{a\in[n]}\langle\psi[x/a]\rangle_n$,
5. $\langle \exists x\psi\rangle_n := \bigvee_{a\in[n]}\langle\psi[x/a]\rangle_n$.

Now, the translation $\langle\phi\rangle_n^{\mathrm{u}}$ is the conjunction of:

1. $\langle\phi\rangle_n$,
2. $F_{\bar{a};1} \vee \cdots \vee F_{\bar{a};n}$ for each $F \in \sigma_F$ and each $\bar{a} \in [n]^{r_F}$,
3. $\overline{F_{\bar{a};b}} \vee \overline{F_{\bar{a};c}}$ for each $F \in \sigma_F$, and each $\bar{a} \in [n]^{r_F}$ and $b, c \in [n]$ with $b \neq c$.

In the particular case that ϕ is a conjunction of standardized sentences, $\phi_1 \wedge \ldots \wedge \phi_m$, where $\phi_i = \forall\bar{x}_i\exists\bar{y}_i(C_i)$, the translation $\langle\phi\rangle_n^{\mathrm{u}}$ particularizes to a CNF-formula with the following clauses:

1. $\bigvee_{\bar{b}\in[n]^\ell}\langle C_i[\bar{x}/\bar{a}, \bar{y}/\bar{b}]\rangle_n$ for each $i \in \{1, \ldots, m\}$ and each $\bar{a} \in [n]^k$,
2. $F_{\bar{a};1} \vee \cdots \vee F_{\bar{a};n}$ for each $F \in \sigma_F$ and each $\bar{a} \in [n]^{r_F}$,
3. $\overline{F_{\bar{a};b}} \vee \overline{F_{\bar{a};c}}$ for each $F \in \sigma_F$, and each $\bar{a} \in [n]^{r_F}$ and $b, c \in [n]$ with $b \neq c$.

Clauses of type 1 are called *matrix clauses*, clauses of type 2 are called *long functional clauses*, and those of type 3 are called *short functional clauses*. Note that the size and the number of variables of $\langle\phi\rangle_n^{\mathrm{u}}$ are bounded by a polynomial in n that depends only on ϕ.

Next we define $\langle\phi\rangle_n^{\mathrm{b}}$ (the b stands for *binary*). The variables are:

1. $R_{\bar{a}}$ for each $R \in \sigma_R$ and each $\bar{a} \in [n]^{r_R}$,
2. $F_{\bar{a};b}$ for each $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$ and $0 \leq b \leq |n| - 1$.

The $R_{\bar{a}}$ still have an obvious correspondence with the relational ground atoms. However, the intended meaning of $F_{\bar{a};b}$ is different. Its meaning is that the b-th less significant bit in the binary encoding of $F(\bar{a})$ is 1. Thus, in this case the base cases of the translation are $\langle R(\bar{a})\rangle'_n := R_{\bar{a}}$ and

$$\langle F(\bar{a}) = a\rangle'_n := \bigwedge_{b=0}^{|n|-1} F_{\bar{a};b}^{(\mathrm{bit}(b,a))}.$$

The translation $\langle\phi\rangle_n^{\mathrm{b}}$ is the conjunction of:

1. $\langle\phi\rangle'_n$,
2. $\bigvee_{b=0}^{|n|-1} F_{\bar{a};b}^{(1-\mathrm{bit}(b,a))}$ for each $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$ and $a \in [2^{|n|}] \setminus [n]$.

The second type of clauses forbid elements outside $[n]$ from being in the range of F. Note that the long and short functional clauses for F are morally implicit.

In the particular case that ϕ is a conjunction of standardized sentences, $\phi_1 \wedge \ldots \wedge \phi_m$, where $\phi_i = \forall\bar{x}_i\exists\bar{y}_i(C_i)$, the translation $\langle\phi\rangle_n^{\mathrm{b}}$ particularizes to a CNF-formula with the following clauses:

1. $\bigvee_{\bar{b}\in[n]^\ell} \langle C_i[\bar{x}/\bar{a}, \bar{y}/\bar{b}]\rangle'_n$ for each $i \in \{1, \ldots, m\}$ and each $\bar{a} \in [n]^k$,
2. $\bigvee_{b=0}^{|n|-1} F_{\bar{a};b}^{(1-\mathrm{bit}(b,a))}$ for each $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$ and $a \in [2^{|n|}] \setminus [n]$.

Note that these are clauses since if C is a ground clause in which all function atoms $F(\bar{x}) = y$ appear negatively, the translation $\langle C\rangle'_n$ is still a clause, since we identify $\neg\bigwedge_i F_i$ with $\bigvee_i \neg F_i$.

2.5 Restrictions and decision trees

A *restriction* ρ is a partial assignment, i.e. a function mapping some atoms into $\{0, 1\}$. For a formula F we let $F \restriction \rho$ denote the formula obtained from F by first replacing every atom in the domain of ρ by its value under ρ and then eliminating constants: repeatedly replace subformulas $G \vee 1$ by 1 and $G \wedge 1$ by G; similarly for 0. Note that if the assignment ρ satisfies a literal in clause C, then $C \restriction \rho = 1$. If ρ falsifies a literal in a term T, then $T \restriction \rho = 0$.

A *decision tree* is a finite, rooted, ordered tree whose inner vertices are labeled by atoms, whose leafs are labeled by 0 or 1, and such that no atom occurs twice in a branch (i.e. a path from the root to some leaf). Each inner vertex has two successors (i.e. immediate successors on a branch). Since the tree is ordered we can distinguish between a *left* and a *right* successor of an inner vertex. By a *0-branch* (*1-branch*) we mean a branch leading to

a leaf labeled 0 (labeled 1). Every path π from the root to some vertex corresponds to the following restriction that we also denote by π: if an atom occurs as a label of a vertex p in the path π, then the restriction sets this atom to 0 if the left successor of p is in π and to 1 if the right successor of p is in π; if π contains no successor of p, then the restriction does not evaluate the atom.

A decision tree T *(strongly) represents* F if $F \restriction \pi \equiv b$ (resp. $F \restriction \pi = b$) for every $b \in \{0,1\}$ and every b-branch π of T. Here, $\equiv$ denotes logical equivalence of formulas. We say F *evaluates to* b *under* π if $F \restriction \pi \equiv b$. Note that $x \vee \neg x \restriction \emptyset \equiv 1$ but $x \vee \neg x \restriction \emptyset \neq 1$. Also, observe that if T represents F and $F \equiv G$, then T also represents G. The minimal height of a decision tree that represents F is denoted $h(F)$.

A decision tree T *represents* a formula F if $F \restriction \pi \equiv b$ for every $b \in \{0,1\}$ and every b-branch π of T. Here, $\equiv$ denotes logical equivalence of formulas. Observe that if T represents F and $F \equiv G$, then T also represents G. The minimal height of a decision tree that represents F is denoted $h(F)$.

Remark 1. The more common definition of representation is stronger than the notion used here in that one demands $F \restriction \pi = b$ for every b-branch π. The choice of the notion of representation is a subtle point; our use of it in the arguments relies on the choice we did, while e.g. some arguments in [50] rely on the stronger notion. ◇

The following lemma is easy to verify.

Lemma 1. *Let F and G be formulas and let T_F and T_G be decision trees of height s_F and s_G that represent F and G, respectively. Then there exists a decision tree T of height at most $s_F + s_G$ that represents $(F \wedge G)$ and such that every 0-branch of T extends some 0-branch of T_F or some 0-branch of T_G.*

Of course, saying that a 0-branch of T extends some 0-branch of T_F means that this holds for the corresponding restrictions.

2.6 Auxiliary lemmas

This section contains some auxiliary lemmas that will be used throughout the thesis.

2.6.1 Completeness, tree-likeness and deduction

The first proof is the following quantitative version of completeness:

Lemma 2. *Let s and n be naturals such that $s \geq n \geq 1$ and let $\Gamma \cup \{F\}$ be a set of propositional formulas each of size at most s and mentioning n variables in total. If $\Gamma \models F$, then F has a proof from Γ of size at most $27 \cdot s^2 \cdot 2^n$. Moreover, the proof is a k-DNF proof if each formula in $\Gamma \cup \{F\}$ is a k-DNF.*

Proof. Fix a set of n variables. For an assignment α to these variables let C_α be the disjunction of all literals falsified by α. Let G be a formula in the fixed variables and α an assignment. We claim that there is a cut-free proof of $G \vee C_\alpha$ or $\neg G \vee C_\alpha$ depending on whether $\alpha \models G$ or not, with at most $4|G|$ inferences. This can be verified by a straightforward induction on G: e.g. assume that $G = H_1 \wedge H_2$ and that the claim holds for H_1 and H_2. If $\alpha \not\models G$ choose $i \in \{1, 2\}$ such that $\alpha \not\models H_i$. Then there is a proof as desired for $\neg H_i \vee C_\alpha$; weakening and a structural inference gives $\neg G \vee C_\alpha$. If otherwise $\alpha \models G$, then there are proofs as desired of $H_1 \vee C_\alpha$ and $H_2 \vee C_\alpha$. From these derive $\neg G \vee C_\alpha$ in 4 steps by an application of (IOC) and structural inferences on premisses and conclusion to bring the formulas into the right form. The case $G = H_1 \vee H_2$ is dual.

Now assume $\Gamma \models F$. For all assignments α that satisfy all $G \in \Gamma$, and hence F, prove $F \vee C_\alpha$. For all assignments α that falsify some $G \in \Gamma$, prove $F \vee C_\alpha$ from Γ by deriving $\neg G \vee C_\alpha$, cutting on G after a structural inference, and finally adding F by (WKG) and a structural inference. In both cases these are at most $4(s + 1)$ many inferences. Take a resolution refutation of the set of clauses C_α, where α ranges over all assignments, with $\sum_{i=0}^{n-1} 2^i < 2^n$ applications of (CUT). Adding F to all formulas occurring in this refutation gives a proof of F from the already derived $F \vee C_\alpha$.

To make it a k-DNF-proof when all formulas in $\Gamma \cup \{F\}$ are k-DNFs argue as follows. In the preceeding paragraph, instead of deriving $\neg G \vee C_\alpha$, derive $C \vee C_\alpha$ in $4|C|$ steps for every clause C of $\neg G$. Let m be the number of such clauses. If $m = 1$ we continue as before. Assume then that $m > 1$. With a structural inference, write G associated to the left, and successively cut all terms with the $C \vee C_\alpha$ (brought into the right form with a structural inference). These are $2m + 1$ inferences to get C_α, hence $2m + 3 \leq 4m$ (as $m > 1$) inferences to get $F \vee C_\alpha$. So this formula is derived with at most $\sum_C 4|C| + 4m = 4(|G| + 1)$ inferences where C ranges over the clauses of $\neg G$.

In total, the proof has at most $2^n \cdot 4(s + 1) + 2^n$ many inferences, and all formulas have size at most $3s$: note $|C_\alpha| = n + (n - 1)$ and $|G \vee C_\alpha| \leq s + 2n$. This implies the lemma. □

The following clarifies the relationship between the tree-like and the dag-like versions. It goes back to [33] and appears in the form stated here in [26, Theorem 16].

Theorem 1. *Let Γ be a set of clauses. If $\Gamma \vdash^{s}_{k,*} \square$, then $\Gamma \vdash^{2s}_{1} \square$.*

The next lemma states the Deduction Theorem for R(k):

Lemma 3. *Let $\Gamma \cup \{F\}$ be a set of k-DNFs and let $C_1, \ldots, C_n$ be clauses with at most k literals each. If $\Gamma, C_1, \ldots, C_n \vdash^{s}_{k} F$, then $\Gamma \vdash^{s'}_{k} \neg C_1 \vee \cdots \vee \neg C_n \vee F$ for $s' = O(s \cdot \sum_{i=1}^{n} |C_i|)$.*

Proof. Let $\Gamma = \{F_1, \ldots, F_m\}$, assume $\Gamma, C_1, \ldots, C_n \vdash^{s}_{k} F$, and let Π be the proof witnessing it. Let $H := \neg C_1 \vee \cdots \vee \neg C_n$. Add H to every formula in the proof to get a proof of $H \vee F$ from axioms $F_i \vee H$ and $C_i \vee H$. Observe that $F_i \vee H$ can be obtained from F_i by weakening and $H \vee C_i$ is a weakening of an axiom $C_i \vee \neg C_i$. $\square$

2.6.2 Distributivity lemmas

In this section we show that we can replace small formulas for variables in proofs and distribute out to obtain either CNFs or DNFs as lines of the proof.

Let D be the distributivity operator on propositional formulas that recursively distributes conjunctions over disjunctions. In other words, the operator converts an arbitrary propositional formula into a DNF by the naive method. The operator is defined inductively by cases on the outermost connective of the formula. Formally:

1. If A is a literal or a conjunction of literals, then $D(A) := A$,
2. If $A = \bigvee_{i=1}^{r} A_i$ where each A_i is not a disjunction, then $D(A) := \bigvee_{i=1}^{r} D(A_i)$,
3. If $A = \bigwedge_{i=1}^{r} \bigvee_{j=1}^{s_i} A_{i,j}$ where each $A_{i,j}$ is not a disjunction, then

$$D(A) := \bigvee_{j_1=1}^{s_1} \cdots \bigvee_{j_r=1}^{s_r} D\left(\bigwedge_{i=1}^{r} A_{i,j_i}\right).$$

Lemma 4 (Distributivity)**.** *Let $\Gamma \cup \{F\}$ be a set of k-DNFs. For every variable x, let G_x and H_x be equivalent t-term-c-DNF and t-clause-c-CNF formulas, respectively, with v variables. For every formula A let A' be the result of replacing each positive occurrence of a variable x by G_x and each negative occurrence of a variable x by H_x. If $\Gamma \vdash^{s}_{k} F$, then $D(\Gamma') \vdash^{s'}_{k'} D(F')$ where $s' = O(s(kct^k)^2\, 2^{kv})$ and $k' = kc$.*

Proof. We may assume that the given proof applies AXM only to atoms since every axiom $A \vee \overline{A}$ has a short derivation from its underlying atom-axioms. It suffices to show that,

whenever C is derived from A and B by a single R(k)-rule, there is a short R(k')-proof of $D(C')$ from $D(A')$ and $D(B')$. We distinguish by cases on the type of rule.

(AXM): We want a proof of $D(x' \vee \overline{x}')$, which is $G_x \vee \overline{H_x}$. Since G_x and H_x are equivalent, this is a tautology, and by completeness it has a proof. Since it is a c-DNF of size at most ct on at most v variables, by Lemma 2 the proof is in R(c) and has size at most $O((ct)^2\ 2^v)$ times larger than the size of $x \vee \overline{x}$.

(WKG): Suppose $A \vee B$ is derived from A by weakening. From $D(A')$ we derive $D(A') \vee D(B') = D(A' \vee B')$ in one weakening step. The derived formula is a kc-DNF and its size is at most kct^k times larger than the size of $A \vee B$.

(CUT): Suppose $A \vee B$ is derived through the cut rule on $A \vee T$ and $B \vee \overline{T}$, where T is a term of at most k literals.We want to obtain $D(A' \vee B')$ from $D(A' \vee T')$ and $D(A' \vee \overline{T}')$. To that end it suffices to derive the empty clause from $D(T')$ and $D(\overline{T}')$, which are contradictory since G_x and H_x are equivalent. By completeness, such a refutation exists. Since both are kc-DNFs of size at most kct^k on at most kv variables, by Lemma 2 the refutation is in R(kc) and has size $O((kct^k)^2\ 2^{kv})$. Overall the corresponding proof of $D(A' \vee B')$ is also this many times larger than the size of $A \vee B$.

(IOC): Suppose $A \vee B \vee (S \wedge T)$ is derived by introduction of conjunction from $A \vee S$ and $B \vee T$, where $S \wedge T$ has at most k literals. We want to derive $D(A' \vee B' \vee (S \wedge T)')$ from $D(A' \vee S')$ and $D(B' \vee T')$. To that end, it suffices to derive $D((S \wedge T)')$ from $D(S')$ and $D(T')$, which can be done by completeness. These formulas are kc-DNFs of size at most kct^k on at most kv variables. Therefore, by Lemma 2, the proof is in R(kc) and has size $O((kct^k)^2\ 2^{kv})$. Overall the corresponding proof of $D(A' \vee B' \vee (S \wedge T)')$ is also this many times larger than the size of $A \vee B \vee (S \wedge T)$.

To complete the proof, note that each simulation step is a proof in R(kc) and that its size is at most $O((kct^k)^2\ 2^{kv})$ times larger than the corresponding formula in the original proof. □

Similarly, we introduce a distributivity operator C, which distributes disjunctions over conjunctions. The operator defined inductively by cases on the outermost connective of the formula. Formally:

1. If A is a literal or a disjunction of literals, then $C(A) := A$,
2. If $A = \bigwedge_{i=1}^{r} A_i$ where each A_i is not a conjunction, then $C(A) := \bigwedge_{i=1}^{r} C(A_i)$,

3. If $A = \bigvee_{i=1}^{r} \bigwedge_{j=1}^{s_i} A_{i,j}$ where each $A_{i,j}$ is not a conjunction, then

$$C(A) := \bigwedge_{j_1=1}^{s_1} \cdots \bigwedge_{j_r=1}^{s_r} C\left(\bigvee_{i=1}^{r} A_{i,j_i}\right).$$

Note that, in 3, if there exists an integer s such that $s_i = s$ for every $i \in [r]$, then C(A) can also be written as

$$C(A) := \bigwedge_{f \in \mathcal{F}} C\left(\bigvee_{i=1}^{r} A_{i,f(i)}\right).$$

where $\mathcal{F} = \{f \mid f : [r] \to [s]\}$. Depending on the context, we identify $C(A)$ either as a conjunction of clauses or a set of clauses.

We show that, given a resolution-refutation, we can substitute a variable by a depth-2 formula, and still obtain a refutation of the substituted formula thanks to the C-operator.

Lemma 5 (C-Distributivity)**.** *Let Γ be a set of clauses. For every variable x, let G_x and H_x be equivalent t-term-c-DNF and t-clause-c-CNF formulas respectively, both with v variables. For every formula A let A' be the result of replacing each positive occurrence of a variable x by G_x and each negative occurrence of a variable x by H_x. If $\Gamma \vdash_1^{s,w} \square$, then $C(\Gamma') \vdash_1^{s'} \square$ where $s' = O(s2^v wtc^{tw})$.*

Proof. Note that we can assume that the given refutation $A_1, A_2, \ldots, \square$ uses only cuts. For this proof only, let a resolution-proof of $C(A'_i)$ from $C(\Gamma')$ be a sequence of clauses, such that every clause is either in $C(\Gamma')$ or is the result of applying a resolution-rule to the previous clauses, and such that all clauses in $C(A'_i)$ occur in the sequence.

Now, we prove by induction on i that there is a resolution-proof of $C(A'_1, \ldots, C(A'_i)$ from $C(\Gamma')$. If A_i is a clause of Γ, then $C(A'_i) \subseteq C(\Gamma')$ and we are done. Otherwise, A_i is obtained by a cut on A_j and A_k with $1 \leq j, k < i$. Let $A_j = A \vee x$ and let $A_k = B \vee \overline{x}$, so that x is the cut variable. Then, by induction hypothesis, the clauses of $C((A \vee x)')$ and $C((B \vee \overline{x})')$ are already in the sequence, and we want to obtain a proof of the clauses of $C((A \vee B)')$. Before showing how to do so, it will be useful to prove the following claim.

Claim 1. *Let F, G be clauses. Then, $C((F \vee G)') = C(C(F') \vee C(G'))$.*

Proof. Note that $F' \vee G'$ is of the form

$$F' \vee G' = \bigvee_{i \in [w_F]} \bigvee_{j \in [t]} \bigwedge_{k \in [c]} \ell_{i,j,k} \vee \bigvee_{i' \in [w_G]} \bigvee_{j' \in [t]} \bigwedge_{k' \in [c]} \ell_{i',j',k'}$$

where w_F and w_G is the width of clauses F and G respectively. Then,

$$C(F' \vee G') = \bigwedge_{f_F \in \mathcal{F}_F} \bigwedge_{f_G \in \mathcal{F}_G} \left(\bigvee_{i' \in [w_G]} \bigvee_{j' \in [t]} \ell_{i,j,f_F(i,j)} \vee \bigvee_{i \in [w_F]} \bigvee_{j \in [t]} \ell_{i',j',f_G(i',j')} \right)$$

where $\mathcal{F}_F = \{f \mid f : [w_F] \times [t] \to [c]\}$ and the same for $\mathcal{F}_G$. Also, note that

$$C(F') = \bigwedge_{f_F \in \mathcal{F}_F} \bigvee_{i \in [w_F]} \bigvee_{j \in [t]} \ell_{i,j,f_F(i,j)}$$

and the same for $C(G')$. Therefore,

$$\begin{aligned}
C((F \vee G)') &= C(F' \vee G') \\
&= \bigwedge_{f_F \in \mathcal{F}_F} \bigwedge_{f_G \in \mathcal{F}_G} \left(\bigvee_{i' \in [w_G]} \bigvee_{j' \in [t]} \ell_{i,j,f_F(i,j)} \vee \bigvee_{i \in [w_F]} \bigvee_{j \in [t]} \ell_{i',j',f_G(i',j')} \right) \\
&= C\left(\left(\bigwedge_{f_F \in \mathcal{F}_F} \bigvee_{i \in [w_F]} \bigvee_{j \in [t]} \ell_{i,j,f_F(i,j)} \right) \vee \left(\bigwedge_{f_G \in \mathcal{F}_G} \bigvee_{i' \in [w_G]} \bigvee_{j' \in [t]} \ell_{i',j',f_G(i',j')} \right) \right) \\
&= C\left(C(F') \vee C(G')\right)
\end{aligned}$$

□

Now, we show how to obtain the clauses of $C((A \vee B)')$ from the clauses of $C((A \vee x)')$ and the clauses of $C((B \vee \overline{x})')$. By Claim 1, this is the same as obtaining the clauses of $C(C(A') \vee C(B'))$ from the clauses of $C(C(A') \vee C(x'))$ and the clauses $C(C(B') \vee C(\overline{x}'))$.

Let D be an arbitrary clause of $C(C(A') \vee C(B'))$. Note that D is of the form $D_A \vee D_B$ where D_A is a clause of $C(A')$ and D_B is a clause of $C(B')$. We show that D can be obtained from the clauses of $C(D_A \vee C(x'))$ and the clauses of $C(D_B \vee C(\overline{x}'))$, which occur in the sequence since $C(D_A \vee C(x')) \subseteq C(C(A') \vee C(x'))$ and $C(D_B \vee C(\overline{x}')) \subseteq C(C(B') \vee C(\overline{x}'))$. Each clause of $C(D_A \vee C(x'))$ is a disjunction of a clause of $C(x')$ with D_A and the same for D_B and $C(\overline{x}')$. Therefore, it is enough to show that the empty clause can be derived from $C(x')$ and $C(\overline{x}')$.

Since $C(x')$ and $C(\overline{x}')$ are contradictory, by completeness of resolution, there is a resolution refutation of size $O(2^v)$. This size is $O(2^v wt)$ when we add a disjunction with D_A and D_B appropriately to the initial clauses of this refutation. The size of the whole step will be the size of the proof of each clause D times the number of clauses we need to obtain, this is

$O(2^v wtc^{tw})$. Finally, the produced refutation will have this size for every step of the original one, this is $O(s2^v wtc^{tw})$. □

2.6.3 Proof translations

Applying the D-distributivity lemma, we show how to translate refutations of the unary encoding to the binary encoding. To do so, we need the following claim.

Claim 2. *Let $x_1, \dots x_n$ be variables. The formula*

$$\phi(n) := \bigvee_{S \in \mathcal{P}([n])} \left(\bigwedge_{i \in S} x_i \wedge \bigwedge_{j \in [n] \setminus S} x_j \right) \tag{2.2}$$

has a tree-like $\mathrm{R}(n)$ *proof of size polynomial in* $n \cdot 2^n$.

Proof. We prove the claim by induction on n. Note that $\phi(1)$ is exactly the axiom $x_1 \vee \overline{x_1}$. If $n > 1$, by induction hypothesis we have a tree-like $\mathrm{R}(n-1)$ proof of the $(n-1)$-DNF formula $\phi(n-1)$ and we want to obtain a $\mathrm{R}(n)$ proof of the n-DNF formula $\phi(n)$. The terms of $\phi(n)$ are precisely the terms of $\phi(n-1)$ conjuncted with either x_n or $\overline{x_n}$. Let $\phi_0(n)$ be the n-DNF with the terms of $\phi(n)$ that contain $\overline{x_n}$ and let $\phi_1(n)$ be the one with the terms of $\phi(n)$ that contain x_n. Note that $\phi(n) = \phi_0(n) \vee \phi_1(n)$. Then, the proof of $\phi(n)$ is as follows: first, repeatedly apply introduction of conjunction to $\phi(n-1)$ and the axiom $x_n \vee \overline{x_n}$ on $\overline{x_n}$ to obtain $\phi_0(n) \vee x_n$. Second, do the same with a different copy of $\phi(n-1)$, this time on x_n to obtain $\phi_1(n) \vee \overline{x_n}$. Finally, cut those two on x_n to obtain $\phi_0(n) \vee \phi_1(n) = \phi(n)$.

The obtained proof is obviously $\mathrm{R}(n)$ and it is also tree-like, since we used different copies of $\phi(n-1)$ for each of its occurrences. We show now that the size is polynomial in $n \cdot 2^n$. Let s_n be the size of the proof of $\phi(n)$. At every step, it uses two copies of the proof of $\phi(n-1)$. It also uses 2^{n-1} axioms and 2^{n-1} intermediate steps to obtain each of $\phi_0(n) \vee x_n$ and $\phi_1(n) \vee \overline{x_n}$ and finally performs a cut. This is:

$$s_n = 2 \cdot s_{n-1} + 4 \cdot 2^{n-1} + 1 = 2^{n-1} + (n-1) \cdot 2^{n+1} + n - 1,$$

this is, size polynomial in $n \cdot 2^n$. □

Now we are ready to prove the unary-to-binary translation lemma.

Lemma 6. *Let ϕ be a standardized universal formula. For every n, if $\langle \phi \rangle_n^{\mathrm{u}} \vdash_{1,*}^{s} \square$, then $\langle \phi \rangle_n^{\mathrm{b}} \vdash_{\log n,*}^{s'} \square$ for s' polynomial in s and n.*

Proof. For a formula G let G' be obtained by replacing every atom $F_{\bar{a};a}$ by $\bigwedge_{b=0}^{|n|-1} F_{\bar{a};b}^{(\mathrm{bit}(b,a))}$. It suffices to show how to prove from the clauses of $\langle\phi\rangle_n^{\mathrm{b}}$, the $|n|$-DNF's of the primed axioms of $\langle\phi\rangle_n^{\mathrm{u}}$. The matrix clauses of $\langle\phi\rangle_n^{\mathrm{b}}$ are exactly the primed matrix clauses of $\langle\phi\rangle_n^{\mathrm{u}}$. We show how to derive the primed functional clauses of $\langle\phi\rangle_n^{\mathrm{u}}$.

The primed long functional clauses of $\langle\phi\rangle_n^{\mathrm{u}}$ read $\bigvee_{a\in[n]}\bigwedge_{b=0}^{|n|-1} F_{\bar{a};b}^{(\mathrm{bit}(b,a))}$ for $F\in\sigma$ and $\bar{a}\in[n]^{r_F}$. If n is a power of 2, note that this has the form of (2.2) with $\log n$ variables. Therefore, by Claim 2, there is a tree-like R($\log n$)-proof of it of size polynomial in n. If n is not a power of 2, then first give a proof of the formula for $2^{|n|}$ and then obtain the formula for n by cuts with the axioms $\bigvee_{b=0}^{|n|-1} F_{\bar{a};b}^{1-\mathrm{bit}(b,a)}$ for $a\in[2^{|n|}]\setminus[n]$. Note that this proof is also tree-like R(log).

Primed short functional clauses of $\langle\phi\rangle_n^{\mathrm{u}}$ read $\bigvee_{b=0}^{|n|-1}\left(F_{\bar{a};b}^{(1-\mathrm{bit}(b,a))}\vee F_{\bar{a};b}^{(1-\mathrm{bit}(b,a'))}\right)$ for $F\in\sigma$, $\bar{a}\in[n]^{r_F}$ and $a,a'\in[n]$ with $a\neq a'$. But such a formula is a weakening of an axiom since the binary representations of a and a' differ in some bit. □

Remark 2. Note that the proof of the substitution instances of the long functional clauses is tree-like R(log). ◇

It will turn out to be useful to have also the opposite result, a translation from refutations of the binary encoding to refutations of the unary encoding. We show the following:

Lemma 7. *Let $k\geq 1$ and ϕ be a standardized universal formula. For every n, if $\langle\phi\rangle_n^{\mathrm{b}}\vdash_k^s\square$, then $\langle\phi\rangle_n^{\mathrm{u}}\vdash_k^{s'}\square$ for s' polynomial in s and n.*

Proof. We show how to transform a refutation of $\langle\phi\rangle_n^{\mathrm{b}}$ into a refutation of $\langle\phi\rangle_n^{\mathrm{u}}$. We define an operator Q on terms. Consider a term T of the form

$$T=\bigwedge_{(F,\bar{a})\in D}\left(\bigwedge_{b\in B^0(F,\bar{a})}\overline{F_{\bar{a},b}}\wedge\bigwedge_{b\in B^1(F,\bar{a})}F_{\bar{a},b}\right)\wedge R \tag{2.3}$$

where $D\subseteq\{(F,\bar{a})\mid F\in\sigma_F\wedge\bar{a}\in[n]^{r_F}\}$ and $B^0(F,\bar{a}),B^1(F,\bar{a})$ are disjoint subsets of $[n]$ for every $(F,\bar{a})\in D$ and R is a term only containing relational atoms. Then we define

$$Q(T):=\bigvee_{g\in G}\bigwedge_{(F,\bar{a})\in D}F_{\bar{a},g(F,\bar{a})}\wedge R. \tag{2.4}$$

where $G=\{g:D\to[n]\mid\forall(F,\bar{a})\in D\ \forall x\in\{0,1\}\ \forall b\in B^x(F,\bar{a})\ \mathrm{bit}(b,g(F,\bar{a}))=x\}$. To extend the operator Q to DNF formulas, let ψ be a DNF of the form $\bigvee_{i\in I}T_i$ where each T_i

is a term. We set $Q(\psi) := \bigvee_{i\in I} Q(T_i)$. We claim that applying Q on each of the lines of the refutation of $\langle\phi\rangle_n^{\mathrm{b}}$ we can obtain a refutation of $\langle\phi\rangle_n^{\mathrm{u}}$ by adding of some intermediate steps.

