The Model-Theoretic Expressiveness of Propositional Proof Systems

Erich Grädel[1], Benedikt Pago[2], and Wied Pakusa[3]

1 **RWTH Aachen University**
graedel@logic.rwth-aachen.de
2 **RWTH Aachen University**
benedikt.pago@rwth-aachen.de
3 **University of Oxford**
wied.pakusa@cs.ox.ac.uk

Abstract

We establish new, and surprisingly tight, connections between propositional proof complexity and finite model theory. Specifically, we show that the power of several propositional proof systems, such as Horn resolution, bounded width resolution, and the polynomial calculus of bounded degree, can be characterised in a precise sense by variants of fixed-point logics that are of fundamental importance in descriptive complexity theory. Our main results are that *Horn resolution* has the same expressive power as *least fixed-point logic*, that *bounded width resolution* captures *existential least fixed-point logic*, and that the (monomial restriction of the) *polynomial calculus of bounded degree* solves precisely the problems definable in *fixed-point logic with counting*.

1998 ACM Subject Classification F.4.1. Mathematical Logic

Keywords and phrases Propositional proof systems, fixed-point logics, resolution, polynomial calculus, generalized quantifiers

Digital Object Identifier 10.4230/LIPIcs.CSL.2017.27

1 Introduction

The question whether there exists an efficient proof system by means of which the validity of *arbitrary propositional formulas* can be verified via *proofs of polynomial size* is equivalent to the closure of NP under complementation. Since Cook and Reckhow [14] made the notion of an efficient propositional proof system precise, a huge body of research on the power of various propositional proof system has been established. In particular, we now have super-polynomial lower bounds on the proof complexity for quite strong proof systems, see [7, 25] for surveys on propositional proof complexity.

In this paper we study *polynomial-time variants* of propositional proof systems, which admit efficient proof search, resulting in proofs of polynomial size, such as restricted variants of resolution and the polynomial calculus. Recall that the resolution proof system Res takes as input a propositional formula φ in conjunctive normal form (CNF), and it refutes the satisfiability of φ if there is a derivation of the empty clause from φ. It is well-known that shortest resolution proofs can be of exponential size, so in general, we provably cannot search for resolution proofs in polynomial time. However, there are interesting restrictions of Res, such as Horn-Res (resolution restricted to Horn clauses) and bounded-width resolution k-Res (resolution restricted to clauses of size $\leq k$) that do admit efficient proof search. Of course, unless P = NP, any proof system that admits efficient proof search is necessarily incomplete for full propositional logic. Nevertheless we can still prove interesting statements

in such systems, and usually have completeness for relevant fragments of propositional logic, such as Horn-logic or 2-CNF. We can now try to solve algorithmic problems by reducing them to provability (or refutability) in some specific polynomial-time proof system, which, if it works successfully for all inputs, would give us a polynomial-time algorithm for the problem. Our goal is to understand how powerful this approach can be, depending on the specific proof system that we use.

Let us illustrate this by two concrete problems. First we consider *graph isomorphism*, a problem which is not known to be solvable in polynomial time although there is strong evidence that it is not NP-complete. Given two graphs $G = (V, E)$ and $H = (W, F)$ we ask whether there is a bijection $\pi\colon V \to W$ such that $\pi(E) = F$. Of course, this can easily be encoded as the satisfiability problem of a propositional CNF-formula. First, for each pair of vertices $v \in V$ and $w \in W$ we introduce a variable X_{vw} with the intended meaning that $X_{vw} = 1$ if $\pi(v) = w$. We add clauses $\bigvee_{w \in W} X_{vw}$ for every $v \in V$ and $\bigvee_{v \in V} X_{vw}$ for every $w \in W$ to ensure that every $v \in V$ has an image and every $w \in W$ has a preimage. Additionally we add for all $v_1, v_2 \in V$ and $w_1, w_2 \in W$ a clause $\neg(X_{v_1 w_1} \wedge X_{v_2 w_2})$ in case that $\{v_1 \mapsto w_1, v_2 \mapsto w_2\}$ is not a partial isomorphism. The resulting CNF-formula, denoted by $\mathrm{Iso}(G, H)$, is satisfiable if, and only if, the two graphs G and H are isomorphic. Following our reasoning from above, we can now use an efficient variant of resolution, or of a stronger proof system, and try to refute the satisfiability of the formula $\mathrm{Iso}(G, H)$. If this is possible, then G are H are not isomorphic. Unfortunately, if we do not find a proof, then we are stuck, because it might still be the case that G and H are not isomorphic, but our proof system is not strong enough to show this. Hence, we get an efficient, sound, but not necessarily complete graph isomorphism test. The question how successful this approach is when based on resolution was studied by Toran in [26]. Unfortunately, he proved that shortest resolution proofs for graph non-isomorphism can be of exponential size (even for graphs with colour class size four). More recently, Grohe and Berkholz showed that also in the stronger system polynomial calculus (PC) one cannot obtain small proofs for graph non-isomorphism [9, 10].

