On Sufficient Conditions for Unsatisfiability of Random Formulas *

Albert Atserias†
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
C/Jordi Girona Salgado, 1-3, Edif. C6.
08034 Barcelona, Spain.
Tel: +34 93 401 69 94
Fax: +34 93 401 70 14
atserias@lsi.upc.es

September 10, 2003

Abstract

A descriptive complexity approach to random 3-SAT is initiated. We show that unsatisfiability of any significant fraction of random 3-CNF formulas cannot be certified by any property that is expressible in Datalog. Combined with the known relationship between the complexity of constraint satisfaction problems and expressibility in Datalog, our result implies that any constraint propagation algorithm working with small constraints will fail to certify unsatisfiability almost always. Our result is a consequence of designing a winning strategy for one of the players in the existential pebble game. The winning strategy makes use of certain extension axioms that we introduce and hold almost surely on a random 3-CNF formula. The second contribution of our work is the connection between finite model theory and propositional proof complexity. To make this connection explicit, we establish a tight relationship between the number of pebbles needed to win the game and the width of the Resolution refutations. As a consequence to our result and the known size-width relationship in Resolution, we obtain new proofs of the exponential lower bounds for Resolution refutations of random 3-CNF formulas and the Pigeonhole Principle.

* A preliminary version of this paper appeared in the Proceedings of the 17th IEEE Annual Symposium on Logic in Computer Science (LICS), pp. 325-334, 2002 under the title "Unsatisfiable Random Formulas are Hard to Certify". The paper was the recipient of the Kleene Award for best student paper and is the base of two of the chapters of the Ph.D. thesis of the author at the University of California, Santa Cruz.

† Partially supported by CICYT TIC2001-1577-C03-02 and ALCOM-FT IST-99-14186.

1 Introduction

The *phase transition* phenomenon that certain combinatorial problems exhibit has inspired a new line of research within the field of computational complexity. The focus is no longer exclusively set to the worst-case complexity of the problem, but also in studying its structure for the typical instances when drawn from certain distributions of interest. This research is motivated by the need to understand the structure of hard-to-solve **NP**-complete problems of practical importance. Moreover, the structural analysis of the typical instance is often followed by an analysis of the running-time of the known algorithms to solve the problem. As a matter of fact, the methods for these two tasks are very often related, and one task can be seen as complementary of the other.

The satisfiability problem for 3-CNF formulas, the most popular **NP**-complete problem, has been intensively studied from this perspective. The distribution of interest in this context is parameterized by the number of variables n and the number of clauses m of the formula. Thus, following the standard notation, we let $\mathcal{F}(n, m, 3)$ denote the probability space of random 3-CNF formulas with n variables and m clauses generated uniformly and independently from the set of all possible clauses on exactly three distinct variables. The ratio m/n is denoted by Δ and is called the *clause density*. It is conjectured that the probability that a random formula F is unsatisfiable exhibits a phase transition from asymptotically 0 to asymptotically 1 around a critical clause density Δ_c [CR92]. As far as partial results are concerned, it has been proven that if $\Delta < 3.42$ then F is almost surely satisfiable [KKL02] and if $\Delta > 4.506$ then F is almost surely unsatisfiable [DBM00]. Moreover, experimental results suggest that Δ_c is around 4.2 [SML96], but the existence of such a constant is only a conjecture for the moment.

As mentioned already, the structural analysis of the typical instance of the 3-CNF satisfiability problem is complemented by an analysis of the known algorithms to solve it. The first results in this respect were only experimental showing that the running-time of the Davis-Putnam-Logemann-Loveland procedure (DPLL) for satisfiability experiments a sharp jump around $\Delta \approx 4.2$, and decays slowly as the ratio Δ increases. These empirical observations are confirmed by two deep theoretical results. First, Broder, Frieze, and Upfal [BFU93] proved that if $\Delta < 1.63$ then a *simple* variant of DPLL running in linear time succeeds in finding a satisfying assignment with high probability. Second, Chvátal and Szeméredi [CS88], as improved by [BKPS98, BS01], proved that every Resolution refutation of an unsatisfiable F almost surely requires size $2^{\Omega(n/\Delta^{2+\epsilon})}$. Thus, since a DPLL run on F provides a Resolution refutation when F is unsatisfiable no matter which heuristic is used for the splitting rule [BKPS98], this provides a proof that all DPLL-based algorithms behave badly in the unsatisfiable region with small Δ. The results of Broder, Frieze, and Upfal were later improved to values of Δ closer to 4.2 (see the nice survey by Achlioptas [Ach01]), and the results of Chvátal and Szeméredi were extended to other proof systems such as the Polynomial Calculus and Res(2) [BSI99, ABE02].

We remark that, at the time of writing, the known lower bounds on the satisfiability threshold Δ_c are algorithmic, while the known upper bounds are not. What this means is that for every density Δ for which it has been proven that a random formula is almost surely satisfiable, there actually is a polynomial-time algorithm that certifies this with probability bounded away from zero. The situation at the upper-end is quite different. All known constant upper bounds on Δ_c follow by a counting argument that does not provide with a polynomial-time algorithm that certifies unsatisfiability. In fact, the theoretical results of Chvátal and Szeméredi about Resolution, and their

extension to other proof systems such as the Polynomial Calculus and Res(2), prove that every algorithm that implicitly or explicitly produces a refutation in such systems will fail to succeed in polynomial-time or with non-zero probability. Hence, these results suggest a pattern in the average-case complexity of the satisfiability problem sometimes referred to informally as the *easy-hard* pattern (see [SK96], and see [CDA$^+$00] for a criticism). It seems that for constant values of Δ beyond the critical point, certifying unsatisfiability on a significant fraction of the inputs becomes computationally hard. A definitive argument in its favor would be the *non-existence* of a property of 3-CNF formulas that (1) is efficiently decidable, (2) it implies unsatisfiability, and (3) holds with probability bounded away from 0 when $\Delta > 4.2$ is constant. Obviously, this hypothesis implies $\mathbf{P} \neq \mathbf{NP}$ when the term "efficiently decidable" is interpreted as decidable in polynomial-time, so its proof seems out of reach with current methods. Thus, it is reasonable to ask whether the hypothesis is true for stricter but reasonable interpretations of the term "efficiently decidable", and this is the approach taken in this paper.

We initiate a study of the descriptive complexity of properties that imply unsatisfiability with non-zero probability. We show the *non-existence* of a property of 3-CNF formulas that (1) is expressible in Datalog when 3-CNF formulas are encoded as finite relational structures, (2) it implies unsatisfiability, and (3) holds with probability bounded away from 0 when Δ is constant. Datalog is a well-known query language studied in the context of database theory (see Ullman [Ull89]), capable of expressing many interesting problems including Graph Reachability, Non-2-Colorability, Unsatisfiability of 2-CNFs, and some $\mathbf{P}$-complete problems such as the Monotone Circuit Value Problem and Path Systems among others (see [KV00a, Pap85, Ull89]). In a nutshell, Datalog may be viewed as the existential positive fragment of first-order logic augmented with a mechanism of recursion. In fact, Chandra and Harel [CH82] proved that Datalog coincides with the existential positive fragment $\exists\mathrm{LFP}$ of least fixed-point logic LFP. Let us add that Datalog captures a significant but proper fraction of polynomial-time (see [Ull89] again), and in the presence of a successor relation and negations on the given predicates, it captures polynomial-time exactly [Pap85].

The main ingredient of our proof is the design of a winning strategy for the Duplicator in the *existential k-pebble game* between the structure encoding a randomly generated formula, and the structure encoding a fixed template formula that we know is satisfiable. The existential k-pebble game is a powerful combinatorial tool introduced by Kolaitis and Vardi [KV95, KV00a] to analyze the expressive power of Datalog with k variables. In fact, the existential k-pebble game captures the expressive power of the stronger logic $\exists L^k_{\infty\omega}$, the existential positive fragment of infinitary logic with k variables. Thus, our lower bound is much stronger than stated above since what we actually prove is that every property that implies unsatisfiability of formulas with n variables and is (non-uniformly) expressible in the infinitary logic $\exists L^k_{\infty\omega}$ when $k \leq n/(\ln(n))^2$ must have asymptotic probability 0 when $n \to \infty$ and Δ is constant. Note that we allow the number of variables k to grow with the number of propositional variables n, a parameter that is not allowed to vary in a uniform Datalog program. The techniques also provide results when Δ is a function of n. We discuss these in Section 5.

The design of the winning strategy for the Duplicator relies on the fact that the *incidence graph* of a random 3-CNF formula, the bipartite graph that relates clauses with the variables that occur in them, satisfies certain extension axioms that we introduce based on the expander properties of the graph. The use of extension axioms is a recurring theme in proving 0-1 laws for several logics and

distributions on the instances [Fag76, Lyn80, SS88, KV90]. Moreover, the expander properties of these graphs have also been used in the context of propositional proof complexity to prove lower bounds in the size of Resolution refutations [BSW01, BSG01]. We make the point that this is not a coincidence since there actually is a connection between the fields of propositional proof complexity and finite model theory that we discuss next.

Propositional proof complexity is devoted to the study of the length of proofs in propositional proof systems. Unquestionably, certain tautologies are more obvious than others, and this is reflected by the length of their proofs in a fixed proof system. In addition, the study of the lengths of proofs yields significant insight into the combinatorial and computational content of the tautology. The field started with the pioneering work of Cook and Reckhow [CR79], and has been developing continuously since then.

One of the well-studied proof systems is propositional Resolution, introduced by Robinson [Rob65]. For a discussion on the origins of Resolution, going back to the work of Blake [Bla37], and Davis and Putnam [DP60], see [CS88] (we thank a referee for pointing this out). Strictly speaking, Resolution is a refutational system, which means that it provides proofs of contradiction of unsatisfiable CNF formulas rather than proofs of tautologies. However, it is well known that both frameworks are easily translatable one into the other without much effort. The first significant lower bounds in propositional proof complexity were for Resolution. Indeed, Haken [Hak85] showed that the propositional encoding of the Pigeonhole Principle proposed by Cook and Reckhow requires Resolution refutations of exponential size. Later on, Chvátal and Szeméredi [CS88] proved that a 3-CNF formula randomly generated from $\mathcal{F}(n, \Delta n, 3)$ requires exponential-size Resolution refutations almost surely.

Our results establish two previously unnoticed links between the fields of finite model theory and propositional proof complexity. The first novelty of this work is the observation that hard-to-prove tautologies yield finite structures on which playing combinatorial games might be easier than expected. It is generally recognized that identifying the finite structures on which to carry over a non-expressibility result via combinatorial games is a difficult task. Our point is that the encodings of combinatorial principles as propositional tautologies could serve for this. As a matter of fact, the idea of playing existential k-pebble games on random 3-CNF formulas came out of the following informal reasoning: if a formula is hard to refute in Resolution, shouldn't it be hard to tell apart from a satisfiable formula by a bounded player that confronts an adversary? In order to make this argument even more explicit, we consider the encoding of the Pigeonhole Principle as an unsatisfiable CNF formula in the spirit of Cook and Reckhow, and show that the Duplicator wins the existential pebble game on the finite structure encoding this formula and a structure encoding a satisfiable formula.

The second link between finite model theory and propositional proof complexity is the observation that there is a tight connection between the number of pebbles needed to win the existential k-pebble game and the concept of width in Resolution. In a Resolution refutation, the width is the maximal number of literals in a clause of the refutation. Ben-Sasson and Wigderson [BSW01] proved the following non-trivial result about the width of Resolution: if a 3-CNF formula has a Resolution refutation of size S, then it has a Resolution refutation of width $O\left(\sqrt{n \log S}\right)$, where n is the number of propositional variables. This interesting result relates the width with the size in a form that is suitable to prove size lower bounds. Indeed, if every Resolution refutation of F requires width w, then every Resolution refutation of F requires size $2^{\Omega(w^2/n)}$. Thus, lower bounds

on width imply lower bounds on size. It is worth noting that a conjecture about a trade-off of this type was formulated already in 1981 [KM81]. The connection that we establish is this: we show the existence of a Datalog program P with $2k$ variables such that (1) if F has a Resolution refutation of width k, then P evaluates true on the structure encoding F, and (2) if P evaluates true on the structure encoding F, then F has a Resolution refutation of width $2k$. Thus, if the Duplicator wins the existential k-pebble game on the structure encoding F and the template structure, then every Resolution refutation of F requires width $k/2$. An immediate corollary of our main result about the Duplicator winning the $n/(\ln n)^2$-pebble game on a random 3-CNF formula F, and the size-width trade-off of [BSW01], is that every Resolution refutation of F requires size $2^{\Omega(n/(\ln n)^4)}$. This shows that our main inexpressibility result for $\exists L^k_{\infty\omega}$ is a generalization of the result of Chvátal and Szeméredi. The same argument can be used to obtain Haken's exponential lower bound for the Pigeonhole Principle. It should be pointed out that Ben-Sasson and Wigderson prove similar lower bounds by means of the width method in a more direct way. Nonetheless, it is quite interesting that non-expressibility results in finite model theory yield lower bounds in proof complexity.

We turn next to a discussion about the implications of our results for the complexity of algorithms for constraint satisfaction problems. It was first pointed out by Feder and Vardi [FV98] that every constraint satisfaction problem can be cast as a homomorphism problem: given two finite structures $\mathbf{A}$ and $\mathbf{B}$ over the same relational vocabulary, is there a homomorphism from $\mathbf{A}$ to $\mathbf{B}$? Intuitively, $\mathbf{A}$ represents the variables and the tuples of constraint variables, $\mathbf{B}$ represents the values and the constraints, and the homomorphisms between $\mathbf{A}$ and $\mathbf{B}$ are the solutions to the constraint satisfaction problem. Note that satisfiability problems fit well in this framework by letting the universe of $\mathbf{B}$ be $\{0, 1\}$ and the relations of $\mathbf{B}$ be the truth-tables of the involved connectives.