First, let C be a matrix-clause of $\langle\phi\rangle_n^{\mathrm{b}}$. Note that C has the form

$$C = \bigvee_{(F,\bar{a})\in D} \bigvee_{a\in A(F,\bar{a})} \bigvee_{b=0}^{|n|-1} F_{\bar{a},b}^{(1-\mathrm{bit}(b,a))} \vee R,$$

where $D \subseteq \{(F,\bar{a}) | F \in \sigma_F \wedge \bar{a} \in [n]^{r_F}\}$, $A(F,\bar{a}) \subseteq [n]$ and R is a clause on relational atoms. Now,

$$Q(C) = \bigvee_{(F,\bar{a})\in D} \bigvee_{a\in A(F,\bar{a})} \bigvee_{\substack{c\in[n]\\ c\neq a}} F_{\bar{a},c} \vee R.$$

On the other hand, the corresponding clause in the unary encoding is of the form

$$\bigvee_{(F,\bar{a})\in D} \bigvee_{a\in A(F,\bar{a})} \overline{F_{\bar{a},a}} \vee R. \tag{2.5}$$

To obtain $Q(C)$ from the clauses of $\langle\phi\rangle_n^{\mathrm{u}}$, we just cut (2.5) with the corresponding long functional clauses of the unary encoding. Second, consider the clauses of $\langle\phi\rangle_n^{\mathrm{b}}$ of the form

$$\bigvee_{b=0}^{|n|-1} F_{\bar{a},b}^{(1-\mathrm{bit}(b,c))}$$

with $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$ and $c > n$. The Q-transformation of these is

$$\bigvee_{\substack{a\in[n]\\ a\neq c}} F_{\bar{a},a}$$

which is exactly the long functional clause of $\langle\phi\rangle_n^{\mathrm{u}}$ for F and $\bar{a}$, since $c > n$.

Next, let us look at the inference steps of the refutation of $\langle\phi\rangle_n^{\mathrm{b}}$ one at a time. As usual, we may assume that axioms are applied over atoms only. Axioms of the type $F_{\bar{a},b} \vee \overline{F_{\bar{a},b}}$, after the Q-transformation, become of the form $\bigvee_{a\in[n]} F_{\bar{a},a}$, that is, instances of the long functional clauses of $\langle\phi\rangle_n^{\mathrm{u}}$. Formulas obtained by weakening are identically obtained in the refutation of $\langle\phi\rangle_n^{\mathrm{u}}$, since $Q(A \vee B) = Q(A) \vee Q(B)$.

Suppose now that $A \vee B$ is obtained from a cut between $A \vee T$ and $B \vee \overline{T}$, where T is a term of at most k literals. It would suffice to derive $\Box$ from $Q(T)$ and $Q(\overline{T})$. We know T is

of the form (2.3) and therefore its negation will be of the form,

$$\overline{T} = \bigvee_{(F,\bar{a})\in D} \left(\bigvee_{b\in B^0(F,\bar{a})} F_{\bar{a},b} \vee \bigvee_{b\in B^1(F,\bar{a})} \overline{F_{\bar{a},b}} \right) \vee \overline{R}.$$

Then, its Q-transformation is

$$Q(\overline{T}) = \bigvee_{(F,\bar{a})\in D} \left(\bigvee_{b\in B^0(F,\bar{a})} \bigvee_{a\in N_b^1} F_{\bar{a},a} \vee \bigvee_{b\in B^1(F,\bar{a})} \bigvee_{a\in N_b^0} F_{\bar{a},a} \right) \vee \overline{R},$$

where $N_b^x = \{a \mid \mathrm{bit}(b,a) = x\}$. Note that this last formula is the same as

$$\bigvee_{(F,\bar{a})\in D} \bigvee_{a\in N(F,\bar{a})} F_{\bar{a},a} \vee \overline{R} \tag{2.6}$$

where $N(F,\bar{a}) = \{a \mid \exists x \in \{0,1\}\ \exists b \in B^x(F,\bar{a})\ \mathrm{bit}(b,a) = 1-x\}$.

Let $g \in G$. Then, $g(F,\bar{a}) \notin N(F,\bar{a})$ and therefore $\langle\phi\rangle_n^{\mathrm{u}}$ contains the short functional clause $\overline{F_{\bar{a},g(F,\bar{a})}} \vee \overline{F_{\bar{a},a}}$ for every $a \in N(F,\bar{a})$. Repeatedly cutting these clauses with (2.6) yields

$$\bigvee_{(F,\bar{a})\in D} \overline{F_{\bar{a},g(F,\bar{a})}} \vee \overline{R}.$$

We can cut these clauses with (2.4) to obtain the empty clause.

Suppose now that $A \vee B \vee (S \wedge T)$ is obtained from an introduction of conjunction between $A \vee S$ and $B \vee T$. In this case, it suffices to show that $Q(S \wedge T)$ can be obtained from $Q(S)$ and $Q(T)$. Recall the that both S and T are of the form (2.3) and that

$$Q(S) = \bigvee_{g\in G_S} \bigwedge_{(F,\bar{a})\in D_S} F_{\bar{a},g(F,\bar{a})} \wedge R_S, \tag{2.7}$$

where $D_S \subseteq \{(F,\bar{a}) \mid F \in \sigma_F \wedge \bar{a} \in [n]^{r_F}\}$ and $G_S = \{g : D_S \to [n] \mid \forall (F,\bar{a}) \in D_S\ \forall x \in \{0,1\}\ \forall b \in B_S^x(F,\bar{a})\ \mathrm{bit}(b, g(F,\bar{a})) = x\}$, and analogously for $Q(T)$. From this, we want to obtain

$$Q(S \wedge T) = \bigvee_{g\in G} \bigwedge_{(F,\bar{a})\in D} F_{\bar{a},g(F,\bar{a})} \wedge R, \tag{2.8}$$

where $G = \{g : D \to [n] \mid \forall (F, \bar{a}) \in D \; \forall x \in \{0,1\} \; \forall b \in B^x(F, \bar{a}) \; \mathrm{bit}(b, g(F, \bar{a})) = x\}$ with $D = D_S \cup D_T$ and $B^x(F, \bar{a}) = B^x_S(F, \bar{a}) \cup B^x_T(F, \bar{a})$ for every $(F, \bar{a}) \in D$, and $R = R_S \wedge R_T$. To obtain (2.8) we need some steps. First, we would like to obtain

$$\bigvee_{g \in G_S} \bigvee_{g' \in G_T} \left(\bigwedge_{(F,\bar{a}) \in D_S} F_{\bar{a}, g(F,\bar{a})} \wedge \bigwedge_{(F',\bar{a}') \in D_T} F'_{\bar{a}', g'(F',\bar{a}')} \wedge R_S \wedge R_T \right). \tag{2.9}$$

Note that $Q(S)$ and $Q(T)$ are of the form $\bigvee_{i \in I} S_i$ and $\bigvee_{j \in J} T_j$ respectively, where S_i and T_j are terms, and that we want to obtain $\bigvee_{i \in I} \bigvee_{j \in J} (S_i \wedge T_j)$. To do so, we proceed the following way: take $\bigvee_{i \in I} S_i$ and introduce conjunction repeatedly with the axiom $T_j \vee \overline{T_j}$ to obtain $\bigvee_{i \in I} (S_i \wedge T_j) \vee \overline{T_j}$ for every $j \in J$. Successive cuts of these with $\bigvee_{j \in J} T_j$ yield (2.9).

Note now that the terms of (2.8) are exactly the terms of (2.9) such that g and g' are consistent for every $(F, \bar{a}) \in D$. Therefore, to obtain (2.8) we only need to get rid of those terms that contain $F_{\bar{a}, g(F,\bar{a})}$ and $F_{\bar{a}, g'(F,\bar{a})}$ with $g(F, \bar{a}) \neq g'(F, \bar{a})$. But each of this terms is the negation of a weakening of a short functional clause $\overline{F_{\bar{a}, g(F,\bar{a})}} \vee \overline{F_{\bar{a}, g'(F,\bar{a})}}$. Cutting (2.9) with those gives us (2.8).

It is left to the reader to check that the refutation constructed is polynomial in the size of the original one and n, and that the terms of the constructed refutation are never larger than those of the original one. □

Chapter 3

Bounded tree-width QBFs and Q-resolution

In this chapter we show that the TQBF problem is PSPACE-complete even when restricted to input formulas whose constraint graph is of bounded tree-width. Later, we introduce the Q-resolution proof system and note lower bounds on this system implied by our result. Finally, we introduce the concept of bounded respectful tree-width as a particular case of bounded tree-width and show an upper bound in this case.

3.1 Tree-width and path-width

Let ϕ be a CNF-formula with variables $X_1, \ldots, X_n$ and clauses $C_1, \ldots, C_m$. The *constraint graph* of ϕ has one vertex for every variable of ϕ and two variables are connected by an edge if and only if there is a clause which contains them both. We identify the variables of a formula with the vertices of its constraint graph.

For a given a graph $G = (V, E)$, a *tree decomposition* of G is a pair (T, L), where T is a tree and L is a function $L : V(T) \to \mathcal{P}(V)$, that satisfies the following properties:

1. $\bigcup_{t \in V(T)} L(t) = V$,
2. for every $(u, v) \in E$, there is a $t \in V(T)$ such that $u, v \in L(t)$,
3. for every $v \in V$, the subgraph of T induced by $\{t \in V(T) \mid v \in L(t)\}$ is a connected subtree.

For later convenience we assume that T is a rooted tree. Note that a graph has multiple tree-decompositions.

Given a tree-decomposition, its *width* is defined as

$$\max_{t \in V(T)} L(t) - 1.$$

The *tree-width* of a graph is the minimum among the widths of its tree-decompositions.

The tree-width of a formula is defined as the tree-width of its constraint graph.

Claim 3. *Let G be a graph and let (T, L) be a tree-decomposition of G. Then, for every $S \subseteq V(G)$ that induces a clique, there is a $t \in V(T)$ such that $S \subseteq L(t)$.*

A *path decomposition* of a graph G is a tree-decomposition (T, L) such that T is a path. The *path-width* of a graph is the minimum among the widths of its path decompositions.

3.2 Quantified boolean formulas

A Quantified boolean formula (QBF) is a formula of the form

$$\phi = Q_1 x_1 \dots Q_q x_q (\phi') \tag{3.1}$$

where $x_1, \dots, x_q$ are propositional variables, the *matrix* ϕ' is a CNF-formula and Q_i is either $\forall$ or $\exists$ for every $i \in [q]$. The size of a QBF as in (3.1) is defined as the size of its matrix ϕ'. The tree-width (path-width) of a QBF is the tree-width (path-width) of its matrix. We say that $Q_1 x_1 \dots Q_q x_q$ is the *prefix* of ϕ.

3.3 Leveled formulas

In this section we state and prove the main lemma. This lemma is a reduction from n-leveled QBFs to $O(\log n)$-leveled QBFs, which is progress in our iterative argument. Before stating the lemma, we formalize the concept of leveled-QBF.

For the rest of the chapter, think of a clause as a *sequence* of literals, and of a CNF-formula as a sequence of clauses. Then, for a CNF-formula ϕ, we can write $\ell_1(\phi), \dots, \ell_s(\phi)$ for the s literals of ϕ in the left-to-right order in which they appear in ϕ. Note that this sequence can contain repeated literals. For example, in

$$\phi = ((x1, \overline{x_2}), (x_2, \overline{x_3}, x_4), (\overline{x_4}))$$

we have $\ell_4(\phi) = \overline{x_3}$. When ϕ is clear from the context we write ℓ_i instead of $\ell_i(\phi)$.

3.3.1 Definition of leveled QBF

Let n be a positive integer. An *n-leveled CNF-formula* is a CNF-formula ϕ in which its sequence of clauses is partitioned into *blocks* $B_1, \ldots, B_\ell$, where each block is a consecutive subsequence of clauses of ϕ, and its set of variables is partitioned into the same number of *groups* $G_1, \ldots, G_\ell$, each containing at most n variables, and such that for every $j \in \{1, \ldots, \ell - 1\}$ we have that every clause C in B_j has all its variables in $G_j \cup G_{j+1}$, and every clause C in B_ℓ has all its variables in G_ℓ. An *n-leveled QBF* is a quantified Boolean formula whose matrix is an n-leveled CNF-formula.

Observe that every QBF with n variables is an n-leveled QBF: put all clauses in a single block and all variables in a single group. However, when the sizes of the groups are limited, we get a nice structure:

Lemma 8. *Let n be a positive integer. Every n-leveled QBF has path-width at most $2n - 1$.*

Proof. Let ϕ be an n-leveled QBF with groups $G_1, \ldots, G_\ell$. Define (T, L) as the path decomposition of the matrix of ϕ where T is a path on vertices $t_1, \ldots, t_\ell$, and $L(t_i) := G_i \cup G_{i+1}$ for $i \in \{1, \ldots, \ell - 1\}$ and $L(t_\ell) := G_\ell$. Since each G_i has cardinality at most n, the claim follows. □

Now, we can formalize the statement of the main lemma.

Lemma 9. *There exist $c, d \geq 1$ and a polynomial-time algorithm that, for every $n, s \geq 1$, given an n-leveled QBF ϕ of size s, computes a $c \cdot |n|$-leveled QBF ψ of size $d \cdot s \cdot |n|$ such that $\phi \leftrightarrow \psi$.*

We devote the rest of the section to the proof of this lemma. In order to improve the readability of Boolean formulas, we use $+$ for disjunction and $\cdot$ for conjunction.

3.3.2 Definition of $\boldsymbol{\theta}$

Let ϕ be a n-leveled QBF as in (3.1) whose matrix ϕ' is an n-leveled CNF-formula of size s with groups $G_1, \ldots, G_\ell$ and blocks $B_1, \ldots, B_\ell$. As a first step towards building ψ we define an intermediate formula θ. The formula θ contains variables $\tau_1, \ldots, \tau_s$, one for each literal in ϕ', and is defined as

$$\theta := Q_1 \boldsymbol{\tau}_1 \cdots Q_q \boldsymbol{\tau}_q (\textsc{ncons}_\forall + (\textsc{cons}_\exists \cdot \textsc{sat}))$$

where

1. each $\boldsymbol{\tau}_j$, for $j \in \{1, \ldots, q\}$, is the tuple of τ-variables corresponding to all the occurrences of the variable x_j in ϕ',

2. CONS_Q, for $Q \in \{\forall, \exists\}$, is a QBF to be defined later that is satisfied by an assignment to $\tau_1, \ldots, \tau_s$ if and only if all the variables from the same $\boldsymbol{\tau}_j$ with $Q_j = Q$ are given the same truth value,

3. NCONS_Q for $Q \in \{\forall, \exists\}$ is a QBF that is equivalent to the negation of CONS_Q,

4. SAT is a QBF to be defined later that is satisfied by an assignment to $\tau_1, \ldots, \tau_s$ if and only if every clause of ϕ' contains at least one literal $\ell_k = x^{(a)}$ such that τ_k is given value a.

This information about the constituents of θ is enough to prove the following claim.

Claim 4. $\phi \leftrightarrow \theta$

Proof. We need to prove both implications. In both cases we use a game in which two players, the existential player and the universal player, take rounds following the order of quantification of the formula to choose values for the variables quantified their way. The aim of the existential player is to show that the matrix of the formula can be made true while the aim of the universal player is to show him wrong.

In the following, for $j \in \{1, \ldots, q\}$, we say that an assignment to the variables of $\boldsymbol{\tau}_j$ is *consistent* if they are given the same truth value, say $a \in \{0, 1\}$. In case the assignment is consistent, we say that a is the *corresponding assignment* for the variable x_j. Conversely, if a is an assignment to the variable x_j, the *corresponding consistent assignment* for the tuple $\boldsymbol{\tau}_j$ is the assignment that sets each variable in $\boldsymbol{\tau}_j$ to a. If an assignment to $\boldsymbol{\tau}_j$ is not consistent we call it inconsistent.

($\rightarrow$): Assume ϕ is true and let α be a winning strategy for the existential player in ϕ. We build another strategy β that guarantees him a win in θ. The construction of β will be based on the observation that, in the course of the game on θ, if the assignment given by the universal player to some $\boldsymbol{\tau}_j$ with $Q_j = \forall$ is inconsistent, then $\text{NCONS}_\forall$ is true irrespective of all other variables, and hence the matrix of θ is true. With this observation in hand, the strategy β is defined as follows: at round j with $Q_j = \exists$, if all $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{j-1}$ have been given consistent assignments up to this point and $a_1, \ldots, a_{j-1} \in \{0, 1\}$ are the corresponding assignments to the variables $x_1, \ldots, x_{j-1}$, let a_j be the assignment given to x_j by the strategy α in this position of the game on ϕ, and let the existential player assign value a_j to every

variable in $\boldsymbol{\tau}_j$. If on the other hand some $\boldsymbol{\tau}_k$ with $k < j$ has been given an inconsistent assignment, let the existential player assign an arbitrary value (say 0) to every variable in $\boldsymbol{\tau}_j$. Using the observation above and the assumption that α is a winning strategy, it is not hard to see that β is a winning strategy.

($\leftarrow$): Assume θ is true and let β be a winning strategy for the existential player in θ. We build a strategy α for the existential player in ϕ. In this case the construction of α will be based on the observation that, in the course of the game on θ, as long as the universal player assigns consistent values to every $\boldsymbol{\tau}_j$ with $Q_j = \forall$, the assignment given by β to each new $\boldsymbol{\tau}_j$ with $Q_j = \exists$ must be consistent. To see this note that, if not, the universal player would have the option of staying consistent all the way until the end of the game in which case both $\text{NCONS}_\forall$ and $\text{CONS}_\exists$ would become false, thus making the matrix of θ false. With this observation in hand, the strategy α is defined as follows: at round j with $Q_j = \exists$, let $a_1, \ldots, a_{j-1} \in \{0, 1\}$ be the assignment given to $x_1, \ldots, x_{j-1}$ up to this point, let $\mathbf{a}_1, \ldots, \mathbf{a}_{j-1}$ be the corresponding consistent assignments for $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{j-1}$, and let $\mathbf{a}_j$ be the assignment given by β to $\boldsymbol{\tau}_j$ in this position of the game on θ. By the observation above, since each $\mathbf{a}_k$ with $k < j$ and $Q_k = \forall$ is consistent by definition and each $\mathbf{a}_k$ with $k < j$ and $Q_j = \exists$ has been assigned according to the strategy β, the assignment $\mathbf{a}_j$ must also be consistent. Thus the existential player can set x_j to its corresponding value a_j and continue with the game.

We need to show that α is a winning strategy for the existential player on ϕ. First, if the existential player plays according to α, then the final assignment $a_1, \ldots, a_q$ that is reached in the game on ϕ is such that the corresponding assignment $\mathbf{a}_1, \ldots, \mathbf{a}_q$ in the game on ψ satisfies the matrix of θ. Since each $\mathbf{a}_j$ is consistent this means that SAT must be made true by $\mathbf{a}_1, \ldots, \mathbf{a}_q$, thus the matrix of ϕ is made true by $a_1, \ldots, a_q$. This shows that the existential player wins. □

Now, we show how to construct the QBF-formulas SAT, $\text{CONS}_\exists$ and $\text{NCONS}_\forall$. These formulas have the τ-variables as free variables and a new set of quantified variables for each literal in ϕ'. Recall that the τ-variables assign a truth value to each variable-ocurrence in ϕ'. The formula SAT will verify that these assignments satisfy all clauses of ϕ', the formula $\text{CONS}_\exists$ will verify that each existentially quantified variable is assigned consistently, and the formula $\text{NCONS}_\forall$ will verify that at least one universally quantified variable is assigned inconsistently.

3.3.3 Definition of SAT

For every $i \in \{1, \ldots, s+1\}$, we have variables μ_i and ν_i. By scanning its literals left-to-right, the formula checks that every clause of ϕ' contains at least one literal $\ell_k = x^{(a)}$ such that τ_k is given value a. To do so, μ_i and ν_i indicate the status of this process when exactly $i-1$ literals have been scanned. The intended meaning of the variables is the following:

- $\mu_i =$ "just before scanning ℓ_i, the clauses already completely scanned are satisfied, and the current clause is not satisfied yet".
- $\nu_i =$ "just before scanning ℓ_i, the clauses already completely scanned are satisfied, and the current clause is satisfied as well".

Note that ℓ_{s+1} is not a literal. Therefore, "just before scanning ℓ_{s+1}" means "just after scanning the last literal" in this case. Also, variables μ_1 and ν_1 are initialized to true and false, respectively. We want to make sure that at position $i = s+1$, i.e. after scanning the last literal, μ_{s+1} is true. Later, we will axiomatize the transition between positions i and $i+1$. That will define μ_{i+1} and ν_{i+1} depending on μ_i, ν_i and ℓ_i according to its intended meaning. We will axiomatize this into the formula $\text{SAT}(i)$. Then, SAT is defined as

$$\text{SAT} := \exists \boldsymbol{\mu} \exists \boldsymbol{\nu} \left(\mu_1 \cdot \overline{\nu_1} \cdot \prod_{i=1}^{s} \text{SAT}(i) \cdot \mu_{s+1} \right)$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{s+1})$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{s+1})$.

Next, we formalize $\text{SAT}(i)$. For every $i \in \{1, \ldots, s\}$, let $a_i \in \{0,1\}$ denote the sign of ℓ_i, the i-th literal of ϕ', and let $k_i \in \{0,1\}$ be the predicate that indicates whether ℓ_i is the last in literal its clause. Then, $\text{SAT}(i)$ is the conjunction of the following formulas:

$$\begin{aligned}
\mu_{i+1} \;&\leftrightarrow\; \overline{k_i}\,\mu_i\,a_i\,\overline{\tau_i} \;+\; \overline{k_i}\,\mu_i\,\overline{a_i}\,\tau_i \;+\; k_i\,\mu_i\,a_i\,\tau_i \;+\; k_i\,\mu_i\,\overline{a_i}\,\overline{\tau_i} \;+\; k_i\,\nu_i, \\
\nu_{i+1} \;&\leftrightarrow\; \overline{k_i}\,\mu_i\,a_i\,\tau_i \;+\; \overline{k_i}\,\mu_i\,\overline{a_i}\,\overline{\tau_i} \;+\; \overline{k_i}\,\nu_i.
\end{aligned}$$

In words, the axiomatization states that μ_{i+1} holds in one of three cases: 1) if ℓ_i is the last literal in its clause and the clause has been satisfied by a previous literal ($k_i\nu_i$), or 2) if ℓ_i is the last literal in its clause, this clause is not yet satisfied by a previous literal, but the truth assignment satisfies the current one ($k_i\mu_i a_i\tau_i + k_i\mu_i\overline{a_i\tau_i}$), or 3) if ℓ_i is not the last literal in its clause, this clause is not yet satisfied by a previous literal, and the truth assignment does not satisfy the current one either ($\overline{k_i}\mu_i a_i\overline{\tau_i} + \overline{k_i}\mu_i\overline{a_i}\tau_i$).

In the case of ν, the axiomatization states that ν_{i+1} holds in one of two cases: 1) if ℓ_i is not the last literal in its clause and the clause has been satisfied by a previous literal ($\overline{k_i}\nu_i$), or 2) if ℓ_i is not the last literal in its clause, this clause is not yet satisfied by a previous literal, but the truth assignment satisfies the current one ($\overline{k_i}\mu_i a_i \tau_i + \overline{k_i}\mu_i \overline{a_i \tau_i}$). Note that if ℓ_i is the last literal in its clause, then ν_{i+1} is always false, as it refers to a clause that cannot be satisfied yet.

Note that these two formulas can be written in CNF by writing $\leftrightarrow$ in terms of conjunctions and disjunctions and by distributing disjunctions over conjunctions. We call *i-link* a clause that contains variables only with indices i and $i+1$. Observe for later use that all clauses in the resulting CNF-formulas for $\text{SAT}(i)$ are i-links. Also, the size of SAT written in CNF is $c \cdot s$ for some constant $c \geq 1$.

3.3.4 Definition of $\text{CONS}_\exists$

The construction of $\text{CONS}_\exists$ is a bit more complicated. It uses universally quantified variables $\{\pi_1, \ldots, \pi_s\}$ as pointers to the literals of ϕ', in one-to-one correspondance with $\{\tau_1, \ldots, \tau_s\}$. We say that pointer π_i points to literal ℓ_i. If x is the underlying variable of ℓ_i, we say that π_i points to x. Pointers that are set to true are called *activated*. We say that a pointer has been scanned if its pointed literal has been scanned. The formula checks the following: whenever exactly two pointers are activated and they point to occurrences of the same existentially quantified variable, then the truth values assigned to the pointed literals are consistent. To refer to a variable, we do not encode its identifier directly. Instead, we encode the parity of its group and its index inside this group. This is enough information to distinguish between different variables in the same or neighbouring blocks. This fact is key to our argument and will be proved later in Claim 5. The point is that this compact encoding uses only $|n|+1$ bits per occurrence, where n is the number of variables per group, which may be much smaller than the total number of variables.

The formula uses the following variables for $i \in \{1, \ldots, s+1\}$:

- ξ_i = "just before scanning ℓ_i, all the activated pointers already scanned point to an existentially quantified variable".
- $\sigma_{i,k}$ = "just before scanning ℓ_i, exactly k activated pointers have been scanned".
- $\chi_{i,k}$ = "just before scanning ℓ_i, exactly one activated pointer has been scanned and there have been k changes of block between the pointed literal and position i, or exactly two

have been scanned and there have been exactly k changes of block between the pointed literals".

- ω_i = "just before scanning ℓ_i, exactly one activated pointer has been scanned and the parity of the group of the pointed variable is equal to the parity of the block of the clause of the pointed literal, or exactly two have been scanned and the groups of the pointed variables are the same".

- κ_i = "just before scanning ℓ_i, exactly one activated pointer has been scanned and the τ-variable at the pointed position is true, or exactly two have been scanned and the truth values of the τ-variables at the pointed positions are the same".

- $\lambda_{i,b}$ = "just before scanning ℓ_i, exactly one activated pointer has been scanned and the b-th bit of the index of the pointed variable in its group is 1, or exactly two have been scanned and the b-th bit of the indices of the pointed variables in their respective groups are the same".

The variables at step $i+1$ will be axiomatized in terms of the variables at step i and ℓ_i in the formula $\text{CONS}_\exists(i)$. The formula $\text{CONS}_\exists$ also requires a consistency condition for all possible combinations of activated pointers. For a given combination of these pointers, the consistency condition holds if: either there is a problem with the pointers (there are not exactly two pointers activated or one is not pointing to an existentially quantified variable), or the pointed variables are not comparable (are not of the same group or do not have the same index in the group) or, they are comparable and both receive the same truth value. This consistency condition will be encoded in the formula $\text{CONS}_\exists^{\text{acc}}$. Also, the value of the variables at position $i=1$ will be encoded in the formula $\text{CONS}_\exists^{\text{ini}}$. Now,

$$\text{CONS}_\exists := \forall\boldsymbol{\pi}\exists\boldsymbol{\xi}\exists\boldsymbol{\sigma}\exists\boldsymbol{\chi}\exists\boldsymbol{\omega}\exists\boldsymbol{\kappa}\exists\boldsymbol{\lambda}\left(\text{CONS}_\exists^{\text{ini}}\cdot\prod_{i=1}^{s}\text{CONS}_\exists(i)\cdot\text{CONS}_\exists^{\text{acc}}\right)$$

where $\boldsymbol{\pi}=(\pi_i\,|\,1\le i\le s)$, $\boldsymbol{\xi}=(\xi_i\,|\,1\le i\le s+1)$, $\boldsymbol{\sigma}=(\sigma_{i,k}\,|\,1\le i\le s+1, 0\le k\le 2)$, $\boldsymbol{\chi}=(\chi_{i,k}\,|\,1\le i\le s+1, 0\le k\le 1)$, $\boldsymbol{\omega}=(\omega_i\,|\,1\le i\le s+1)$, $\boldsymbol{\kappa}=(\kappa_i\,|\,1\le i\le s+1)$ and $\boldsymbol{\lambda}=(\lambda_{i,b}\,|\,1\le i\le s+1, 1\le b\le |n|)$.

Next we axiomatize the introduced variables, but before that we need to introduce some notation.