Our second example is directed graph reachability: Given a directed graph $G = (V, E)$ with two distinguished vertices $s, t \in V$, we want to know whether there is a path from s to t in G. Again, it is easy to encode this as a satisfiability problem in propositional logic, by taking the conjunction of all implication clauses $X_v \to X_w$, for all edges $(v, w) \in E$, together with the two clauses $1 \to X_s$ and $X_t \to 0$. Clearly the resulting formula $\mathrm{NonReach}(G, s, t)$ is unsatisfiable if, and only if, t is reachable from s in G. However, in clear contrast to the formulas $\mathrm{Iso}(G, H)$ from above, we can easily prove unsatisfiability for the formulas $\mathrm{NonReach}(G, s, t)$ in efficient variants of resolution such as Horn-Res and k-Res for $k \geq 2$.

Our two examples demonstrate the following: while certain problems, such as directed graph reachability, allow for small and efficient resolution proofs, other problems, such as the graph isomorphism problem, provably require proofs of super-polynomial size even in quite strong proof systems. This leads to the main question that we want to address in this paper: is there a *classification* for those problems which can be solved in fundamental polynomial-time propositional proof systems such as Horn-Res, k-Res and degree-k (monomial)-PC, denoted by mon-PC_k. It came as a surprise to us that there is, indeed, a very clear and tight classification of the power of all of these proof systems in terms of definability in important fixed-point logics which are well-studied in the area of descriptive complexity theory.

Before we can state our results in detail, we have to explain what we mean by saying that a problem, such as directed graph reachability, can be solved by a propositional proof system Prop. As usual, each decision problem can be identified with a membership problem "$\mathfrak{A} \in \mathcal{K}$?" for some class of structures $\mathcal{K}$. For instance, the graph reachability problem from above is

identified with the class $\mathcal{K}_{\text{Reach}} = \{(V, E, s, t) : \text{there is a path from } s \text{ to } t \text{ in } G = (V, E)\}$. Then we naturally want to say that a problem $\mathcal{K}$ can be solved by the proof system Prop if we can find a reduction function f which maps structures $\mathfrak{A}$ to inputs $f(\mathfrak{A})$ for Prop such that $\mathfrak{A} \in \mathcal{K}$ if, and only if, Prop can prove that $f(\mathfrak{A})$ is not satisfiable. It is clear that we only want to allow *simple* reduction functions f, because otherwise the computation of the encoding could already contain part of the work to solve the problem. Coming from the area of finite model theory the obvious and natural formalisation for "f being simple" is to say that f is definable in *first-order logic* (FO). We introduce the precise technical definition of such reductions, which is the notion of a *first-order interpretation*, in Section 2. Note that for the two examples we discussed above the encoding functions are FO-definable.

Having established this definition it turns out that our classification problem is really about understanding the expressive power of the *Lindström extensions* of first-order logic by *generalised quantifiers* for propositional proof systems Prop. We denote these logics by FO(Prop). The basic idea of the logic FO(Prop) is just to extend first-order logic by new quantifiers $\mathcal{Q}_{\text{Prop}}$ which are capable of simulating Prop. In other words, we just incorporate into first-order logic the power to simulate Prop in an explicit way. Note that the logics FO(Prop) are really nothing more than a formalisation of the concept of oracle Turing-machines with access to Prop in the world of first-order logic (the oracle calls to the proof system Prop correspond to applications of the new generalised quantifiers). Again, the precise technical definitions of the Lindström extensions FO(Prop) can be found in Section 2. Having defined these logics, we can now say that a problem $\mathcal{K}$ can be solved in a proof system Prop if, and only if, it is definable in FO(Prop). For instance, we saw that $\mathcal{K}_{\text{Reach}} \in \text{FO}(\text{Horn-Res}) \cap \text{FO}(2\text{-Res})$.