Several concepts of *consistency* have been popularized by the AI community as powerful algorithmic tools to deal with the intractability of constraint satisfaction problems. The underlying common idea of all these concepts is that of identifying some structural property on the instances that guarantees success of a fixed constraint propagation algorithm in reasonable time. All these algorithms start with the initial constraints of the problem, and propagate them by inferring new and possibly tighter constraints, until the problem is either solved, or proved unsatisfiable.

Kolaitis and Vardi [KV00b] shed light on the connections between the concepts of consistency and the concept of winning strategy for the Duplicator in the existential k-pebble game. These seemingly unrelated concepts turned out to be tightly connected through a theorem stating that a constraint satisfaction problem given by the structures $\mathbf{A}$ and $\mathbf{B}$ can be solved by *establishing strong k-consistency* if and only if the Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$. As a matter of fact, what Kolaitis and Vardi pointed out is that solvability by the usual constraint propagation algorithms can be cast, in one form or another, as conditions on winning strategies for the Duplicator. With this interesting re-interpretation, our result about the Duplicator winning the existential k-pebble game on a random 3-CNF formula admits the following intuitive interpretation: any usual constraint propagation algorithm that attempts to solve satisfiability with small constraints will succeed on a negligible fraction of all 3-CNF formulas only. We note that resolution is only one very particular form of constraint propagation.

2 About Datalog and Infinitary Logic

Let us start by defining the basic concepts of logic that we will use. Our terminology and notation are standard. Any introductory textbook to the topic such as [EFT84] or [BM77] would elaborate further, if need be. A *finite relational vocabulary* L is a finite set of *relation symbols* $\{R_1, \ldots, R_r\}$, each with a non-negative integer arity $r_1, \ldots, r_r$. A *structure over* L is a tuple $\mathbf{M} = (U, R_1^{\mathbf{M}}, \ldots, R_r^{\mathbf{M}})$, where U is a set, called the *universe*, and each $R_i^{\mathbf{M}}$ is a relation of arity r_i over U, called the *interpretation of* R_i. The standard example is the relational language of graphs $L = \{E\}$, with a single binary relation symbol E. The structures over L are exactly the directed graphs, possibly with loops. The undirected graphs are those structures over L whose interpretation of E is a symmetric and irreflexive relation.

We assume the existence of countably many *first-order variables* that range over elements of the universe. We also assume the existence of countably many *second-order variables* of each non-negative arity that range over relations of that arity over the universe. The *atomic formulas* are those of the form $R(x_1, \ldots, x_r)$, $X(x_1, \ldots, x_s)$, or $x_1 = x_2$, where $x_1, x_2, \ldots$ are first-order variables that are not necessarily distinct, R is a relation symbol of arity r, and X is a second-order variable of arity s. The class of *first-order formulas* is the smallest class of formulas that contains the atomic formulas and is closed under conjunction, disjunction, negation, and existential and universal quantification over first-order variables. The semantics of the first-order formulas over a structure is defined in the usual way (see [EFT84]). If $\varphi(x_1, \ldots, x_n, X_1, \ldots, X_m)$ is a first-order formula with free variables as shown, $\mathbf{M}$ is a structure over its vocabulary, $a_1, \ldots, a_n$ are elements of the universe of $\mathbf{M}$, and $A_1, \ldots, A_m$ are relations over the universe of $\mathbf{M}$ of the appropriate arity, we use the notation $\mathbf{M} \models \varphi[a_1, \ldots, a_n, A_1, \ldots, A_m]$ to denote the fact that φ is satisfied in $\mathbf{M}$ under the given interpretation of its free variables. For example, if $G = (V, E)$ is an undirected graph, and $v \in V$ is an isolated node, then $G \models (\forall x)(\neg E(x, v))$.

Next we define Datalog. Good introductory texts on Datalog are [Ull89] and [EF95]. Let $L = \{R_1, \ldots, R_r\}$ be a relational vocabulary and let $X_1, \ldots, X_s$ be second-order variables. Let r_i be the arity of R_i, and let s_i be the arity of X_i. A Datalog program over $L \cup \{X_1, \ldots, X_s\}$ is a set of rules of the form

$$t_0 \quad :- \quad t_1, \ldots, t_p,$$

where t_0 is an atomic formula of the form $X_i(x_1, \ldots, x_{s_i})$, and each t_j with $j > 0$ is an atomic formula of the form $R_i(x_1, \ldots, x_{r_i})$ or $X_i(x_1, \ldots, x_{s_i})$. Here the x_i are first-order variables that are not necessarily different.

Note that the *heads* of the rules (left part) are always predicates formed from second-order variables. These are called the *intensional database* predicates (IDBs). The predicates in the vocabulary L always appear in the *bodies* of the rules (right part), and are called the *extensional database* predicates (EDBs). One of the IDBs is called the *goal* of the program, and it might be of arity zero (a propositional letter). A Datalog program is a recursive specification of the IDBs from the EDBs formally defined below. Informally, if $\mathbf{M}$ is a structure over the vocabulary L of the EDBs, the IDBs are given the following semantics: initialize the IDBs to be empty, and repeatedly apply the rules of the program whose bodies (right part) are satisfied by $\mathbf{M}$ and the current value of the IDBs, until no new facts can be added to the IDBs. One way of formally defining the semantics of Datalog is through the concept of system of simultaneous inductions that we introduce next.

Let

$$\varphi_1(x_1,\ldots,x_{r_1},X_1,\ldots,X_s),\ldots,\varphi_s(x_1,\ldots,x_{r_s},X_1,\ldots,X_s)$$

be a sequence of first-order formulas of $L \cup \{X_1,\ldots,X_s\}$ each of which is *positive* in each X_i. In order words, each X_i occurs within an even number of negations. On every structure $\mathbf{M}$ over L, the sequence of formulas $(\varphi_1,\ldots,\varphi_s)$ defines s mappings $F_i : M^{s_1} \times \cdots \times M^{s_s} \to M^{s_i}$ by setting

$$F_i(A_1,\ldots,A_s) = \{(a_1,\ldots,a_{s_i}) \in M^{s_i} : \mathbf{M} \models \varphi_i[a_1,\ldots,a_{s_i},A_1,\ldots,A_s]\}.$$

Since the formulas are positive, it is not hard to see that each F_i is a *monotone operator*, that is, if $A_i \subseteq B_i$ for each $i \in \{1,\ldots,s\}$, then $F_i(A_1,\ldots,A_s) \subseteq F_i(B_1,\ldots,B_s)$ for each $i \in \{1,\ldots,s\}$. The *stages* are defined by setting $F^0 = (\emptyset,\ldots,\emptyset)$ and $F^{m+1} = (F_1(F^m),\ldots,F_s(F^m))$. If $A = (A_1,\ldots,A_s)$ is such that $A = (F_1(A),\ldots,F_s(A))$, we say that A is a *fixed-point*. If in addition every other fixed-point $B = (B_1,\ldots,B_s)$ is such that $A_i \subseteq B_i$ for every $i \in \{1,\ldots,s\}$, we say that A is the *least fixed-point*. The least fixed-point always exists, and if $\mathbf{M}$ is finite, some finite stage F^m reaches it. This follows from the Knaster-Tarki Theorem (see [EF95]). The least fixed-point is denoted by $F^\infty(\mathbf{M}) = (F_1^\infty(\mathbf{M}),\ldots,F_s^\infty(\mathbf{M}))$.

Each Datalog program specifies a system of monotone operators, also known as *simultaneous inductions*. Indeed, each IDB predicate X_i has an associated positive formula φ_i that is the disjunction of the existential closure of the bodies of the rules that have X_i in its head. The proper construction of φ_i requires some renaming of variables in order for each head to have the form $X_i(x_1,\ldots,x_{s_i})$. Then, the interpretation of a program with goal predicate X_g on a structure $\mathbf{M}$ over L is simply $\varphi_g^\infty(\mathbf{M})$, where $(\varphi_1^\infty(\mathbf{M}),\ldots,\varphi_s^\infty(\mathbf{M}))$ is the least fixed-point of the system $(\varphi_1,\ldots,\varphi_s)$. The interpretation of a Datalog program P on $\mathbf{M}$ is denoted by $P^{\mathbf{M}}$. This defines the semantics of Datalog. Let us see an example.

Example 1 Consider the following Datalog program over the vocabulary of directed graphs $\{E\}$, where X and Y are binary and unary second-order variables respectively:

$$\begin{aligned} X(x,y) &:- E(x,y) \\ X(x,y) &:- X(x,z), E(z,y) \\ Y(x) &:- X(x,x). \end{aligned}$$

The Datalog program recursively defines X as the set of all pairs of nodes (x,y) such that y is reachable from x in the graph. Also, it defines Y in terms of X as those nodes that are reachable from themselves. Thus, if Y is the goal predicate of the program, its interpretation on a graph is the set of nodes that belong to a (non-necessarily simple) cycle. The corresponding system of simultaneous inductions is

$$\begin{aligned} \varphi_X(x,y,X,Y) &= E(x,y) \vee (\exists z)(X(x,z) \wedge E(z,y)) \\ \varphi_Y(x,X,Y) &= X(x,x). \end{aligned}$$

We note that in this particular case, the system is quite trivial since, in fact, X and Y variables do not mix. Nonetheless, we will have genuine simultaneous inductions in the general case. Example 3 below is a genuine one. □

A careful writing of the system of simultaneous inductions corresponding to a Datalog program reveals that the defining formulas are existential positive. That is, each component of the system is a first-order formula that is formed from the atomic formulas by means of conjunctions, disjunctions and existential quantification only; no negations, and no universal quantification.

We are interested in the number of first-order variables of a Datalog program as a measure of its complexity. Thus, following Kolaitis and Vardi [KV00a], let k-Datalog be the class of all Datalog programs with at most k variables and IDB predicates of arity at most k. In the following Theorem we show that each k-Datalog program is equivalent to one component of a system of simultaneous inductions defined by an existential positive formula with at most k variables. This was already noted without proof by Kolaitis and Vardi [KV00a]. We provide a full proof for completeness.

Theorem 1 ([KV00a]) *Let $L = \{R_1, \ldots, R_r\}$ be a relational vocabulary, let $X_1, \ldots, X_s$ be second-order variables, and let s_i be the arity of X_i. Let P be a k-Datalog program over $L \cup \{X_1, \ldots, X_s\}$ with goal predicate X_1 such that $k \geq s_i$ for every $i \in \{1, \ldots, s\}$. Then, there exists a system of existential positive formulas*

$$\varphi_1(x_1, \ldots, x_{s_1}, X_1, \ldots, X_s), \ldots, \varphi_s(x_1, \ldots, x_{s_s}, X_1, \ldots, X_s)$$

with all variables among $\{x_1, \ldots, x_k\}$ such that for every structure $\mathbf{M}$ over L it holds that $P^{\mathbf{M}} = \varphi_1^\infty(\mathbf{M})$.

Proof: Let $V = \{x_1, \ldots, x_k\}$ be a set of variables that contains all variables that occur in the program. The main trick in the proof is to rename the variables of the rules in such a way that all heads with the same second-order variable are identical. Technically speaking, this is done as follows.

Suppose that R is a rule $t_0 :\!- t_1, \ldots, t_p$. The head of the rule is of the form $X_i(x_{\pi(1)}, \ldots, x_{\pi(s_i)})$ for some function $\pi : \{1, \ldots, s_i\} \to \{1, \ldots, k\}$. Now, let $\rho : \{1, \ldots, k\} \to \{1, \ldots, k\}$ be any permutation such that $\rho(\pi(j)) = \min\{j' : \pi(j') = \pi(j)\}$, where j and j' range over $\{1, \ldots, s_i\}$. Such a permutation exists because the minimal j' such that $\pi(j') = p$ is clearly unique for any $p \in \{\pi(1), \ldots, \pi(s_i)\}$, and because $k \geq s_i$. This is the only place in the proof where we need that k is at least as big as the arity of any IDB predicate. We let

$$\varphi_R = (\exists x_{i_1}) \cdots (\exists x_{i_q})(t'_1 \wedge \cdots \wedge t'_p),$$

where t'_i is the result of renaming each variable x_j by $x_{\rho(j)}$ in t_i. Here, $i_1, \ldots, i_q$ is the set of all indices of variables in $\{1, \ldots, k\} - \{\rho(\pi(1)), \ldots, \rho(\pi(s_i))\}$. In other words, φ_R is the result of renaming each variable of the rule according to the permutation ρ, and existentially quantifying all variables that are not within the new names $x_{\rho(\pi(1))}, \ldots, x_{\rho(\pi(s_i))}$ of the variables $x_{\pi(1)}, \ldots, x_{\pi(s_i)}$ in the head. Finally, let $C = \{(j, j') : \pi(j) = \pi(j')\}$ be the set of clashes of π, and let

$$\psi_R(x_1, \ldots, x_{s_i}) = \varphi_R \wedge \bigwedge_{(j,j') \in C} x_j = x_{j'}.$$

By now, we have succeeded to rename the variables of the rule R in such a way that the *head*, if we were to write the rule again, has the form $X_i(x_1, \ldots, x_{s_i})$. Now, if $R_1, \ldots, R_t$ are all the rules of the program having X_i in its head, we let

$$\varphi_i(x_1, \ldots, x_{s_i}, X_1, \ldots, X_s) = \psi_{R_1} \vee \cdots \vee \psi_{R_t}.$$

From its very construction, the first component of the system $(\varphi_1^\infty(\mathbf{M}), \ldots, \varphi_s^\infty(\mathbf{M}))$ is defining the semantics of the program P. □

Following [KV00a] we will denote the existential positive fragment of first-order logic with k variables by $\exists\mathrm{FO}^k$. Similarly, we let $\exists L^k_{\infty\omega}$ be the existential positive fragment of infinitary logic with k variables. That is, $\exists L^k_{\infty\omega}$ is the smallest class of formulas with at most k variables that is obtained by closing the atomic formulas under infinitary conjunctions, infinitary disjunctions, and existential quantification only. It was shown in [KV00a] that every component of the least fixed-point of a system of $\exists\mathrm{FO}^k$ formulas is expressible in $\exists L^k_{\infty\omega}$. In view of Theorem 1 we obtain:

Corollary 1 ([KV00a]) *Let k be a positive integer. Then k-Datalog $\subseteq \exists L^k_{\infty\omega}$, where k-Datalog is the class of all Datalog programs with at most k variables and IDB predicates of arity at most k.*

This result will be used to obtain non-expressibility results for Datalog from non-expressibility results for $\exists L^k_{\infty\omega}$. It turns out, as we will see, that the latter is a more amenable task because it has a nice combinatorial reformulation.