Let $g_i\in\{1,\dots,\ell\}$ be the group-number of the variable underlying literal ℓ_i, let $n_i\in\{1,\dots,|G_{g_i}|\}$ be the index of this variable within G_{g_i}, and recall $a_i\in\{0,1\}$ denotes the sign

of ℓ_i. For every $i \in \{1, \ldots, s\}$, let $h_i \in \{0, 1\}$ be the predicate that indicates whether the i-th literal ℓ_i is the last in its block or not (recall that the blocks are subsequences of consecutive clauses that partition the sequence of clauses), and recall that $k_i \in \{0, 1\}$ is the predicate that indicates whether the i-th literal ℓ_i is the last in its clause or not. Next we encode the quantification of ϕ in a way that the type of quantification of each variable can be recovered from each of its occurrences: for every $i \in \{1, \ldots, s\}$, let $q_i \in \{0, 1\}$ be the predicate that indicates whether the variable that underlies the i-th literal ℓ_i is universally or existentially quantified in ϕ.

Finally, observe that the definition of leveled formula implies that if $b_i \in \{1, \ldots, \ell\}$ is the number of the block that contains the clause to which the i-th literal belongs, then the group-number g_i is either b_i or $b_i + 1$ whenever $1 \leq b_i \leq \ell - 1$, and is equal to ℓ if $b_i = \ell$. Accordingly, let $e_i \in \{0, 1\}$ be such that $g_i = b_i - e_i + 1$ for every $i \in \{1, \ldots, s\}$. In other words, e_i indicates whether the parities of g_i and b_i agree or not.

The following claim shows that, although the number ℓ of groups is in general unbounded, a constant number of bits of information are enough to tell if the underlying variables of two literals belong to the same group:

Claim 5. *Let i, j be such that $1 \leq i < j \leq s$. Then, the underlying variables of ℓ_i and ℓ_j belong to the same group if and only if one of the following conditions holds:*

1. *$e_i = e_j$ and $b_i = b_j$, or*

2. *$e_i = 0$, $e_j = 1$, and $b_i = b_j - 1$.*

Proof. For the only if side, we have $g_i = g_j$. Then, $b_i - e_i = b_j - e_j$ and also b_i is either b_j or $b_j - 1$. If $b_i = b_j$, then $e_i = e_j$. If $b_i = b_j - 1$, then necessarily $e_i = 0$ and $e_j = 1$.

For the if side, in the first case, $g_i = b_i - e_i + 1 = b_j - e_j + 1 = g_j$. In the second case, $g_i = b_i - e_i + 1 = b_j - 1 + 1 = b_j - e_j + 1 = g_j$. Therefore, $g_i = g_j$. □

Using this claim, we axiomatize $\text{CONS}_\exists(i)$ as the conjunction of the following formulas on the previously introduced variables:

$$\xi_{i+1} \;\leftrightarrow\; \xi_i \,(\,\overline{\pi_i} \;+\; \pi_i \, q_i)$$

This is, ξ_{i+1} holds only when the activated pointers already scanned point to existentially quantified variables (ξ_i) and, either the pointer on position i is not activated ($\overline{\pi_i}$) or it is

activated and the underlying variable of ℓ_i is quantified existentially ($\pi_i q_i$).

$$\begin{aligned}
\sigma_{i+1,0} &\leftrightarrow \sigma_{i,0}\,\overline{\pi_i} \\
\sigma_{i+1,1} &\leftrightarrow \sigma_{i,0}\,\pi_i \;+\; \sigma_{i,1}\,\overline{\pi_i} \\
\sigma_{i+1,2} &\leftrightarrow \sigma_{i,1}\,\pi_i \;+\; \sigma_{i,2}\,\overline{\pi_i}
\end{aligned}$$

This is, $\sigma_{i+1,k}$ holds in one of two cases: 1) if k activated pointers were already scanned and the pointer to ℓ_i is not activated ($\sigma_{i,k}\overline{\pi_i}$), or 2) if $k > 1$, there were $k-1$ activated pointers scanned up to position i, and the pointer to ℓ_i is activated ($\sigma_{i,k-1}\pi_i$). Note that, since the consistency of the assigned truth values is checked for every pair of occurrences of a variable, it makes no sense that more than two pointers are activated, and therefore, there is no need to keep counting beyond two.

$$\begin{aligned}
\chi_{i+1,0} &\leftrightarrow \sigma_{i,0}\,\pi_i\,\overline{h_i} \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\chi_{i,0}\,\overline{h_i} \;+\; \sigma_{i,1}\,\pi_i\,\chi_{i,0} \;+\; \sigma_{i,2}\,\chi_{i,0} \\
\chi_{i+1,1} &\leftrightarrow \sigma_{i,0}\,\pi_i\,h_i \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\chi_{i,0}\,h_i \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\chi_{i,1}\,\overline{h_i} \;+\; \sigma_{i,1}\,\pi_i\,\chi_{i,1} \;+\; \sigma_{i,2}\,\chi_{i,1}
\end{aligned}$$

This is, $\chi_{i+1,0}$ holds in one of two cases: 1) if, including the current literal, exactly one activated pointer has been scanned, no change of block has been found after the activated pointer and ℓ_i is not the last literal of its block ($\sigma_{i,0}\pi_i\overline{h_i} + \sigma_{i,0}\overline{\pi_i}\chi_{i,0}\overline{h_i}$), 2) or if, including the current literal, exactly two activated pointer have been scanned, and no changes of block have been found inbetween ($\sigma_{i,1}\pi_i\chi_{i,0} + \sigma_{i,2}\chi_{i,0}$).

Also, $\chi_{i+1,1}$ holds in one of three cases: 1) if, including the current literal, exactly one activated pointer has been scanned, no change of block had been found after the first activated pointer and ℓ_i is the last literal of its block ($\sigma_{i,0}\pi_i h_i + \sigma_{i,0}\overline{\pi_i}\chi_{i,0}h_i$), or 2) if exactly one activated pointer has been scanned, the current one is not activated, there has been exactly one change of block since the first activated pointer, and ℓ_i is not the last literal of its block ($\sigma_{i,0}\overline{\pi_i}\chi_{i,1}\overline{h_i}$), or 3) if, including the current literal, exactly two activated pointers have been scanned and exactly one change of block has been found between them ($\sigma_{i,1}\pi_i\chi_{i,1} + \sigma_{i,2}\chi_{i,1}$).

$$\omega_{i+1} \;\leftrightarrow\; \sigma_{i,0}\,\pi_i\,e_i \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\omega_i \;+\; \sigma_{i,1}\,\pi_i\,(\,\chi_{i,0}\,\omega_i\,e_i \;+\; \chi_{i,0}\,\overline{\omega_i}\,\overline{e_i} \;+\; \chi_{i,1}\,\overline{\omega_i}\,e_i) \;+\; \sigma_{i,2}\,\omega_i$$

This is, ω_{i+1} holds in one of four cases: 1) if no activated pointer has been scanned yet, the current one is activated and the parity of the group and the block of ℓ_i agree ($\sigma_{i,0}\pi_i e_i$), or 2) if exactly one activated pointer has been scanned, the current one is not activated and ω_i holds, to propagate tha value of ω_i ($\sigma_{i,1}\overline{\pi_i}\omega_i$), or 3) if exactly one activated pointer has been

scanned, the current one is activated and the groups of the pointed variables are the same ($\sigma_{i,1}\pi_i(\chi_{i,0}\omega_i e_i + \chi_{i,0}\overline{\omega_i e_i} + \chi_{i,1}\overline{\omega_i} e_i)$), or 4) if exactly two pointers have been scanned and ω_i holds, again to propagate the value of ω_i in these conditions. Note that in 4) the groups of the pointed variable are the same since the second of the pointed variables is either in the same block as the first one, or the following.

$$\kappa_{i+1} \;\leftrightarrow\; \sigma_{i,0}\,\pi_i\,\tau_i \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\kappa_i \;+\; \sigma_{i,1}\,\pi_i\,\kappa_i\,\tau_i \;+\; \sigma_{i,1}\,\pi_i\,\overline{\kappa_i}\,\overline{\tau_i} \;+\; \sigma_{i,2}\,\kappa_i$$

This is, κ_{i+1} holds in one of four cases: 1) if no activated pointer has been scanned yet, the current pointer is activated and ℓ_i has been assigned truth-value true, that is, we store the truth value assigned to ℓ_i ($\sigma_{i,0}\pi_i\tau_i$), or 2) if exactly one activated pointer has been scanned, the current pointer is not activated and κ_i holds, this is, we propagate the value of κ_i ($\sigma_{i,1}\,\overline{\pi_i}\,\kappa_i$), or 3) if exactly one activated pointer has been scanned, the current pointer is activated and the value of κi agrees with the truth-value assigned to ℓ_i ($\sigma_{i,1}\pi_i\kappa_i\tau_i + \sigma_{i,1}\pi_i\overline{\kappa_i\tau_i}$), or 4) if exactly two activated pointers have been scanned and κ_i holds, that is again, we propagate the value of κ_i ($\sigma_{i,2}\kappa_i$).

Let $n_{i,b}$ be the b-th bit of the binary encoding of n_i. For all $b \in \{1, \ldots, |n|\}$,

$$\lambda_{i+1,b} \;\leftrightarrow\; \sigma_{i,0}\,\pi_i\,n_{i,b} \;+\; \sigma_{i,1}\,\overline{\pi_i}\,\lambda_{i,b} \;+\; \sigma_{i,1}\,\pi_i\,\lambda_{i,b}\,n_{i,b} \;+\; \sigma_{i,1}\,\pi_i\,\overline{\lambda_{i,b}}\,\overline{n_{i,b}} \;+\; \sigma_{i,2}\,\lambda_{i,b}$$

This is, $\lambda_{i+1,b}$ holds in one of four cases: 1) if no activated pointer has been scanned yet, the current pointer is activated and the b-th bit of the binary encoding of n_i is 1, that is, we store the value of this bit ($\sigma_{i,0}\pi_i n_{i,b}$), or 2) if exactly one activated pointer has been scanned, the current pointer is not activated and $\lambda_{i,b}$ holds, this is, we propagate the value of $\lambda_{i,b}$ ($\sigma_{i,1}\,\overline{\pi_i}\,\lambda_{i,b}$), or 3) if exactly one activated pointer has been scanned, the current pointer is activated and the value of $\lambda_{i,b}$ agrees with b-th bit of the binary encoding of n_i ($\sigma_{i,1}\pi_i\lambda_{i,b}n_{i,b} + \sigma_{i,1}\pi_i\overline{\lambda_{i,b}}\overline{n_{i,b}}$), or 4) if exactly two activated pointers have been scanned and $\lambda_{i,b}$ holds, that is again, we propagate the value of κ_i ($\sigma_{i,2}\lambda_{i,b}$). Note that the conjunction of these variables for all bits check that the n_i's of the pointed variables are equal.

Also, we define $\textsc{cons}^{\text{ini}}_{\exists}$ as the conjunction of the following unit clauses:

$$\xi_1, \sigma_{1,0}, \overline{\sigma_{1,1}}, \overline{\sigma_{1,2}}, \overline{\chi_{1,0}}, \overline{\chi_{1,1}}, \overline{\omega_1}, \overline{\kappa_1}, \overline{\lambda_{1,1}}, \ldots, \overline{\lambda_{1,|n|}}.$$

Note that this inizialization is consistent with the intended meaning of the variables just before scanning the first literal.

Furthermore, we define $\text{CONS}_\exists^{\text{acc}}$ as the following clause:

$$\overline{\xi_{s+1}} + \overline{\sigma_{s+1,2}} + \overline{\omega_{s+1}} + \sum_{b=1}^{|n|} \overline{\lambda_{s+1,b}} + \kappa_{s+1}.$$

This is, after reading the last literal, if every activated pointer points to an existentially quantified variable, and exactly two activated pointers have been scanned, and the two pointed literals belong to the same group, and their n_i inside the group is the same, then their assigned truth-value is also the same.

Again, note that each of these formulas can be written in CNF just by writing $\leftrightarrow$ in terms of conjunctions and disjunctions and by distributing disjunctions over conjunctions, and that the clauses in the resulting CNF-formulas for $\text{CONS}_\exists(i)$ are i-links: the (first) index of the variables they contain is either i or $i+1$. Also, the size of $\text{CONS}_\exists$ written in CNF is $c \cdot s \cdot |n|$ for some constant $c \geq 1$.

3.3.5 Definition of $\text{NCONS}_\forall$

The formula $\text{NCONS}_\forall$ is very similar to $\text{CONS}_\exists$, since it verifies for universally quantified variables exactly the opposite of what $\text{CONS}_\exists$ verifies for existentially quantified variables. For this reason, we proceed to its axiomatization directly.

The formula $\text{NCONS}_\forall$ is defined as

$$\text{NCONS}_\forall := \exists\boldsymbol{\pi}\exists\boldsymbol{\xi}\exists\boldsymbol{\sigma}\exists\boldsymbol{\chi}\exists\boldsymbol{\omega}\exists\boldsymbol{\kappa}\exists\boldsymbol{\lambda}\left(\text{NCONS}_\forall^{\text{ini}} \cdot \prod_{i=1}^{s} \text{NCONS}_\forall(i) \cdot \text{NCONS}_\forall^{\text{acc}}\right)$$

where $\boldsymbol{\pi}$, $\boldsymbol{\xi}$, $\boldsymbol{\sigma}$, $\boldsymbol{\chi}$, $\boldsymbol{\omega}$, $\boldsymbol{\kappa}$, $\boldsymbol{\lambda}$ are defined as before, $\text{NCONS}_\forall^{\text{ini}} := \text{CONS}_\exists^{\text{ini}}$, the formula $\text{NCONS}_\forall(i)$ is axiomatized identically to $\text{CONS}_\exists(i)$ except by replacing every occurrence of q_i by $\overline{q_i}$ for every $i \in \{1, \ldots, s\}$, and the formula $\text{NCONS}_\forall^{\text{acc}}$ is the negation of $\text{CONS}_\exists^{\text{acc}}$, i.e. the following set of unit clauses:

$$\xi_{s+1}, \sigma_{s+1,2}, \omega_{s+1}, \lambda_{s+1,1}, \ldots, \lambda_{s+1,|n|}, \overline{\kappa_{s+1}}.$$

In CNF, the formula $\text{NCONS}_\forall(i)$ is again a set of i-links, and its size is $c \cdot s \cdot |n|$ for some $c \geq 1$.

3.3.6 Converting θ to leveled-QBF

Recall that θ was defined as $Q_1\boldsymbol{\tau}_1 \cdots Q_q\boldsymbol{\tau}_q(\text{NCONS}_\forall + (\text{CONS}_\exists \cdot \text{SAT}))$. By writing this formula in prenex form, we obtain the equivalent formula

$$\mathbf{Qz}\ (\text{NCONS}'_\forall + (\text{CONS}'_\exists \cdot \text{SAT}'))$$

where $\mathbf{Qz}$ is the appropriate prefix of quantified variables and the primed formulas are the matrices of the corresponding non-primed QBFs. We would like to write it as a leveled-QBF.

Let a and b be two new variables and let ϑ be the conjunction of the following formulas:

$$\begin{aligned} &a + \text{NCONS}'_\forall \\ &b + \text{NCONS}'_\forall \\ &\bar{a} + \text{CONS}'_\exists \\ &\bar{b} + \text{SAT}' \end{aligned}$$

It is easy to see that

$$\exists a \exists b(\vartheta) \leftrightarrow \text{NCONS}'_\forall + (\text{CONS}'_\exists \cdot \text{SAT}').$$

We write ϑ in CNF. For the first disjunction $a + \text{NCONS}'_\forall$, it is enough to add a to every clause of $\text{NCONS}'_\forall$, and similarly for the others. Note that, except for the variables a and b, the result is a conjunction of i-links.

In order to make them proper i-links, we introduce new variables $\{a_1, \ldots, a_{s+1}\}$ and $\{b_1, \ldots, b_{s+1}\}$, and clauses $a_i \leftrightarrow a_{i+1}$ and $b_i \leftrightarrow b_{i+1}$ for every $i \in \{1, \ldots, s\}$ to mantain consistency between the introduced variables. Now, we replace each occurrence of a and b in an improper i-link by a_i and b_i respectively. Let ψ' be the resulting formula.

Finally, define

$$\psi := \mathbf{Qz}\exists\mathbf{a}\exists\mathbf{b}(\psi')$$

where $\mathbf{a} = (a_1, \ldots, a_{s+1})$ and $\mathbf{b} = (b_1, \ldots, b_{s+1})$. Note that the construction guarantees $\psi \leftrightarrow \theta$, and by Claim 4, $\psi \leftrightarrow \phi$.

We partition the variables of ψ in groups $H_1, \ldots, H_{s+1}$ where group H_i is the set of variables with (first) index i. We also partition the clauses of ψ in blocks $C_1, \ldots, C_{s+1}$ where block C_i is the set of i-links of ψ. Note that, by the definition of i-link, all variables in C_i are contained in $H_i \cup H_{i+1}$. Therefore, ψ is a leveled-QBF with groups $H_1, \ldots, H_{s+1}$ and blocks $C_1, \ldots, C_{s+1}$.

Now, for every $i \in \{1, \ldots, s+1\}$, the size of H_i is the number of variables with index i

in ψ, namely $c \cdot |n|$ for some constant $c \geq 1$. Also, the size of ψ is $d \cdot s \cdot |n|$ for some constant $d \geq 1$. Therefore, ψ is a $c \cdot |n|$-leveled QBF of size $d \cdot s \cdot |n|$ such that $\phi \leftrightarrow \psi$.

Finally, it is clear that all the steps to produce ψ from ϕ can be performed in time polynomial in s, thus finishing the proof.

3.4 Bounded-width TQBF

In this section we prove the main result of the chapter.

Theorem 2. *There exists an integer $w \geq 1$ such that TQBF on inputs of path-width at most w is PSPACE-complete.*

Proof. We show that there exists a constant $n_0 \geq 1$ and a polynomial-time reduction from the canonical PSPACE-complete problem TQBF to the restriction of TQBF itself to n_0-leveled QBFs. Then the result will follow by setting the path-width to $w = 2n_0 - 1$ and applying Lemma 8.

Let c and d be the constants from the end of section 3.3. We choose the constant n_0 large enough so that whenever $N \geq n_0$ the following conditions are satisfied:

1. $c \cdot |N| < N$,
2. $c \cdot |c \cdot |N|| \leq \log N$,
3. $(2 \log^* N)(\log |N|) \leq \log N$,
4. $d^{2 \log^* N} \leq \log N$.

All these conditions can be met simultaneously. The idea of the reduction is to start with an arbitrary QBF formula ϕ_0 with N_0 variables and size S_0, view it as an N_0-leveled QBF, and apply Lemma 9 repeatedly until we get a n_0-leveled QBF for the large fixed constant n_0. Since the final formula will be equivalent to ϕ_0, we just need to make sure that this process terminates in a small number of iterations and that the size of the resulting formula is polynomial in S_0. We formalize this below.

Let ϕ_0 be an arbitrary QBF formula with N_0 variables and size S_0. In particular ϕ_0 is an N_0-leveled QBF of size S_0. If $N_0 \leq n_0$ then ϕ_0 is already n_0-leveled and there is nothing to do. Assume then $N_0 > n_0$. We apply Lemma 9 to get an N_1-leveled QBF of size S_1 where $N_1 = c \cdot |N_0|$ and $S_1 = d \cdot S_0 \cdot |N_0|$. By condition 1 on n_0 we get $N_1 < N_0$, which is progress. Repeating this we get a sequence of formulas $\phi_0, \phi_1, \ldots, \phi_t$, where ϕ_i is an N_i-leveled QBF of size S_i with

1. $N_i = c \cdot |N_{i-1}|$, and
2. $S_i = d^i \cdot S_0 \cdot \prod_{j=0}^{i-1} |N_j|$,

for $i \geq 1$. We stop the process at the first $i = t$ such that $N_t \leq n_0$. We claim that $t \leq 2 \log^* N_0$ and that $S_t \leq S_0 \cdot N_0 \cdot \log N_0$. This will be enough, since then the algorithm that computes ϕ_t from ϕ_0 is the required reduction as it runs in time polynomial in the size of the formula, and $\phi_0 \leftrightarrow \phi_t$.

Claim 6. *It holds that* $t \leq 2 \log^* N_0$.

Proof. First, by conditions 1 and 2 on n_0 we have

1. $N_i = c \cdot |N_{i-1}| < N_{i-1}$, and
2. $N_{i+1} = c \cdot |N_i| = c \cdot |c \cdot |N_{i-1}|| \leq \log N_{i-1}$

for every $i \geq 1$ such that $N_{i-1} > n_0$. In particular, this means that the process terminates and t exists. Unfolding the second inequality gives

$$N_{t-1} \leq \log^{(\lfloor (t-1)/2 \rfloor)} N_0.$$

However, by the choice of t we have $N_{t-1} > n_0 \geq 1$, which means that $\lfloor (t-1)/2 \rfloor < \log^* N_0$ and therefore $t \leq 2 \log^* N_0$. □

Given this bound on t, we bound S_t. We have

$$S_t = d^t \cdot S_0 \cdot \prod_{j=0}^{t-1} |N_j| \leq d^t \cdot S_0 \cdot |N_0|^t,$$

where in the inequality we used the fact that $N_i \leq N_{i-1}$ for every $i \geq 1$ such that $N_{i-1} > n_0$, by condition 1 on n_0. Now:

$$|N_0|^t \leq 2^{(2 \log^* N_0)(\log |N_0|)} \leq 2^{\log N_0} = N_0.$$

In the first inequality we used the bound on t, and in the second we used the assumption that $N_0 \geq n_0$ and condition 3 on n_0. Altogether, this gives

$$S_t \leq d^{2 \log^* N_0} \cdot S_0 \cdot N_0 \leq S_0 \cdot N_0 \cdot \log N_0,$$

which concludes the proof. Again, we used the assumption that $N_0 \geq n_0$ and condition 4 on n_0. □

3.5 The Q-resolution proof system

In this section we define and compare some proof systems for QBFs. In order to state their rules, it is useful to note that a QBF can be written as

$$\phi = Q_1 X_1 \cdots Q_q X_q(\phi') \tag{3.2}$$

where $X_1, \ldots, X_q$ are disjoint sequences of propositional variables, and $Q_i \neq Q_{i+1}$ for $1 \leq i < q$. Of course $Q_i X_i$ means $Q_i x_1^i \ldots Q_i x_\ell^i$ for $X_i := (x_1^i, \ldots, x_\ell^i)$. Also, we say that X_i is a *quantifier block* of ϕ. Note that logical equivalence is preserved upon reordering of the variables within the same quantifier block. To establish an order between the variables in the prefix of a QBF that accounts for the quantifier blocks, we say that x *is after* y in ϕ for $x, y \in \mathrm{var}(\phi)$ if x and y belong to quantifier blocks X_i and X_j, respectively, with $i > j$. For the rest of the chapter, all the literals in a clause have different underlying variables and, in particular, all clauses are non-tautological.

In [13], in an attempt to generalize resolution to QBFs, Büning et al. introduced the Q-resolution proof system, consisting of the following rules:

1. $\dfrac{C}{\square}$, if every $x \in \mathrm{var}(C)$ is quantified universally.
2. $\dfrac{C \vee x \quad D \vee \overline{x}}{(C' \vee D')''}$, if x is quantified existentially, where
   (a) C' (resp. D') is equal to C (resp. D) except for the literals whose underlying variable is quantified universally and is after every existentially quantified variable y in $\mathrm{var}(C)$ (resp. $\mathrm{var}(D)$) in ϕ, and
   (b) $(C' \vee D')''$ is 1 if $(C' \vee D')$ is tautological and, otherwise, is equal to $(C' \vee D')$ except for the literals whose underlying variable is quantified universally and is after every existentially quantified variable y in $\mathrm{var}(C') \cup \mathrm{var}(D')$ in ϕ.

Later, Pan and Vardi [39] extended the symbolic quantifier elimination approach from CNF formulas to QBFs. They introduce a QBF solver that produces multi-resolution [15] refutations. Even though they use OBDDs to represent the clauses, the proof system that is implicit in their algorithm has the following two rules:

1. $\dfrac{C \vee x}{C}$, if x is quantified universally and no $y \in \mathrm{var}(C)$ is after x in ϕ.
2. $\dfrac{C \vee x \quad D \vee \overline{x}}{C \vee D}$, if x is quantified existentially and no $y \in \mathrm{var}(C) \cup \mathrm{var}(D)$ is after x in ϕ.

In this work, we will call this proof system *weak Q-resolution.*

We introduce a simpler proof system, in the mold of weak Q-resolution, with the following rules:

1. $\dfrac{C \vee x}{C}$, if x is quantified universally and no $y \in \mathrm{var}(C)$ is after x in ϕ.
2. $\dfrac{C \vee x \quad D \vee \overline{x}}{C \vee D}$, if x is quantified existentially.

For the moment, let us call this system *Q*-resolution.* Note that it is stronger than weak Q-resolution, since their only difference is that Q*-resolution weakens the restrictions to apply its second rule.

We say that a proof system P' *p-simulates* a proof system P if, whenever a contradiction has a P-refutation size s, it also has a P'-refutation of size polynomial in s. Also, we say that two proof systems are *p-equivalent* if they p-simulate each other. We show that Q*-resolution is, in fact, p-equivalent to Q-resolution:

Lemma 10. *The proof systems Q-resolution and Q*-resolution are p-equivalent.*

Proof. Let R_1 and R_2 be rules 1. and 2. of Q-resolution, and let R_1^* and R_2^* be rules 1. and 2. of Q*-resolution. First, we show that Q*-resolution p-simulates Q-resolution. To do so, we show that every Q-resolution step can be simulated by several Q*-resolution steps. To simulate R_1, if C is a purely universal clause, we obtain a Q*-resolution refutation of C by applying R_1^* repeatedly $|C|$ times, always on the literal whose underlying variable is the right-most in the prefix. To simulate R_2 on clauses C and D, again, repeatedly apply R_1^* on the universally quantified variables of C and D that are after every existentially quantified variable in its clause in right-to-left order, then apply R_2^* on the resulting clauses, and finally, repeteadly apply R_1^* on the universally quantified variables of the resulting clause that are after every existentially quantified variable, in right-to-left order. These are, at most, $|C| + |D|$ steps.

Second, we show that Q-resolution p-simulates Q*-resolution. Let $C_1^*, \ldots, C_\ell^*$ be a Q*-resolution refutation. For $i \in \{1, \ldots, \ell\}$ let

$$C_i := \begin{cases} C_i^* & \text{if } C_i^* \text{ is an initial clause,} \\ C_j & \text{if } C_i^* = R_1^*(C_j^*) \text{ for some } j < i, \\ R_2(C_j, C_k) & \text{if } C_i^* = R_2^*(C_j^*, C_k^*) \text{ with } j, k < i. \end{cases}$$

First we want to see that $C_1, \ldots, C_\ell$ is a valid Q-resolution derivation. It is clear by definition that every C_i is either an initial clause or the result of applying R_2, since in the second case

C_i is already in the refutation. It remains to be seen that $\square$ can be derived from $C_1, \ldots, C_\ell$ in one more step. For that, it is enough to show that, for every $i \in \{1, \ldots, \ell\}$, we have that $C_i = C_i^*$ if C_i^* is an initial clause and that, otherwise, C_i subsumes $C_i^* \vee A_i$ for some purely universal clause A_i whose literals are after every existentially quantified variable of C_i^* in the prefix. If we suceed, just note that C_ℓ subsumes $(C_\ell^* \vee A_\ell) = (\square \vee A_\ell) = A_\ell$, and, since A_ℓ is a purely universal clause, we apply R_1 to C_ℓ to obtain $\square$. We are left to prove the claim. We will proceed by cases according to the definition of C_i^*. First, it is clear by definition that $C_i = C_i^*$ if C_i^* is an initial clause. Second, if $C_i^* = R_1^*(C_j^*)$ for some $j < i$, let l_i be the universally quantified literal that is in C_j^* and not in C_i^* and let $A_i := A_j \vee l_i$. Since C_j subsumed $C_j^* \vee A_j$, it is clear that C_i subsumes $C_i^* \vee A_i$. Third, we have that C_j subsumes $C_j^* \vee A_j$ and C_k subsumes $C_k^* \vee A_k$. By the definition of the rule and the conditions on A_j and A_k, we have that $R_2(C_j, C_k) = R_2(C_j^* \vee A_j, C_k^* \vee A_k)$. Therefore, $R_2(C_j, C_k)$ subsumes $R_2^*(C_j^* \vee A_j, C_k^* \vee A_k)$, this is, C_i subsumes C_i^* and, therefore, $C_i^* \vee A_i$ for $A_i := \emptyset$. $\square$

Since both proof systems are p-equivalent, to simplify notation, we will refer to both as Q-resolution for the rest of the chapter.

Now, for a QBF ϕ with matrix ϕ' and for variables x, y quantified existentially and universally respectively in ϕ, we define

$$\phi'^{(\exists x)} := \{C \vee D \mid C \vee x \in \phi' \text{ and } D \vee \overline{x} \in \phi'\} \cup \{C \in \phi' \mid x \notin \text{var}(C)\}, \text{ and}$$
$$\phi'^{(\forall x)} := \{C \mid C \vee x \in \phi' \text{ or } C \vee \overline{x} \in \phi'\} \cup \{C \in \phi' \mid x \notin \text{var}(C)\}.$$

We write $\phi'^{(Q_1 x_1, Q_2 x_2)}$ instead of $(\phi'^{(Q_2 x_2)})^{(Q_1 x_1)}$. Note that $x \notin \text{var}(\phi^{(Qx)})$. We prove the following lemma:

Lemma 11. *Let ψ be a CNF formula and let $\mathbf{Q}$ be any prefix. Then,*

1. $\mathbf{Q}\psi^{(\exists x)} \models \mathbf{Q}\exists x\psi$, *and*
2. $\mathbf{Q}\psi^{(\forall x)} \models \mathbf{Q}\forall x\psi$.