We are prepared to state our main results in a formal way. We first look at the restrictions of resolution we mentioned before, Horn-Res and k-Res, for $k \geq 2$. It turns out that Horn-Res can solve precisely those problems which are definable in least-fixed point logic (LFP), that is FO(Horn-Res) = LFP. This follows by the well-known result that the problem of computing winning positions in reachability games (known as GAME or alternating reachability) is complete for LFP with respect to FO-reductions. More interestingly, we proceed to show that k-Res, for every $k \geq 2$, is less powerful than Horn-Res. In fact, FO(2-Res) = FO(TC), where FO(TC) is the extension of first-order logic by a transitive closure operator. Moreover, we prove that, for every $k \geq 3$, FO(k-Res) = EFP, where EFP is the *existential* fragment of least fixed-point logic which is known to be a strict fragment of full least fixed-point logic. One can also show that the Lindström extensions for Horn resolution and width-k resolution have different structural properties. While for FO(Horn-Res) a single application of a $\mathcal{Q}_{\text{Horn-Res}}$ quantifier suffices to obtain the full expressive power, nesting of $\mathcal{Q}_{k\text{-Res}}$ quantifiers is needed for the logics FO(k-Res). For lack of space, details will be deferred to the full version of this paper.

We then turn our attention to the monomial variant of the polynomial calculus (mon-PC), which is a proof system based on algebraic reasoning techniques. Its restriction to polynomials of degree at most k, denoted by mon-PC$_k$, gives us an interesting polynomial-time proof system which is known to be much stronger than bounded-width resolution and Horn resolution. Accordingly, we can prove that the logic FO$^+$(mon-PC$_k$) is more powerful than all logics based on restrictions of resolution that we considered before. In fact, we show that FO$^+$(mon-PC$_k$), for $k \geq 2$, has the same expressive power as fixed-point logic with counting (FPC) which is a very expressive logic well-studied in descriptive complexity theory [15, 23] (here, FO$^+$ denotes the extension of FO by a numeric sort to match the setting of FPC).

Finally, we discuss applications of our model-theoretic characterisations of propositional

proof systems. For instance, we can make statements about computational problems with regards to their solvability in one of these proof systems by using known (un-)definability results for fixed-point logics. Furthermore, we show how one can apply our logical characterisations of proof systems in order to transfer lower bounds from finite model theory to propositional proof complexity. In particular, we can easily reprove many lower bounds on the resolution sizes and widths for various families of propositional formulas.

Related work Let us briefly discuss some related work. Probably the most relevant result to mention here is the elegant characterisation by Atserias and Dalmau of resolution width in terms of the number of pebbles required to win an existential pebble game played on a given CNF-formula and a structural encoding of truth assignments [2, 4]. This somehow resembles our result saying that bounded width resolution corresponds to *existential* least fixed-point logic. Using their game-theoretic characterisation of resolution width, Atserias and Dalmau can reprove many of the known lower bounds on the resolution width. Again, this is similar to what we achieve in Section 5.1.

On the other hand, what makes our setting quite different from the approach of Atserias and Dalmau is that we always consider the power of proof systems, such k-Res, *up to logical reductions*. This reflects, for example, in our result saying that FO(3-Res) = FO(4-Res), i.e. that 3-Res has the same expressive power as 4-Res. But, certainly, this only holds if we allow first-order reductions to transform inputs between 4-Res and 3-Res. As a consequence, our analysis provides a much less precise characterisation of resolution width than the one obtained by Atserias and Dalmau. However, our more general point of view has some advantages. For instance, in the situation of lower bound proofs, we can avoid playing pebble games directly on the inputs to proof systems, such as CNF-formulas, but instead it suffices to play such games on pairs of structures in which these inputs interpret. This can make the description of winning strategies much simpler. Furthermore, our setting allows us to prove lower bound results much more independently from a concrete encoding of a problem, because of the fact that our logics are closed under logical interpretations, cf. discussion on the graph isomorphism problem in Section 5.1. Indeed, we can obtain lower bounds not only for one concrete family of inputs, but for any other family to which this family reduces to. For example, it is easy to see that our arguments in Corollary 20 about the lower bound for the 3-colourability problem actually go through for *any* other family of k-CNF-formulas to which one can reduce, in first-order logic say, the problem of solving a linear equation system over the two-element field (in which each equation has at most three variables) by using k-CNF-formulas with linearly many propositional variables.