3 Formulas as Structures and Satisfiability as Homomorphism

Let $\{v_1, \ldots, v_n\}$ be a set of propositional variables. A *literal* is a variable v_i or the negation of a variable $\neg v_i$. A *clause* is a multiset of literals. A k-CNF *formula* is a multiset of clauses each of which has at most k literals. It is customary to write clauses $\{l_1, \ldots, l_k\}$ as disjunctions $l_1 \vee \cdots \vee l_k$, and CNF formulas as conjunctions of clauses $C_1 \wedge \cdots \wedge C_m$. A k-CNF formula F is *satisfiable* if there exists a *truth assignment* $f : \{v_1, \ldots, v_n\} \to \{0, 1\}$ that satisfies every clause C of F. That is, if $C = \{\neg v_{i_1}, \ldots, \neg v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}\}$ is a clause of C, then either $f(v_{i_p}) = 0$ for some $p \in \{1, \ldots, s\}$, or $f(v_{j_p}) = 1$ for some $p \in \{1, \ldots, k-s\}$.

There is a natural encoding of k-CNF formulas into finite relational structures. Let $L_k = \{R_0, \ldots, R_k\}$ be the vocabulary of $k+1$ relations of arity k. We encode a k-CNF formula F with variables $v_1, \ldots, v_n$ as a finite structure $\mathbf{M} = (M, R_0^\mathbf{M}, \ldots, R_k^\mathbf{M})$ over L_k. The universe M is the set of variables $\{v_1, \ldots, v_n\}$, and for each $s \in \{0, \ldots, k\}$ the relation $R_s^\mathbf{M}$ encodes the set of clauses of F with exactly s negated variables. More precisely, $R_s^\mathbf{M}$ consists of all k-tuples of the form

$$(v_{i_1}, \ldots, v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}) \in \{v_1, \ldots, v_n\}^k$$

such that $\{\neg v_{i_1}, \ldots, \neg v_{i_s}, v_{j_1}, \ldots, v_{j_{k-s}}\}$ is a clause of F. The encoding of F by such a finite structure will be denoted by $M(F)$.

Example 2 Consider the 3-CNF formula

$$F = (v_1 \vee \neg v_2 \vee v_3) \wedge (\neg v_2 \vee \neg v_3 \vee v_4) \wedge (v_1 \vee \neg v_3 \vee \neg v_4)$$

over the variables $v_1, \ldots, v_4$. Recall that, strictly speaking, the clauses of F are sets of literals. Thus, $C_1 = \{\neg v_2, v_1, v_3\}$, $C_2 = \{\neg v_2, \neg v_3, v_4\}$ and $C_3 = \{\neg v_3, \neg v_4, v_1\}$. The structure encoding F is $M(F) = (M, R_0, R_1, R_2, R_3)$ where $M = \{v_1, v_2, v_3, v_4\}$, $R_0 = \emptyset$, $R_1 = \{(v_2, v_1, v_3), (v_2, v_3, v_1)\}$, $R_2 = \{(v_2, v_3, v_4), (v_3, v_2, v_4), (v_3, v_4, v_1), (v_4, v_3, v_1)\}$ and $R_3 = \emptyset$. □

The encoding by means of finite structures allows us to use logic to define properties of k-CNF formulas. For example, the simple property that no clause is tautological is expressed by the following first-order formula:

$$(\forall x_1)\cdots(\forall x_k)\left(\bigwedge_{s=0}^{k}\left(R_s(x_1,\ldots,x_k)\rightarrow\bigwedge_{i\leq s<j}x_i\neq x_j\right)\right).$$

More sophisticated properties can be expressed in more powerful logics as our next example reveals.

Example 3 The purpose of this example is to show that the property "F is an unsatisfiable 2-CNF formula" is definable by a 3-Datalog program. The main idea amounts to the well-known fact that clauses of two literals may be seen as arcs of a directed graph, and that the unsatisfiability of a 2-CNF formula is characterized by a reachability property of that graph (see [Pap95, page 184]). Let F be a 2-CNF formula on the variables $v_1,\ldots,v_n$. Let $G=(V,E)$ be the following directed graph. The set of nodes V is the set of all possible literals $v_1,\ldots,v_n,\neg v_1,\ldots,\neg v_n$. The set of arcs is the set of all pairs of literals (l_1,l_2) such that $\{\neg l_1,l_2\}$ is a clause of F. Then, it is not hard to check that F is unsatisfiable if and only if there exists a variable v_i such that $\neg v_i$ is reachable from v_i and v_i is reachable from $\neg v_i$. We express this as a 3-Datalog program over the vocabulary L_2 of 2-CNF formulas with four binary IDB predicates PP, PN, NP, NN, and a propositional IDB goal P. The meaning of $PP(v_i,v_j)$ will be that v_j is reachable from v_i, the meaning of $PN(v_i,v_j)$ will be that $\neg v_j$ is reachable from v_i, the meaning of $NP(v_i,v_j)$ will be that v_j is reachable from $\neg v_i$, and the meaning of $NN(v_i,v_j)$ will be that $\neg v_j$ is reachable from $\neg v_i$. Thus, the names PP, PN, NP and NN stand for Positive-Positive, Positive-Negative, Negative-Positive and Negative-Negative respectively. The recursive definitions of each of these predicates by means of a Datalog program follows:

$$\begin{aligned}
PP(x,y) &:- R_1(x,y)\\
PN(x,y) &:- R_2(x,y)\\
NP(x,y) &:- R_0(x,y)\\
NN(x,y) &:- R_1(y,x)\\
PP(x,y) &:- PP(x,z),PP(z,y)\\
PP(x,y) &:- PN(x,z),NP(z,y)\\
PN(x,y) &:- PP(x,z),PN(z,y)\\
PN(x,y) &:- PN(x,z),NN(z,y)\\
NP(x,y) &:- NP(x,z),PP(z,y)\\
NP(x,y) &:- NN(x,z),NP(z,y)\\
NN(x,y) &:- NP(x,z),PN(z,y)\\
NN(x,y) &:- NN(x,z),NN(z,y)\\
P &:- PN(x,x),NP(x,x).
\end{aligned}$$

The recursive definition of $PP(v_i,v_j)$, for instance, reflects the fact that there are exactly two ways of reaching v_j from v_i by a path of length at least two: either we first reach a positive literal v_k,

and from that we reach v_j, or we first reach a negative literal $\neg v_k$, and from that we reach v_j. The interpretation of the rest of recursive definitions is similar. □

Before we conclude this section, let us introduce an alternative way of interpreting satisfiability of k-CNF formulas. Recall that a *homomorphism* between two relational structures $\mathbf{M}$ and $\mathbf{N}$ over the same vocabulary L is a mapping $f : M \to N$ such that if $(a_1, \ldots, a_r) \in R^{\mathbf{M}}$ for some R in L, then $(f(a_1), \ldots, f(a_r)) \in R^{\mathbf{N}}$. Now, let F be a k-CNF formula. Recall that a *truth assignment* is a mapping from the set of variables of F to $\{0, 1\}$. Moreover, a truth assignment satisfies a clause $v_{j_1} \vee \cdots \vee v_{j_k}$ exactly when f does not map the tuple $(v_{j_1}, \ldots, v_{j_k})$ to the tuple $(0, \ldots, 0)$, and similarly for the other types of clauses that have one, two, or more negations. Thus, we observe, as Feder and Vardi [FV98] do, that the satisfying truth assignments of F are exactly the homomorphisms between $M(F)$, the structure encoding F, and the following structure:

$$\begin{aligned} T_k &= (\{0,1\}, R_0^T, R_1^T, \ldots, R_k^T) \\ R_0^T &= \{0,1\}^k - \{(0,0,\ldots,0)\} \\ R_1^T &= \{0,1\}^k - \{(1,0,\ldots,0)\} \\ &\ldots \\ R_k^T &= \{0,1\}^k - \{(1,1,\ldots,1)\}. \end{aligned}$$

The structure T_k will be called the *template* structure from now on, and since most often in this paper $k = 3$, we will drop the subindex k in such a case. This interpretation of satisfiability in terms of homomorphisms will be exploited later.

4 Random Model of CNF Formulas

Let n, m and k be positive natural numbers. Let $F(n, m, k)$ be the set of all k-CNF formulas on the variables $\{v_1, \ldots, v_n\}$ with exactly m clauses each of which has exactly k literals on distinct variables. We let $\mathcal{F}(n, m, k)$ be the uniform distribution on the sample space $F(n, m, k)$. Observe that a particular formula $F = C_1 \wedge \ldots \wedge C_m$ of the sample space has probability

$$\left[2^k \binom{n}{k}\right]^{-m}.$$

Alternatively, $\mathcal{F}(n, m, k)$ can be described as the result of independently repeating the following experiment m times: choose exactly k variables from $\{v_1, \ldots, v_k\}$, and negate each variable independently with probability $1/2$. We will use this interpretation whenever it is convenient. The ratio m/n is denoted by Δ, and is called the *clause density*. Usually, Δ is fixed to a constant and therefore $m = \Delta n$ is determined by n. We are interested in studying the asymptotic properties of a randomly chosen formula $F \sim \mathcal{F}(n, \Delta n, k)$ as n approaches infinity.

It is well known that when the clause density Δ exceeds a certain constant that only depends on k, a randomly chosen formula is almost surely unsatisfiable. Let us remind the simple argument. Let $f : \{0, 1\}^n \to \{0, 1\}$ be a truth assignment. Observe that probability that a randomly chosen clause is falsified by f is $1/2^k$ since all chosen variables need to be assigned the wrong polarity.

Therefore, the probability that the clause is satisfied by f is $(1 - 1/2^k)$. Now, by independence of the clauses, the probability that a randomly chosen formula F is satisfied by f is

$$\left(1 - \frac{1}{2^k}\right)^m \leq e^{-m/2^k}.$$

Here we used the fact that $1 + x \leq e^x$ for every real number x. If we let X be the random variable that counts the number of satisfying assignments of a randomly chosen formula F, we have that $\mathrm{E}[X] = 2^n e^{-m/2^k}$ by linearity of expectation. By Markov's inequality

$$\Pr[X > 0] \leq \mathrm{E}[X] = 2^n e^{-m/2^k} = 2^n e^{-\Delta n/2^k}.$$

Finally, if $\Delta > 2^k \ln(2)$, the formula is unsatisfiable with probability approaching 1 as n approaches ∞. When $k = 3$, this bound gives the well-known number 5.19.

5 Main Result and Discussion of Proof Techniques

Let $3CNF_n$ be the class of all structures of the form $M(F)$ where F is a 3-CNF formula on at most n variables. Let $3SAT_n \subseteq 3CNF_n$ be the class of all structures of the form $M(F)$ where F is a satisfiable 3-CNF formula on at most n variables. Similarly, let $3UNSAT_n \subseteq 3CNF_n$ be the class of all structures of the form $M(F)$ where F is an unsatisfiable 3-CNF formula on at most n variables. The main result of this chapter is the following:

Theorem 2 *For all constants $\delta > 0$, $\Delta > 0$, and for all sufficiently large n, if $C_n \subseteq 3CNF_n$ is such that*

(i) $C_n \subseteq 3UNSAT_n$ *and*

(ii) $\Pr[M(F) \in C_n] \geq \delta$ *when F is drawn from $\mathcal{F}(n, \Delta n, 3)$,*

then C_n is not definable in $\exists L^k_{\infty\omega}$ on $3CNF_n$ for $k \leq n/(\ln(n))^2$.

In words, what this result says is that every property of 3-CNF formulas that implies unsatisfiability and is expressible in $\exists L^k_{\infty\omega}$ with $k \leq n/(\ln(n))^2$ must have asymptotic probability 0. Let us point out that the hypothesis that δ and Δ are constants is only enforced to simplify the calculations and to obtain a cleaner statement. If δ and Δ were monotone functions tending to 0 and ∞ respectively, we would obtain a similar result with these parameters appearing in the bound on k. For example, if δ is constant and Δ is an arbitrary increasing function, we obtain the same result with k bounded by $n/M^{2+\gamma}$ for arbitrarily small $\gamma > 0$, where $M = \max(\Delta, \ln(n))$. We return to this example after the proof of the theorem in Section 7.