Proof. For the first claim, let A be an assignment that satisfies $\psi^{(\exists x)}$. Let A_0, A_1 be extensions of A that assign $x := 0$ and $x := 1$ respectively. If neither satisfies ψ, then ψ contains at least a pair of clauses $C \vee x$ and $D \vee \overline{x}$ such that $A_0(C) = 0$ and $A_1(D) = 0$. But then, $C \vee D$ belongs to $\psi^{(\exists x)}$ and $A(C \vee D) = 0$ causing a contradiction. Therefore, since at least one of A_0 and A_1 satisfies ψ, we have that A satisfies $\exists x\psi$. For the second claim, just note that $\psi^{(\forall x)} \models \psi$. $\square$

Note that completeness of weak Q-resolution (and therefore, also Q-resolution) is proved by repeated applications of this lemma: let $\mathbf{Q}$ be the prefix of ϕ. Then, $\phi'^{(\mathbf{Q})}$ is either empty, and therefore the formula is true, or contains just $\square$, and therefore the formula is false.

Various efforts have been directed to determine families of QBFs for which the Q-resolution proof system is polynomially bounded. Aspvall et al. [6] showed that (weak) Q-resolution is polynomially bounded for bijunctive-QBFs, that is, formulas with at most two literal per clause. Later, Büning et al. [13] showed the same for Horn-QBFs. They also proved that extended-Horn QBFs, that is, QBFs in which the existentially quantified part of each clause is Horn and the universal part is arbitrary, require exponential-size Q-resolution refutations.

Observe that Theorem 2 implies that, unless NP=PSPACE, no proof-system is polynomially bounded for QBFs of bounded tree-width (and even path-width), as otherwise one could guess a polynomial-size refutation and check it in polynomial time. However, some families of QBFs with bounded tree-width have polynomial-size Q-resolution refutations. For example, if we allow only existential quantifiers, the problem becomes equivalent to boolean satisfiability of CNFs (CNF-SAT), and Alekhnovich and Razborov [3] showed that CNFs of bounded branch-width (which is equivalent to bounded tree-width) have polynomial-size resolution (and therefore, (weak) Q-resolution) refutations. We devote the rest of the chapter to describe a larger family of QBFs with bounded tree-width for which (weak) Q-resolution is polynomially bounded.

3.6 Respectful tree-width

As defined in section 3.2, the tree-width of a QBF is the tree-width of its matrix, and therefore, it is independent of its prefix. Multiple algorithms on CNFs that are tractable on instances with bounded tree-width are not applicable to QBFs, since the tree decompositions that they use do not mesh well with the quantification of the variables. To tackle this problem, Chen and Dalmau [17] introduced what we call here *respectful tree-width*, a concept analogous to tree-width, but on tree decompositions that are, in some sense, respectful with the prefix of the formula, so that the algorithms for CNFs make sense.

Let ϕ be a QBF and let (T, L) be a tree decomposition of its matrix. Let r be the root of T. Define t_x as the closest vertex to r in T such that $x \in L(t_x)$. For a pair of variables $x, y \in \text{var}(\phi)$, we say that x *is under* y if $t_x \neq t_y$ and t_y is in the (unique) path from r to t_x in T. We say that (T, L) is *respectful* with the prefix of ϕ if, for every pair of variables

$x, y \in \mathrm{var}(\phi)$, if x is under y, then y is not after x. A *respectful tree decomposition* of ϕ is one that is respectful with its prefix. The *respectful tree-width* of ϕ is the minimum width among its respectful tree decompositions.

The main result of this section is that Q-resolution is polynomially bounded on QBFs of bounded respectful tree-width. The proof of this lemma makes use of a construction on graphs defined by Dechter and Pearl [24] named *induced graph*.

A pair $(H, \prec)$ is an induced graph of G if $\prec$ defines a strict total order on the vertices of G, and H is the closure of G under the following operation: for every $x, y, z \in V(H)$ such that $x \prec z$ and $y \prec z$, if (x, z) and (y, z) are edges, add (x, y) as an edge. The *width* of an induced graph is $\max_{x \in V(H)} |\{(x, y) \in E(H) \mid y \prec x\}|$. The *induced width* of a graph is the minimum among the widths of its induced graphs.

Given G and $\prec$, the usual way to obtain H, as proposed by Dechter and Pearl, is through the following process: one vertex of $V(H)$ at the time and in order opposite to $\prec$, add edges (x, y) for every pair x, y of neighbors of the current vertex z such that $x \prec z$ and $y \prec z$.

Let ϕ be a QBF and let $(H, \prec)$ be an induced graph of its matrix. We say that $(H, \prec)$ is *respectful* with the prefix of ϕ if, for every pair of variables $x, y \in \mathrm{var}(\phi)$, if $x \prec y$ then x is not after y. A *respectful induced graph* of ϕ is one that is respectful with its prefix. The *respectful induced width* of ϕ is the minimum width among its respectful induced graphs.

Observe the following claim:

Claim 7. *Let ϕ be a QBF as in (3.2) and let $(H, \prec)$ be a respectful induced graph of ϕ. Let S be the sequence of variables in ϕ in the order defined by $\prec$. Then, $S = Y_1, \ldots, Y_q$, where Y_i is a permutation of X_i for every $i \in \{1, \ldots, q\}$. Moreover, $Q_1 Y_1 \ldots Q_q Y_q(\phi')$ is logically equivalent to ϕ.*

In [4], Arnborg et al. show that a QBF has a tree decomposition of width w if and only if its constraint graph is a partial w-tree. Along the same lines, Freuder [27] shows that a QBF has an induced graph of width w if and only if its constraint graph is a partial w-tree. By composing these theorems, we obtain that a QBF has a tree decomposition of width w if and only if it has an induced graph of width w. In [23], Dechter gives a direct proof of the *if* side of this statement in terms of bucket elimination. Using the construction by Decther, we present a direct proof of the whole statement in graph-theoretic terms and show that our constructions preserve respectfulness.

Lemma 12. *Let ϕ be a QBF. Then ϕ has a respectful tree decomposition of width w if and only if it has a respectful induced graph of width w.*

Proof. Let G be the constraint graph of ϕ. First, let (T, L) be a respectful tree decomposition of ϕ of width w. We will construct a respectful induced graph of ϕ of the same width. Define $\prec$ as $x \prec y$ if y is under x in (T, L) and arbitrarily if neither is under the other. Let H be such that $(H, \prec)$ is an induced graph of G. We show that (T, L) is also a tree decomposition of H by induction on the number of edges of H. If $|E(H)| = |E(G)|$, then $H = G$ and we are done. If $|E(H)| > |E(G)|$, let (x, y) be an edge of $E(H) \setminus E(G)$. By definition of H, there is a $z \in V(H)$ such that $x \prec z$ and $y \prec z$ and both (x, z) and (y, z) belong to $E(H)$. By induction hypothesis (T, L) is a tree decomposition of $H - (x, y)$. We have to show that $x, y \in L(t)$ for some $t \in V(T)$. Let T_z be the connected subtree of T induced by the vertices $t \in V(T)$ such that $z \in L(t)$ and let t_z be the root of T_z. We will show that, in fact, both x and y belong to t_z. Let $T_z^x := \{t \in V(T_z) \mid x \in L(t)\}$. Since (T, L) is a tree decomposition of $H - (x, y)$ and $(x, z) \in E(H)$, we have that T_z^x is non-empty. Let $t_z^x \in T_z^x$ be the closest vertex to t_z among them. If $t_z^x \neq t_z$, then x is under z and, by the definition of $\prec$, we have that $z \prec x$, which is a contradiction. Therefore, $t_z^x = t_z$, which implies $x \in L(t_z)$. The same argument can be made to show that $y \in L(t_z)$, proving the claim. Define $\prec_R$ as $x \prec_R y$ if y is under x in (T, L) and $x \prec_R y$ if x occurs before y in the prefix of ϕ and neither is under the other. Let H_R be such that $(H_R, \prec_R)$ is an induced graph of G. Note that, since $\prec_R$ is a particular case of $\prec$, we have that (T, L) is also a tree decomposition of H_R. To see that $(H_R, \prec_R)$ is respectful we show that, if $x \prec_R y$, then x is not after y. We have two cases: first, if y is under x, then, since (T, L) is respectful, we have that x is not after y; and second, if x occurs before y in the prefix, of course x is not after y. Finally, for every $x \in V(H_R)$, by definition of H_R, the vertices of $V_x := \{x\} \cup \{y \mid y \prec x \text{ and } (x, y) \in E(H_R)\}$ form a clique. By Claim 3, for every $x \in V(H_R)$ there is a $t \in V(T)$ such that $V_x \subseteq L(t)$. Therefore,

$$\max_{x \in V(H_R)} |\{(x, y) \in E(H_R) \mid y \prec x\}| \leq \max_{x \in V(H_R)} |V_x| - 1 \leq \max_{t \in V(T)} |L(t)| - 1 \leq w.$$

Second, let $(H, \prec)$ be a respectful induced graph of ϕ of width w. We will construct a respectful tree decomposition of the same width. Let T be a graph with one vertex t_x for every $x \in V(H)$ and one edge (t_y, t_x) where y is the biggest (with respect to $\prec$) neighbor of x in H such that $y \prec x$. Note that T is acyclic, since for every $x \in V(H)$, we have that t_x is connected to at most one vertex t_y such that $y \prec x$. As defined, T is not rooted and may not be connected, but we will fix this at the end of the proof. Let L be defined by $L(t_x) := \{x\} \cup \{y \mid (y, x) \in E(H) \text{ and } y \prec x\}$ for every $x \in V(H)$. Next, we show that (T, L) is a respectful tree decomposition of G of width w. First, we have that $\bigcup_{t \in V(T)} L(t) = V(G)$

since $V(G) = V(H)$ and, for every $x \in V(H)$, we have that $x \in L(t_x)$. Second, for every $(x, y) \in E(G)$, we have $x, y \in L(t_x)$ if $y \prec x$ and $x, y \in L(t_y)$ if $x \prec y$. Third, we have to show that for every $x \in V(G)$, the subgraph of T induced by $\{t \in V(T) \mid x \in L(t)\}$ is a connected subtree. Recall that H has the property that, for every $x, y, z \in V(H)$ such that $x \prec z$ and $y \prec z$, if (x, z) and (y, z) are in $E(H)$, also (x, y) is in $E(H)$. It is enough to see that, if $x \in L(t)$, the unique shortest path $t_1, \ldots, t_\ell$ with $t_1 = t$ and $t_\ell = t_x$ is such that $x \in L(t_i)$ for every $i \in \{1, \ldots, \ell\}$. We prove this by induction on i. If $i = 1$, by hypothesis we have $x \in L(t_1)$. Now, let $i > 1$ and, as induction hypothesis, assume x belongs to $L(t_1), \ldots, L(t_{i-1})$. We want to show that x belongs to $L(t_i)$ also. Let $y, z \in V(H)$ be such that $t_{i-1} = t_z$ and $t_i = t_y$. Since, by induction hypothesis, $x \in L(t_z)$, we have that $x \prec z$ and that (x, z) is in $E(H)$. Also, since $(t_y, t_z) \in E(T)$, we have that (y, z) is in $E(H)$. Now, we show that $x \prec y$ by cases: if $z \prec y$, then $x \prec y$, since $x \prec z$. If $y \prec z$, then also $x \prec y$, since otherwise y would not have been the biggest (with respect to $\prec$) neighbour of z such that $y \prec z$ (x would satisfy the conditions and would be bigger than y). Note that, by the construction of T, every vertex $t_u \in V(T)$ has at most one neighbour in $V_u(T) := \{t_v \in V(T) \mid v \prec u\}$. Suppose, for the sake of contradiction, that $z \prec y$. Then, (y, z) is the single edge that connects y to $V_y(T)$. But since the path does not repeat edges, it cannot lead to any other vertex in $V_y(T)$. Since t_x is in $V_y(T)$, this is a contradiction. Therefore, we have that $y \prec z$. Finally, since $x \prec z$ and $y \prec z$ and (x, z) and (y, z) both belong to $E(H)$ and $(H, \prec)$ is an induced graph, also (x, y) belongs to $E(H)$. And then, since $x \prec y$, we have that $x \in L(t_i)$. We make sure now that the graph that we built is rooted and connected. Let $T_1, \ldots, T_k$ be the connected components of T. For $i \in \{1, \ldots, k\}$, let r_i be the unique vertex of T_i such that $|L(r_i)| = 1$. Let r be a fresh vertex and let $T_C = (V_C, E_C)$ with $V_C := V(T) \cup \{r\}$ and $E_C := V(E) \cup \bigcup_{i \in [k]} \{(r, r_i)\}$ be a rooted tree with r in the root. Note that T_C is connected. Also, let L_C be the extension of L to V_C such that $L_C(r) = \emptyset$. Note that (T_C, L_C) is respectful, since if x is under y, by construction of T_C surely $y \prec x$, and, since $(H, \prec)$ is respectful, y is not after x. Finally, (T_C, L_C) has width

$$\max_{t \in V(T_C)} |L_C(t)| - 1 = \max_{t \in V(T)} |L(t)| - 1 = \max_{x \in V(H)} |\{y \mid (x, y) \in E(H) \text{ and } y \prec x\}| \leq w.$$

□

Corollary 1. *Let ϕ be a QBF. Then ϕ has respectful tree-width w if and only if it has respectful induced width w.*

In a different setting, Chen and Dalmau [17] show that quantified constraint satisfac-

tion problems, which generalize QBFs to unbounded domains, are tractable if they have bounded respectful tree-width. We show here the corresponding result for Q-resolution: it is polynomially bounded for QBFs of bounded respectful tree-width.

Lemma 13. *Let ϕ be a false QBF sentence with n variables, m clauses and respectful tree-width w. Then, there is a weak Q-resolution refutation of ϕ of size $O(m + n \cdot 3^w)$.*

Proof. By Lemma 12, we have that ϕ has respectful induced width w. Let $(H, \prec)$ be a respectful induced graph of ϕ of width w. Let $Y := (R_1 y_1, \ldots, R_n y_n)$ be the sequence of variables of ϕ in order $\prec$ together with its quantifier in ϕ, and, for $i \in \{1, \ldots, n\}$, let $Y_i := (R_i y_i, \ldots, R_n y_n)$ be the i-th suffix of Y.

Since $(H, \prec)$ is respectful with the prefix of ϕ, by Claim 7 we have that $R_1 y_1 \ldots R_n y_n\,(\phi')$ is equivalent to ϕ. Moreover, since $\phi \equiv \square$, by Lemma 11 we have that $\phi'^{(Y)} \models \square$ and also that $\square \in \phi'^{(Y)}$. Then, the sequence $(\phi', \phi'^{(Y_n)}, \ldots, \phi'^{(Y_1)})$ makes a valid Q-resolution refutation of ϕ.

Finally, note that every $\phi'^{(Y_i)}$ has at most 3^w clauses not already in the sequence, since every variable is connected to at most w variables of $\phi'^{(Y_{i-1})}$, and there are a total of 3^w possible clauses that can be formed with w variables. Therefore, the size of the refutation is $O(m + n \cdot 3^w)$. □

3.7 Formulas with bounded respectful tree-width

In the previous section we have shown that false QBFs with bounded respectful tree-width have short Q-resolution refutations. In this section we introduce a family of formulas with this property and show some formulas that belong to this family and may have real-world applications.

3.7.1 QBFs with bounded number of variables

Let $x_1, \ldots, x_k$ be propositional variables. A k-QBF is defined recursively as follows:

1. any clause on variables $x_1, \ldots, x_k$ is a k-QBF,
2. if ϕ and ψ are k-QBFs, then $\phi \wedge \psi$ is a k-QBF,
3. if ϕ is a k-QBF, then $\exists x_i$ is a k-QBF, where $i \in \{1, \ldots, k\}$, and
4. if ϕ is a k-QBF, then $\forall x_i$ is a k-QBF, where $i \in \{1, \ldots, k\}$.

Notice that we allow a variable to be quantified more than once. The recursive construction of a k-QBF defines a (rooted) labeled tree, whose leaves are labeled with the clauses of the formula and whose internal vertices are either labeled with a $\wedge$ and have two children, or labeled with $\exists$ or $\forall$ and have a single child. For a k-QBF ϕ, we say that (T_ϕ, K_ϕ) is its *associated tree* in the sense described above, where T_ϕ is a tree of the indicated form and $K_\phi : V(T_\phi) \to \mathcal{C} \cup \{\wedge, \exists x_1, \ldots, \exists x_s, \forall x_1, \ldots, \forall x_s\}$ where $\mathcal{C}$ is the set of all clauses on the variables $x_1, \ldots, x_k$. We say ϕ is the *associated formula* of the pair (T_ϕ, K_ϕ).

This family of formulas is the propositional version of one introduced by Dalmau et al. in [20], extended by allowing universal quantification. Their framework allows, given a QBF, to rewrite it as a logically equivalent k-QBF. Here we want to achieve exactly the opposite: given a k-QBF, rewrite it as a logically equivalent QBF. To do so, given a k-QBF ϕ, consider the following rewriting rules:

1. **A-Rule**: Associativity of conjunction is applied to subformulas of ϕ.
2. **C-Rule**: Commutativity of conjunction is applied to subformulas of ϕ.
3. **$\exists$-Rule**: a subformula of ϕ of the form $(\psi \wedge (\exists x \theta))$ is replaced by the formula $(\exists x(\psi \wedge \theta))$, provided the variable x does not occur in ψ.
4. **$\forall$-Rule**: a subformula of ϕ of the form $(\psi \wedge (\forall x \theta))$ is replaced by the formula $(\forall x(\psi \wedge \theta))$, provided the variable x does not occur in ψ.
5. **R-$\exists$-Rule**: a subformula of ϕ of the form $(\exists x \psi)$ is replaced by the formula $(\exists y)\psi[x/y]$, where y does not occur in ψ and $\psi[x/y]$ is obtained from ψ by replacing all free occurrences of x in ψ by y.
6. **R-$\forall$-Rule**: a subformula of ϕ of the form $(\forall x \psi)$ is replaced by the formula $(\forall y)\psi[x/y]$, where y does not occur in ψ and $\psi[x/y]$ is obtained from ψ by replacing all free occurrences of x in ψ by y.

It is clear that the application of these rules preserves logical equivalence.

Note that every k-QBF of size s can be rewritten as a QBF as in (3.1) by the following steps: first, repeatedly apply the R-Rules with fresh variables $x_{k+1}, \ldots, x_s$ until no variable in the formula occurs quantified more than once. Second, repeatedly apply $\exists$-Rule and $\forall$-Rule, always on the outermost possible quantifier (and A-Rule and C-Rule, as necessary, to reorder the conjunctions in order to apply the other rules) until we obtain the form (3.1). It is clear that this can be done in a number of steps polynomial in s and that the resulting formula ϕ^R will be over the variables $x_1, \ldots, x_s$. Also, let K_ϕ^R be equal to K_ϕ but appropiately applying the renaming performed by the R-Rules on the clauses at the leaves.

For a tree T and $t \in V(T)$, let T^t be the subtree of T rooted at t. Let ϕ_t be the associated formula of (T^t_ϕ, K_ϕ) and let ϕ^R_t be the associated formula of (T^t_ϕ, K^R_ϕ). Now, define $L_\phi : V(T_\phi) \to \mathcal{P}(\{x_1, \ldots, x_s\})$ as

$$L_\phi(t) := \{x \mid x \text{ is free in the formula } \phi^R_t\}$$

for every $t \in V(T_\phi)$.

We prove the following claim:

Claim 8. *The pair (T_ϕ, L_ϕ) is a respectful tree decomposition of ϕ^R of width $k-1$.*

Proof. First, note that every clause of ϕ^R is precisely $K^R_\phi(t)$ for some leaf t of T_ϕ. Since $t = T^t_\phi$, the associated formula of (t, K^R_ϕ) is precisely the clause $K^R_\phi(t)$, and therefore, all of its variables are free in it. Therefore, for every clause C of ϕ^R, there is a leaf t of T_ϕ for which $L(t) = \text{var}(C)$, and also, $\bigcup_{t \in V(T_\phi)} L(t) = \text{var}(\phi'^R)$. Second, for $x \in \text{var}(\phi^R)$, let t_x be the (unique) child of the (unique) vertex of t of T_ϕ such that $K^R_\phi(t)$ is of the form Qx for $Q \in \{\exists, \forall\}$, and the root of T if there is none. Note that $x \in L(t)$ if and only if both $t \in V(T^{t_x})$ and for some leaf t' of T^t, we have $x \in L(t')$. Then, the subgraph of T_ϕ induced by $\{t \in \text{var}(\phi^R) \mid x \in L(t)\}$ is precisely the union of the (unique) paths from t_x to a leaf t' of T_ϕ such that $x \in L(t')$. Since all of these paths have their beginning at t_x, this is a connected subtree. Finally, note that for every $t \in T_\phi$, we have $|L(t)| \leq k$ since, in case $|L(t)| > k$ for some $t \in T_\phi$, that would imply that ϕ^R_t has more than k free variables, which is not possible, since, before renaming, ϕ (and therefore, ϕ_t) has only k variables in total. □

Corollary 2. *Every k-QBF is logically equivalent to a QBF with respectful tree-width $k-1$.*

Note that, together with Lemma 13, this gives that, as long as $k \leq c \cdot \log n$ for some constant c, for every false k-QBF we can obtain a logically equivalent QBF and a short Q-resolution refutation of the second. Next, we see examples of k-QBFs for which this result may be useful.

3.7.2 Bounded model checking

An alternating finite state machine is a nondeterministic state machine whose states are of two types: ∃-states or ∀-states. On a given input, the machine accepts if there is at least one transition leaving every ∃-state such that for every transition leaving every ∀-state, the machine ends up reaching an accepting state. Consider an alternating finite state machine

with n states and with m transitions leaving each state, in which every transition leaving an $\exists$-state leads to a $\forall$-state and viceversa. States and transitions leaving each state are labeled with a number encoded in binary as $\bar{x} = x_1, \ldots, x_{|n|}$ and $\bar{y} = y_1, \ldots, y_{|m|}$, respectively. Also, define the ternary relation R as $R(\bar{x}, \bar{y}, \bar{x}')$ if, from state $\bar{x}$, using transition $\bar{y}$, we can reach state $\bar{x}'$ in a single step. Let $I(\bar{x})$ indicate that $\bar{x}$ is an initial state, and let $Z(\bar{x})$ indicate that $\bar{x}$ is a Z-state.

We want to obtain a proof of the following statement, common in the context of bounded model checking: no Z-state is accessible from an I-state in at most ℓ steps. We call this statement $P_{\leq \ell}$. Note that this problem can be reduced to obtaining a proof of P_t for every $0 \leq t \leq \ell$. We focus on this last problem, which is equivalent to finding a refutation of $\neg P_t$, which is equal to $\exists \bar{x}(I(\bar{x}) \wedge \psi_t(\bar{x}))$ where

$$\begin{aligned}
&\psi_0(\bar{x}) = Z(\bar{x}), \\
&\psi_{i+1}(\bar{x}) = \exists \bar{y} \exists \bar{x}'(R(\bar{x}, \bar{y}, \bar{x}') \wedge \psi_i(\bar{x}')) && \text{for odd } i \geq 0, \\
&\psi_{i+1}(\bar{x}') = \forall \bar{y} \exists \bar{x}(R(\bar{x}', \bar{y}, \bar{x}) \wedge \psi_i(\bar{x})) && \text{for even } i \geq 0.
\end{aligned}$$

Observe that, by writing $I(\bar{x})$, $Z(\bar{x})$ and $R(\bar{x}, \bar{y}, \bar{x}')$ as CNFs, the formula $\neg P_t$ that we obtain is a $(2|n| + |m|)$-QBF. Therefore, if P_t is true, we can obtain a Q-resolution refutation of a QBF expressing $\neg P_t$ of size exponential in $2|n| + |m|$, that is, polynomial in the number of states and the size of the formula.

By defining the formulas encoding $I(\bar{x})$, $Z(\bar{x})$ and $R(\bar{x}, \bar{y}, \bar{x}')$ appropiately, we can use this to model multiple real-world situations. We present a couple of examples:

Verification of software with human interaction In this case, the alternating finite state machine models the interaction between a user and a computer interface: $\exists$-states are those waiting for a response of the system and $\forall$-states are those waiting for a response from the user. The initial state is the initial configuration of the software, the Z-states are those in which the software crashes or reaches an undesired point. Finally, the relation R is defined by the work-flow of the program. We want to make sure that, from the initial state of the program, for every input of the user into the interface, there is a possible response of the program in such a way that the user cannot crash the system before ℓ interactions.

Two-player games by turns In this case, the alternating finite state machine models the strategies of the players: the $\exists$-states model the positions in which the first player has to move and the $\forall$-states model the positions in which his adversary has to move. The

initial state is the initial configuration of the game and the Z-state is a winning or losing configuration, depending on what we want to prove. The relation R defines the legal moves of the players. What we can prove here is that, starting with the initial configuration of the game, the first player cannot win the game (or lose it) before $\ell + 1$ rounds have been played.

Chapter 4

Definability and interpretability

In this chapter we show that the class of first-order principles whose propositional encoding has short R(const)-refutations is closed under definability. Using this result, we obtain a similar result for interpretability, for both the unary and the binary propositional encodings. As examples, we add several definitions and interpretations of well-known first-order principles into others and show a systematic technique to convert some Σ_1-definitions into quantifier-free interpretations.

4.1 Quantifier-free definitions

Let σ and τ be first-order vocabularies, split into σ_R and σ_F (τ_R and τ_F, respectively) for relation and function symbols, respectively. Recall that we assume that both vocabularies have at least one constant symbol denoted by 0.

Let Θ be a collection of flattened first-order formulas. A Θ-*translation from* σ *to* τ consists of Θ-formulas $\theta_R(x_1, \ldots, x_{r_R}; \bar{p})$ and $\theta_F(x_1, \ldots, x_{r_F}, y; \bar{p})$, where R and F range over σ_R and σ_F respectively. The variables in $\bar{p}$ are called *parameters*. We often omit writing down the parameters.

Let ϕ be a flattened formula over σ and let D be a translation from σ to τ. The *translation of* ϕ *through* D, denoted by $D(\phi)$, is the formula over τ that is obtained from ϕ as follows: replace every atom $R(x_1, \ldots, x_{r_R})$ for $R \in \sigma_R$ by $\theta_R(x_1, \ldots, x_{r_R})$ and every atom $F(x_1, \ldots, x_{r_F}) = y$ for $F \in \sigma_F$ by $\theta_F(x_1, \ldots, x_{r_F}, y)$. Note that this is well-defined because ϕ is flattened.

Let ϕ and ψ be flattened sentences over σ and τ, respectively. We say that D *defines* ϕ *in* ψ if the following conditions are satisfied. Let $\delta = \delta(\bar{p})$ be the formula $\bigwedge_{i \neq j} p_i \neq p_j$.

Then:

1. $\psi \models \delta \implies \forall \bar{x} \exists y(\theta_F(\bar{x}, y))$,
2. $\psi \models \delta \implies \forall \bar{x} \forall y \forall z(\theta_F(\bar{x}, y) \wedge \theta_F(\bar{x}, z) \implies y = z)$,
3. $\psi \models \delta \implies D(\phi)$.

for every $F \in \sigma_F$. The first two conditions ensure that θ_F is *functional* in the structures where ψ and δ are true (for any particular choice of parameters). Say that ϕ *is* Θ*-definable in* ψ, or that ψ Θ*-defines* ϕ, if there is a Θ-translation that defines ϕ in ψ.

Here we show an example of definability:

Example 1. The functional pigeonhole principle (FPHP_n^m) says that there is no injective function from $[m]$ to $[n]$ whenever $m > n$. In first order logic, $\text{FPHP}_n^{n^2}$ is expressed by the fact that the sentence

$$\forall x \forall x' \forall y \forall y' \forall t(F(x, y) = t \wedge F(x', y') = t \implies x = x' \wedge y = y').$$

has no models of cardinality n. Note that in order to represent n^2 in a universe of size n we use a binary function F,since then $\text{Dom}(F) = [n]^2$, and therefore, $|\text{Dom}(F)| = n^2$.

Similarly, $\text{FPHP}_n^{n^3}$ can be expressed by the fact that

$$\forall x \forall x' \forall y \forall y' \forall z \forall z' \forall t(G(x, y, z) = t \wedge G(x', y', z') = t \implies x = x' \wedge y = y' \wedge z = z').$$

has no models of cardinality n. Note that these two sentences are not standardized, but can be easily expressed as a standardized sentence by distributing over the conjunction at the right-hand side of the implication. From now on, to simplify, we will treat sentences like this as standardized sentences, even though they are not strictly of the appropiate form.

Define

$$\theta_F(x, y, t) := (G(x, y, y) = t),$$

and let D be the quantifier-free translation with no parameters defined by this θ-formula. Since G is a function in $\text{FPHP}_n^{n^3}$, it is clear that θ_F is functional. Also, it is clear that $D(\text{FPHP}_n^{n^2})$, which reads

$$\forall x \forall x' \forall y \forall y' \forall t(\theta_F(x, y, t) \wedge \theta_F(x', y', t) \implies x = x' \wedge y = y')$$

and is equal to

$$\forall x \forall x' \forall y \forall y' \forall t (G(x,y,y) = t \wedge G(x',y',y') = t \implies x = x' \wedge y = y')$$

is implied by $\mathrm{FPHP}_n^{n^3}$. Therefore, D defines $\mathrm{PHP}_n^{n^2}$ in $\mathrm{PHP}_n^{n^3}$.