Besides this, we also want to mention the series of papers [5, 9, 22, 21] which establish surprisingly tight connections between the equivalence of graphs in finite-variable logic with counting and their indistinguishability by linear programming techniques (Sherali-Adams relaxiations of graph isomorphism polytopes) and algebraic propositional proof system. Similar to the applications that we give in this paper, these results also allow the transfer of known lower bounds from finite model theory to get lower bounds on proof complexity. In particular, we use notions and ideas of [9] in Section 4.

2 Preliminaries

Logical interpretations and Lindström quantifiers Let $\mathcal{L}$ be a logic and σ, τ be signatures with $\tau = \{S_1, ..., S_\ell\}$. Let s_i denote the arity of S_i. An $\mathcal{L}[\sigma, \tau]$-*interpretation* is a tuple

$$I(\overline{z}) = (\varphi_\delta(\overline{x}, \overline{z}), \varphi_\approx(\overline{x}_1, \overline{x}_2, \overline{z}), \varphi_{S_1}(\overline{x}_1, ..., \overline{x}_{s_1}, \overline{z}), ..., \varphi_{S_\ell}(\overline{x}_1, ..., \overline{x}_{s_\ell}, \overline{z}))$$

where $\varphi_\delta, \varphi_\approx, \varphi_{S_1}, ..., \varphi_{S_\ell} \in \mathcal{L}[\sigma]$ and $\overline{x}, \overline{x}_1, ..., \overline{x}_{s_\ell}$ are tuples of pairwise distinct variables of the same length d and $\overline{z}$ is a tuple of variables pairwise distinct from the x-variables. We call d the *dimension* and $\overline{z}$ the *parameters* of $\mathcal{I}(\overline{z})$.

A d-dimensional $\mathcal{L}[\sigma,\tau]$-interpretation $\mathcal{I}(\overline{z})$ defines a partial mapping $\mathcal{I} : Str(\sigma, \overline{z}) \to Str(\tau)$ in the following way: For $(\mathfrak{A}, \overline{z} \mapsto \overline{a}) \in Str(\sigma, \overline{z})$ we obtain a τ-structure $\mathfrak{B}$ over the universe $\{\overline{b} \in A^d \mid \mathfrak{A} \models \varphi_\delta(\overline{b}, \overline{a})\}$, setting $S_i^{\mathfrak{B}} = \{(\overline{b}_1, .., \overline{b}_{s_i}) \in B^{s_i} \mid \mathfrak{A} \models \varphi_{S_i}(\overline{b}_1, ..., \overline{b}_{s_i}, \overline{a})\}$ for each $S_i \in \tau$. Moreover let $\mathcal{E} = \{(\overline{b}_1, \overline{b}_2) \in A^d \times A^d \mid \mathfrak{A} \models \varphi_\approx(\overline{b}_1, \overline{b}_2, \overline{a})\}$. Now we define

$$\mathcal{I}(\mathfrak{A}, \overline{z} \mapsto \overline{a}) := \begin{cases} \mathfrak{B}/\mathcal{E} & \text{if } \mathcal{E} \text{ is a congruence relation on } \mathfrak{B} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We say that $\mathcal{I}$ interprets $\mathfrak{B}/\mathcal{E}$ in $\mathfrak{A}$.

Let $\mathcal{L}$ be a logic and $\mathcal{K} \subseteq Str(\tau)$ a class of τ-structures with $\tau = \{S_1, ..., S_\ell\}$. The *Lindström extension* $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ of $\mathcal{L}$ by *Lindström quantifiers* for the class $\mathcal{K}$ is obtained by extending the syntax of $\mathcal{L}$ by the following formula creation rule:

> Let $\varphi_\delta, \varphi_\approx, \varphi_{S_1}, ..., \varphi_{S_\ell}$ be formulas in $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ that form an $\mathcal{L}[\sigma,\tau]$-interpretation $\mathcal{I}(\overline{z})$. Then $\psi(\overline{z}) = \mathcal{Q}_\mathcal{K}\mathcal{I}(\overline{z})$ is a formula in $\mathcal{L}(\mathcal{Q}_\mathcal{K})$ over the signature σ, with $(\mathfrak{A}, \overline{z} \mapsto \overline{a}) \models \mathcal{Q}_\mathcal{K}\mathcal{I}(\overline{z})$, iff $\mathfrak{B} := \mathcal{I}(\mathfrak{A}, \overline{z} \mapsto \overline{a})$ is defined and $\mathfrak{B} \in \mathcal{K}$.