Before we discuss how Theorem 2 is proved, let us state the consequence for Datalog. Let $3CNF = \bigcup_{n \geq 1} 3CNF_n$ and $3UNSAT = \bigcup_{n \geq 1} 3UNSAT_n$. Since k-Datalog is included in $\exists L^k_{\infty\omega}$ by Corollary 1, the following holds:

Corollary 2 *Let $\Delta > 0$ be a constant. Every Datalog Boolean query on $3CNF$ that implies $3UNSAT$ must have asymptotic probability 0 in $\mathcal{F}(n, \Delta n, 3)$.*

Theorem 2 is an inexpressibility result since it asserts that certain Boolean queries are not expressible in $\exists L^k_{\infty\omega}$ on $3CNF$. The purpose of the rest of this section is to discuss the techniques that are available to prove such a result.

Consider the following game played by two players, Spoiler and Duplicator, on a board formed by two structures $\mathbf{A}$ and $\mathbf{B}$ over the same relational vocabulary. Each player has k pebbles numbered $\{1, \ldots, k\}$. At each round of the game, Spoiler can make one of two different moves: either he places a free pebble over an element of $\mathbf{A}$, or he removes a pebble from a pebbled element of $\mathbf{A}$. To each move of Spoiler, Duplicator must respond by placing her corresponding pebble over an element of $\mathbf{B}$, or removing her corresponding pebble from $\mathbf{B}$ respectively. By corresponding pebble we mean the pebble that has the same number as that chosen by Spoiler. If a round is reached in which the correspondence $a_i \mapsto b_i$ that assigns pebbled elements of $\mathbf{A}$ to the corresponding pebbled elements of $\mathbf{B}$ is not a partial homomorphism between $\mathbf{A}$ and $\mathbf{B}$, then we say that Spoiler wins. Otherwise, Duplicator wins. Recall that a partial homomorphism from $\mathbf{A}$ to $\mathbf{B}$ is a function $f : A' \to B'$, where $A' \subseteq A$ and $B' \subseteq B$, such that if $a_1, \ldots, a_r \in A'$ and $(a_1, \ldots, a_r) \in R^{\mathbf{A}}$, then $(f(a_1), \ldots, f(a_r)) \in R^{\mathbf{B}}$. The following definition makes the game formal:

Definition 1 *Let L be a relational vocabulary, and let $\mathbf{A}$ and $\mathbf{B}$ be two structures over L. A winning strategy for Duplicator in the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$ is a non-empty set $\mathcal{H}$ of partial homomorphisms from $\mathbf{A}$ to $\mathbf{B}$ such that:*

(i) If $f \in \mathcal{H}$, then $|\mathrm{Dom}(f)| \leq k$.

(ii) If $f \in \mathcal{H}$ and $g \subseteq f$, then $g \in \mathcal{H}$.

(iii) If $f \in \mathcal{H}$, $|\mathrm{Dom}(h)| < k$ and $a \in A$, then there is some $b \in B$ and $g \in \mathcal{H}$ such that $f \subseteq g$ and $b \in \mathrm{Dom}(g)$.

The main result about existential k-pebble games is that they characterize expressibility in the logic $\exists L^k_{\infty\omega}$ in the following sense:

Theorem 3 ([KV00a]) *Let L be a relational vocabulary, let $\mathcal{C}$ be a class of finite structures over L, and let Q be a Boolean query on $\mathcal{C}$. Then, the following are equivalent:*

(i) Q is definable in $\exists L^k_{\infty\omega}$ on $\mathcal{C}$,

(ii) For every two structures $\mathbf{A}, \mathbf{B} \in \mathcal{C}$, if $\mathbf{A} \models Q$ and Duplicator wins the existential k-pebble game on $\mathbf{A}$ and $\mathbf{B}$, then $\mathbf{B} \models Q$.

Equipped with the tool of the existential pebble games, the proof of Theorem 2 will proceed as follows. Suppose that C_n satisfies conditions (i) and (ii) in the hypothesis of the theorem, and that C_n is definable on $3CNF_n$ by an $\exists L^k_{\infty\omega}$-sentence φ with $k \leq n/(\ln(n))^2$. The core of the proof will be a probabilistic argument that, using (ii), will show that there exists some 3-CNF formula F such that $M(F) \in C_n$ and Duplicator wins the existential k-pebble game on $M(F)$ and T, where T is the template structure defined in Section 3. Then we apply Theorem 3 and infer that $T \models \varphi$, and so $T \in C_n$ because $T \in 3CNF_n$ and φ defines C_n on $3CNF_n$. By (i), T must be the encoding of an unsatisfiable 3-CNF formula. However, this is absurd because the identity mapping is a trivial homomorphism from T to T. Indeed, T is the encoding of the unique 3-CNF formula on the variables $\{v_1, v_2\}$ that contains every clause except those falsified by the truth assignment $v_1 \mapsto 0$ and $v_2 \mapsto 1$. Consequently, T is the encoding of a satisfiable 3-CNF formula.

6 Extension Axioms and the Matching Game

In order to execute the proof idea that we outlined in Section 5 we will need to develop some combinatorial machinery. We do it in this section.

A directed graph $G = (M, E)$ is called *bipartite* if there is a partition of the set of nodes M into two sets U and V such that $E \subseteq U \times V$. A *partial matching π of G* is a subset of the edges not sharing an endpoint. If $(u, v) \in \pi$, we write $\pi(u) = v$. Note that this notation is not ambiguous. We say that *π is defined on u*, and we define the *domain* $\mathrm{Dom}(\pi)$ *of* π as the set of nodes of U on which π is defined. We also define the *range* $\mathrm{Ran}(\pi)$ *of* π as $\{\pi(u) : u \in \mathrm{Dom}(\pi)\}$. If $A \subseteq U$, we say that *A is matchable into V in G* if there exists a partial matching π of G such that $\mathrm{Dom}(\pi) = A$. If $u \in U$, we let $N_G(u) = \{v \in V : (u, v) \in E\}$, and if $v \in V$, we let $N_G(v) = \{u \in U : (u, v) \in E\}$. The left-degree (resp. right-degree) of G is the maximum size of $N_G(u)$ as u ranges over U (resp. V). If $A \subseteq U$, then $N_G(A) = \bigcup_{u \in A} N_G(u)$. For $\epsilon > 0$, we say that *G is an (s, ϵ)-bipartite expander* if for every $A \subseteq U$ of size at most s, the size of $N_G(A)$ is at least $(1 + \epsilon)|A|$. Observe that in an (s, ϵ)-bipartite expander, every subset of U of size at most s is matchable into V. This is true by Hall's Theorem:

Theorem 4 (Hall's Theorem) *If $G = (U \cup V, E)$ is a bipartite graph on U and V, and $A \subseteq U$, then A is matchable into V in G if and only if $|N_G(B)| \geq |B|$ for every $B \subseteq A$.*

Observe also that an immediate consequence of Hall's Theorem is that if $A \subseteq U$ is minimally non-matchable into V, then $|N_G(A)| < |A|$. We will use this observation later.

Let $G = (U \cup V, E)$ be a bipartite graph on U and V. The *matching game on G with k fingers* was introduced by Ben-Sasson and Galesi [BSG01]. The game is played by two players: Prover and Disprover. Each player has k fingers numbered $\{1, \ldots, k\}$. In each round of the game, Prover may place a finger over an uncovered node in U or remove a finger from a covered node in U. If Prover places a finger over node $u \in U$, Disprover must place her corresponding finger over an uncovered node in $N_G(u) \subseteq V$. If Prover removes a finger from a node in U, Disprover must remove her corresponding finger from V. The game is over when Disprover is not able to answer to a move of the Prover. In that case, we say that Prover wins the game. If Disprover can make the game go on forever, we say that Disprover wins the game. Notice that at every non-final round, the fingers placed on G determine a partial matching of G. The goal of Disprover is to *maintain* a partial matching forever.

Ben-Sasson and Galesi found a sufficient condition on the graph G that guarantees a win for Disprover. They proved that if G is an (s, ϵ)-bipartite expander, then Disprover wins the matching game with $r \leq \epsilon s/(2 + \epsilon)$ fingers. Although our main result would already follow from this, we prefer to give an alternative and self-contained proof of a similar result with the aim of isolating the extension axioms on which the strategy of Disprover relies. The result that we obtain is that if G is an (s, ϵ)-bipartite expander of left-degree at most l, then Disprover wins the matching game with $r \leq \epsilon s/(l + \epsilon)$ fingers. Although our result is weaker than the one of Ben-Sasson and Galesi, our proof is somewhat simpler (Claim 4.6 and 4.7 in [BSG01] are avoided). Moreover, we will consider bipartite graphs whose left-degree is 3, and therefore the weakness of our result is really minor.

Let us isolate an important property of bipartite graphs.

Definition 2 *Let $r \leq s$, let $G = (U \cup V, E)$ be a bipartite graph on U and V, and let $A \subseteq U$ and $B \subseteq V$ be such that $|A| = |B| \leq r$. We say that (G, A, B) satisfies the (r, s)-matching property if the following holds: For every $C \subseteq U - A$ of size $s - |B|$, there exists a partial matching from C to $V - B$ in G.*

The following is the main result of this section.

Theorem 5 ([BSG01]) *Let G be an (s, ϵ)-bipartite expander of left-degree at most l, and let r be a positive integer bounded by $\epsilon s/(l + \epsilon)$. Then, Disprover wins the matching game on G with r fingers.*

Proof: The strategy of Disprover is to guarantee that $(G, \mathrm{Dom}(\pi_t), \mathrm{Ran}(\pi_t))$ satisfies the (r, s)-matching property, where π_t is the partial matching of the game at the end of round t. Observe that $(G, \emptyset, \emptyset)$ satisfies the (r, s)-matching property because in an (s, ϵ)-bipartite expander, every subset of U of size s is matchable into V. The following two lemmas will be used to show that Disprover can reply to the moves that Prover makes according to her strategy.

Lemma 1 (Extension Axiom) *If (G, A, B) satisfies the (r, s)-matching property, $u \in U - A$, and $|A| = |B| < r$, then there exists $v \in N_G(u) \cap (V - B)$ such that $(G, A \cup \{u\}, B \cup \{v\})$ satisfies the (r, s)-matching property.*

Proof: Let $\{v_1, \ldots, v_e\} = N_G(u) \cap (V - B)$. Clearly $e \leq l$ because the left-degree of G is at most l. Also $e \geq 1$ because $u \in U - A$ and every subset of $U - A$ of size $s - |B| > s - r \geq 0$ is matchable into $V - B$. Let $A' = A \cup \{u\}$, and for every $i \in \{1, \ldots, e\}$ let $B^i = B \cup \{v_i\}$. Note that $|A'| = |B^i| \leq r$ since $|A| = |B| < r$. Suppose for contradiction that (G, A', B^i) does not satisfy the (r, s)-matching property for any $i \in \{1, \ldots, e\}$. Let $C^i \subseteq U - A'$ be of size $s - |B^i|$ and non-matchable into $V - B^i$. Let $D^i \subseteq C^i$ be minimally non-matchable into $V - B^i$. Necessarily, $|N_G(D^i) \cap (V - B^i)| < |D^i|$ by Hall's Theorem. The rest of neighbors of D^i can only be in B^i, so

$$|D^i| + |B^i| > |N_G(D^i)| \geq (1 + \epsilon)|D^i| \tag{1}$$

for every $i \in \{1, \ldots, e\}$. Recall that $|D^i| \leq |C^i| = s - |B^i| \leq s$ and the expansion property of G for the second inequality. We derive a lower bound on $|D^i|$ for some $i \in \{1, \ldots, e\}$. We first claim that $\bigcup_{i=1}^{e} D^i \cup \{u\}$ is not matchable into $V - B$. Indeed, suppose that $\pi \subseteq E$ is such a matching, and let E^i be the image $\pi(D^i)$ of D^i. Then $v_i \in E^i$ since D^i is minimally non-matchable into $V - B^i$. Therefore, u cannot be matched to any of its neighbors in $V - B$; a contradiction. Now, since $\bigcup_{i=1}^{e} D^i \cup \{u\} \subseteq U - A$ and (G, A, B) satisfies the (r, s)-matching property, we conclude that

$$\left| \bigcup_{i=1}^{e} D^i \cup \{u\} \right| > s - |B|.$$

Therefore, using $|B^i| = |B| + 1$, there must exist some $i \in \{1, \ldots, e\}$ such that

$$|D^i| > \frac{s - |B^i|}{e} \geq \frac{s - |B^i|}{l}.$$

Plugging this bound into (1) we get

$$|B^i| > \frac{\epsilon s}{l + \epsilon} \geq r,$$

a contradiction. □

Lemma 2 (Retraction Axiom) *If (G, A, B) satisfies the (r, s)-matching property and $(u, v) \in E \cap (A \times B)$, then $(G, A - \{u\}, B - \{v\})$ satisfies the (r, s)-matching property.*