Note that the translation D lacks a definition of $\theta_0(y)$. In our examples, we will understand that θ_0 is defined trivially as $\theta_0(y) := (y = 0)$ if it is not explicitly defined. To do the converse, we use a Σ_1-definition, i.e. a definition that allows existentially quantified θ-formulas. Let D be the following Σ_1-definition:

$$\theta_G(x,y,z,t) = \exists w (F(x,y) = w \wedge F(z,w) = t).$$

To see that this is a valid definition of $\mathrm{FPHP}_n^{n^3}$ into $\mathrm{FPHP}_n^{n^2}$, first note that θ_G is functional since F is functional and w is uniquely determined by F in the definition of θ_G. Moreover, $D(\mathrm{FPHP}_n^{n^3})$, which is equivalent to

$$\begin{aligned}\forall x \forall x' \forall y \forall y' \forall z \forall z' \forall t \forall w \forall w' (F(x,y) = w \wedge F(z,w) = t \wedge \\ F(x',y') = w' \wedge F(z',w') = t \implies x = x' \wedge y = y' \wedge z = z')\end{aligned}$$

is implied by $\mathrm{FPHP}_n^{n^2}$. In section 4.3.1 we will show how to avoid the use of these Σ_1-formulas by using the power of quantifier-free interpretations. ◇

We now show that, under the unary encoding, short refutations in R(const) transfer through quantifier-free definability. We will make use of the *upper-bound half* of Riis' Gap Theorem [47] for tree-like resolution as stated below:

Theorem 3 (Riis, 2001)**.** *Let $\phi_1(\bar{x}), \ldots, \phi_t(\bar{x})$ be standardized universal formulas with r free variables $\bar{x}$. If $\bigwedge_{i=1}^t \phi_i(\bar{x})$ is unsatisfiable, then for every natural n and every $\bar{a} \in [n]^r$ we have $\langle \phi_1[\bar{x}/\bar{a}] \rangle_n^{\mathrm{u}}, \ldots, \langle \phi_t[\bar{x}/\bar{a}] \rangle_n^{\mathrm{u}} \vdash_{1,*}^s \square$ for s polynomial in n.*

The following consequence of this result will be used several times:

Lemma 14. *Let ψ be a conjunction of standardized universal sentences and let $\phi(\bar{x}, \bar{y})$ be a flattened quantifier-free DNF-formula, where $\bar{x}$ has length r and $\bar{y}$ has length t. If $\psi \models \forall \bar{x} \exists \bar{y} \phi(\bar{x}, \bar{y})$, then for every natural n and every $\bar{a} \in [n]^r$ there is a natural k such that $\langle \psi \rangle_n^{\mathrm{u}} \vdash_k^s \bigvee_{\bar{b} \in [n]^t} \langle \phi[\bar{x}/\bar{a}, \bar{y}/\bar{b}] \rangle_n$ for s polynomial in n.*

Proof. Write ϕ as $\bigvee_{i\in[\ell]} t_i$ where each t_i is a term. Then, $\psi \wedge \bigwedge_{i\in[\ell]} \forall\bar{y}(\neg t_i)$ is unsatisfiable. Note that $\bar{x}$ is free in this unsatisfiable formula. Since ψ is a conjunction of standardized universal sentences and $\bigwedge_i \forall\bar{y}(\neg t_i)$ is a conjunction of standardized universal formulas with free variables, we apply Theorem 3. We get

$$\langle\psi\rangle^{\mathrm{u}}_n, \langle\forall\bar{y}(\neg t_1[\bar{x}/\bar{a}])\rangle^{\mathrm{u}}_n, \ldots, \langle\forall\bar{y}(\neg t_\ell[\bar{x}/\bar{a}])\rangle^{\mathrm{u}}_n \vdash^{\mathrm{poly}}_1 \square$$

for every $\bar{a} \in [n]^r$. Here, $\vdash^{\mathrm{poly}}$ refers to size polynomial in n. Now, since the vocabulary of both premises is the same, the functional clauses of both encoded formulas are already present in $\langle\psi\rangle^{\mathrm{u}}_n$. We get

$$\langle\psi\rangle^{\mathrm{u}}_n, \{\langle\neg t_1[\bar{x}/\bar{a},\bar{y}/\bar{b}]\rangle_n \mid \bar{b}\in[n]^t\}, \ldots, \{\langle\neg t_\ell[\bar{x}/\bar{a},\bar{y}/\bar{b}]\rangle_n \mid \bar{b}\in[n]^t\} \vdash^{\mathrm{poly}}_1 \square$$

Now, by Lemma 3 we obtain

$$\langle\psi\rangle^{\mathrm{u}}_n \vdash^{\mathrm{poly}}_k \bigvee_{\bar{b}\in[n]^t}\bigvee_{i\in[\ell]} \langle t_i[\bar{x}/\bar{a},\bar{y}/\bar{b}]\rangle_n = \bigvee_{\bar{b}\in[n]^t} \langle\phi[\bar{x}/\bar{a},\bar{y}/\bar{b}]\rangle_n$$

for some natural k. □

We now prove our main lemma concerning definability.

Lemma 15. *Let ϕ be a conjunction of standardized universal-existential sentences and ψ be a conjunction of standardized universal sentences. For all naturals t, $n \geq t$, s and k, if ϕ is quantifier-free definable in ψ with t parameters and $\langle\phi\rangle^{\mathrm{u}}_n \vdash^s_k \square$, then $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{s'}_{k'} \square$ for s' polynomial in s, n and 2^k and k' linear in k.*

Proof. Let σ be the vocabulary of ϕ split into σ_R and σ_F as usual. Let τ be the vocabulary of ψ. Let T be a quantifier-free translation that defines ϕ in ψ.

By the definition of definability we have $\psi \models T(\phi)$. The underlying idea of the proof is simple: from the short propositional refutation of the encoding of ϕ, get a propositional refutation of the encoding of $T(\phi)$, and combine it with a propositional proof of $\psi \models T(\phi)$ to get a propositional refutation of the encoding of ψ. However the details are not so easy because $T(\phi)$ does not have the right form and cannot be converted directly into sets of clauses. We need to do more work.

The conditions 1., 2. and 3. of the definition of definability hold for ψ and ϕ and the θ-formulas that compose T. The formulas for conditions 1. and 2. on the right of δ have the form $\forall\bar{x}\exists\bar{y}\gamma$, where γ is a single clause on the θ-formulas. For 3. the formula is a conjunction

of formulas of such a form. Since the θ-formulas are flattened quantifier-free, each can be written equivalently as a flattened quantifier-free DNF and a flattened quantifier-free CNF. If we replace each positive occurrence of a θ-formula by its DNF and each negative occurrence of a θ-formula by its CNF, what we obtain is a collection of formulas of the form $\forall\bar{x}\exists\bar{y}\gamma$ where γ is a quantifier-free flattened DNF formula. Note that $\bar{y}$ is trivial in 2.

For $q \in \{1, 2\}$ and $F \in \sigma_F$, let $\gamma_{q,F}$ be the quantifier-free part in the formula for condition q. Also, let $\gamma_{3,i}$ be the quantifier-free part of the i-th conjunct of ϕ. By Lemma 14,

1. $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{\mathrm{poly}}_{\mathrm{const}} \bigvee_{b\in[n]}\langle\gamma_{1,F}[\bar{x}/\bar{a}, y/b]\rangle_n$ for every $F \in \sigma_F$ and $\bar{a} \in [n]^{r_F}$,
2. $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{\mathrm{poly}}_{\mathrm{const}} \langle\gamma_{2,F}[\bar{x}/\bar{a}, y/b, z/c]\rangle_n$ for every $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$, and $b, c \in [n]$ with $b \neq c$,
3. $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{\mathrm{poly}}_{\mathrm{const}} \bigvee_{\bar{b}\in[n]^\ell}\langle\gamma_{3,i}[\bar{x}/\bar{a}, \bar{y}/\bar{b}]\rangle_n$ for every i and $\bar{a} \in [n]^k$.

Here, we omit mentioning a tuple $\bar{c}$ of pairwise distinct elements substituted for the parameters $\bar{p}$. Note that $\langle\delta(\bar{p}/\bar{c})\rangle_n$ evaluates to true for such a tuple $\bar{c}$ and to false for any tuple $\bar{c}$ with some components that are equal. Further, with $\vdash^{\mathrm{poly}}_{\mathrm{const}}$ we refer to a polynomial size (in n) R(k) proof for some constant k.

Let $\Gamma := \langle\phi\rangle^{\mathrm{u}}_n$. Our next goal is to show that there is a substitution into Γ that takes every clause in Γ into one of the DNF formulas on the right-hand sides above. The clauses in Γ can be split into four types; namely:

1. $C_{1,F}(\bar{a})$: long functional clause for $F \in \sigma_F$ and $\bar{a} \in [n]^{r_F}$,
2. $C_{2,F}(\bar{a}, b, c)$: short functional clause for $F \in \sigma_F$, $\bar{a} \in [n]^{r_F}$, and $b, c \in [n]$ with $b \neq c$,
3. $C_{3,i}(\bar{a})$: clause encoding the i-th conjunct of ϕ for tuple $\bar{a} \in [n]^k$.

For each clause $C \in \Gamma$, let C' be the result of replacing each positive occurrence of an atom by the propositional translation of the DNF formula of the corresponding θ-formula, and each negative occurrence of an atom by the propositional translation of the CNF formula of the corresponding θ-formula. Direct inspection shows that:

1. if $C = C_{1,F}(\bar{a})$, then $C' = \bigvee_{b\in[n]}\langle\gamma_{1,F}[\bar{x}/\bar{a}, y/b]\rangle_n$,
2. if $C = C_{2,F}(\bar{a}, b, c)$, then $C' = \langle\gamma_{2,F}[\bar{x}/\bar{a}, y/b, z/c]\rangle_n$.
3. if $C = C_{3,i}(\bar{a})$, then $C' = \bigvee_{\bar{b}\in[n]^\ell}\langle\gamma_{3,i}[\bar{x}/\bar{a}, \bar{y}/\bar{b}]\rangle_n$.

This covers all cases. Let $\Gamma' := \{C' \mid C \in \Gamma\}$ and note that since each C' is a DNF we have $D(\Gamma') = \Gamma'$. Therefore, applying Lemma 4 to the hypothesis that $\Gamma \vdash^s_k \Box$ we get

$$\Gamma' = D(\Gamma') \vdash^{s''}_{k''} D(\Box) = \Box$$

for s'' polynomial in s, n and 2^k and for k'' linear in k. Composing proofs we get $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{s'}_{k'} \Box$ for s' and k' as claimed. $\Box$

Remark 3. The previous lemma fails if one requires the implication 3. in the definition of definability to hold only with respect to finite structures. Indeed, with this weaker notion of definability $\phi = \bot$ would become quantifier-free definable in any ψ without finite models. Then $\Box \in \langle\phi\rangle^{\mathrm{u}}_n$, but the clauses $\langle\psi\rangle^{\mathrm{u}}_n$ may not admit polynomial size R(const) refutations (e.g. ψ could be the pigeonhole principle). $\diamond$

Remark 4. One might be tempted to try to prove Lemma 15 by the following sort of argument. Assume that ϕ has no finite models and has a quantifier-free definition T in ψ, say for simplicity, with empty parameter tuple $\bar{p}$. Further assume that $\Gamma := \langle\phi\rangle^{\mathrm{u}}_n$ has short R(k)-refutations. Then, using the refutation as a strategy, a prover playing against an adversary must be able to construct a partial assignment falsifying a clause of Γ by querying only some few k-terms on the atoms of Γ. If instead of querying these terms, prover's queries were about the truth value of the formulas in which the atoms are replaced by their definitions through T, then the position reached by the prover would hold a partial assignment falsifying a DNF from Γ', where Γ' is the propositional translation of $T(\phi)$ written as a set of DNF formulas. Now, since $\psi \models T(\phi)$, we have $\langle\psi\rangle^{\mathrm{u}}_n \models \Gamma'$, so some clause of $\langle\psi\rangle^{\mathrm{u}}_n$ must be false under any assignment extending the partial one constructed by the prover. It is thus tempting to define prover's position as winning in the game for $\langle\psi\rangle^{\mathrm{u}}_n$. The problem this has is that the falsified clause in $\langle\psi\rangle^{\mathrm{u}}_n$ may vary with the extension chosen. In other words, the partial assignment constructed has no need to falsify a particular clause of $\langle\psi\rangle^{\mathrm{u}}_n$. To find such a clause one would like to use a short proof of $\langle\psi\rangle^{\mathrm{u}}_n \models \Gamma'$ but that might not be available.

Note that the wrong argument above needs $\psi \models T(\phi)$ to hold only in the finite, and the previous remark showed that this cannot work. What makes our argument work is that it exploits the fact that $\psi \models T(\phi)$ is much stronger than $\langle\psi\rangle^{\mathrm{u}}_n \models \Gamma'$, which is equivalent to $\psi \models T(\phi)$ holding in the finite; so much stronger that the latter becomes provable through short proofs. $\diamond$

The following is a corollary of Lemma 15:

Corollary 3. *The class of conjunctions of standardized universal sentences whose unary encoding has polynomial-size* R(const)*-refutations is downward closed under quantifier-free definitions.*

That the unary encoding of ϕ has polynomial-size R(const)-refutations means that there is an R(k)-refutation of $\langle\phi\rangle^{\mathrm{u}}_n$ of size n^c for every n, for constants k and c.

Using the proof translations between encodings of section 2.6.3, we can also obtain the result analogous to Lemma 15 in the case of binary encoding. However, note that k' is no longer independent of n.

Lemma 16. *Let ϕ a be conjunction of standardized universal-existential sentences and ψ be a conjunction of standardized universal sentences. For all naturals t, $n \geq t$, s and k, if ϕ is quantifier-free definable in ψ with t parameters and $\langle\phi\rangle^{\mathrm{b}}_n \vdash^s_k \square$, then $\langle\psi\rangle^{\mathrm{b}}_n \vdash^{s'}_{k'} \square$ for s' polynomial in s and n^k and k' linear in $k \log n$.*

Proof. Since $\langle\phi\rangle^{\mathrm{b}}_n \vdash^s_k \square$, by Lemma 7, $\langle\phi\rangle^{\mathrm{u}}_n \vdash^{s'''}_k \square$ for s''' polynomial in s and n. Then, by Lemma 15, $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{s''}_{k''} \square$ for s'' polynomial in s, n and 2^k and for k'' linear in k. Finally, by Lemma 6, $\langle\psi\rangle^{\mathrm{b}}_n \vdash^{s'}_{k'} \square$ for s' polynomial in s and n^k and k' linear in $k \log n$. □

Just as before, from Lemma 16 we obtain the following corollary:

Corollary 4. *The class of conjunctions of standardized universal sentences whose binary encoding has quasipolynomial-size* R(log)*-refutations is downward closed under quantifier-free definitions.*

Here, that the binary encoding of ϕ has quasipolynomial-size R(log)-refutations means that there is an R($c \cdot (\log n)^2$)-refutation of $\langle\phi\rangle^{\mathrm{b}}_n$ of size $n^{c \cdot \log n}$ for every n, for a constant c.

4.2 Quantifier-free interpretations

Let ϕ be a flattened first-order formula over σ and let r be a natural with $r \geq 1$. Let σ_r be the vocabulary that includes a r-ary relation symbol U, a $2r$-ary relation symbol E, a $(r \cdot r_R)$-ary relation symbol $\bar{R}$ for every $R \in \sigma_R$ and a $(r \cdot (r_F + 1))$-ary relation symbol $\bar{F}$ for every $F \in \sigma_F$. By ϕ^{T_r} we denote the result of fixing for every variable x a new r-tuple $\bar{x} = (x_1, \ldots, x_r)$, and simultaneously replacing in ϕ the following:

$$
\begin{array}{lcl}
F(x_1, \ldots, x_{r_F}) = y & \mapsto & \bar{F}(\bar{x}_1, \ldots, \bar{x}_{r_F}, \bar{y}), \\
R(x_1, \ldots, x_{r_R}) & \mapsto & \bar{R}(\bar{x}_1, \ldots, \bar{x}_{r_R}), \\
x = y & \mapsto & E(\bar{x}, \bar{y}), \\
\forall x(\cdots) & \mapsto & \forall \bar{x}(U(\bar{x}) \implies \cdots), \\
\exists x(\cdots) & \mapsto & \exists \bar{x}(U(\bar{x}) \wedge \cdots)
\end{array}
$$

for every R in σ_R and $F \in \sigma_F$. Note that $\bar{F}$ is now a relation symbol (not function) and that we need ϕ to be flattened.

Remark 5. Note that, if ϕ is a conjunction of standardized universal formulas, then ϕ^{T_r} also is. However, if ϕ is a conjunction of standardized universal-existential formulas, then ϕ^{T_r} is not necessarily one, since $\exists x(C)$ becomes $\exists x(U(x) \wedge C)$ and $U(x) \wedge C$ is no longer a clause. Note that if the unary relation U is the trivial true relation, then $U(x) \wedge C$ is a clause and ϕ^{T_r} is a conjunction of standardized universal-existential formulas. $\diamond$

Let I be a Θ-translation from σ_r to τ and let ϕ and ψ be flattened sentences over σ and τ, respectively. We say that I *interprets* ϕ *in* ψ if the following conditions are satisfied. Let $\delta = \delta(\bar{p})$ be the formula $\bigwedge_{i \neq j} p_i \neq p_j$. Then:

1. $\psi \models \delta \implies \forall \bar{\bar{x}} \exists \bar{y}(\theta_{\bar{F}}(\bar{\bar{x}}, \bar{y}))$,
2. $\psi \models \delta \implies \forall \bar{\bar{x}} \forall \bar{y} \forall \bar{z}(\theta_{\bar{F}}(\bar{\bar{x}}, \bar{y}) \wedge \theta_{\bar{F}}(\bar{\bar{x}}, \bar{z}) \implies \theta_E(\bar{y}, \bar{z}))$,
3. $\psi \models \delta \implies \forall \bar{\bar{x}} \forall \bar{y} \forall \bar{z}(\theta_U(\bar{x}_1) \wedge \cdots \wedge \theta_U(\bar{x}_{r_F}) \wedge \theta_{\bar{F}}(\bar{\bar{x}}, \bar{y}) \implies \theta_U(\bar{z}))$,
4. $\psi \models \delta \implies \forall \bar{x}(\theta_E(\bar{x}, \bar{x}))$,
5. $\psi \models \delta \implies \forall \bar{x} \forall \bar{y}(\theta_E(\bar{x}, \bar{y}) \implies \theta_E(\bar{y}, \bar{x}))$,
6. $\psi \models \delta \implies \forall \bar{x} \forall \bar{y} \forall \bar{z}(\theta_E(\bar{x}, \bar{y}) \wedge \theta_E(\bar{y}, \bar{z}) \implies \theta_E(\bar{x}, \bar{z}))$,
7. $\psi \models \delta \implies \forall \bar{\bar{x}} \forall \bar{\bar{y}}(\theta_{\bar{R}}(\bar{\bar{x}}) \wedge \theta_E(\bar{x}_1, \bar{y}_1) \wedge \ldots \wedge \theta_E(\bar{x}_{r_R}, \bar{y}_{r_R}) \implies \theta_{\bar{R}}(\bar{\bar{y}}))$,
8. $\psi \models \delta \implies \forall \bar{\bar{x}} \forall \bar{\bar{y}} \forall \bar{z} \forall \bar{t}(\theta_{\bar{F}}(\bar{\bar{x}}, \bar{z}) \wedge \theta_{\bar{F}}(\bar{\bar{y}}, \bar{t}) \wedge \theta_E(\bar{x}_1, \bar{y}_1) \wedge \ldots \wedge \theta_E(\bar{x}_{r_F}, \bar{y}_{r_F}) \implies \theta_E(\bar{z}, \bar{t}))$.
9. $\psi \models \delta \implies I(\phi^{\mathrm{T}_r})$,

for every $R \in \sigma_R \cup \{U\}$, $F \in \sigma_F$. Say that ϕ *is* Θ*-(r-)interpretable in* ψ, or that ψ Θ*-(r-)interprets* ϕ, if there is a Θ-translation from σ^r to τ that interprets ϕ in ψ for some $r \geq 1$.

We would like to have a result similar to Lemma 15 also in this case. To do that, we define the following operation on formulas: Let ϕ be a flattened formula over σ. By ϕ^{I_r}, we denote the conjunction of ϕ^{T_r} with the following formulas:

1. $\forall \bar{\bar{x}} \exists \bar{y}(\bar{F}(\bar{\bar{x}}, \bar{y}))$,
2. $\forall \bar{\bar{x}} \forall \bar{y} \forall \bar{z}(\bar{F}(\bar{\bar{x}}, \bar{y}) \wedge \bar{F}(\bar{\bar{x}}, \bar{z}) \implies E(\bar{y}, \bar{z}))$,
3. $\forall \bar{\bar{x}} \forall \bar{y} \forall \bar{z}(U(\bar{x}_1) \wedge \cdots \wedge U(\bar{x}_{r_F}) \wedge \bar{F}(\bar{\bar{x}}, \bar{y}) \implies U(\bar{z}))$,
4. $\forall \bar{x}(E(\bar{x}, \bar{x}))$,

5. $\forall\bar{x}\forall\bar{y}(E(\bar{x},\bar{y}) \implies E(\bar{y},\bar{x}))$,
6. $\forall\bar{x}\forall\bar{y}\forall\bar{z}(E(\bar{x},\bar{y}) \wedge E(\bar{y},\bar{z}) \implies E(\bar{x},\bar{z}))$,
7. $\forall\bar{\bar{x}}\forall\bar{\bar{y}}(\bar{R}(\bar{\bar{x}}) \wedge E(\bar{x}_1,\bar{y}_1) \wedge \ldots \wedge E(\bar{x}_{r_R},\bar{y}_{r_R}) \implies \bar{R}(\bar{\bar{y}}))$,
8. $\forall\bar{\bar{x}}\forall\bar{\bar{y}}\forall\bar{\bar{z}}\forall\bar{\bar{t}}(\bar{F}(\bar{\bar{x}},\bar{\bar{z}}) \wedge \bar{F}(\bar{\bar{y}},\bar{\bar{t}}) \wedge E(\bar{x}_1,\bar{y}_1) \wedge \ldots \wedge E(\bar{x}_{r_F},\bar{y}_{r_F}) \implies E(\bar{\bar{z}},\bar{\bar{t}}))$.

for every $R \in \sigma_R \cup \{U\}$, $F \in \sigma_F$.

By the previous definition, the following fact is clear:

Fact 2. *Let ϕ and ψ be flattened formulas. Then, ϕ is quantifier-free interpretable in ψ if and only if ϕ^{I_r} is quantifier-free definable in ψ for some r.*

By Lemma 15 and the previous fact, we have the following result:

Theorem 4. *Let ϕ and ψ be conjunctions of standardized universal sentences. For all naturals r, t, $n \geq t$, s and k, if ϕ is quantifier-free r-interpretable in ψ with t parameters and $\langle\phi^{\mathrm{I}_r}\rangle^{\mathrm{u}}_n \vdash^s_k \square$, then $\langle\psi\rangle^{\mathrm{u}}_n \vdash^{s'}_{k'} \square$ for s' polynomial in s, n and 2^k and k' linear in k.*

Note that, to apply Lemma 15, we need ϕ^{I_r} to be a conjunction of standardized universal-existential sentences. By Remark 5, that forces us to require ϕ to be a conjunction of standardized universal sentences, whereas in Lemma 15, we could have a conjunction of standardized universal-existential sentences. Again by Remark 5 note that, if in the interpretation of ϕ in ψ the formula θ_U is the trivial true relation, then the result is valid also when ϕ is a conjunction of standardized universal-existential sentences.

Finally, by the same proof as in the definability case, we have the following result in the case of binary encoding:

Corollary 5. *Let ϕ and ψ be conjunctions of standardized universal sentences. For all naturals r, t and $n \geq t$, if ϕ is quantifier-free r-interpretable in ψ with t parameters and there is a* R(log)*-refutation of $\langle\phi^{\mathrm{I}_r}\rangle^{\mathrm{b}}_n$ of size quasipolynomial in n, then there is also a* R(log)*-refutation of $\langle\psi\rangle^{\mathrm{b}}_n$ of size quasipolynomial in n.*

Next, we show two examples of interpretations. First, we show one in which the flexibility offered by the parameters is key:

Example 2. Recall that the least number principle (LNP) is the sentence saying that $<$ is a strict partial order and that F is a function mapping each element x to some y with $y < x$. Also recall that the dense linear order principle (DLOP) is the sentence saying that $<$ is a strict linear order, and that G is a function mapping any two different elements x and y to

some z that lies strictly between x and y. Both LNP and DLOP are written as conjunctions of standardized universal sentences. We show that LNP is quantifier-free interpretable in DLOP. The interpretation has arity 1 and uses two parameter variables p and q. Define:

1. $\theta_U(x) := (p < q \wedge p < x \leq q) \vee (q < p \wedge q < x \leq p)$,
2. $\theta_E(x, y) := x = y$,
3. $\theta_{\bar{<}}(x, y) := x < y$,
4. $\theta_{\bar{0}}(y) := (p < q \wedge y = q) \vee (q < p \wedge y = p)$,
5. $\theta_{\bar{F}}(x, y) := (p < q \wedge G(p, x) = y) \vee (q < p \wedge G(q, x) = y$.

The correctness relies on the fact that the interpretation is required to work only when $p \neq q$. Note that this formula is flattened. $\diamond$

The second example is about relativized formulas. The *relativization* of a formula ϕ, written ϕ^{R} is a formula that is satisfiable in a domain D if and only if ϕ is satisfiable in some non-empty subdomain of D. The formula ϕ^{R} is defined as the particular case of ϕ^{I_r} in which $r = 1$ and the binary relation E is the equality relation.

Example 3. Let ϕ be a conjunction of standardized universal formulas. We show that ϕ and ϕ^{R} are mutually interpretable by quantifier-free translations.

Let σ be the vocabulary of ϕ split into σ_R and σ_F as usual. Recall that we assume that $0 \in \sigma_F$. Define $\theta_U(x) := U(x)$, $\theta_E(x, y) := x = y$, $\theta_{\bar{R}}(\bar{x}) := R(\bar{x})$ for every $R \in \sigma_R$, and $\theta_{\bar{F}}(\bar{x}, y) := F(\bar{x}) = y$ for every $F \in \sigma_F$. It is straightforward to check that this is an interpretation of ϕ in ϕ^{R}.

In the reverse direction, let us first rename the relativizing predicate from ϕ^{R} to V to avoid the overloading of names. Then define $\theta_U(x) := x = x$, $\theta_E(x, y) := x = y$, $\theta_{\bar{V}}(x) := x = x$, $\theta_{\bar{R}}(\bar{x}) := R(\bar{x})$ for every $R \in \sigma_R$, and $\theta_{\bar{F}}(\bar{x}, y) := F(\bar{x}) = y$ for every $F \in \sigma_F$. It is straightforward to check that this is an interpretation of ϕ^{R} in ϕ. $\diamond$

Using the previous example and as an application of Theorem 4, we show that the class of relativized formulas whose propositional translations have short R(const) refutations is closed under quantifier-free interpretations.

Theorem 5. *Let ϕ^{R} and ψ^{R} be conjunctions of relativized, standardized universal sentences. For all naturals t, $n \geq t$, s and k, if ϕ^{R} is quantifier-free interpretable in ψ^{R} with t parameters and $\langle \phi^{\mathrm{R}} \rangle^{\mathrm{u}}_n \vdash^s_k \square$, then $\langle \psi^{\mathrm{R}} \rangle^{\mathrm{u}}_n \vdash^{s'}_{k'} \square$ for s' polynomial in s, n and 2^k and k' linear in k.*

Proof. Assume $\langle \phi^{\mathrm{R}} \rangle^{\mathrm{u}}_n \vdash^s_k \Box$. First note that, by the interpretation in the previous example and transitivity of interpretability, ϕ is quantifier-free interpretable in $\phi^{\mathrm{R}^{\mathrm{R}}}$. Therefore, by Theorem 4, we have $\langle \phi^{\mathrm{R}^{\mathrm{R}}} \rangle^{\mathrm{u}}_n \vdash^{s''}_{k''} \Box$ for s'' polynomial in s, n and 2^k and for k'' linear in k. Now, if ϕ^{R} is quantifier-free interpretable in ψ^{R}, then again by Theorem 4 we get $\langle \psi^{\mathrm{R}} \rangle^{\mathrm{u}}_n \vdash^{s'}_{k'} \Box$ for s' and k' as claimed. $\Box$

4.3 Further examples

In example 1 we showed that $\mathrm{FPHP}^{n^2}_n$ is quantifier-free definable in $\mathrm{FPHP}^{n^3}_n$ and that $\mathrm{FPHP}^{n^3}_n$ is Σ_1-definable $\mathrm{FPHP}^{n^2}_n$. However, we do not have results in this thesis for Σ_1-definability. In this section we demonstrate the power of interpretations by proving that $\mathrm{FPHP}^{n^3}_n$ is interpretable $\mathrm{FPHP}^{n^2}_n$ through a quantifier-free translation. Then, we express other stronger versions of the functional pigeonhole principle and show they are mutually quantifier-free interpretable by using this same technique.

4.3.1 $\mathrm{FPHP}^{n^2}_n$ vs. $\mathrm{FPHP}^{n^3}_n$

Recall the first-order sentences used to express $\mathrm{FPHP}^{n^2}_n$ and $\mathrm{FPHP}^{n^3}_n$ and the Σ_1-definition of $\mathrm{FPHP}^{n^3}_n$ in $\mathrm{FPHP}^{n^2}_n$ of example 1. Here we use a quantifier-free binary interpretation that simulates the role of the existential quantification in the Σ_1-formulas. In all interpretations used in this section we will have that θ_U is the trivial true relation. From now on, we omit it from the definitions.