Fixed-point logic with counting We assume that the reader is familiar with least fixed-point logic, denoted LFP. In finite model theory, a very important extension of LFPis fixed-point logic with counting, FPC. FPC is evaluated on *two-sorted structures*. For any finite, one-sorted σ-structure $\mathfrak{A}$ with universe A, we define the two-sorted extension $\mathfrak{A}^+ := \mathfrak{A} \uplus (\{0, ..., |A|\}; <)$, where $<$ is the canonical ordering on $\{0, ..., |A|\}$. We call the thus extended vocabulary σ^+. The elements of A form the *point sort* and $\{0, ..., |A|\}$ is called the *numeric sort*. Fixed-point logic with counting (FPC) is the extension of least-fixed point logic over such two-sorted structure by counting quantifiers, so that we have formulas $\exists^{\geq\lambda} x\varphi$, where λ is a numeric variable, saying that there exist at least λ many points $a \in A$ making $\varphi(a)$ true. The importance of FPC comes from the fact that it can express many fundamental algorithmic techniques and comes very close to being a logic for polynomial time. For more details on FPC, we refer to [15, 24].

Representing propositional formulas as relational structures Propositional formulas (in CNF) can be represented as structures of some fixed vocabulary in several ways. We shall briefly discuss two possibilities. Since these, and others, are mutually interpretable into each other by simple formulas, it does not really matter which representation we choose; the corresponding Lindström extensions of FO will all have the same expressive power. Perhaps the most obvious representation of a CNF-formula ψ as a structure $\mathfrak{A}(\psi)$ is based on the vocabulary $\{C, V, P, N\}$; the universe of $\mathfrak{A}(\psi)$ consists of the variables and the clauses of ψ, the monadic relations V and C identify the variables and clauses, respectively, and the binary relations P and N specify which variables appear positively and negatively in which clauses; so Pvc is true in $\mathfrak{A}(\psi)$ if the variable v appears positively in the clause c, and analogously for N. A different representation, that sometimes leads to more elegant logical descriptions works with the set L of literals and with a self-inverse bijection $\neg : L \to L$, so that ψ would be represented by $\mathfrak{A}(\psi) = (A, C, L, \neg, \in)$ where A is the set of clauses and literals, $\neg(x)$ is the complementary literal to x, and $x \in c$ means that the literal x occurs in the clause c.

3 Resolution and Fixed-Point Logics

3.1 Horn Resolution captures Least Fixed-Point Logic

Let posLFP be the fragment of LFP-formulas that are in negation normal form (i.e. negation is applied only to input atoms), in which each fixed-point variable is bound only once, and that do not make use of greatest fixed points. Further, let EFP_0 be the basic existential fragment of LFP; it consists of those formulas in posLFP whose quantifiers are all existential.

It is known that, on finite structures (but not in general), every LFP-formula can be effectively translated into an equivalent one in posLFP. On the other side EFP_0 is strictly weaker; it has the same expressive power as Datalog with negation of input atoms.

▶ **Theorem 1.** *For every $\varphi \in \mathrm{posLFP}[\tau]$ there is a first-order interpretation I_φ that maps finite τ-structures to propositional Horn formulas $\psi_{\mathfrak{A},\varphi}$ such that $\mathfrak{A} \models \varphi$ if, and only if, $\psi_{\mathfrak{A},\varphi}$ is unsatisfiable. Further, if φ is in EFP_0 then all clauses in $\psi_{\mathfrak{A},\varphi}$ have width at most three.*

Proof. Fix a formula $\varphi \in \mathrm{posLFP}[\tau]$. For every finite τ-structure $\mathfrak{A}$, with universe A, we construct the propositional Horn formula $\psi_{\mathfrak{A},\varphi}$ as follows. An *instantiated subformula* of φ is an expression $\beta(\overline{a})$ which is obtained by taking some subformula $\beta(\overline{x})$ of φ and by instantiating every free variable x by some element $a \in A$. We now take for every instantiated subformula β of φ a propositional variable X_β, and inductively define a set $C(\mathfrak{A},\varphi)$ of clauses as follows.