Proof: Let $A' = A - \{u\}$ and $B' = B - \{v\}$. Clearly $|A'| = |B'| \leq r$. Let $C \subseteq U - A'$ be of size $s - |B'|$. Suppose first that $u \in C$. Then $C - \{u\} \subseteq U - A$ and has size $s - |B'| - 1 = s - |B|$. Therefore, $C - \{u\}$ is matchable into $V - B$ since (G, A, B) satisfies the (r, s)-matching property. It follows that C is matchable into $V - B'$ by adding the edge (u, v) to that matching. Suppose next that $u \notin C$. For every $w \in C$, the set $C - \{w\} \subseteq U - A$ is matchable into $V - B \subseteq V - B'$ since (G, A, B) satisfies the (r, s)-matching property. It follows that if C is not matchable into $V - B'$, then it is minimally non-matchable, and so $|N_G(C) \cap (V - B')| < |C|$ by Hall's Theorem. The rest of neighbors of C can only be in B' and therefore

$$|C| + |B'| > |N_G(C)| \geq (1 + \epsilon)|C|.$$

Recall that $|C| = s - |B'| \leq s$ and the expansion property of G for the second inequality. We conclude that

$$|B'| > \frac{\epsilon s}{1 + \epsilon} \geq \frac{\epsilon s}{l + \epsilon} \geq r,$$

a contradiction. □

We complete the proof. Suppose that $(G, \mathrm{Dom}(\pi_t), \mathrm{Ran}(\pi_t))$ satisfies the (r, s)-matching property. Consider the move of Prover at time $t + 1$. If Prover places a finger over an uncovered node $u \in U - \mathrm{Dom}(\pi_t)$, Disprover answers by some $v \in N_G(u) \cap (V - \mathrm{Ran}(\pi_t))$ such that $(G, \mathrm{Dom}(\pi_t) \cup \{u\}, \mathrm{Ran}(\pi_t) \cup \{v\})$ satisfies the (r, s)-matching property. Such a v exists by Lemma 1 because $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| < r$. If Prover removes a finger from a node $u \in \mathrm{Dom}(\pi_t)$, Disprover removes the corresponding finger from $v = \pi_t(u)$, and $(G, \mathrm{Dom}(\pi_t) - \{u\}, \mathrm{Ran}(\pi_t) - \{v\})$ satisfies the (r, s)-matching property by Lemma 2. □

7 Proof of Main Result

Let $F = C_1 \wedge \ldots \wedge C_m$ be a 3-CNF formula on the variables $v_1, \ldots, v_n$. We define the *incidence graph $G(F)$ of F* as follows: $G(F)$ is a bipartite graph on the sets of nodes $\{C_1, \ldots, C_m\}$ and $\{v_1, \ldots, v_n\}$, and (C_i, v_j) is an edge in $G(F)$ if and only if $v_j \in Var(C_i)$. Here $Var(C_i)$ is used to denote the set of variables that occur in C_i. Notice that the left-degree of $G(F)$ is at most 3.

Let F be a 3-CNF formula, let V be a set of variables from F, and let $\mathcal{C} \subseteq F$ be the subset of the clauses of F that mention some variable from V. In symbols,

$$\mathcal{C} = \{C \in F : Var(C) \cap V \neq \emptyset\}.$$

Let π be a partial matching of $G(F)$ such that $\mathrm{Dom}(\pi) = \mathcal{C}$. We define the *partial truth assignment ρ corresponding to π and V* as follows: For every $x \notin V$, the partial truth assignment ρ is undefined on x. If $x \in V$ and $\pi(C) = x$ for some $C \in \mathcal{C}$, then $\rho(x) = 0$ if x occurs both positively and negatively in C, $\rho(x) = 1$ if x occurs positively in C only, and $\rho(x) = 0$ if x occurs negatively in C only. Finally, if $x \in V$ but $x \notin \mathrm{Ran}(\pi)$, then $\rho(x) = 0$. Note that ρ is well-defined because π is a partial matching. In the following, we say that a partial truth assignment ρ *falsifies a clause C* if ρ sets all literals of C to 0.

Lemma 3 *Let F be a 3-CNF formula and let ρ be the partial truth assignment corresponding to some partial matching π of $G(F)$ and some set of variables V. Then ρ does not falsify any clause from F.*

Proof: Let $\mathcal{C}$ be the set of clauses from F that mention some variable in V. Let C be an arbitrary clause from F. If $C \notin \mathcal{C}$, then every variable of C is left undefined by ρ and C cannot be falsified. If $C \in \mathcal{C}$, then $\pi(C)$ occurs in C and $\rho(\pi(C))$ is set to satisfy C. □

The next result connects the matching game and the existential pebble game.

Theorem 6 *Let F be a 3-CNF formula such that $G(F)$ has right-degree at most d. If Disprover wins the matching game with r fingers on $G(F)$, then Duplicator wins the existential $\lfloor r/d \rfloor$-pebble game on $M(F)$ and T.*

Proof: Let $G = G(F)$ and $r' = \lfloor r/d \rfloor$. Thus, G is a bipartite graph between the clauses and the variables of F. Duplicator will simulate a play of the matching game on G on the side trying to satisfy the following condition: clause C has a finger on it in the matching game at time t if and only if some variable $x \in Var(C)$ has a pebble on it in the existential r'-pebble game at time t. In other words, if V_t is the set of pebbled variables in the existential r'-pebble game at the end of round t, and $\mathcal{C}_t$ is the set of clauses of F that mention some variable in V_t, then $\mathcal{C}_t$ is the set of fingered clauses in the matching game at the end of round t. Since at most r' variables are pebbled simultaneously in the existential r'-pebble game, and since the right degree of G is at most d, at most $r'd \leq r$ fingers will be required. Duplicator will make use of the winning strategy of Disprover in the matching game on G with r fingers. In the following, let π_t be the partial matching in the matching game at the end of round t. Note that $\pi_0 = \emptyset$ and $\mathrm{Dom}(\pi_t) = \mathcal{C}_t$. We consider the two possible moves of Spoiler at time $t+1$ separately.

If Spoiler puts a pebble on variable x, Duplicator responds as follows. Let $\mathcal{D} = \{C_1, \ldots, C_l\}$ be the set of clauses of F on which variable x occurs. Duplicator considers each clause of $\mathcal{D}$ one at a time, starting by C_1. If C_1 does not have a finger on it in the matching game at time t, Duplicator simulates the winning strategy of Disprover as if Prover had placed a finger over C_1, and proceeds to C_2. Otherwise, she proceeds to C_2 directly. This procedure is repeated until C_l is considered and treated. Let π_{t+1} be the partial matching in the matching game at that point. Then Duplicator replies in the existential r'-pebble game as follows: If some C_i exists such that $\pi_{t+1}(C_i) = x$, Duplicator answers 0 if x occurs both positively and negatively in C_i, answers 1 if x occurs positively in C_i only, and answers 0 if x occurs negatively in C_i only. Otherwise, Duplicator answers 0. Note for the record that Duplicator is answering consistently with the partial truth assignment corresponding to π_{t+1} and V_{t+1}, the partial matching and the set of pebbled variables in the existential r'-pebble game at time $t+1$.

If Spoiler removes a pebble from variable x, the answer of Duplicator is clear: she removes the corresponding pebble from T. However, she will need to do some bookkeeping in the matching game: Let $\mathcal{D} = \{C_1, \ldots, C_l\}$ be the set of clauses of F on which variable x occurs. Duplicator considers each clause of $\mathcal{D}$ one at a time, starting by C_1. If some other variable $y \in Var(C_1) - \{x\}$ has a pebble on it in the existential r'-pebble game at time t, Duplicator proceeds to C_2. If there is no $y \in Var(C_1) - \{x\}$ as above, Duplicator simulates the matching game as if Prover had removed his finger from C_1. Then Duplicator proceeds to C_2. This procedure is repeated until C_l is considered and treated. Let π_{t+1} be the partial matching in the matching game at that point.

We need to argue two points: (1) that Duplicator can follow this strategy, and (2) that if she does, she wins the existential r'-pebble game on $M(F)$ and T. For (1), recall that each variable occurs in at most d clauses, and therefore, since at most $r' = \lfloor r/d \rfloor$ variables are pebbled in the existential r'-pebble game, at most r fingers are used in the matching game. It follows that Duplicator can simulate the winning strategy of Disprover in the matching game. For (2), suppose that Spoiler has not won at time t; we show that he does not win at time $t+1$. When Spoiler removes a pebble, Duplicator cannot lose since she removes the corresponding pebble from T. When Spoiler puts a new pebble, Duplicator answers according to the partial truth assignment corresponding to the partial matching π_{t+1} and the set of pebbled variables V_{t+1}, and this cannot falsify a clause from F by Lemma 3. This completes the proof of the Theorem. □

Finally, we are ready for the proof of the main result.

Proof of Theorem 2: Fix $\delta > 0$, $\Delta > 0$, and let n be large. Let $k \leq n/(\ln(n))^2$. Suppose for contradiction that C_n satisfies (i) and (ii), and φ is an $\exists L^k_{\infty\omega}$ sentence defining C_n on $3CNF_n$. We will show that there exists a 3-CNF formula F such that $M(F) \in C_n$ and $G(F)$ is an $(n/\ln(n), 1/4)$-bipartite expander of right-degree at most $d = \ln(n)/13$ and left-degree at most 3. This will be enough because then: (1) $M(F) \models \varphi$ and (2) Duplicator wins the existential k-pebble game on $M(F)$ and T by Theorem 5 and Theorem 6. This means that $T \models \varphi$ by Theorem 3, so that $T \in C_n \subseteq \text{3UNSAT}_n$; a contradiction since T is isomorphic to a satisfiable 3-CNF formula on at most n variables (in fact, on two variables).

Let us prove the existence of F as above by the probabilistic method. First we bound the probability that the right degree of $G(F)$ is bigger than d. Let $d_R(G(F))$ denote the right-degree of the bipartite graph $G(F)$. The proof of the following Lemma can be found in the Appendix.

Lemma 4 *Let F be drawn from $\mathcal{F}(n, \Delta n, 3)$. Then,*

$$\Pr\left[d_R(G(F)) \geq d\right] \leq n\left(\frac{3e\Delta}{d}\right)^d.$$

Next we bound the probability that $G(F)$ is not an (s,ϵ)-bipartite expander. Similar calculations have been done many times, starting with [CS88]. Our proof is in the Appendix too.

Lemma 5 *Let $s > 0$ and $\epsilon > 0$ be such that $(1+\epsilon)s \leq n$ and let F be drawn from $\mathcal{F}(n, \Delta n, 3)$. Then,*

$$\Pr\left[G(F) \textit{ not } (s,\epsilon)\textit{-bipartite expander}\right] \leq \frac{C(1-C^s)}{1-C},$$

where $C = \Delta e^{2+\epsilon}(1+\epsilon)^{2-\epsilon}(s/n)^{1-\epsilon}$.

Finally, we bound the probability that the right-degree of $G(F)$ is bigger than d, or $G(F)$ is not an (s,ϵ)-expander, or $M(F)$ does not belong to C_n by

$$n\left(\frac{3e\Delta}{d}\right)^d + \frac{C(1-C^s)}{1-C} + 1 - \delta. \tag{2}$$

From our choice of $d = \ln(n)/13$, $s = n/\ln(n)$, and $\epsilon = 1/4$, we immediately see that (2) is strictly smaller than 1 for sufficiently large n (recall that $\delta > 0$ and $\Delta > 0$ are constants). Thus,

some F must exist for which $G(F)$ is an (s, ϵ)-expander of right-degree at most d, and $M(F)$ belongs to C_n. This completes the proof of the Theorem. □

Before we close this section, we discuss how the proof would be modified to obtain the result mentioned after the statement of the theorem in Section 5. Namely, we want to show that if δ is still constant but Δ is an increasing function of n we obtain the same result with k bounded by $n/M^{2+\gamma}$ for arbitrarily small $\gamma > 0$, where $M = \max(\Delta, \ln(n))$. The proof would proceed by cases according to whether Δ is bounded by $3\ln(n)$ or not. In the former case we proceed as before. In the latter case, for an arbitrarily small $\gamma > 0$, we choose the parameter $d = 6\Delta$, let $\epsilon > 0$ be smaller than $\gamma/(1+\gamma)$, and let $s = 6n/\Delta^{1+\gamma}$. Then we replace Lemma 4 by an application of Chernoff bounds [HR89], and check that the resulting expression replacing (2) is strictly smaller than 1 for sufficienly large n.

8 Easier Games

The aim of this section is to show that certain combinatorial statements such as the Pigeonhole Principle, when encoded as contradictory sets of clauses in the spirit of Cook and Reckhow [CR79], may yield structures on which playing pebble games is easier than expected.