Let I be the binary interpretation defined by the translation

$$\theta_E(\bar{x}, \bar{y}) := (x_1 = y_1), \text{ and}$$
$$\theta_{\bar{G}}(\bar{x}, \bar{y}, \bar{z}, \bar{t}) := (F(x_1, y_1) = t_2 \wedge F(z_1, t_2) = t_1).$$

Of course, $\bar{x} = (x_1, x_2)$ and $\bar{y} = (y_1, y_2)$. First, $\theta_{\bar{G}}$ is functional since both t_1 and t_2 are uniquely determined by F, which is a function. Second, it is clear that θ_E is an equivalence relation. Third, it is easy to see that θ_E is a congruence of the function with graph $\theta_{\bar{G}}$, since two sets of E-equivalent arguments have the same first components, and thus, the same t_1 and t_2. Finally, we check that $I((\mathrm{FPHP}^{n^3}_n)^{\mathrm{T}_2})$, which reads

$$\forall\bar{x}\forall\bar{x}'\forall\bar{y}\forall\bar{y}'\forall\bar{z}\forall\bar{z}'\forall\bar{t}(\neg\theta_{\bar{G}}(\bar{x}, \bar{y}, \bar{z}, \bar{t}) \vee \neg\theta_{\bar{G}}(\bar{x}', \bar{y}', \bar{z}', \bar{t}) \vee (\theta_E(\bar{x}, \bar{x}') \wedge \theta_E(\bar{y}, \bar{y}') \wedge \theta_E(\bar{z}, \bar{z}')))$$

which is equal to

$$\forall\bar{x}\forall\bar{x}'\forall\bar{y}\forall\bar{y}'\forall\bar{z}\forall\bar{z}'\forall\bar{t}(F(x_1,y_1)\neq t_2\vee F(z_1,t_2)\neq t_1\vee F(x_1',y_1')\neq t_2\vee F(z_1',t_2)\neq t_1\vee (x_1=x_1'\wedge y_1=y_1'\wedge z_1=z_1'))$$

is implied by $\mathrm{FPHP}_n^{n^2}$. To argue why this is correct, assume $\mathrm{PHP}_n^{n^2}$ and let $\bar{x}$, $\bar{x}'$, $\bar{y}$, $\bar{y}'$, $\bar{z}$, $\bar{z}'$, $\bar{t}$ be such that $F(x_1,y_1)=t_2$, $F(z_1,t_2)=t_1$, $F(x_1',y_1')=t_2$, $F(z_1',t_2)=t_1$. We have to see that $x_1=x_1'$, $y_1=y_1'$ and $z_1=z_1'$. By the first and third equalities we have that $x_1=x_1'$ and that $y_1=y_1'$, and by the second and fourth we have that $z_1=z_1'$, as claimed. Therefore, I is a quantifier-free interpretation of $\mathrm{PHP}_n^{n^3}$ in $\mathrm{PHP}_n^{n^2}$.

4.3.2 FPHP_n^{2n} vs. FPHP_n^{3n}

FPHP_n^{2n} is expressed by the fact that the conjunction of the sentences

$$\begin{aligned}
&\forall x\forall x'\forall y(F(x)=y\wedge F(x')=y\implies x=x'),\\
&\forall x\forall x'\forall y(G(x)=y\wedge G(x')=y\implies x=x'),\text{ and}\\
&\forall x\forall x'\forall y(F(x)\neq y\vee G(x')\neq y).
\end{aligned}$$

has no models of cardinality n. Note that the first two equations ensure that F and G are injective functions, and the last one ensures that their ranges are disjoint. Also, note how $2n$ is represented in a universe of size n by tagging each $x\in[n]$ with one of two labels F and G. Similarly, FPHP_n^{3n} is expressed by the fact that the conjunction of the following sentences has no models of cardinality n.

$$\begin{aligned}
&\forall x\forall x'\forall y(F'(x)=y\wedge F'(x')=y\implies x=x'),\\
&\forall x\forall x'\forall y(G'(x)=y\wedge G'(x')=y\implies x=x'),\\
&\forall x\forall x'\forall y(H'(x)=y\wedge H'(x')=y\implies x=x'),\\
&\forall x\forall x'\forall y(F'(x)\neq y\vee G'(x')\neq y),\\
&\forall x\forall x'\forall y(F'(x)\neq y\vee H'(x')\neq y),\text{ and}\\
&\forall x\forall x'\forall y(G'(x)\neq y\vee H'(x')\neq y).
\end{aligned}$$

The following translation is a quantifier-free definition of FPHP_n^{2n} in FPHP_n^{3n}:

$$\theta_F(x,y):=(F'(x)=y),$$

$$\theta_G(x,y) := (G'(x) = y).$$

For the converse, one could consider the following Σ_1-translation D:

$$\begin{aligned}
&\theta_{F'}(x,y) := \exists w(F(x) = w \wedge F(w) = y),\\
&\theta_{G'}(x,y) := \exists w(F(x) = w \wedge G(w) = y), \text{ and}\\
&\theta_{H'}(x,y) := \exists w(G(x) = w \wedge G(w) = y).
\end{aligned}$$

This is a Σ_1-definition of FPHP_n^{3n} in FPHP_n^{2n}: all the θ-formulas are functional, in the sense that, first w and then y are uniquely determined by x (since F and G are functions). Also $D(\mathrm{FPHP}_n^{3n})$, which is equivalent to

$$\begin{aligned}
&\forall x\forall x'\forall y\forall w\forall w'(F(x) = w \wedge F(w) = y \wedge F(x') = w' \wedge F(w') = y \implies x = x'),\\
&\forall x\forall x'\forall y\forall w\forall w'(F(x) = w \wedge G(w) = y \wedge F(x') = w' \wedge G(w') = y \implies x = x'),\\
&\forall x\forall x'\forall y\forall w\forall w'(G(x) = w \wedge G(w) = y \wedge G(x') = w' \wedge G(w') = y \implies x = x'),\\
&\forall x\forall x'\forall y\forall w\forall w'(F(x) \neq w \vee F(w) \neq y \vee F(x') \neq w' \vee G(w') \neq y),\\
&\forall x\forall x'\forall y\forall w\forall w'(F(x) \neq w \vee F(w) \neq y \vee G(x') \neq w' \vee G(w') \neq y), \text{ and}\\
&\forall x\forall x'\forall y\forall w\forall w'(F(x) \neq w \vee G(w) \neq y \vee G(x') \neq w' \vee G(w') \neq y).
\end{aligned}$$

is implied by FPHP_n^{2n}, as it is easy to see.

Using the same technique as before, we show how to use a quantifier-free interpretation to reach the same goal. Let I be the binary interpretation defined by the translation:

$$\begin{aligned}
&\theta_E(\bar{x},\bar{y}) := (x_1 = y_1),\\
&\theta_{\bar{F}'}(\bar{x},\bar{y}) := (F(x_1) = y_2 \wedge F(y_2) = y_1),\\
&\theta_{\bar{G}'}(\bar{x},\bar{y}) := (F(x_1) = y_2 \wedge G(y_2) = y_1), \text{ and}\\
&\theta_{\bar{H}'}(\bar{x},\bar{y}) := (G(x_1) = y_2 \wedge G(y_2) = y_1).
\end{aligned}$$

We show that I is an interpretation of FPHP_n^{3n} in FPHP_n^{2n}. First, all of $\theta_{\bar{F}'}$, $\theta_{\bar{G}'}$ and $\theta_{\bar{H}'}$ are functional since both y_2 and y_1 are uniquely determined by F and G, which are functions. Second, it is clear that θ_E is an equivalence relation. Third, it is easy to see that θ_E is a congruence of the functions with graphs $\theta_{\bar{F}'}$, $\theta_{\bar{G}'}$ and $\theta_{\bar{H}'}$, since E-equivalent arguments will have the same first component, and thus, the same y_2 and y_1, as F and G are functions.

Finally, it is easy to check that $I((\mathrm{FPHP}_n^{3n})^{\mathrm{T}_2})$, which is the conjunction of

$$\begin{aligned}
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1)\neq y_2\vee F(y_2)\neq y_1\vee F(x_1')\neq y_2\vee F(y_2)\neq y_1\vee x_1=x_1'),\\
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1)\neq y_2\vee G(y_2)\neq y_1\vee F(x_1')\neq y_2\vee G(y_2)\neq y_1\vee x_1=x_1'),\\
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(G(x_1)\neq y_2\vee G(y_2)\neq y_1\vee G(x_1')\neq y_2\vee G(y_2)\neq y_1\vee x_1=x_1'),\\
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1)\neq y_2\vee F(y_2)\neq y_1\vee F(x_1)\neq y_2\vee G(y_2)\neq y_1),\\
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1)\neq y_2\vee F(y_2)\neq y_1\vee G(x_1)\neq y_2\vee G(y_2)\neq y_1),\text{ and}\\
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1)\neq y_2\vee G(y_2)\neq y_1,\vee G(x_1)\neq y_2\vee G(y_2)\neq y_1).
\end{aligned}$$

is implied by FPHP_n^{2n}. Therefore, I is a quantifier-free interpretation of $FPHP_n^{3n}$ in FPHP_n^{2n}.

4.3.3 FPHP_n^{n+1} vs. FPHP_n^{n+2}

We can express FPHP_n^{n+1} as the fact that the conjunction of sentences

$$\begin{aligned}
&\forall x\forall x'\forall y(F(x)=y\wedge F(x')=y\implies x=x'),\text{ and}\\
&\forall x\forall y(F(x)=y\implies 0\neq y).
\end{aligned}$$

has no models of cardinality $n+1$. Note that the first sentence ensures that F is injective and the second ensures that the range of F does not contain one of the elements of the universe.

Similarly, FPHP_n^{n+2} is expressed by the fact that the conjunction of

$$\begin{aligned}
&\forall x\forall x'\forall y(G(x)=y\wedge G(x')=y\implies x=x'),\\
&\forall x\forall y(G(x)=y\implies 0\neq y),\\
&\forall x\forall y(G(x)=y\implies 1\neq y),\text{ and}\\
&\forall x(0=x\implies 1\neq x).
\end{aligned}$$

has no models of cardinality $n+2$. It is easy to see that the translation

$$\begin{aligned}
&\theta_F(x,y):=(G(x)=y),\\
&\theta_0(y):=(0=y)
\end{aligned}$$

is a quantifier-free definition of FPHP_n^{n+1} in FPHP_n^{n+2}. For the converse, consider the fol-

lowing Σ_1-translation D:

$$\begin{aligned}
&\theta_G(x,y) := \exists z(F(x) = z \wedge F(z) = y),\\
&\theta_0(y) := (0 = y),\\
&\theta_1(y) := \exists z(0 = z \wedge F(z) = y).
\end{aligned}$$

It is clear that all of the θ-formulas are functional. Also, it is easy to see that $D(\mathrm{FPHP}_n^{n+2})$, which is equivalent to

$$\begin{aligned}
&\forall x\forall x'\forall y\forall z\forall z'(F(x) = z \wedge F(z) = y \wedge F(x') = z' \wedge F(z') = y \implies x = x'),\\
&\forall x\forall y\forall z(F(x) = z \wedge F(z) = y \implies 0 \neq y),\\
&\forall x\forall y\forall z\forall z'(F(x) = z \wedge F(z) = y \implies (0 \neq z' \vee F(z') \neq y)),\\
&\forall x\forall z(0 = x \implies (0 \neq z \vee F(z) \neq x)).
\end{aligned}$$

is implied by FPHP_n^{n+1}. Therefore D is a Σ_1-definition of FPHP_n^{n+2} in FPHP_n^{n+1}.

Again, we turn this definition into a quantifier-free interpretation using the same technique as before. Let I be the quantifier-free binary interpretation defined by the translation

$$\begin{aligned}
&\theta_E(\bar{x},\bar{y}) := (x_1 = y_1),\\
&\theta_{\bar{G}}(\bar{x},\bar{y}) := (F(x_1) = y_2 \wedge F(y_2) = y_1),\\
&\theta_{\bar{0}}(\bar{y}) := (0 = y_1),\\
&\theta_{\bar{1}}(\bar{y}) := (0 = y_2 \wedge F(y_2) = y_1).
\end{aligned}$$

It is easy to see that all of them are functional and that θ_E is an equivalent relation. Also, θ_E is a congruence of the functions with graph $\theta_{\bar{G}}$, $\theta_{\bar{0}}$ and $\theta_{\bar{1}}$. Moreover, $I((\mathrm{FPHP}_n^{n+2})^{\mathrm{T}_2})$, which reads

$$\begin{aligned}
&\forall\bar{x}\forall\bar{x}'\forall\bar{y}(F(x_1) \neq y_2 \vee F(y_2) \neq y_1 \vee F(x_1') \neq y_2 \vee F(y_2) \neq y_1 \vee x_1 = x_1'),\\
&\forall\bar{x}\forall\bar{y}(F(x_1) \neq y_2 \vee F(y_2) \neq y_1 \vee 0 \neq y_1),\\
&\forall\bar{x}\forall\bar{y}(F(x_1) \neq y_2 \vee F(y_2) \neq y_1 \vee 0 \neq y_2 \vee F(y_2) \neq y_1),\\
&\forall\bar{x}(0 \neq x_1 \vee 0 \neq x_2 \vee F(x_2) \neq x_1).
\end{aligned}$$

is implied by FPHP_n^{n+1}.

Remark 6. Note that the three examples on FPHP_n^m that have been presented (n^2 vs. n^3,

$2n$ vs. $3n$ and $n+1$ vs. $n+2$) can be easily generalized to n^r vs. n^s, rn vs. sn and $n+r$ vs. $n+s$ for all naturals r, s with $r < s$ by the same sort of arguments. $\diamond$

4.3.4 FPHP_n^{2n} vs. $\mathrm{FPHP}_n^{n^2,2n}$

In a previous example we already introduced a first-order sentence that expresses FPHP_n^{2n} by the fact that it has no models of cardinality n. The principle $\mathrm{FPHP}_n^{n^2,2n}$ says that there is no injective function from R to $[n]$, where R is a subset of $[n^2]$ of size at least $2n$. The conjunction of the following sentences expresses $\mathrm{FPHP}_n^{n^2,2n}$ by the fact that it has no models of cardinality n.

$$\begin{aligned}
&\forall x\forall x'\forall y\forall z(A_0(x) = y \wedge A_1(x) = z \wedge A_0(x') = y \wedge A_1(x') = z \implies x = x'),\\
&\forall x\forall x'\forall y\forall z(B_0(x) = y \wedge B_1(x) = z \wedge B_0(x') = z \wedge B_1(x') = y \implies x = x'),\\
&\forall x\forall x'\forall y\forall z(A_0(x) \neq y \vee A_1(x) \neq z \vee B_0(x') \neq y \vee B_1(x') \neq z),\\
&\forall x\forall y\forall z(A_0(x) = y \wedge A_1(x) = z \implies R(y,z)),\\
&\forall x\forall y\forall z(B_0(x) = y \wedge B_1(x) = z \implies R(y,z)),\\
&\forall x\forall y\forall x'\forall y'\forall z(R(x,y) \wedge R(x',y') \wedge P(x,y) = z \wedge P(x',y') = z \implies x = x' \wedge y = y').
\end{aligned}$$

Note that the first three formulas ensure that the functions $x \mapsto (A_0(x), A_1(x))$ and $x \mapsto (B_0(x), B_1(x))$ are injective and have disjoint ranges. We are interested in this specific principle, as it is central to the main result of Chapter 5.

We start by showing that there is a Σ_1-definition of FPHP_n^{2n} in $\mathrm{FPHP}_n^{n^2,2n}$. Consider the following Σ_1-translation D:

$$\begin{aligned}
\theta_F(x,y) &:= \exists w\exists z(A_0(x) = w \wedge A_1(x) = z \wedge P(w,z) = y),\\
\theta_G(x,y) &:= \exists w\exists z(B_0(x) = w \wedge B_1(x) = z \wedge P(w,z) = y).
\end{aligned}$$

Since A_0, A_1, etc. are functions, w and z are uniquely determined by x, and y is uniquely determined by w and z. Therefore, both θ-formulas are functional. Also, $D(\mathrm{FPHP}_n^{2n})$, which is equivalent to

$$\begin{aligned}
\forall x\forall x'\forall y\forall w\forall z\forall w'\forall z'(&A_0(x) = w \wedge A_1(x) = z \wedge P(w,z) = y \wedge\\
&A_0(x') = w' \wedge A_1(x') = z' \wedge P(w',z') = y \implies x = x'),\\
\forall x\forall x'\forall y\forall w\forall z\forall w'\forall z'(&B_0(x) = w \wedge B_1(x) = z \wedge P(w,z) = y \wedge
\end{aligned}$$

$$B_0(x') = w' \wedge B_1(x') = z' \wedge P(w', z') = y \implies x = x'),$$
$$\forall x \forall x' \forall y \forall w \forall z \forall w' \forall z' (A_0(x) \neq w \vee A_1(x) \neq z \vee P(w, z) \neq y \vee$$
$$B_0(x') \neq w' \vee B_1(x') \neq z' \vee P(w', z'))$$

is implied by $\mathrm{FPHP}_n^{n^2,2n}$.

Again, we can turn this into a quantifier-free interpretation. Let I be a quantifier-free ternary interpretation defined by the following translation:

$$\theta_E(\bar{x}, \bar{y}) := (x_1 = y_1),$$
$$\theta_{\bar{F}}(\bar{x}, \bar{y}) := (A_0(x_1) = y_2 \wedge A_1(x_1) = y_3 \wedge P(y_2, y_3) = y_1),$$
$$\theta_{\bar{G}}(\bar{x}, \bar{y}) := (B_0(x_1) = y_2 \wedge B_1(x_1) = y_3 \wedge P(y_2, y_3) = y_1).$$

It is clear that all of the θ-formulas are functional and that θ_E is an equivalence relation. Also, θ_E is a congruence of the functions with graphs $\theta_{\bar{F}}$ and $\theta_{\bar{G}}$. Moreover, $I((\mathrm{FPHP}_n^{2n})^{\mathrm{T}_3})$ reads

$$\forall \bar{x} \forall \bar{x}' \forall \bar{y} (A_0(x_1) \neq y_2 \vee A_1(x_1) \neq y_3 \vee P(y_2, y_3) \neq y_1 \vee$$
$$A_0(x_1') \neq y_2 \vee A_1(x_1') \neq y_3 \vee P(y_2, y_3) \neq y_1 \vee x_1 = x_1'),$$
$$\forall \bar{x} \forall \bar{x}' \forall \bar{y} (B_0(x_1) \neq y_2 \vee B_1(x_1) \neq y_3 \vee P(y_2, y_3) \neq y_1 \vee$$
$$B_0(x_1') \neq y_2 \vee B_1(x_1') \neq y_3 \vee P(y_2, y_3) \neq y_1 \vee x_1 = x_1'),$$
$$\forall \bar{x} \forall \bar{x}' \forall \bar{y} (A_0(x_1) \neq y_2 \vee A_1(x_1) \neq y_3 \vee P(y_2, y_3) \neq y_1 \vee$$
$$B_0(x_1') \neq y_2 \vee B_1(x_1') \neq y_3 \vee P(y_2, y_3) \neq y_1).$$

and is implied by $\mathrm{FPHP}_n^{n^2,2n}$. Therefore, I is a quantifier-free definition through quotients of FPHP_n^{2n} in $\mathrm{FPHP}_n^{n^2,2n}$.

4.3.5 Non-functional pigeonhole principles

All the examples in this section have been about functional pigeonhole principles, in which pigeons cannot split into multiple holes. In the first three cases we could obtain their non-functional counterparts just by substituting every function atom $F(\bar{x}) = y$ by a relation atom $F(\bar{x}, y)$ and adding only one of the two functional clauses for F, namely

$$\forall \bar{x} \exists y (F(\bar{x}, y))$$

for every pigeon function F.

In the case of $\mathrm{FPHP}_n^{n^2,2n}$, we can obtain its non-functional counterpart $\mathrm{PHP}_n^{n^2,2n}$ by substituting every atom $P(x,y)=z$ by a relation atom $P(x,y,z)$ and adding a relativized long functional clause. This is:

$$\forall x\forall x'\forall y\forall z(A_0(x)=y\wedge A_1(x)=z\wedge A_0(x')=y\wedge A_1(x')=z \implies x=x'),$$
$$\forall x\forall x'\forall y\forall z(B_0(x)=y\wedge B_1(x)=z\wedge B_0(x')=z\wedge B_1(x')=y \implies x=x'),$$
$$\forall x\forall x'\forall y\forall z(A_0(x)\neq y\vee A_1(x)\neq z\vee B_0(x')\neq y\vee B_1(x')\neq z),$$
$$\forall x\forall y\forall z(A_0(x)=y\wedge A_1(x)=z \implies R(y,z)),$$
$$\forall x\forall y\forall z(B_0(x)=y\wedge B_1(x)=z \implies R(y,z)),$$
$$\forall x\forall y\forall x'\forall y'\forall z(R(x,y)\wedge R(x',y')\wedge P(x,y,z)\wedge P(x',y',z) \implies x=x'\wedge y=y'),$$
$$\forall x\forall y\exists z(R(x,y) \implies P(x,y,z)).$$

Chapter 5

Lower bounds for DNF-refutations

In this chapter we prove a lower-bound on the refutations of the unary translation of $\mathrm{PHP}_n^{n^2,2n}$, the first-order principle introduced in section 4.3.5. This translation has variables $P_{u,v}$ to encode the flights of the pigeons and additional propositional variables R_u for $u \in [n^2]$ intended to express that pigeon u decides to fly. It also has clauses

$$\begin{aligned} &\neg R_u \vee \neg R_{u'} \vee \neg P_{u,v} \vee \neg P_{u',v} && \text{for } u, u' \in [n^2] \text{ with } u \neq u' \text{ and } v \in [n] \\ &\neg R_u \vee \bigvee_{v \in [n]} P_{u,v} && \text{for } u \in [n^2], \end{aligned}$$

together with a set of *threshold* clauses that encode the rest of the first-order formula. For the purposes of this chapter we do not need to make them explicit. We call the conjunction of these clauses

$$\mathrm{TH}_{2n}(\bar{R}, \bar{X}),$$

a CNF-formula in the R_u-variables $\bar{R}$ and some auxiliary variables $\bar{X}$ that expresses that at least $2n$ pigeons decide to fly. More precisely, $\mathrm{TH}_{2n}(\bar{R}, \bar{X})$ is a polynomial-size (in n) set of clauses such that for every assignment α to the variables $\bar{R}$ the following holds: there exists an assignment ξ to the auxiliary variables $\bar{X}$ such that $\alpha \cup \xi$ satisfies $\mathrm{TH}_{2n}(\bar{R}, \bar{X})$ if and only if α sets at least $2n$ many variables in $\bar{R}$ to true. The following is the main result of this chapter:

Theorem 6. *For every real $\epsilon > 0$ and every sufficiently large n, every DNF-refutation of $\mathrm{PHP}_n^{n^2,2n}$ has size at least $2^{(\log n)^{3/2-\epsilon}}$.*

To prove this theorem we use a special class of expander bipartite graphs. First, we will define these graphs and prove their existence, and later we will prove the desired lower-bound by using them.

5.1 Resilient expanders

In this section we discuss the sort of expander graphs that we need. In short, these are unbalanced low-degree bipartite expanders that satisfy an additional robustness condition: for at least half the subsets of vertices of some fixed size on the right-hand side, the graph remains an expander if these vertices are removed. Let us note that a similar definition was implicit in [8] which was later revisited in [50]. However, both these concepts were very tied to their specific application to Proof Complexity. Here we provide a more systematic and general treatment.

5.1.1 Definition and some basic properties

Let $G = (U, V, E)$ be a bipartite graph with $|U| = t$ and $|V| = n$ where $t \geq n$. Let b be a positive real and let q and r be naturals such that $0 \leq q \leq n/(1+b)$ and $0 \leq r \leq n$. Recall that G is a (q, b)-expander if $|N_G(S)| \geq (1+b)|S|$ for every q-element subset $S \subseteq U$. We say that G is a (q, b, r)-*resilient expander* if for a random r-element subset $\mathbf{B} \subseteq V$ we have that $G \setminus \mathbf{B}$ is a (q, b)-expander with probability bigger than $1/2$. The choice of $1/2$ here is arbitrary; any constant in the open interval $(0, 1)$ would do. However, observe that if we were to require that $G \setminus \mathbf{B}$ is a (q, b)-expander with probability 1 over the choice of $\mathbf{B}$, then the minimum degree of G would have to exceed r. Later we will see that for the less demanding requirement of probability strictly smaller than 1 we can afford a much smaller degree.

A first property to note is that if G is a (q, b, r)-resilient expander, then $G \cap C$ is also a (q, b, r)-resilient expander for every $C \subseteq U$. In other words, the property is hereditary under taking subsets of the left-hand side. Similarly, if it is a (q, b, r)-resilient expander then it also is a (q', b', r')-resilient expander for all $q' \leq q$, all positive $b' \leq b$, and all $r' \leq r$. The next lemma proves the only non-trivial case of this statement.

Lemma 17. *If G is a (q, b, r)-resilient expander, then G is a (q, b, s)-resilient expander for all $s \leq r$.*

Proof. Fix $s \leq r$. Call a set $B \subseteq V$ good if $G \setminus B$ is a (q, b)-expander. Observe that any subset of a good set is good. Assume at least half the r-element subsets of V are good. Each good r-element set contains exactly $\binom{r}{s}$ many good s-element sets, and each such s-element set appears in at most $\binom{n-s}{r-s}$ many good r-element sets. Therefore, the number of good s-element sets is at least $\frac{1}{2}\binom{n}{r}\binom{r}{s}/\binom{n-s}{r-s}$. Expanding the binomials, one sees this is precisely $\frac{1}{2}\binom{n}{s}$. □

5.1.2 Existence

We prove that random bipartite graphs with the appropriate parameters are resilient expanders. For naturals t, n and d, let $\mathbf{G} = \mathbf{G}(t, n, d)$ be the random bipartite graph (U, V, E) with $U = [t]$ and $V = [n]$ defined by the following random experiment: for each $u \in U$ choose a d-element subset N_u of V uniformly and independently at random, and declare each $v \in N_u$ a neighbor of u.

Lemma 18. *Let ε and b be positive reals, let t, n, q, r and d be naturals such that $t \geq n > 1 + 2/\varepsilon$, $q \leq n/12(1+b)$, $r \leq n/12$, and $n \geq d \geq (\log t + (3+b)\log n)/(\log n - \log(3(1+b)q + 3r))$, and let $\mathbf{G} = \mathbf{G}(t, n, d)$. Then*

$$\Pr[\ \mathbf{G} \text{ is a } (q, b, r)\text{-resilient expander }] > 1 - \varepsilon.$$

Before we prove this, let us look at some special cases to illustrate the complicated expressions in the hypothesis. Think of ε and b as positive constants and think of all other parameters as functions of n. If $t = O(n)$, $q = \Omega(n)$ and $r = \Omega(n)$, then the required lower bound on the degree d is $O(\log n)$. On the other hand, if still $t = O(n)$ but $q = n^{1-\Omega(1)}$ and $r = n^{1-\Omega(1)}$, then the required lower bound on the degree is only $O(1)$. For our application we will have $t = 2n$, $q = n^{1-\Omega(1)}$ and $r = \Theta(n/\log n)$, in which case the required lower bound on the degree is $O(\log n/\log\log n)$.

To prove Lemma 18 we rely on the following probabilistic fact. Let X be a random variable that takes all of its values x with positive probability. Given an event $\mathcal{E}$, recall that $\Pr[\ \mathcal{E} \mid X\ ]$ is the random variable $f \circ X$ where f is the function defined by $f(x) = \Pr[\ \mathcal{E} \mid X = x\ ]$ for every value x of X.

Lemma 19. *Let p be a real such that $0 < p < 1$, let $\mathcal{E}$ be an event and let X be a random variable. Then*

$$\Pr[\ \Pr[\ \mathcal{E} \mid X\ ] > p\ ] \geq \frac{1}{1-p} \cdot (\Pr[\ \mathcal{E}\ ] - p)\,.$$

Proof. Since $\Pr[\ \mathcal{E} \mid X\ ]$ takes values in $[0, 1]$ we have

$$\mathbb{E}[\ \Pr[\ \mathcal{E} \mid X\ ]\ ] \leq \Pr[\ \Pr[\ \mathcal{E} \mid X\ ] > p\ ] \cdot 1 + (1 - \Pr[\ \Pr[\ \mathcal{E} \mid X\ ] > p\ ]) \cdot p.$$

On the other hand, direct calculation shows $\mathbb{E}[\ \Pr[\ \mathcal{E} \mid X\ ]\ ] = \Pr[\ \mathcal{E}\ ]$. This implies the lemma. □

Proof of Lemma 18. Let $\mathbf{B}$ be an r-element subset of V chosen uniformly at random and independently from $\mathbf{G}$. In the following we let B range over values of $\mathbf{B}$. Let $\mathcal{E}$ be the event that $\mathbf{G} \setminus \mathbf{B}$ is a (q, b)-expander. By Lemma 19 it suffices to show that

$$\Pr[\ \mathcal{E}\ ] > 1 - \frac{\varepsilon}{2}. \tag{5.1}$$

Fix B and let $\mathcal{E}^B$ denote the event that $\mathbf{G} \setminus B$ is a (q, b)-expander. Further, fix two sets $S \subseteq U$ and $T \subseteq V \setminus B$ of cardinalities $i \leq q$ and $j < (1+b)i$ respectively. Recall that $N_{\mathbf{G}}(S)$ denotes the neighbors of S in the random graph $\mathbf{G}$. Then

$$\Pr[\ N_{\mathbf{G}}(S) \subseteq T \cup B\ ] \leq \left(\frac{\binom{j+r}{d}}{\binom{n}{d}}\right)^i \leq \left(\frac{(j+r)e}{n}\right)^{di};$$

here we use $\binom{j+r}{d} \leq ((j+r)e/d)^d$ and $\binom{n}{d} \geq (n/d)^d$. By the union bound over (non-empty) $S \subseteq U$ and $T \subseteq V \setminus B$ of the appropriate cardinalities we have

$$\Pr[\ \overline{\mathcal{E}^B}\ ] \leq \sum_{i=1}^{q} \binom{t}{i} \sum_{j=1}^{\lfloor (1+b)i \rfloor} \binom{n}{j} \cdot \left(\frac{(j+r)e}{n}\right)^{di}. \tag{5.2}$$

The term $\binom{n}{j} \cdot ((j+r)e/n)^{di}$ in the internal sum in (5.2) is bounded by $n^j \cdot ((j+r)e/n)^{di}$, which is an increasing function of j. Plugging in the largest possible j and multiplying by the number of terms, the internal sum in (5.2) is at most

$$(1+b)i \cdot n^{(1+b)i} \cdot \left(\tfrac{(1+b)ie+re}{n}\right)^{di} \leq \left(n^{2+b} \cdot \left(\tfrac{3(1+b)q+3r}{n}\right)^d\right)^i.$$

Here we use $1 \leq i \leq q$ and $q \leq n/12(1+b)$ so that $(1+b)i \leq n$ and $(1+b)i \cdot n^{(1+b)i} \leq n^{(2+b)i}$. Crudely bounding $\binom{t}{i}$ by t^i, we conclude that (5.2) is bounded by

$$\sum_{i=1}^{q} \left(t \cdot n^{2+b} \cdot \left(\frac{3(1+b)q+3r}{n}\right)^d\right)^i.$$

From $q \leq n/12(1+b)$ and $r \leq n/12$ we conclude that the fraction is bounded by $1/2$ and hence is strictly smaller than 1. From $d \geq (\log t + (3+b)\log n)/(\log n - \log(3(1+b)q + 3r))$

we conclude that (5.2) is bounded by

$$\sum_{i=1}^{\infty} \left(\frac{1}{n}\right)^i = \frac{1}{n-1}.$$

At this point we proved that $\Pr[\ \overline{\mathcal{E}^B}\ ] \leq 1/(n-1)$ for every B. This implies (5.1), because

$$\Pr[\ \overline{\mathcal{E}}\ ] = \sum_B \Pr[\ \overline{\mathcal{E}^B} \text{ and } \mathbf{B} = B\ ] = \sum_B \Pr[\ \overline{\mathcal{E}^B}\ ] \cdot \Pr[\ \mathbf{B} = B\ ] \leq \frac{1}{n-1} < \frac{\varepsilon}{2}.$$

Here, the second displayed equality is due to the independence of the events $\overline{\mathcal{E}^B}$ and $\mathbf{B} = B$, and the last inequality is due to $n > 1 + 2/\varepsilon$. □

5.1.3 Left and right degrees

Besides being a resilient-expander, we often need our graph to have low right-degree. This is guaranteed in a random graph by the following easy calculation:

Lemma 20. *Let ε be a positive real, let t, n, d and d' be naturals satisfying $t \geq n \geq d$ and $n(tde/nd')^{d'} < \varepsilon$, and let $\mathbf{G} = \mathbf{G}(t, n, d)$. Then*

$$\Pr[\ \mathbf{G} \textit{ has right-degree smaller than } d'\ ] > 1 - \varepsilon.$$

Proof. For fixed vertices $u \in U$ and $v \in V$, the probability that (u, v) is an edge in $\mathbf{G}$ is $\binom{n-1}{d-1}/\binom{n}{d} = d/n$. Moreover, for fixed $v \in V$, these events are mutually independent as u ranges over U. By the union bound over all d'-element subsets of U, this means that the probability that the degree of v is at least d' is bounded by $\binom{t}{d'}(d/n)^{d'}$. By the union bound over v, the probability that the right-degree is at least d' is bounded by $n\binom{t}{d'}(d/n)^{d'}$. The lemma follows from the bound $\binom{t}{d'} \leq (te/d')^{d'}$ and the hypothesis that $n(tde/nd')^{d'} < \varepsilon$. □

As mentioned earlier, in our application of Lemma 18 we will have $b = O(1)$, $t = 2n$, $q = n^{1-\Omega(1)}$ and $r = \Theta(n/\log n)$, in which case the required lower bound on d is $O(\log n/\log\log n)$. Setting $d = \lceil \log n \rceil$ satisfies this lower bound and Lemma 20 gives right-degree $d' = O(\log n)$. Therefore, for the setting of parameters b, t, q and r of our interest, there exists a (q, b, r)-resilient expander with left-degree $O(\log n)$ and right-degree $O(\log n)$. Let us argue now that having a (q, b, r)-resilient expander with right-degree $O(\log n)$ but left-degree $o(\log n/\log\log n)$ is impossible.

Suppose G is an (t, n, d_L, d_R)-graph that is a (q, b, r)-resilient expander where b, t, q and r are as above and $d_R = O(\log n)$. Then there exist at least $t/(d_L \cdot d_R)$ vertices in U with pairwise disjoint neighborhoods in V. Let $\tilde{\mathbf{B}}$ be a random subset of V obtained by placing each vertex in it independently with probability r/n. For a fixed vertex $u \in U$, the probability that $\tilde{\mathbf{B}}$ contains all the neighbors of u is at least $(r/n)^{d_L}$. Moreover, these events are mutually independent for vertices from U that have pairwise disjoint neighborhoods in V. Therefore, the probability that $\tilde{\mathbf{B}}$ does not contain all the neighbors of any vertex in U is bounded by

$$\left(1 - \left(\frac{r}{n}\right)^{d_L}\right)^{\frac{t}{d_L \cdot d_R}} \leq \exp\left(-\left(\frac{r}{n}\right)^{d_L} \cdot \frac{t}{d_L \cdot d_R}\right).$$

The probability of this event for a random r-element subset $\mathbf{B} \subseteq V$ is at most a multiplicative factor $3\sqrt{r}$ bigger (see equation (5.5) in Section 5.4). Since G is a (q, b, r)-resilient expander, the probability of this event for $\mathbf{B}$ is at least $1/2$. But since $t \geq n$, $r = \Omega(n/\log n)$ and $d_R = O(\log n)$, this is possible only if d_L is $\Omega(\log n/\log\log n)$.

Now we are ready to prove Theorem 6 as outlined in the introduction.

5.2 Killing large conjunctions

Let t be a natural such that $n < t < m$. Let $\boldsymbol{\rho} = \boldsymbol{\rho}(t)$ be the random restriction[1] on the variables of $\mathrm{PHP}_n^{m,t}$ defined by the following random experiment:

1. choose a subset $\mathbf{A} \subseteq [m]$ uniformly at random among all t-element subsets of $[m]$.
2. let $\boldsymbol{\rho}$ be the restriction that, for every $u \in [m]$, sets R_u to 1 if $u \in \mathbf{A}$ and to 0 otherwise;
3. extend $\boldsymbol{\rho}$ to set the auxiliary variables $\bar{X}$ such that $\mathrm{TH}_t(\bar{R}, \bar{X})$ is satisfied;
4. extend $\boldsymbol{\rho}$ to set every $P_{u,v}$ with $u \in [m] \setminus \mathbf{A}$ and $v \in [n]$ to 1 independently with probability $1/2$ and to 0 otherwise.

Here, by a *pigeon variable* we mean a variable $P_{u,v}$ for $u \in [m]$ and $v \in [n]$; we say $P_{u,v}$ *mentions* pigeon u; a formula *mentions* a pigeon if so does some variable occuring in it. For later use, note that if ρ is a realization of $\boldsymbol{\rho}$ and A is the corresponding realization of $\mathbf{A}$, then $\mathrm{PHP}_n^{m,t} \restriction \rho$ and PHP_n^t are the same formula up to renaming of pigeons.

[1] Of course, by a random restriction we mean a random variable whose values are restrictions.

Lemma 21. *Let p be a natural such that $p < t$ and $p < m-t$, and T be a term that mentions at least p many pigeons. Then*

$$\Pr\left[\ T \restriction \boldsymbol{\rho} \neq 0\ \right] \leq \left(\frac{1}{2} + \frac{t}{m-p}\right)^p.$$

Proof. Choose p literals in T mentioning pairwise different pigeons. Let P be the set of pigeons mentioned by these literals, and for every $u \in P$ let ℓ_u be the literal chosen for pigeon u. Consider the events $\mathcal{E} :=$ "$\boldsymbol{\rho}(\ell_u) \neq 0$ for all $u \in P \setminus \mathbf{A}$", and $\mathcal{F}_i :=$ "$|P \setminus \mathbf{A}| = i$", where $i \in \{0, \ldots, p\}$. Note that $\Pr[\ T \restriction \boldsymbol{\rho} \neq 0\ ] \leq \Pr[\ \mathcal{E}\ ]$ and

$$\Pr[\ \mathcal{E}\ ] = \sum_{i=0}^{p} \Pr[\ \mathcal{E} \mid \mathcal{F}_i\ ] \cdot \Pr[\ \mathcal{F}_i\ ] = \sum_{i=0}^{p} \frac{1}{2^i} \cdot \frac{\binom{p}{i}\binom{m-p}{t-p+i}}{\binom{m}{t}}.$$

For naturals $m \geq k$ we write $m^{\underline{k}}$ for the falling factorial $m^{\underline{k}} := m \cdot (m-1) \cdots (m-k+1)$. Note that our assumptions on p ensure $m - p > t - p + i > 0$. Using $0 \leq i \leq p$ and noting $m^{\underline{p}} = m^{\underline{i}} \cdot (m-i)^{\underline{p-i}}$, we have

$$\frac{\binom{m-p}{t-p+i}}{\binom{m}{t}} = \frac{(m-t)^{\underline{i}}}{m^{\underline{i}}} \cdot \frac{t^{\underline{p-i}}}{(m-i)^{\underline{p-i}}} \leq \frac{t^{\underline{p-i}}}{(m-i)^{\underline{p-i}}} \leq \left(\frac{t}{m-p}\right)^{p-i}.$$

Replacing, and using the binomial formula, the probability we want is bounded by

$$\sum_{i=0}^{p} \binom{p}{i} \cdot \left(\frac{1}{2}\right)^i \cdot \left(\frac{t}{m-p}\right)^{p-i} = \left(\frac{1}{2} + \frac{t}{m-p}\right)^p.$$

□

Lemma 22. *Let p and s be naturals such that $s < p < t$, and T be a term that mentions at most p many pigeons. Then*

$$\Pr\left[\ T \restriction \boldsymbol{\rho} \text{ mentions more than } s \text{ many pigeons}\ \right] \leq \binom{p}{s+1}\left(\frac{t}{m}\right)^{s+1}.$$

Proof. For any $s+1$ pigeon variables in T mentioning pairwise different pigeons, the probability that they all remain unset by $\boldsymbol{\rho}$ is

$$\frac{\binom{m-s-1}{t-s-1}}{\binom{m}{t}} = \frac{t^{\underline{s+1}}}{m^{\underline{s+1}}} \leq \left(\frac{t}{m}\right)^{s+1}.$$

The claim thus follows by the union bound. □

5.3 Restriction to a graph and binary encoding

Let t be a natural such that $n < t < m$ and let $G = (U, V, E)$ be a bipartite graph with $U = [t]$ and $V = [n]$. Consider the following restriction θ_G: it sets every variable $P_{u,v}$ to 0 if $(u, v) \notin E$ and is undefined on all other variables. Then $\mathrm{PHP}^t_n \restriction \theta_G$ is the CNF with clauses (1 and)

$$\begin{aligned} &P_{u,v_1} \vee \cdots \vee P_{u,v_d} && \text{for } u \in U \text{ with } N_G(u) = \{v_1, \ldots, v_d\}, \\ &\neg P_{u,v} \vee \neg P_{u',v} && \text{for } (u, v), (u', v) \in E \text{ with } u \neq u'. \end{aligned}$$

This formula is commonly denoted by $\mathrm{PHP}(G)$ (cf. [11, 49]).

Now assume that G is a (U, V, d_L, d_R)-graph with associated function ϕ_G. Write $\ell := |d_L - 1|$ for the length of the binary representation of the largest number in $[d_L]$. We introduce *binary pigeon variables* $P_{u;b}$ for $u \in U$ and $b \in [\ell]$. Again, we say that $P_{u;b}$ *mentions* pigeon u, and that a formula mentions the pigeons mentioned by some atom occuring in it. The intuitive meaning of a truth assignment to the binary pigeon variables is that pigeon u flies to hole $\phi_G(u, j)$, where j is the number whose binary representation is given by the truth values $P_{u;\ell-1}, \ldots, P_{u;0}$. The formula $\mathrm{BPHP}(G)$ has *domain clauses* and *collision clauses*:

$$\begin{aligned} &\textstyle\bigvee_{b\in[\ell]} \neg^{\mathrm{bit}(b,j)} P_{u;b} && \text{for } (u, j) \in U \times [2^\ell] \text{ s.t. } (u, j) \notin \mathrm{Dom}(\phi_G), \\ &\textstyle\bigvee_{b\in[\ell]} \neg^{\mathrm{bit}(b,j)} P_{u;b} \vee \bigvee_{b\in[\ell]} \neg^{\mathrm{bit}(b,j')} P_{u';b} && \text{for } (u, j) \in \mathrm{Dom}(\phi_G) \text{ and } (u', j') \in \mathrm{Dom}(\phi_G) \\ &&& \text{such that } u \neq u' \text{ and } \phi_G(u, j) = \phi_G(u', j'). \end{aligned}$$

Here, for a variable X we write $\neg^0 X := X$ and $\neg^1 X := \neg X$. The unary encoding $\mathrm{PHP}(G)$ and the binary encoding $\mathrm{BPHP}(G)$ are closely related. Indeed, the formula obtained from $\mathrm{PHP}(G)$ by substituting every variable $P_{u,v}$ by the term $\bigwedge_{b\in[\ell]} \neg^{1-\mathrm{bit}(b,j)} P_{u;b}$, where $j \in [2^\ell]$ is such that $\phi(u, j) = v$, is the conjunction of the collision clauses of $\mathrm{BPHP}(G)$ and *sporadic axioms*:

$$\textstyle\bigvee_{j\in J_G(u)} \bigwedge_{b\in[\ell]} \neg^{1-\mathrm{bit}(b,j)} P_{u;b} \qquad \text{for } u \in U \text{ with } J_G(u) := \{j \in [2^\ell] \mid (u, j) \in \mathrm{Dom}(\phi_G)\}.$$

The following lemma states that these sporadic axioms are redundant.

Lemma 23. *Every sporadic axiom has a DNF-proof from the domain clauses of* $\mathrm{BPHP}(G)$ *of size at most* $112 \cdot \ell^2 \cdot 8^\ell$ *and such that every term appearing in the proof mentions one pigeon.*

Proof. Observe that for $u \in U$ the formula

$$\bigvee_{j \in [2^\ell]} \bigwedge_{b \in [\ell]} \neg^{1-\mathrm{bit}(b,j)} P_{u;b}$$

is a tautology in the ℓ variables that mention pigeon u and has size $2^\ell \cdot (\ell + (\ell - 1)) + (2^\ell - 1) \leq \ell \cdot 2^{\ell+1}$. By Lemma 2 it has a DNF-proof of size at most $27 \cdot \ell^2 \cdot 2^{3\ell+2}$. The sporadic axiom is obtained from this tautology, written appropriately via one structural inference, by at most 2^ℓ many cuts with domain clauses of size at most 2ℓ each. This adds a factor of at most $(1 + 2^\ell) \cdot \ell \cdot 2^{\ell+1} \cdot 2\ell \leq \ell^2 \cdot 2^{2\ell+3}$ in size. In total, the proof has size at most $28 \cdot \ell^2 \cdot 2^{3\ell+2}$. □

5.4 Killing large disjunctions

Let t be a natural such that $n < t < m$ and let $G = (U, V, E)$ be a (t, n, d_L, d_R)-graph with associated function ϕ_G. Let r be a natural such that $1 \leq r \leq n$. We define a random restriction $\boldsymbol{\mu} = \boldsymbol{\mu}(G, r)$ on the variables of BPHP(G) by the following random experiment:

1. independently for every $v \in V$, choose a pigeon $\mathbf{Q}_v \in N_G(v)$ uniformly at random;
2. independently, choose a subset $\mathbf{B} \subseteq V$ uniformly at random among all r-element subsets of V;
3. let $\mathbf{M} := \{(\mathbf{Q}_v, v) \mid v \in \mathbf{B} \text{ and } \mathbf{Q}_v \neq \mathbf{Q}_{v'} \text{ for all } v' \in \mathbf{B} \setminus \{v\}\}$;
4. let $\boldsymbol{\mu}$ be the partial assignment associated with the matching $\mathbf{M}$.

Here, the partial assignment μ associated with a matching M of G is the assignment that, for every $(u, v) \in M$, sets $P_{u;b}$ to $\mathrm{bit}(b, j)$ for every $b \in [\ell]$, where j is such that $\phi_G(u, j) = v$, and leaves the other variables unset. Call a formula F *matching-satisfiable (in G)* if $F \restriction \mu = 1$ for some such partial assignment μ. Two formulas F and F' are *very disjoint (in G)* if $N_G(P)$ and $N_G(P')$ are disjoint, where $P \subseteq U$ and $P' \subseteq U$ are the sets of pigeons mentioned by F and F' respectively.

Lemma 24. *Let s and w be naturals such that $r \geq s \geq 1$ and $w \geq 1$. Further, let $F = \bigvee \Gamma$ where Γ contains at least w matching-satisfiable, pairwise very disjoint formulas each mentioning at most s pigeons. Then*

$$\Pr\left[\ F \restriction \boldsymbol{\mu} \neq 1\ \right] \leq 3\sqrt{r} \cdot \exp\left(-w \cdot \left(\frac{r}{d_R \cdot n}\right)^s \cdot \left(1 - \frac{r}{n}\right)^{d_L \cdot s}\right).$$

Proof. Define the random variables $\tilde{\mathbf{B}}$, $(\tilde{\mathbf{Q}}_v)_{v\in V}$, $\tilde{\mathbf{M}}$, $\tilde{\boldsymbol{\mu}}$ similarly as above but letting $\tilde{\mathbf{B}}$ be the random subset of V that contains every $v \in V$ independently with probability r/n. Let $\tilde{\mathbf{B}}_v$ denote the indicator variable for the event that $v \in \tilde{\mathbf{B}}$; note that the indicator variables are independent.

Fix a matching-satisfiable formula $F' \in \Gamma$ mentioning at most s pigeons. Choose a minimal matching M such that $F' \restriction \mu = 1$ where μ is the partial assignment associated with M. Write $M_0 := \mathrm{Dom}(M)$ and $M_1 := \mathrm{Im}(M)$. Then, by minimality of M, the domain M_0 is included in the set of pigeons $P \subseteq U$ mentioned by F'. Observe that the event that $F' \restriction \tilde{\boldsymbol{\mu}} = 1$ is implied by the event that $M \subseteq \tilde{\mathbf{M}}$. The latter event is implied by the intersection of

$$\mathcal{E}_1 := \text{“}\tilde{\mathbf{B}}_v = 1 \text{ for every } v \in M_1\text{”, and}$$
$$\mathcal{E}_2 := \text{“}\tilde{\mathbf{Q}}_v = M^{-1}(v) \text{ for every } v \in M_1\text{”}$$

and the event that $\tilde{\mathbf{Q}}_v \notin M_0$ for every $v \in \tilde{\mathbf{B}} \setminus M_1$. Thus it is implied by the intersection of $\mathcal{E}_1$, $\mathcal{E}_2$ and

$$\mathcal{E}_3 := \text{“}\tilde{\mathbf{B}}_v = 0 \text{ for every } v \in N_G(M_0) \setminus M_1\text{”}.$$

Now, the probability of $\mathcal{E}_1$ is at least $(r/n)^s$, the probability of $\mathcal{E}_2$ is at least $(1/d_R)^s$, and the probability of $\mathcal{E}_3$ is at least $(1-r/n)^{d_L\cdot s}$, the last because $|N_G(M_0) \setminus M_1| \leq d_L \cdot s$. These three events are independent. Hence

$$\Pr[\ \mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3\ ] \geq \left(\frac{r}{n}\right)^s \cdot \left(\frac{1}{d_R}\right)^s \cdot \left(1-\frac{r}{n}\right)^{d_L\cdot s} =: p.$$

The event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3$ depends only on the variables $\tilde{\mathbf{Q}}_v$ and $\tilde{\mathbf{B}}_v$ with $v \in N_G(M_0) \subseteq N_G(P)$. Thus, for a family of pairwise very disjoint formulas in Γ, the events are independent. Using the assumption of the lemma,

$$\Pr[\ F \restriction \tilde{\boldsymbol{\mu}} \neq 1\ ] \leq (1-p)^w \leq \exp(-wp). \tag{5.3}$$

Writing $B(m,q)(k) = \binom{m}{k} q^k (1-q)^{m-k}$ for the binomial distribution, we have

$$\Pr[\ F \restriction \tilde{\boldsymbol{\mu}} \neq 1\ ] \geq \Pr[\ |\tilde{\mathbf{B}}| = r\ ] \cdot \Pr[\ F \restriction \tilde{\boldsymbol{\mu}} \neq 1 \mid |\tilde{\mathbf{B}}| = r\ ] = B\left(n, \frac{r}{n}\right)(r) \cdot \Pr[\ F \restriction \boldsymbol{\mu}\ ]. \tag{5.4}$$

Using Robbins' [48] version of Stirling's formula, one can derive the following bound (see

also [12, p.4, Eq. (1.5)]):

$$B\left(n, \frac{r}{n}\right)(r) \geq \frac{1}{e^{1/6}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \left(\frac{n}{r(n-r)}\right)^{1/2} \geq \frac{1}{3}\frac{1}{\sqrt{r}}. \tag{5.5}$$

Combining (5.3), (5.4) and (5.5) yields the lemma. □

Again, we write $\ell := |d_L - 1|$.

Lemma 25. *Let s, s_0 and s_1 be naturals such that $s \geq 1$ and $s_0 \geq s_1 \geq 2\ell$. If* BPHP(G) *has a refutation of size at most s_0 such that every formula in it has the form $\bigvee \Gamma$ for some set[2] Γ of ℓ-CNFs each of which has size at most s_1 and mentions at most s pigeons, then* BPHP(G) *has a refutation of size at most $s_0 \cdot 729 \cdot s_1^4 \cdot 4^{s \cdot \ell}$ such that every formula in it has the form $\bigvee \Gamma$ for some set Γ of ℓ-CNFs each of which mentions at most s pigeons and is matching-satisfiable.*

Proof. Consider an (IOC)-application that introduces a ℓ-CNF F which is not matching-satisfiable. Let Δ be the set of clauses from BPHP(G) that mention exactly the at most s many pigeons mentioned by F. Then $\Delta \models \neg F$ because any assignment to the pigeon variables appearing in Δ satisfies every clause in Δ only if it is associated to some matching. Since there are at most $s \cdot \ell$ variables mentioning the s many pigeons in F, by Lemma 2 there is a proof of $\neg F$ from Δ of size at most $27 \cdot s_1^2 \cdot 2^{s \cdot \ell}$. Add this proof to the refutation; a structural inference on $\neg F$ and two cuts with the premisses of the (IOC) application derives the formula without F; this formula can be used to continue the proof. Proceed like this for all (IOC)-applications in the original proof. For each F eliminated in this way we added a proof of the ℓ-DNF $\neg F$ and this proof may contain new formulas which are not matching-satisfiable. But this proof can be chosen as an ℓ-DNF-proof where each ℓ-term mentions at most s many pigeons. As above, eliminate all the new ℓ-terms T which are not matching-satisfiable. The required proofs of the clause $\neg T$ can now be chosen as resolution proofs of size at most $27 \cdot (\ell + (\ell - 1))^2 \cdot 2^\ell$. In these resolution proofs all formulas are disjunctions of literals and every literal is matching-satisfiable – at least if every pigeon u has at least one neighbor in G. This we can assume because otherwise already the domain clauses for u are contradictory and have a resolution refutation of size at most $27 \cdot (\ell + (\ell - 1))^2 \cdot 2^\ell$. □

[2]We allow Γ to be a singleton and understand that $\bigvee\{F\} = F$.

5.5 Switching lemma

Associate with a DNF F the hypergraph $\mathcal{H}(F)$ which has as universe the set of variables of F and which has for each term T in F a hyperedge consisting in the variables of T. The *covering number* $\mathrm{cv}(F)$ of F is the size of the smallest hitting set of $\mathcal{H}(F)$.

Lemma 26. *Let F be a k-DNF in the binary pigeon variables. Then F contains at least $\frac{\mathrm{cv}(F)}{\ell \cdot k \cdot d_L \cdot d_R}$ many pairwise very disjoint terms.*

Proof. Let $\mathcal{T}$ be a maximal family of very disjoint terms in F. Let P be the set of pigeons mentioned by $\bigvee \mathcal{T}$. Then the set of all pigeon variables mentioning pigeons in $N_G(N_G(P))$ is a hitting set of $\mathcal{H}(F)$. Noting that $N_G(N_G(P))$ has cardinality at most $|\mathcal{T}| \cdot d_L \cdot d_R$ we get

$$\mathrm{cv}(F) \leq |N_G(N_G(P))| \leq |\mathcal{T}| \cdot \ell \cdot k \cdot d_L \cdot d_R$$

and the lemma follows. □

Interest in the covering number stems from the following lemma proved in by Segerlind, Buss and Impagliazzo [50] (see also the survey [49, Corollary 9.3]).

Lemma 27 ([50])**.** *Let $k, h, c > 0$ be naturals and $\gamma > 0$ a real. Let Γ be a set of k-DNFs that is closed under restrictions and assume that $\boldsymbol{\sigma}$ is a random restriction such that $\Pr[\ F \restriction \boldsymbol{\sigma} \neq 1\ ] \leq c \cdot 2^{-\gamma \cdot \mathrm{cv}(F)}$ for every $F \in \Gamma$. Then for every $F \in \Gamma$ we have*

$$\Pr\left[\ h(F \restriction \boldsymbol{\sigma}) > h\ \right] \leq c \cdot k \cdot 2^{-(\gamma/4)^k \cdot h}.$$

Recall, $h(F)$ denotes the minimal height of a decision tree representing the formula F.

5.6 Matching game

In the next section we show that if G is a good expander, then all the refutations of $\mathrm{BPHP}(G)$ involve some formula that cannot be represented by a shallow decision tree. For its proof we use the *matching games* from [10] later simplified in [7]. Here we provide even cleaner proofs.

Let G be a (U, V, d_L, d_R)-graph. For $S \subseteq U$ and $T \subseteq V$, we say that S *is matchable into* T if there exists a matching M of G with $S \subseteq \mathrm{Dom}(M)$ and $\mathrm{Im}(M) \subseteq T$. If S is not matchable into T but every proper subset of S is, we call it *minimally non-matchable*. For a

matching M and a natural $q > 0$, we say that M is *q-extendible* if every $S \subseteq U \setminus \mathrm{Dom}(M)$ of cardinality at most q is matchable into $V \setminus \mathrm{Im}(M)$.

Lemma 28. *Let $q > 0$ be a natural. If M is a q-extendible matching and (u, v) is an edge in M, then $M \setminus \{(u, v)\}$ is a q-extendible matching.*

Proof. Write $M_0 := \mathrm{Dom}(M)$ and $M_1 := \mathrm{Im}(M)$ and note that $u \in M_0$ and $v \in M_1$. Let S' be a subset of $U \setminus (M_0 \setminus \{u\})$ of cardinality at most q. We need to show that S' is matchable into $V \setminus (M_1 \setminus \{v\})$. We consider two cases: $u \in S'$ and $u \notin S'$. In case $u \in S'$, using that $u \in M_0$, we have that $S' \setminus \{u\}$ is a subset of $U \setminus M_0$ of cardinality at most q. Since M is q-extendible, $S' \setminus \{u\}$ is matchable into $V \setminus M_1$. But then, using that $v \in M_1$, the set S' is also matchable into $V \setminus (M_1 \setminus \{v\})$ by adding (u, v) to the matching that witnesses this. In case $u \notin S'$ then S' is a subset of $U \setminus M_0$ of cardinality at most q. Since M is q-extendible we conclude that S' is matchable into $V \setminus M_1$, and hence into $V \setminus (M_1 \setminus \{v\})$. □

For a natural $q > 0$ and a real $b > 0$, the graph G is a *(q, b)-expander* if $|N_G(S)| \geq (1+b)|S|$ for every $S \subseteq U$ of cardinality at most q.