The Pigeonhole Principle states that there is no one-to-one mapping from a set of $n+1$ pigeons into a set of n holes. This statement may be expressed in propositional logic as follows. For $i \in \{1, \ldots, n+1\}$ and $j \in \{1, \ldots, n\}$, let $p_{i,j}$ be a propositional variable meaning that pigeon i is sitting in hole j. The clause $P_i = p_{i,1} \vee \ldots \vee p_{i,n}$ says that pigeon i sits in some hole. The clause $H_k^{i,j} = \neg p_{i,k} \vee \neg p_{j,k}$ for $i \neq j$ says that pigeons i and j cannot sit in the same hole k. The conjunction PHP_n^{n+1} of all the clauses P_i and $H_k^{i,j}$ is an unsatisfiable CNF since it expresses the negation of the Pigeonhole Principle. Note that PHP_n^{n+1} is not a 3-CNF formula since the clauses P_i involve n literals. However, it is easy to turn it into an equivalent 3-CNF as follows. For every $i \in \{1, \ldots, n+1\}$ and $j \in \{0, \ldots, n\}$, let $y_{i,j}$ be a new propositional variable. Then EP_i is the following 3-CNF formula: $\neg y_{i,0} \wedge \bigwedge_{j=1}^{n}(y_{i,j-1} \vee p_{i,j} \vee \neg y_{i,j}) \wedge y_{i,n}$. Finally, the 3-CNF formula $EPHP_n^{n+1}$ expressing the negation of the Pigeonhole Principle is the conjunction of all clauses EP_i and all clauses $H_k^{i,j}$. Here is our promised result about pebble games and the Pigeonhole Principle:

Theorem 7 *Duplicator wins the existential n-pebble game on $M(EPHP_n^{n+1})$ and T.*

Proof: Consider the following strategy for Duplicator. She will keep track of a partial matching π_t in the complete bipartite graph K_n^{n+1} that satisfies the following property: Pigeon $i \in \{1, \ldots, n+1\}$ belongs to $\mathrm{Dom}(\pi_t)$ if and only if some variable of pigeon i is pebbled at the end of round t. The variables of pigeon i are $p_{i,1}, \ldots, p_{i,n}$ and $y_{i,0}, \ldots, y_{i,n}$. Notice that if there are p pebbles on the board at the end of round t, then $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| \leq p$. Initially $\pi_0 = \emptyset$. Let us consider the possible moves of Spoiler at time $t+1$ separately:

If Spoiler places a pebble over a variable $p_{i,j}$, Duplicator acts as follows: If $i \in \mathrm{Dom}(\pi_t)$, Duplicator answers 1 if $j = \pi_t(i)$ and 0 otherwise. In that case, $\pi_{t+1} = \pi_t$. If $i \notin \mathrm{Dom}(\pi_t)$, Duplicator finds some $k \in \{1, \ldots, n\} - \mathrm{Ran}(\pi_t)$ and sets $\pi_{t+1} = \pi_t \cup \{(i,k)\}$. Note that such a k exists since at most $n-1$ pebbles are placed on the board at the end of round t, and therefore $|\mathrm{Dom}(\pi_t)| = |\mathrm{Ran}(\pi_t)| \leq n-1$. Then she answers 1 if $j = k$ and 0 otherwise.

If Spoiler places a pebble over a variable $y_{i,j}$, Duplicator acts as follows: If $i \in \mathrm{Dom}(\pi_t)$, Duplicator answers 1 if $j \geq \pi_t(i)$ and 0 otherwise. In that case, $\pi_{t+1} = \pi_t$. If $i \notin \mathrm{Dom}(\pi_t)$, Duplicator finds some $k \in \{1, \ldots, n\} - \mathrm{Ran}(\pi_t)$ and sets $\pi_{t+1} = \pi_t \cup \{(i,k)\}$. Such a k exists for the same reason as above. Then she answers 1 if $j \geq k$ and 0 otherwise.

If Spoiler removes a pebble from a variable, the answer of Duplicator is clear: she removes the corresponding pebble in T. However, she will need to do some bookkeeping in the partial matching π_t. Suppose that the variable from which Spoiler removed the pebble belongs to pigeon i. If no other variable of pigeon i is pebbled at time t, Duplicator sets $\pi_{t+1} = \pi_t - \{(i, \pi_t(i))\}$. Otherwise, Duplicator sets $\pi_{t+1} = \pi_t$.

It is not hard to see that Duplicator wins the existential n-pebble game when playing according to this strategy since a clause is never falsified. □

We can do the same with a form of the Pigeonhole Principle called $G\text{-}PHP$ that was introduced by Ben-Sasson and Wigderson [BSW01]. Let U and V be disjoint sets of sizes $n+1$ and n respectively. Let $G = (U \cup V, E)$ be a bipartite graph on U and V. Obviously, there is no matching from U into V. This is expressed by the conjunction of the following set of clauses on the variables $\{x_{u,v} : (u,v) \in E\}$. For every $u \in U$, let $P_u = \bigvee_{(u,v)\in E} x_{u,v}$, and for every $(u,w),(v,w) \in E$, $u \neq v$, let $H_w^{u,v} = \neg x_{u,w} \vee \neg x_{v,w}$. We write $G\text{-}PHP$ for the conjunction of these clauses. Observe that if $G = K_n^{n+1}$, then $G\text{-}PHP$ coincides with PHP_n^{n+1}. Observe also that if G has left-degree at most k, then $G\text{-}PHP$ is a k-CNF.

We assign a partial truth assignment ρ to every partial matching π in G. For every $u \in \mathrm{Dom}(\pi)$, let $\rho(x_{u,\pi(u)}) = 1$ and $\rho(x_{u,v}) = 0$ for every $v \in V - \{\pi(u)\}$ such that $(u,v) \in E$.

Lemma 6 *Let $G = (U \cup V, E)$ be a bipartite graph, let π be a partial matching in G, and let ρ be the partial truth assignment corresponding to π. Then, ρ does not falsify any clause from $G\text{-}PHP$.*

Proof: The clauses of the form $\neg x_{u_1,v} \vee \neg x_{u_2,v}$ are not falsified by ρ since at most one of (u_1,v) or (u_2,v) is in the partial matching π. Similarly, the clauses of the form $x_{u,v_1} \vee \ldots \vee x_{u,v_d}$ are not falsified by ρ since if $\rho(x_{u,v_j}) = 0$ for some $j \in \{1,\ldots,d\}$, then $\rho(x_{u,v_i}) = 1$ for some other $i \in \{1,\ldots,d\}$. □

The proof of the following result is very similar to the proof of Theorem 6.3 in [BSG01]. Nonetheless, for all we know at this point, neither statement seems to imply the other[1].

Theorem 8 *Let $G = (U \cup V, E)$ be an (s,ϵ)-expander of left-degree at most l, and let r be a positive integer bounded by $\epsilon s/(l+\epsilon)$, then Duplicator wins the existential r-pebble game on $M(G\text{-}PHP)$ and T.*

Proof: We design a strategy for Duplicator to win the existential r-pebble game on the structures $M(G\text{-}PHP)$ and T. Duplicator will simulate the play of a matching game on G on the side trying to satisfy the following condition: Node $u \in U$ has a finger on it in the matching game at time t if and only if there exists some $v \in N_G(u)$ such that $x_{u,v}$ has a pebble on it in the existential r-pebble game at time t. This condition is clearly satisfied at time $t = 0$. Duplicator will make use of the

[1]But see the paragraph on recent developments in Section 10. It turns out that our statement does imply the statement in [BSG01] in a rather indirect way. The converse is not known.

winning strategy of Disprover in the matching game on G with r fingers. In the following, let π_t be the partial matching in the matching game at time t, so that $\pi_0 = \emptyset$. We consider the two possible moves of Spoiler at time $t+1$ separately.

If at time $t+1$ Spoiler removes a pebble from variable $x_{u,v}$ in the existential r-pebble game, Duplicator acts as follows: Note that u must have a finger on it at time t in the matching game since $x_{u,v}$ has a pebble at time t in the existential r-pebble game. Thus, $\pi_t(u)$ is well-defined. If some $w \in V - \{\pi_t(u)\}$ exists such that the variable $x_{u,w}$ is pebbled in the existential r-pebble game, Duplicator does nothing in the matching game, so that $\pi_{t+1} = \pi_t$. Then she removes the pebble from T corresponding to $x_{u,v}$ in the existential r-pebble game. On the other hand, if no $w \in V - \{\pi_t(u)\}$ exists as above, Duplicator removes the fingers from nodes u and $\pi_t(u)$ in the matching game as if Prover had removed his finger from node u and Disprover had responded accordingly. Thus, $\pi_{t+1} = \pi_t - \{(u, \pi_t(u))\}$ in that case. Then, she removes the pebble from T corresponding to $x_{u,v}$ in the existential r-pebble game.

If at time $t+1$ Spoiler places a pebble over variable $x_{u,v}$ in the existential r-pebble game, Duplicator acts as follows: If u has a finger on it in the matching game already, Duplicator does nothing in that game, so that $\pi_{t+1} = \pi_t$. Then, in the existential r-pebble game, she answers 1 if $v = \pi_t(u)$, and 0 otherwise. If u did not have a finger in the matching game, Duplicator finds some $w \in V$ to match u according to the winning strategy of Disprover as if Prover had placed a finger over u in the matching game. Thus, $\pi_{t+1} = \pi_t \cup \{(u, w)\}$ in that case. Then, in the existential r-pebble game, she answers 1 if $w = v$ and 0 otherwise.

We need to argue two points: (1) that Duplicator can follow this strategy, and (2) that if she does, she wins the existential r-pebble game on $M(F)$ and T. For (1), observe that since at most r pebbles are involved in the existential r-pebble game, at most r nodes from U have a finger in the matching game. It follows that Duplicator can always find the required w when needed according to the winning strategy of Disprover in the matching game. For (2), suppose that Spoiler has not won at time t; we show that he does not win at time $t+1$. If Spoiler removed a pebble from variable $x_{u,v}$, say, Duplicator cannot lose there since she removed the corresponding pebble in T. If Spoiler placed a pebble over variable $x_{u,v}$, say, Duplicator will always answer according to the truth values inferred by a partial matching on G, and these never falsify a clause from G-PHP by Lemma 6. This completes the proof of the Theorem. □

9 On the Complexity of Resolution

Resolution is probably the most popular proof system for propositional logic. There is only one rule: from clauses $C \cup \{x\}$ and $D \cup \{\neg x\}$ derive $C \cup D$. A Resolution refutation of a CNF formula F is a sequence of clauses ending with the empty clause $\{\}$, each of which is either (1) a clause of F, (2) a clause appearing earlier in the sequence, or (3) follows from two previous clauses in the sequence by the Resolution rule. The length of a Resolution refutation is the length of the sequence of clauses. The width of a Resolution refutation is the maximum number of distinct literals in a clause of the refutation.

Theorem 9 *Let $k \geq 3$ be an integer. There exists a $2k$-Datalog sentence φ such that for every 3-CNF formula F, the following holds:*

(i) If F has a Resolution refutation of width k, then $M(F) \models \varphi$.

(ii) If $M(F) \models \varphi$, then F has a Resolution refutation of width $2k$.

Proof: Consider the following set of Datalog rules. For every w and i such that $0 \leq i \leq w \leq 2k$, let S_i^w be a new predicate symbol of arity w. Consider the following rule that we call (A):

$$S_i^3(x_1, x_2, x_3) \ :- \ R_i(x_1, x_2, x_3)$$

The meaning of (A) is that every clause of F is derivable. Add also the following rules that we call (B) and (C) respectively:

$$\begin{aligned} S_i^w(x_1, \ldots, x_p/x_q, \ldots, x_q/x_p, \ldots, x_i, y_1, \ldots, y_{w-i}) \ &:- \ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \\ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{p'}/y_{q'}, \ldots, y_{q'}/y_{p'}, \ldots, y_{w-i}) \ &:- \ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \end{aligned}$$

where $1 \leq p < q \leq i$ in (B), $1 \leq p' < q' \leq w - i$ in (C). The meaning of (B) is that if the clause $\{\neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i}\}$ is a derivable, then any result of permuting two x-literals in it is also derivable. Similarly for (C). The following four clauses are called (D), (E), (F) and (G) respectively:

$$\begin{aligned} S_i^w(x, \ldots, x, x, x_1, \ldots, x_a, y_1, \ldots, y_{w-i}) \ &:- \ S_i^w(x, \ldots, x, x_1, x_1, \ldots, x_a, y_1, \ldots, y_{w-i}) \\ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_b, y, y, \ldots, y) \ &:- \ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_b, y_b, y, \ldots, y) \\ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}, y_{w-i-1}) \ &:- \ S_{i+1}^w(x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}) \\ S_{i+1}^w(x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}) \ &:- \ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i-1}, y_{w-i-1}) \end{aligned}$$

where $0 < a < i - 1$ in (D), $0 < b < w - i - 1$ in (E), and $0 < i < w - 1$ in (F) and (G). The meaning of rule (D) is that if $\{\neg x, \ldots, \neg x, \neg x_1, \neg x_1, \ldots, \neg x_a, y_1, \ldots, y_{w-i}\}$ is derivable, so is the result of contracting one $\neg x_1$ and adding one $\neg x$. Similarly for (E). The meaning of (F) is that if $\{\neg x_1, \neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i-1}\}$ is derivable, so is the result of contracting one $\neg x_1$ and adding one y_{w-i-1}. Similarly for (G). We introduce two rules that, together with (B),...,(G) can be used to eliminate repeated literals. The rules are called (H), (I):

$$\begin{aligned} S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \ &:- \ S_i^{w+p}(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}, y_{w-i}, \ldots, y_{w-i}) \\ S_i^w(x_1, \ldots, x_i, y_1, \ldots, y_{w-i}) \ &:- \ S_{i+p}^{w+p}(x_1, \ldots, x_1, x_1, \ldots, x_i, y_1, \ldots, y_{w-i}), \end{aligned}$$

as long as $w + p \leq 2k$, $0 \leq i < w$ in (H) and $0 < i \leq w$ in (I). Next we introduce the rule that simulates the Resolution rule that we call (J):

$$\begin{aligned} &S_{r_1+r_2-1}^{w_1+w_2-2}(x_2, \ldots, x_{r_1}, x'_1, \ldots, x'_{r_2}, y_1, \ldots, y_{w_1-r_1}, y'_1, \ldots, y'_{w_2-r_2-1}) : - \\ &\quad :- \ S_{r_1}^{w_1}(z, x_2, \ldots, x_{r_1}, y_1, \ldots, y_{w_1-r_1}), S_{r_2}^{w_2}(x'_1, \ldots, x'_{r_2}, y'_1, \ldots, y'_{w_2-r_2-1}, z), \end{aligned}$$

where $w_1 + w_2 - 2 \leq 2k$, $0 < r_1 \leq w_1$ and $0 \leq r_2 < w_2$. The meaning of (J) is that if the clauses $\{\neg z, \neg x_2, \ldots, \neg x_{r_1}, y_1, \ldots, y_{w_1-r_1}\}$ and $\{\neg x'_1, \ldots, \neg x'_{r_2}, y'_1, \ldots, y'_{w_2-r_2-1}, z\}$ have been derived, then the resolvent clause is also derivable by an application of the Resolution rule. Note that the resulting clause may have repeated literals. Note also that we do not insist that variable z appears only once in each clause in order to apply the Resolution rule. The soundness of the rule is not lost and we will make use of this fact later. Finally, add a new predicate symbol G and the rule (K):

$$G \ :- \ S_0^1(x), S_k^1(x).$$

The meaning of (K) is that if $\{x\}$ and $\{\neg x\}$ are derivable, then F is refutable.

Let φ be the conjunction of all these Datalog rules (ABCDEFGHIJK). Simple inspection reveals that each rule can be written with at most $2k$ variables. Thus, φ is a $2k$-Datalog program. It is a routine task (but somewhat tedious) to check that if F has a Resolution refutation of width k then $M(F) \models \varphi$. Similarly, if $M(F) \models \varphi$ then F has a Resolution refutation of width $2k$. The proof can be found in the Appendix. □

An immediate corollary of the proof of Theorem 2 and Theorem 9 is that every Resolution refutation of a random 3-CNF formula requires width $n/2(\ln n)^2$ with probability approaching 1 as n approaches ∞. Now, Ben-Sasson and Wigderson [BSW01] proved that if F is a 3-CNF on n variables that has a Resolution refutation of length S, then F has a Resolution refutation of width $O\left(\sqrt{n \log S}\right)$. Thus, we obtain a slightly weaker bound than that of Chvátal and Szeméredi.

Corollary 3 ([CS88]) *Let F be drawn from $\mathcal{F}(n, 6n, 3)$. Then, almost surely, F is unsatisfiable and every Resolution refutation of F requires size $2^{\Omega(n/(\ln n)^4)}$.*

From Theorem 8 we can obtain Haken's lower bound:

Corollary 4 ([Hak85]) *Every resolution refutation of PHP_n^{n+1} requires length $2^{\Omega(n)}$.*

Proof: It is known that there exist $(\Omega(n), 1/4)$-bipartite expanders of left-degree 3 on sets of size $n+1$ and n (an easy probabilistic argument as in the proof Theorem 2 does it). Therefore, for such a bipartite expander $G = (U \cup V, E)$, every Resolution refutation of the set of clauses G-PHP requires width $\Omega(n)$ by Theorem 8. Since the number of variables of G-PHP is $O(n)$, we conclude from the size-width trade-off of Ben-Sasson and Wigderson that every Resolution refutation of G-PHP requires size $2^{\Omega(n)}$. Although the lower bound for PHP_n^{n+1} follows from this at once, we review the proof for completeness.

Suppose for contradiction that PHP_n^{n+1} has a resolution refutation with $2^{o(n)}$ clauses. Let $C_1, \ldots, C_r$ be such a refutation. Let ρ be the partial truth assignment to the variables $\{p_{i,j}\}$ that sets $\rho(p_{i,j}) = 0$ if $(i,j) \notin E$. Here we are assuming without loss of generality that $U = \{1, \ldots, n+1\}$ and $V = \{1, \ldots, n\}$. For every $i \in \{1, \ldots, r\}$, let $C'_i = \rho(C_i)$ be the result of applying ρ to C_i. That is, C'_i is 1 if ρ sets some literal of C_i to 1, and the subset of non-falsified literals of C_i otherwise. We claim that the sequence of clauses $C'_1, \ldots, C'_r$ is a Resolution refutation with weakening of G-$PHP \cup \{1\}$. If this is true, then G-PHP has a Resolution refutation of length $2^{o(n)}$ since the clause 1 cannot be used in any cut.

We will show, by induction on $m \in \{1, \ldots, r\}$, that $C'_1, \ldots, C'_m$ is a valid Resolution derivation from G-$PHP \cup \{1\}$. For the base case $m = 1$, necessarily C_m is an initial clause of PHP_n^{n+1}, and it is easily seen that either $C'_m = 1$, or C'_m is an initial clause of G-PHP. Suppose that $m > 1$ and that the claim is valid for every smaller value. If C_m is an initial clause of PHP_n^{n+1}, argue as in the base case. Suppose next that C_m is derived by the resolution rule from C_k and C_l with $k < l < m$. Suppose that the cut variable is $p_{i,j}$. The immediate case is when $C'_k \neq 1$ and $C'_l \neq 1$ because then C'_m is simply the result of applying the resolution rule on them. Also, if $C'_k = 1$ and $C'_l = 1$, then $C'_m = 1$ because some satisfied literal must be inherited by C_m. Finally, if $C'_k = 1$ and $C'_l \neq 1$, say, there are two cases two consider. If the satisfied literal in C_k is not about variable $p_{i,j}$, then this literal is inherited in C_m and so $C'_m = 1$. If the satisfied literal in C_k is about variable $p_{i,j}$ indeed, then it must be $\neg p_{i,j}$. In that case, C'_l does not contain the literal $p_{i,j}$ because $\rho(p_{i,j}) = 0$. We conclude that C'_m is a weakening of C'_l and we are done. □

10 Conclusions

While Datalog falls short to capturing any complexity class defined in terms of the standard computational models, many important algorithmic techniques can be "implemented" as Datalog programs. In particular, this holds true for algorithmic techniques capable of solving such problems as unsatisfiability of 2-CNF formulas, graph reachability, or the circuit value problem. Note that the latter problem is $\mathbf{P}$-complete but Datalog is not closed under logspace reductions, and that is why it falls short to capturing $\mathbf{P}$.

The fact that natural classes of algorithms can be implemented in Datalog makes inexpressibility results powerful. A clear example of this is provided by the results in Section 9, where we showed that a $2k$-Datalog program is able to implement the proof search algorithm for Resolution of width k. This algorithm was studied already in 1977 by Galil [Gal77], and revisited by Ben-Sasson and Wigderson [BSW01]. Moreover, the important results of Ben-Sasson and Wigderson that relate Resolution width with Resolution size, together with our inexpressibility result for Datalog, allowed us to re-derive the known lower bounds for Resolution size. In this sense then, inexpressibility results for logics that fall short to capturing complexity classes can still be used to derive complexity-theoretic lower bounds.

The main issue that remains open from our paper is that of establishing further connections between inexpressibility results and lower bounds for proof complexity. The first observation we want to make in this respect is related to a potential application of proof-complexity techniques to finite-model-theory techniques. The second observation is related to a synergy in the reverse direction.

From proof-complexity to finite model theory Inexpressibility results are usually proved by designing a winning strategy for the Duplicator, one of the players of the appropriate Ehrenfeucht-Fraïssé game for the logic under consideration. For example, inexpressibility results for $\exists L^k_{\infty\omega}$ are proved by exhibiting a winning strategy for the Duplicator in the existential k-pebble game. It turns out that the design of winning strategies for the Duplicator is often a highly non-trivial task. To make things worse, there is a clear lack of techniques to describe the winning strategies in a precise mathematical form, other than by enumerating essentially all possible cases. However, there is an important exception: when extension axioms are available, winning strategies for the Duplicator are easy to describe. The proof of our main result made strong use of extension axioms, which, in turn, were derived from the expansion properties of certain graphs. This suggests that further extension-type axioms are yet to be discovered, and the experience in using expander properties in propositional proof complexity may turn out to be helpful.

This brings the opportunity to compare our result to a result in finite model theory due to Dawar [Daw98]. The result is that the class of finite non-3-colorable graphs is not definable in the logic $L^k_{\infty\omega}$, the full infinitary logic with k variables, for any $k \geq 1$. The relationship with our result is the fact that 3-colorability is a constraint satisfaction problem, and can thus be expressed as a homomorphism problem to a fixed template structure as we did for the satisfiability problem. Indeed, a graph is 3-colorable if and only if it can be mapped homomorphically to the triangle. We note that $L^k_{\infty\omega}$ is a stronger logic than the one we studied, $\exists L^k_{\infty\omega}$, since negations are allowed and there is no restriction in the allowed quantifiers. On the other hand, the inexpressibility result due to Dawar is a *worst-case* negative result, as opposed to ours which is an *average-case* negative

result and thus stronger.

In turn, this suggests a few open problems. The phase transition phenomenon also occurs in the 3-colorability problem and has been studied intensively in recent years [AF99]. Can we prove average-case negative results similar to ours, or maybe for the stronger logic $L^k_{\infty\omega}$, near the threshold of 3-colorability? We point out that Beame et al. (personal communication) have studied the problem also recently from the proof complexity perspective.

From finite model theory to proof-complexity The second observation that we want to make with respect to the interaction between finite model theory and proof complexity is the following: it seems intuitively clear that if Datalog relates to Resolution in the precise sense of Theorem 9 of Section 9, then stronger fixed-point logics should relate to stronger proof systems. This suggests two problems. The first one is in finite model theory, proper.

Open Problem 1 *Extend the inexpressibility results of Datalog on random 3-CNF formulas to stronger fixed-point logics, such as Least Fixed-Point Logic LFP, or even full infinitary logic with finitely many variables $L^\omega_{\infty\omega}$.*

Clearly, the same problem could be posed for other logics such as first-order logic, existential monadic second-order logic (monadic $\mathbf{NP}$), or logics with counting quantifiers. The second open problem is stated in slightly more general terms. It should be considered as a family of open problems, one for each logic and proof system, rather than as concrete open problem with a precise statement.

Open Problem 2 *Establish further connections between inexpressibility results for natural logics and lower bounds for propositional proof systems.*

We hope that our results, together with the solution to some of these open problems, may contribute to the synergy between the fields of finite model theory and propositional proof complexity.

Recent developments The second part of this paper established a tight connection between the minimum Resolution width and the minimum number of pebbles for Duplicator to win the game. We showed that the width is at least half this number of pebbles. On the other hand, our proofs rest on the techniques that were employed for proving Resolution space complexity lower bounds in [BSG01]. A few obvious questions arise. What is the exact connection? Is the minimum width equal to the minimum number of pebbles for Duplicator to win? Is it the minimum space? Is the space bigger than the width, or conversely? Recent work by the author and Dalmau [AD03] answered some of these questions. It turns out that the minimum width coincides, indeed, with the minimum number of pebbles for Duplicator to win. The proof idea also led to a proof that the minimum space is lower bounded by the minimum width minus the width of the formula, resolving an open problem in [ET01, ABSRW02, BSG01]. These results show that width and number of pebbles are two names for the same thing, and that all space lower bounds already follow from width lower bounds. It remains open whether width can be smaller than space for some formula.

Acknowledgments I am grateful to Phokion Kolaitis for his constructive criticism and suggestions. Thanks also to José Balcázar, María Bonet, Steve Cook, Víctor Dalmau, Moshe Vardi, and the anonymous referees for their comments.

Appendix: Proofs

Let us start by proving Lemma 4 on the right-degree of the incidence graph of a random formula.

Proof of Lemma 4: Note that a variable occurs in a random clause with probability $\binom{n-1}{2}\binom{n}{3}^{-1} = 3/n$. For every $i \in \{1,\ldots,n\}$, let X_i be the random variable that counts the number of clauses of F in which variable v_i occurs. By the above, the expectation of X_i is $\Delta n \cdot 3/n = 3\Delta$. More important,

$$\Pr[X_i \geq d] \leq \binom{\Delta n}{d}\left(\frac{3}{n}\right)^d \leq \left(\frac{\Delta n e}{d}\right)^d \left(\frac{3}{n}\right)^d = \left(\frac{3e\Delta}{d}\right)^d.$$

Here we used the well-known inequality $\binom{n}{k} \leq (ne/k)^k$ that is derivable using the fact that $k! \geq (k/e)^k$. Now, let X be the random variable that counts the number of $i \in \{1,\ldots,n\}$ such that $X_i \geq d$. The expectation of X is at most $n(3e\Delta/d)^d$. Therefore, the probability that $X > 0$ is at most $n(3e\Delta/d)^d$ by Markov's inequality. □

Next we prove Lemma 5 on the probability that the incidence graph is an expander.

Proof of Lemma 5: Write $G = G(F) = (U_1 \cup U_2, E)$. Let $\gamma = 1+\epsilon$. We bound the probability that for a fixed set $U \subseteq U_1$ of size $i \leq s$ and a fixed set $V \subseteq U_2$ of size γi it holds that $N_G(U) \subseteq V$. For a fixed $u \in U$, the probability that $N_G(u) \subseteq V$ is bounded by $\binom{\gamma i}{3}\binom{n}{3}^{-1}$. Since each clause is drawn independently of the rest, the probability that $N_G(U) \subseteq V$ is bounded by

$$\left[\frac{\binom{\gamma i}{3}}{\binom{n}{3}}\right]^i \leq \left(\frac{\gamma i}{n}\right)^{3i}.$$

Here we used the fact that $\binom{n}{k} / \binom{m}{k} \leq (n/m)^k$ if $n \leq m$. The probability that G is not an (s,ϵ)-bipartite expander is bounded by the probability that there exist sets U and V as above. Since there are $\binom{\Delta n}{i}$ possible U and $\binom{n}{\gamma i}$ possible V, this probability is bounded by

$$\begin{aligned}\sum_{i=1}^{s}\binom{\Delta n}{i}\binom{n}{\gamma i}\left(\frac{\gamma i}{n}\right)^{3i} &\leq \sum_{i=1}^{s}\left(\frac{\Delta n e}{i}\right)^i\left(\frac{ne}{\gamma i}\right)^{\gamma i}\left(\frac{\gamma i}{n}\right)^{3i} = \\ &= \sum_{i=1}^{s}\left[\Delta e^{2+\epsilon}(1+\epsilon)^{2-\epsilon}\left(\frac{i}{n}\right)^{1-\epsilon}\right]^i.\end{aligned}$$

The largest term into brackets is achieved when $i = s$. Thus, we can bound the sum by

$$\sum_{i=1}^{s} C^i \leq \frac{C(1-C^s)}{1-C}$$

as was to be proved. □

The next proof establishes that the Datalog program defined in Theorem 9 achieves its goal.