Lemma 29. *Let $q > 0$ be a natural and $b > 0$ a real. If G is a (q, b)-expander, M is a q-extendible matching with $|M| < \lfloor qb/d_L \rfloor$ and $u \in U \setminus \mathrm{Dom}(M)$, then there exists $v \in N_G(u) \setminus \mathrm{Im}(M)$ such that $M \cup \{(u, v)\}$ is a q-extendible matching.*

Proof. Again write $M_0 := \mathrm{Dom}(M)$ and $M_1 := \mathrm{Im}(M)$. Let $v_1, \ldots, v_l$ be an enumeration of $N_G(u) \setminus M_1$. Since M is q-extendible and $q \geq 1$, we have that $\{u\}$ is matchable into $V \setminus M_1$, so $l \geq 1$. Clearly, $M \cup \{(u, v_i)\}$ is a matching for every $i \in \{1, \ldots, l\}$. Assume for contradiction that $M \cup \{(u, v_i)\}$ is not q-extendible for any $i \in \{1, \ldots, l\}$. For every $i \in \{1, \ldots, l\}$ let S_i be a subset of $U \setminus (M_0 \cup \{u\})$ of cardinality at most q that is minimally non-matchable into $V \setminus (M_1 \cup \{v_i\})$. By Hall's Theorem and the minimality of S_i we have $|N_G(S_i) \setminus (M_1 \cup \{v_i\})| < |S_i|$, and hence $|N_G(S_i)| < |S_i| + (qb/d_L - 1) + 1$. On the other hand $|S_i| \leq q$, and hence $|N_G(S_i)| \geq (1 + b)|S_i|$ by expansion of G. These together imply $|S_i| < q/d_L$ and hence $|S_i| < q/l$ because $1 \leq l \leq d_L$. Since this holds for every $i \in \{1, \ldots, l\}$ we get $|S| \leq q$ for $S := \bigcup_{i=1}^{l} S_i \cup \{u\}$. Since M is q-extendible and $S \subseteq U \setminus M_0$ we conclude that S is matchable into $V \setminus M_1$. A matching M' witnessing this matches u to v_i for some $i \in \{1, \ldots, l\}$. As M' matches S_i into $V \setminus M_1$ while S_i is non-matchable into $V \setminus (M_1 \cup \{v_i\})$, necessarily M' matches some $u_i \in S_i$ to v_i. But this contradicts M' to be a matching because $u_i \neq u$ as $u \notin S_i$. □

5.7 Adversary argument

Let G be a (U, V, d_L, d_R)-graph. We derive a lower bound on the height of formulas in a refutation of BPHP(G) provided G is suitably expanding. This is done by an adversary argument (cf. [43]) based on Lemma 29.

Lemma 30. *Let $q > 0$ be a natural and $b > 0$ a real. If G is a (q, b)-expander, then every refutation of* BPHP(G) *contains a formula F with*

$$h(F) > \tfrac{1}{3}\lfloor qb/d_L \rfloor.$$

Proof. For the sake of contradiction assume $F_0, \ldots, F_{s-1}$ is a refutation of BPHP(G) such that $h(F_i) \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ for all $i \in [s]$; let T_i be a decision tree of height $\leq \frac{1}{3}\lfloor qb/d_L \rfloor$ representing F_i and assume T_{s-1} is the tree with one node labeled 0. We can assume that every F_i contains only variables occurring in BPHP(G): otherwise substitute 0 for all other variables and "answer" in T_i all queries on these variables by 0.

For a matching M let μ_M denote the restriction associated with it (cf. Section 5.4).

Claim. Let M be a matching and $i \in [s]$. Then

1. if F_i is a clause in BPHP(G) or an axiom, then $F_i \restriction \mu_M \not\equiv 0$,
2. if M is q-extendible and such that $|M| \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ and $F_i \restriction \mu_M \equiv 0$, then there exists $1 \leq i' < i$ and a q-extendible matching M' such that $|M'| \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ and $F_{i'} \restriction \mu_{M'} \equiv 0$.

Proof of Claim. The first item is trivial if F_i is an axiom. Assume F_i is a domain clause for $(u, j) \notin \mathrm{Dom}(\phi_G)$. If $u \notin \mathrm{Dom}(M)$, then F_i is untouched by μ_M. Otherwise there is j' such that $\phi(u, j') = M(u)$. Then $j \neq j'$ and there is a $b \in [\ell]$ such that $\mathrm{bit}(b, j) \neq \mathrm{bit}(b, j')$. Then μ_M evaluates $P_{u;b}$ to $\mathrm{bit}(b, j')$, and hence $\neg^{\mathrm{bit}(b,j)} P_{u;b} \restriction \mu_M = 1$. Then $F_i \restriction \mu_M = 1$, so $F_i \restriction \mu_M \not\equiv 0$.

Assume F_i is a collision clause for u, u', j, j' with $u \neq u'$ and $\phi_G(u, j) = \phi_G(u, j')$. If not both u and u' are in $\mathrm{Dom}(M)$, then clearly $F_i \restriction \mu_M \not\equiv 0$. Otherwise, as M is a matching, $M(u) \neq \phi_G(u, j)$ or $M(u') \neq \phi_G(u', j')$. Assume the first and choose j'' such that $M(u) = \phi_G(u, j'')$. Then $j \neq j''$, so $\mathrm{bit}(b, j) \neq \mathrm{bit}(b, j'')$ for some $b \in [\ell]$. As above, this implies $\neg^{\mathrm{bit}(b,j)} P_{u;b} \restriction \mu_M = 1$, so $F_i \restriction \mu_M = 1$ and $F_i \restriction \mu_M \not\equiv 0$.

We now prove the second item. Let i and M accord its assumption. By the first item, F_i is not a clause in BPHP(G) nor an axiom. Then there are $i_0, i_1 < i$ such that F_i is logically

implied by $(F_{i_0} \wedge F_{i_1})$. By Lemma 1 there is a decision tree T of height $\leq \frac{2}{3}\lfloor qb/d_L \rfloor$ that represents $(F_{i_0} \wedge F_{i_1})$.

We call a matching *appropriate for* a path π in T if it is q-extendible, contains M, its associated restriction extends π (as a restriction, cf. Section 2.5), and its domain is $\mathrm{Dom}(M) \cup U(\pi)$, where $U(\pi)$ is the set of pigeons mentioned by some variable queried in π.

Subclaim. There exists a branch π of T and a matching M_π appropriate for π.

The subclaim implies the Claim: if π were a 1-branch, then $(F_{i_0} \wedge F_{i_1}) \restriction \mu_{M_\pi} \equiv 1$ (since μ_{M_π} extends π), so $F_i \restriction \mu_{M_\pi} \equiv 1$ and this contradicts $M \subseteq M_\pi$ and $F_i \restriction \mu_M \equiv 0$. Hence π is a 0-branch and thus extends a 0-branch π' of T_{i_0} or T_{i_1}. Choose accordingly $i' := i_0$ or $i' := i_1$ and let M' be the restriction of M_π to $U(\pi')$. Then M' is q-extendible (by Lemma 28), $|M'| \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ (since $|U(\pi')| \leq \frac{1}{3}\lfloor qb/d \rfloor$) and $F_{i'} \restriction \mu_{M'} \equiv 0$ (since $\mu_{M'}$ extends π').

Observe that M is an appropriate matching for the path π consisting only in the root of T. To prove the subclaim it thus suffices to show that if we have a path π with appropriate matching M_π such that π that does not lead to a leaf of T then we can extend π by one node t such that there is an appropriate matching $M_{\pi t}$ for πt.

So let π and M_π be as stated, say, π leads to an inner node t of T querying the variable $P_{u;b}$. We distinguish two cases. In case $u \in \mathrm{Dom}(M_\pi)$ then μ_{M_π} evaluates $P_{u;b}$; in this case we prolongue π by the corresponding successor t' of t and let $M_{\pi t'} := M_\pi$. In case $u \notin \mathrm{Dom}(M_\pi)$ we look for some v such that $M_\pi \cup \{(u, v)\}$ is a q-extendible matching and then proceed as in the first case. Such a v can be found because $\mathrm{Dom}(M_\pi) = \mathrm{Dom}(M) \cup U(\pi)$ has cardinality at most

$$|\mathrm{Dom}(M)| + |U(\pi)| \leq \tfrac{1}{3}\lfloor qb/d_L \rfloor + \tfrac{2}{3}\lfloor qb/d_L \rfloor - 1 < \lfloor qb/d_L \rfloor,$$

and Lemma 29 applies. Here we use that $|U(\pi)|$ is bounded by the length of π, and this is at most $\frac{2}{3}\lfloor qb/d_L \rfloor - 1$ because π leads to an internal node of T, whose height is at most $\frac{2}{3}\lfloor qb/d_L \rfloor$. ⊣

The Claim implies that there are no i and M that satisfy the assumption of the second item. But $i := s - 1$ and $M := \emptyset$ do: using Hall's Theorem it is easy to see that $\emptyset$ is q-extendible, and obviously $0 \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ and $F_{s-1} \restriction \emptyset \equiv 0$ hold because $F_{s-1} = 0$. □

5.8 Proof size lower bound

We prove Theorem 6. Let $\epsilon > 0$ be arbitrary and write

$$m := n^2,\ t := 2n,\ s := (\log n)^{1/2-\epsilon}.$$

Assume for the sake of contradiction that there exists infinitely many n such that $\mathrm{PHP}_n^{m,t}$ has a DNF-refutation $R = R_n$ of size at most n^s. For the next claim recall the random restriction $\boldsymbol{\rho} = \boldsymbol{\rho}(t)$ from Section 5.2.

Claim 1. There exists a realization ρ of $\boldsymbol{\rho}$ such that every term in every DNF in $R \restriction \rho$ mentions at most s pigeons.

Proof of Claim 1: Call a term *long* if it mentions more than $p := 2s\log(n)$ pigeons, and *short* otherwise. By Lemma 21, a long term T does not restrict to 0 (under $\boldsymbol{\rho}$) with probability at most

$$\left(\frac{1}{2} + \frac{t}{m-p}\right)^p \leq \frac{1}{2^p} \cdot e^{\frac{tp}{2(m-p)}}.$$

But this is smaller than $n^{-s} \cdot 1/2$ noting $\frac{tp}{2(m-p)} \approx 0$ for large enough n. By the union bound, with probability bigger than $1/2$ every long term of R restricts under $\boldsymbol{\rho}$ to 0.

By Lemma 22, a short term restricts to one mentioning more than s many pigeons with probability at most

$$\binom{p}{s+1} \cdot \left(\frac{t}{m}\right)^{s+1} \leq \left(\frac{pt}{m}\right)^{s+1}.$$

But this is smaller than $n^{-s} \cdot 1/2$ for sufficiently large n. By the union bound, with probability bigger than $1/2$ every short term of R restricts to one mentioning at most s pigeons. The claim follows. $\dashv$

Choose ρ according Claim 1. We already observed in Section 5.2 that, up to some renaming of pigeons, $R \restriction \rho$ is a DNF-refutation of PHP_n^t size at most n^s.

Set

$$b := 1,\ q := \lceil\sqrt{n}\rceil,\ r := \lceil n/\log n\rceil,\ d_L := \lceil\log n\rceil,\ d_R := 7\lceil\log n\rceil.$$

Recall for later use that $\ell := |d_L - 1|$ and therefore ℓ is $O(\log\log n)$. Assuming n is sufficiently large the hypotheses of Lemmas 18 and 20 are satisfied for $\varepsilon := 1/2$ and imply the existence of a (U, V, d_L, d_R)-graph G that is a (q, b, r)-resilient expander where $U = [t]$ and $V = [n]$.

Recall the restriction θ_G from Section 5.3. There we observed that $\mathrm{PHP}_n^t \restriction \theta_G$ is $\mathrm{PHP}(G)$, so $(R \restriction \rho) \restriction \theta_G$ is a refutation of $\mathrm{PHP}(G)$ of size at most n^s. Let ϕ_G be a map associated with G as in Section 2.2. All over the refutation substitute the variable $P_{u,v}$ by the ℓ-term $\bigwedge_{b\in[\ell]} \neg^{1-\mathrm{bit}(b,j)} P_{u;b}$, where j is such that $\phi_G(u, j) = v$. Of course, the result is again a refutation. By the discussion just before Lemma 23 it refutes sporadic axioms and collision clauses of $\mathrm{BPHP}(G)$. By Lemma 23 we can add proofs of the sporadic axioms from the

domain clauses of BPHP(G); this way we get a refutation R' of BPHP(G) of size $n^{c_1 \cdot s}$ for some constant c_1.

Every term in every DNF in $(R \restriction \rho) \restriction \theta_G$ mentions at most s pigeons and becomes after the substitution an ℓ-CNF mentioning at most s pigeons. The additional proofs added for the sporadic axioms mention only one pigeon. Hence, R' is a refutation of BPHP(G) all of whose formulas are disjunctions of ℓ-CNFs each mentioning at most s pigeons. Applying Lemma 25 we move to a refutation R'' of size $n^{c_2 \cdot s}$ for some constant c_2, where additionally all these ℓ-CNFs are matching-satisfiable.

For the next claim, let $\mathbf{B}$ and $\boldsymbol{\mu}$ be random variables defined for G as in Section 5.4.

Claim 2. There exists a realization (B, μ) of $(\mathbf{B}, \boldsymbol{\mu})$ such that

(a) $h(F \restriction \mu) \leq \frac{1}{3}\lfloor qb/d_L \rfloor$ for all F in R'', and

(b) $G \setminus B$ is a $(q, 1)$-expander.

Proof of Claim 2. Note a random $\mathbf{B}$ satisfies (b) with probability bigger than $1/2$ because G is (q, b, r)-resilient. Hence it suffices to show that for any disjunction F of matching-satisfiable ℓ-CNFs each mentioning at most s pigeons

$$n^{c_2 \cdot s} \cdot \Pr[\ h(F \restriction \boldsymbol{\mu}) > \tfrac{1}{3}\lfloor qb/d_L \rfloor\ ] \leq \tfrac{1}{2}. \tag{5.6}$$

A matching-satisfiable ℓ-CNF mentioning at most s pigeons is logically equivalent to a DNF with matching-satisfiable terms each mentioning at most s pigeons. Since there are at most $s \cdot \ell$ binary pigeon variables mentioning some fixed set of $\lfloor s \rfloor$ pigeons, this DNF can be chosen as an $\lfloor s \cdot \ell \rfloor$-DNF. Thus, a formula F as above is logically equivalent to a $\lfloor s \cdot \ell \rfloor$-DNF F' where each term mentions at most s pigeons and is matching-satisfiable. In (5.6) we can equivalently replace[3] F by F'.

To bound the probability in (5.6) we intend to apply Lemma 27. By Lemmas 24 and 26, the random restriction $\boldsymbol{\mu}$ satisfies the assumptions of Lemma 27 with

$$k := \lfloor s \cdot \ell \rfloor,\ h := \lfloor \tfrac{1}{3} qb/d_L \rfloor,\ c := \lceil 3\sqrt{r} \rceil,$$

and

$$\gamma := \left(\frac{r}{d_R \cdot n}\right)^s \cdot \left(1 - \frac{r}{n}\right)^{d_L \cdot s} \cdot \frac{\log(e)}{\ell \cdot k \cdot d_L \cdot d_R}.$$

By Lemma 27 we have $\Pr[\ h(F \restriction \boldsymbol{\mu}) > \frac{1}{3}\lfloor qb/d_L \rfloor\ ]$ is at most $c \cdot k \cdot 2^{-(\gamma/4)^k \cdot h}$. Note that if n is sufficiently large, then $(1 - r/n)^{d_L \cdot s} \geq (1/e)^{c_3 \cdot s}$ for some constant $c_3 > 0$. It is then easy

[3] Thanks to our weaker notion of representation – cf. Remark 1.

to see that $\gamma/4 \geq (1/\log n)^{c_4 \cdot s}$, and hence $(\gamma/4)^k \geq (1/\log n)^{c_4 \cdot s^2 \cdot \ell} \geq n^{-1/(\log n)^\epsilon}$ for some other constant $c_4 > 0$. As $h \geq n^{1/3}$ we get $(\gamma/4)^k \cdot h \geq n^{1/4}$ for sufficiently large n. Noting $c \cdot k \leq n$, then (5.6) follows. $\dashv$

Choose (B, μ) according to Claim 2, say, μ is associated with the matching M of G. Recall that R'' refutes BPHP(G). We claim $R'' \restriction \mu$ is a refutation of BPHP(G') for

$$G' := G \setminus (\mathrm{Dom}(M) \cup \mathrm{Im}(M)).$$

We have to show that every clause C of BPHP(G) restricts under μ to 1 or to a clause of BPHP(G'). If C does not mention a pigeon in $\mathrm{Dom}(M)$, then C is a clause of BPHP(G') and $C \restriction \mu = C$. If C mentions only pigeons in $\mathrm{Dom}(M)$, then $C \restriction \mu = 1$. Finally, assume C is a collision clause for $(u, j) \in \mathrm{Dom}(\phi_G)$ and $(u', j') \in \mathrm{Dom}(\phi_G)$ with $u \neq u'$ and $\phi_G(u, j) = \phi_G(u', j')$, and exactly one pigeon, say u, in $\mathrm{Dom}(M)$. If j is such that $\phi_G(u, j) \neq M(u)$, then $C \restriction \mu = 1$; otherwise, $C \restriction \mu = \bigvee_{b \in [\ell]} \neg^{\mathrm{bit}(b, j')} P_{u';b}$ and this is a domain clause of BPHP(G'): note $\phi_G(u, j) = \phi_G(u', j') = M(u) \in \mathrm{Im}(M)$, so $(u', j') \notin \mathrm{Dom}(\phi_{G'})$. This is ensured by definition of the map associated to a restricted graph (see Section 2.2).

Since $\mathrm{Im}(M) \subseteq B$, Claim 2 (b) implies that G' is a $(q, 1)$-expander. Hence $R'' \restriction \mu$ contradicts Lemma 30 by Claim 2 (a).

Chapter 6

Conclusions

In this chapter we present several open problems that we consider as the next steps in the research lines presented in this thesis. At the end of the chapter, we include a brief note on the publication status of the presented results.

6.1 Open problems

6.1.1 A characterization of QBFs with short refutations

In Chapter 3 we have shown that false QBFs with bounded respectful tree-width have polynomial-size Q-resolution refutations. Also, we mentioned some classes of QBFs that have short Q-resolution refutations, e.g. 2-QBFs and Horn-QBFs. Additionally, purely-existential QBFs with bounded tree-width and QBFs with bounded respectful tree-width also have short Q-resolution refutations.

It would be nice to find other classes of QBFs with short Q-resolution refutations. In particular, we would like to identify other classes of QBFs with bounded tree-width that have short Q-resolution refutations and, ultimately, characterize these kind of formulas.

6.1.2 A stronger interpretability result

Corollary 3 states that the class of standardized universal sentences with short R(const)-refutations is closed under quantifier-free definitions. We would like to have the same result for quantifier-free interpretations. To do so, we would need a stronger version of Theorem 4 similar to the following:

Statement 1. *Let ϕ and ψ be conjunctions of standardized universal sentences. For all naturals r, t, $n \geq t$, s and k, if ϕ is quantifier-free interpretable in ψ with t parameters and $\langle\phi\rangle_n^{\mathrm{u}} \vdash_k^s \square$, then $\langle\psi\rangle_n^{\mathrm{u}} \vdash_{k'}^{s'} \square$ for s' polynomial in s, n and 2^k and k' linear in k.*

Note that this statement has been obtained by weakening the second hypothesis of Theorem 4. We find this statement hard to believe. However, we identify some intermediate results that look more attainable and can be of independent interest. On one side, we may strengthen the second hypothesis of Statement 1 so that it is still weaker than in Theorem 4. This hypothesis may be a natural, particular case of ϕ^{I_r}. One of these is the aforementioned ϕ^R. Another is the *quotient* of ϕ, called ϕ^Q, which is the particular case of ϕ^{I_r} in which $r = 1$ and U is the trivial true relation.

On the other side, we may strengthen the first hypothesis of Statement 1 by adding requirements on the θ-translation that must interpret ϕ in ψ. One way to do that is to fix some of the θ-formulas. For example, we may require that $\theta_E(\bar{x}, \bar{y}) = (x_1 = y_1)$, as in some of our examples. Note that these two ways of strengthening the hypotheses in Statement 1 may be combined to obtain different intermediate results.

One of those is of special interest to us. This is the one that strengthens the first hypothesis by requiring, that ϕ is interpretable in ψ through a translation in which $\theta_U(\bar{x}) = (x_1 = x_1)$ and $\theta_E(\bar{x}, \bar{y}) = (x_1 = y_1)$, and leaves the second hypothesis as is. Such a theorem, together with our quantifier-free interpretation of FPHP_n^{2n} in $\mathrm{FPHP}_n^{n^2,2n}$ in section 4.3.4, could be used to show that known upper-bounds on the size of the $\mathrm{R}(k)$-refutations of FPHP_n^{2n} are also true for $\mathrm{FPHP}_n^{n^2,2n}$. By slightly modifying our quantifier-free interpretation, we could show these same upper-bounds for its non-functional version, PHP_n^{2n}, for which we also show lower-bounds in Chapter 5.

6.1.3 Complete principles for R(const)

In Chapter 4, inspired by the concept of reducibility among computational problems, we proposed quantifier-free definability as a way to determine if a principle is as hard to prove as another in a given proof system. Taking this analogy further, we can say that a principle P is *hard* for a class of principles $\mathcal{C}$ if every principle in $\mathcal{C}$ defines P through a quantifier-free translation. Further, we can say that P is *complete* for $\mathcal{C}$ if it is hard for $\mathcal{C}$ and $P \in \mathcal{C}$.

We would be interested in determining the existence of a complete principle for the class of those whose propositional translations have polynomial-size R(const) refutations. In [21], Dantchev and Martin suggest that the relativization of the Least Number Principle could be a candidate for this role.

6.1.4 Super-polynomial lower bounds for PHP_n^{2n}

In Chapter 5 we have shown super-polynomial lower bounds on the size of the DNF-refutations of the relativized principle $\mathrm{PHP}_n^{n^2,2n}$. We would be interested in extending this result to the more classical PHP_n^{2n}. As mentioned in the introduction, determining if this principle has bounded-depth polynomial-size proofs is a longstanding open problem. Extending our result would answer the question for depth two.

Even more: our current technique, using random restrictions, could possibly be iterated to yield results for higher depths. One impediment for doing so is that the formula is relativized. Therefore, extending our result to the unrelativized PHP_n^{2n} could be a step towards proving super-polynomial lower-bounds for all bounded-depth proof systems.

6.2 Publications related to this thesis

Each of the technical chapters of this thesis is based on a paper as detailed in the following:

- The main result of Chapter 3 is from the paper *Bounded-width QBF is PSPACE-complete*, a joint work with A. Atserias. This paper has been accepted at the 30th Symposium on Theoretical Aspects of Computer Science (STACS 2013) in Kiel, Germany, February 2013.

- The result in Chapter 5 is contained in the paper *Lower Bounds for DNF-refutations of a Relativized Weak Pigeonhole Principle*, a joint work with A. Atserias and M. Müller. This paper has been accepted at the 27th Annual IEEE Conference on Computational Complexity (CCC 2013) in Palo Alto, USA, June 2013.

- The definability and interpretability results in Chapter 4, as well as their auxiliary lemmas from Chapter 2, will be part of the paper *Reducibility Among First-order Principles*, a joint work with A. Atserias and M. Müller. This paper is currently in preparation.

Bibliography

[1] M. Ajtai. The complexity of the pigeonhole principle. Proceedings of the 29th Symposium on the Foundations of Computer Science (FOCS), 346-355, 1988.

[2] M. Ajtai. Approximate counting with uniform constant-depth circuits. Advances in Computational Complexity Theory, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 13:1-20, 1993.

[3] M. Alekhnovich and A.A. Razborov. Satisfiability, branch-width and Tseitin tautologies. Proceedings of the 43rd Symposium on Foundations of Computer Science (FOCS), 593-603, 2002.

[4] S. Arnborg, D.G. Corneil and A. Proskurowski. Complexity of finding embeddings in a k-tree. SIAM Journal on Algebraic Discrete Methods 8(2):277-284, 1987.

[5] S. Arora and B. Barak. Computational Complexity: a modern approach. Cambridge University Press, 2009.

[6] B. Aspvall, M.F. Plass and R.E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. Information Processing Letters 8(3):121-123, 1979.

[7] A. Atserias. On sufficient conditions for unsatisfiability of random formulas. Journal of the ACM 51(2):281-311, 2004.

[8] A. Atserias, M.L. Bonet and J.L. Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. Information and Computation 176(2):136-152, 2002.

[9] P. Beame, R. Impagliazzo and T. Pitassi. Exponential lower bounds for the pigeonhole principle. Computational Complexity 3(2):97-140, 1993.

[10] E. Ben-Sasson and N. Galesi. Space complexity of random formulae in resolution. Random Structures and Algorithms 23(1):92-109, 2003.

[11] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – resolution made simple. Journal of the ACM 48(2):149-169, 2001.

[12] B. Bollobás. Random graphs. 2nd edition, Cambridge University Press, 2001.

[13] H.K. Büning, A. Flögel and M. Karpinski. Resolution for quantified Boolean formulas. Information and Computation 117(1):12-18, 1995.

[14] S. R. Buss. Polynomial size proofs of the propositional pigeonhole principle. Journal of Symbolic Logic 52(4):916-927, 1987.

[15] P. Chatalic and L. Simon. Multi-resolution on compressed sets of clauses. Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2-10, 2000

[16] H. Chen. Quantified constraint satisfaction and bounded treewidth. Proceedings of the 16th European Conference on Artificial Intelligence (ECAI), 161-165, 2004.

[17] H. Chen and V. Dalmau. From pebble games to tractability: An ambidextrous consistency algorithm for quantified constraint satisfaction. Proceedings of the 19th Conference on Computer Science Logic (CLS). 232-247, 2005.

[18] H. Chen and V. Dalmau. Decomposing quantified conjunctive (or disjunctive) formulas. Proceedings of the 27th Symposium on Logic in Computer Science (LICS), 205-214, 2012.

[19] S. A. Cook, R. A. Reckhow. The relative efficiency of propositional proof systems. Journal of Symbolic Logic 44(1):36-50, 1979.

[20] V. Dalmau, P. Kolaitis and M.Y. Vardi. Constraint satisfaction, bounded treewidth and finite-variable logics. Proceedings of the 8th International Conference in Principles and Practice of Constraint Programming (CP), 223-254, 2006.

[21] S. Dantchev and B. Martin. The limits of tractability in resolution-based propositional proof systems. 6th Conference on Computability in Europe. Lecture Notes in Computer Science 6158:98-107, 2010.

[22] S. Dantchev and S. Riis. On relativization and complexity gap for resolution-based proof systems. 17th Annual Conference of the European Association for Computer Science Logic (CSL), Lecture Notes in Computer Science 2803:142-154, 2003.

[23] R. Dechter. Constraint processing. Morgan Kaufmann, 2003.

[24] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems. Artificial Intelligence 34(1):1-38, 1987.

[25] R. Dechter and J. Pearl. Tree clustering for constraint networks. Artificial Intelligence 38:353-366, 1989.

[26] J. L. Esteban, N. Galesi and J. Messner. On the complexity of resolution with bounded conjunctions. Theoretical Computer Science, 321:347-370, 2004.

[27] E. Freuder. Complexity of k-tree structured constraint satisfaction problems. Proceedings of the 8th National Conference on Artificial Intelligence (AAAI) 1:4-9, 1990.

[28] M. Furst, J. B. Saxe and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. Theory of Computing Systems 17(1):13-27, 1984.

[29] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. Proceedings of the 17th Symposium on Logic in Computer Science (LICS), 215-224, 2002.

[30] G. Gottlob, G. Greco, and F. Scarcello. The complexity of quantified constraint satisfaction problems under structural restrictions. Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI), 150-155, 2005.

[31] A. Haken. The intractability of resolution, Theoretical Computer Science 39(2-3):297-308, 1985.

[32] N. Immerman. Descriptive complexity. Springer Verlag, 1999.

[33] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. Journal of Symbolic Logic 59(1):73-86, 1994.

[34] J. Krajíček. On the weak pigeonhole principle. Fundamenta Mathematicae 170:123-140, 2001.

[35] J. Krajíček, P. Pudlák and A. Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. Random Structures & Algorithms 7(1):15-39, 1995.

[36] A. Maciel, T. Pitassi and A. R. Woods. A new proof of the weak pigeonhole principle. Journal of Computer and System Sciences 64(4):843-872, 2002.

[37] J. Marques-Silva and J. Planes. Algorithms for maximum satisfiability using unsatisfiable cores. Proceedings of the 11th Conference on Design, Automation and Test in Europe (DATE), 408-413, 2008.

[38] A. Meyer. Weak monadic second order theory of succesor is not elementary recursive. Logic Colloquium. Lecture Notes in Mathematics 453:132-154, 1975.

[39] G. Pan and M.Y. Vardi. Symbolic decision procedures for QBF. Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming (CP), 453-467, 2004.

[40] G. Pan and M.Y. Vardi. Fixed-parameter hierarchies inside PSPACE. Proceedings of the 21th Symposium on Logic in Computer Science (LICS), 27-36, 2006.

[41] C. Papadimitriou. Computational Complexity. Addison-Wesley, 1994.

[42] J.B. Paris, A.J. Wilkie and A.R. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. Journal of Symbolic Logic 53(4):1235-1244, 1988.

[43] P. Pudlák. Proofs as games. American Mathematical Monthly 107(6):541-550, 2000.

[44] R. Raz. Resolution lower bounds for the weak pigeonhole principle. Journal of the ACM 51(2): 115-138, 2004.

[45] A. A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus. Unpublished, 2003.

[46] A. A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. Theoretical Computer Science 1(303):233-243, 2003.

[47] S. Riis. A complexity gap for tree-resolution. Comp. Complexity, 10:179-209, 2001.

[48] H. Robbins. A remark on Stirling's formula. The American Mathematical Monthly 62(1):26-29, 1955.

[49] N. Segerlind. The complexity of propositional proofs. The Bulletin of Symbolic Logic 13(4):417-481, 2007.

[50] N. Segerlind, S. Buss and R. Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. Journal on Computing 33(5):1171-1200, 2004.