Proof of Theorem 9 (continued): We first prove that if F has a Resolution refutation of width k, then $M(F) \models \varphi$. Indeed, suppose that $C_1, C_2, \ldots, C_m$ is a Resolution refutation of F of width k. We may assume without loss of generality that $C_{m-2} = \{x\}$, $C_{m-1} = \{\neg x\}$ and $C_m = \{\}$. For $j \in \{1, \ldots, m-1\}$, let us write $C_j = \{\neg x_{j,1}, \ldots, \neg x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j}\}$ where all displayed variables are different and $r_j + s_j \leq k$. Let $w_j = r_j + s_j$ be the width of C_j. We will show, by induction on $j \in \{1, \ldots, m-1\}$, that

$$S^{w_j}_{r_j}(x_{j,1}, \ldots, x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j})$$

is derivable from $M(F)$ and the Datalog rules. The base case $j = 1$ is clear: C_j is necessarily a clause of F, and the conclusion is immediate from rule (A), and rules (B),...,(I). Fix $j > 1$ and suppose that the claim holds for all smaller values of j. If C_j is a clause of F, the conclusion is again immediate. Suppose then that C_j follows from C_h and C_l by the Resolution rule. Here $h < l < j$. By induction hypothesis, we know that

$$S^{w_h}_{r_h}(x_{h,1}, \ldots, x_{h,r_h}, y_{h,1}, \ldots, y_{h,s_h})$$
$$S^{w_l}_{r_l}(x_{l,1}, \ldots, x_{l,r_l}, y_{l,1}, \ldots, y_{l,s_l})$$

are derivable. Since C_h and C_l are resolved together to obtain C_j, some variable must appear negatively in C_h and positively in C_l, or conversely. Without loss of generality, let us assume that the former holds. Using the rules (B) and (C), we may also assume that $x_{h,1} = y_{l,s_l}$ is the cut variable z. Now let us apply the rule (J) to obtain

$$S^{w_h+w_l-2}_{r_h+r_l-1}(x_{h,2}, \ldots, x_{h,r_h}, x_{l,1}, \ldots, x_{l,r_l}, y_{h,1}, \ldots, y_{h,s_h}, y_{l,1}, \ldots, y_{l,s_l-1}).$$

Then apply the rules (B),...,(I) to obtain

$$S^{w_h+w_l-2}_{r_j}(x_{j,1}, \ldots, x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j}, y_{j,s_j}, \ldots, y_{j,s_j}).$$

or

$$S^{w_h+w_l-2}_{w_h+w_l-2}(x_{j,1}, \ldots, x_{j,1}, x_{j,1}, \ldots, x_{j,r_j})$$

depending on whether $s_j > 0$ or $s_j = 0$ respectively. Finally, apply the rule (H) or (I) to obtain

$$S^{w_j}_{r_j}(x_{j,1}, \ldots, x_{j,r_j}, y_{j,1}, \ldots, y_{j,s_j}).$$

We have proved that $S^1_0(x)$ and $S^1_1(x)$ are derivable. Thus, G is derivable, so $M(F) \models \varphi$.

Let us prove the reverse direction, namely that if $M(F) \models \varphi$, then F has a Resolution refutation of width $2k$. In order to prove that, we will need to introduce a mild extension of Resolution as an intermediary step. The new proof system incorporates the so-called weakening rule: from clause C, derive $C \cup D$ where D is an arbitrary clause. In Lemma 7 below, we prove that weakenings may be eliminated without increasing the width, and therefore allowing them cannot hurt. Next we prove that if

$$S^w_i(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$$

holds, then the clause

$$\{\neg x_1, \ldots, \neg x_i, y_1, \ldots, y_{w-i}\} \tag{3}$$

has a Resolution derivation with weakening from F in width $2k$. The proof is by induction on the stage s at which the tuple $(x_1, \ldots, x_i, y_1, \ldots, y_{w-i})$ entered in the least fixed-point of the Datalog rules. The base case $s = 1$ is easy: necessarily, the tuple entered following rule (A), and the claim becomes obvious since then (3) is an initial clause of F. Fix $s > 1$ and suppose that the claim holds for all smaller values of s. If the tuple entered following one of the rules (B), . . . ,(I), the claim is clear since clauses are sets of literals. If the tuple entered following the rule (J), we proceed as follows: Let us rename $x_1, \ldots, x_i, y_1, \ldots, y_{w-i}$ in the form of the conclusion of rule (J); that is

$$(x'_2, \ldots, x'_{r_1}, x''_1, \ldots, x''_{r_2}, y'_1, \ldots, y'_{w_1-r_1}, y''_1, \ldots, y''_{w_2-r_2-1}), \tag{4}$$

where both

$$S^{w_1}_{r_1}(z, x'_2, \ldots, x'_{r_1}, y'_1, \ldots, y'_{w_1-r_1})$$
$$S^{w_2}_{r_2}(x''_1, \ldots, x''_{r_2}, y''_1, \ldots, y''_{w_2-r_2-1}, z)$$

hold and have entered in a previous stage. If z does not occur in (4), then (3) is simply the result of resolving the clauses

$$\{\neg z, \neg x'_2, \ldots, \neg x'_{r_1}, y'_1, \ldots, y'_{w_1-r_1}\}$$
$$\{\neg x''_1, \ldots, \neg x''_{r_2}, y''_1, \ldots, y''_{w_2-r_2-1}, z\}$$

on z. If on the contrary, z occurs in (4), then we may obtain (3) by weakening over one of these two clauses. Which clause depends on whether z occurs positively or negatively. This completes the proof of the Theorem. □

Finally, it remains to address the following lemma that we left without proof in the preceeding argument. The result it asserts is well-known, but we include a complete proof for completeness.

Lemma 7 *Let $C_1, \ldots, C_r$ be clauses. If $\{C_1, \ldots, C_r\}$ has a Resolution refutation with weakening of width w and length s, then it also has a Resolution refutation without weakening of width at most w and length at most s.*

Proof: We prove, by induction on m, that if $D_1, \ldots, D_m$ is a Resolution derivation with weakening, then there exist a Resolution derivation $D'_1, \ldots, D'_m$ without weakening such that $D'_i \subseteq D_i$ for every $i \in \{1, \ldots, m\}$. The base case $m = 1$ is clear: D_m is necessarily an initial clause, so $D'_m = D_m$ works. Let $m > 1$ and suppose that the claim holds for all smaller values. Let $D'_1, \ldots, D'_{m-1}$ be the derivation claimed by the induction hypothesis. We consider three cases according to the nature of D_m. If D_m is an initial clause, let $D'_m = D_m$ and we are done. If D_m is derived from D_k with $k < m$ by weakening, we let $D'_m = D'_k$ and we are also done because $D'_k \subseteq D_k \subseteq D_m$. If D_m is derived from D_k and D_j with $k < j < m$ by a cut with $x \in D_k$ and $\neg x \in D_j$, we consider three cases. If x appears in D'_k and $\neg x$ appears in D'_j, we let D'_m be the result of cutting D'_k and D'_j on x. Then

$$D'_m = D'_k \cup D'_j - \{x, \neg x\} \subseteq D_k \cup D_j - \{x, \neg x\} \subseteq D_m.$$

If x appears in D'_k but $\neg x$ does not appear in D'_j, we let $D'_m = D'_j$. Then

$$D'_m = D'_j = D'_j - \{x, \neg x\} \subseteq D_j - \{x, \neg x\} \subseteq D_m.$$

Finally, if x does not appear in D'_k, we let $D'_m = D'_k$ and again

$$D'_m = D'_k = D'_k - \{x, \neg x\} \subseteq D_k - \{x, \neg x\} \subseteq D_m .$$

Since we covered all possible cases, this completes the proof of the lemma. □

References

[ABE02] 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.

[ABSRW02] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Space complexity in propositional calculus. *SIAM Journal of Computing*, 31(4):1184–1211, 2002. A preliminary version appeared in STOC'00.

[Ach01] D. Achlioptas. A survey of lower bounds for random 3-SAT via differential equations. *Theoretical Computer Science*, 265(1–2):159–185, 2001.

[AD03] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. In *18th IEEE Conference on Computational Complexity*, 2003.

[AF99] D. Achlioptas and E. Friedgut. A sharp threshold for k-colorability. *Random Structures and Algorithms*, 14(1):63–70, 1999.

[BFU93] A. Z. Broder, A. M. Frieze, and E. Upfal. On the satisfiability and maximum satisfiability or random 3-CNF formulas. In *4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 322–330, 1993.

[BKPS98] P. Beame, R. Karp, T. Pitassi, and M. Saks. On the complexity of unsatisfiability proofs for random k-CNF formulas. In *30th Annual ACM Symposium on the Theory of Computing*, pages 561–571, 1998.

[Bla37] A. Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BM77] J. L. Bell and M. Machover. *A Course in Mathematical Logic*. North-Holland, 1977.

[BS01] E. Ben-Sasson. *Expansion in Proof Complexity*. PhD thesis, Hebrew University, 2001.

[BSG01] E. Ben-Sasson and N. Galesi. Space complexity of random formulas in resolution. In *16th IEEE Conference on Computational Complexity*, pages 42–51, 2001.

[BSI99] E. Ben-Sasson and R. Impagliazzo. Random CNF's are hard for the polynomial calculus. In *40th Annual IEEE Symposium on Foundations of Computer Science*, pages 415–421, 1999.

[BSW01] E. Ben-Sasson and A. Wigderson. Short proofs are narrow–resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[CDA+00] Cristian Coarfa, D. D. Demopoulos, A. San Miguel Aguirre, D. Subramanian, and M. Y. Vardi. Random 3-SAT: The plot thickens. In *6th International Conference on Principles and Practice of Constraint Programming*, volume 1894 of *Lecture Notes in Computer Science*, pages 143–159. Springer, 2000.

[CH82] A. Chandra and D. Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25:99–128, 1982.

[CR79] S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44:36–50, 1979.

[CR92] V. Chvátal and B. Reed. Mick gets some (the odds are on his side). In *33rd Annual IEEE Symposium on Foundations of Computer Science*, pages 620–627, 1992.

[CS88] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, 1988.

[Daw98] A. Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143:154–174, 1998.

[DBM00] O. Dubois, Y. Boufkhad, and J. Mandler. Typical random 3-SAT formulae and the satisfiability threshold. In *11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 126–127, 2000.

[DP60] M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.

[EF95] H. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[EFT84] H. Ebbinghaus, J. Flum, and W. Thomas. *Mathematical Logic*. Springer-Verlag, 1984.

[ET01] J. L. Esteban and J. Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. A preliminary version appeared in STACS'99.

[Fag76] R. Fagin. Probabilities on finite models. *Journal of Symbolic Logic*, 41:50–58, 1976.

[FV98] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction: A study through Datalog and group theory. *SIAM Journal of Computing*, 28(1):57–104, 1998.

[Gal77] Z. Galil. On resolution with clauses of bounded size. *SIAM Journal of Computing*, 6:444–459, 1977.

[Hak85] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[HR89] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1989.

[KKL02] A. C. Kaporis, L. M. Kirousis, and E. G. Lalas. The probabilistic analysis of a greedy satisfiability algorithm. In *10th Annual European Symposium on Algorithms*, volume 2461 of *Lecture Notes in Computer Science*, pages 574–585. Springer, 2002.

[KM81] B. Krishnamurthy and R. N. Moll. Examples of hard tautologies in the propositional calculus. In *13th Annual ACM Symposium on the Theory of Computing*, pages 28–37, 1981.

[KV90] Ph. G. Kolaitis and M. Y. Vardi. 0-1 laws and decision problems for fragments of second-order logic. *Information and Computation*, 87:302–338, 1990.

[KV95] Ph. G. Kolaitis and M. Y. Vardi. On the expressive power of Datalog: tools and a case study. *Journal of Computer and System Sciences*, 51:110–134, 1995.

[KV00a] Ph. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. *Journal of Computer and System Sciences*, 61(2):302–332, 2000.

[KV00b] Ph. G. Kolaitis and M. Y. Vardi. A game-theoretic approach to constraint satisfaction. In *7th National Conference on Artificial Intelligence*, pages 175–181, 2000.

[Lyn80] J. F. Lynch. Almost sure theories. *Annals of Mathematical Logic*, 18:91–135, 1980.

[Pap85] C. H. Papadimitriou. A note on the expressive power of Prolog. *Bullentin of the EATCS*, 26:21–23, 1985.

[Pap95] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1995.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[SK96] B. Selman and S. Kirckpatrick. Critical behavior in the computational cost of satisfiability testing. *Artificial Intelligence*, 81(1–2):17–29, 1996.

[SML96] B. Selman, D. G. Mitchell, and H. J. Levesque. Generating hard satisfiability problems. *Artificial Intelligence*, 81:17–29, 1996.

[SS88] S. Shelah and J. Spencer. Zero-one laws for sparse random graphs. *Journal of the American Mathematical Society*, 1(1):97–115, 1988.

[Ull89] J. D. Ullman. Database and knowledge-base systems. *Computer Science Press*, 1989.