(1) If β is a τ-literal then we add $1 \to X_\beta$ in case that $\mathfrak{A} \models \beta$ and $X_\beta \to 0$ in case $\mathfrak{A} \not\models \beta$.
(2) If $\beta = \eta \vee \vartheta$ we add the clauses $X_\eta \to X_\beta$ and $X_\vartheta \to X_\beta$.
(3) If $\beta = \eta \wedge \vartheta$ we add the clause $X_\eta \wedge X_\vartheta \to X_\beta$.
(4) If $\beta = \exists x \eta(x)$ then we add all clauses $X_{\eta(a)} \to X_\beta$ for $a \in A$.
(5) If $\beta = \forall x \eta(x)$ then we add the clause $(\bigwedge_{a\in A} X_{\eta(a)}) \to X_\beta$.
(6) If $\beta = [\mathbf{lfp}\, Rx\,.\,\eta](\overline{a})$ or $\beta = R\overline{a}$, then we add the clause $X_{\eta(\overline{a})} \to X_\beta$.

By induction, it readily follows that the minimal model of all these clauses sets the variable X_β to true if, and only if $\mathfrak{A} \models \beta$ (with fixed-point variables interpreted by their least fixed-point on $\mathfrak{A}$). Let now $\psi_{\mathfrak{A},\varphi}$ be defined as the conjunction of all clauses in $C(\mathfrak{A},\varphi)$ together with $X_\varphi \to 0$. Then $\psi_{\mathfrak{A},\varphi}$ is unsatisfiable if, and only if, $\mathfrak{A} \models \varphi$.

We observe that the only clauses of size larger than three are those coming from universal quantifiers. Hence, if there are no universal quantifiers, the formula only has clauses of size at most three. Finally it is clear that, for every fixed $\varphi \in \mathrm{posLFP}[\tau]$, we can interpret (a representation of) the formula $\psi(\mathfrak{A},\varphi)$ inside $\mathfrak{A}$, by using an FO-interpretation I_φ. ◀

This shows that $\mathrm{LFP} \leq \mathrm{FO}(\textsc{Horn-Res})$. Actually we established a stronger result.

▶ **Theorem 2.** *For every formula $\varphi \in \mathrm{LFP}$ there exists a first-order interpretation J_φ such that $\mathcal{Q}_{\textsc{Horn-Res}}(J_\varphi)$ is equivalent to φ on finite structures. In particular, each* LFP*-formula can be translated into an equivalent* FO(Horn-Res)*-formula with a single application of the generalised quantifier $\mathcal{Q}_{\textsc{Horn-Res}}$.*

We are ready to prove that FO(Horn-Res) has the same expressive power as LFP.

▶ **Theorem 3.** *On finite structures,* $\mathrm{LFP} = \mathrm{FO}(\textsc{Horn-Res})$.

It remains to show that $\mathrm{FO}(\textsc{Horn-Res}) \leq \mathrm{LFP}$, that is we have to express Horn resolution in LFP. Recall that a propositional Horn formula ψ admits a derivation of the empty clause if, and only if, ψ contains a clause in which all variables appear negatively,

written $X_1 \dots X_k \to 0$, such that all unit clauses $\{X_i\}$ for $i = 1, \dots, k$ can be derived from ψ by Horn resolution.

Let ψ be presented as a structure $\mathfrak{A}(\psi)$ with universe $C \cup V$ and vocabulary $\{C, V, P, N\}$. Let D be the set of variables $v \in V$ such that the clause $\{v\}$ can be derived from ψ by Horn resolution. Then ψ is unsatisfiable if, and only if, $\mathfrak{A}(\psi) \models \exists c(Cc \wedge \neg\exists x Pxc \wedge \forall x(Nxc \to Dx))$. The set D is definable by the LFP-formula $[\mathbf{lfp}\, Dx\,.\, \exists c(Pxc \wedge \forall y(Nyc \to Dy))](x)$.

3.2 Bounded Width Resolution and Existential Least Fixed-Point Logic

Intuitively, *existential least fixed-point logic* (EFP) extends EFP_0 by stratified negation. This means that it permits fixed-point formulas over existential formulas which may depend on closed fixed-point relations, defined in a lower stratum, and these can be used also in negated form. Thus, negation (and hence, implicitly, also universal quantifiers) are present in a limited form, but least fixed-point recursions may never go through negation or universal quantification. In fact, EFP is equivalent to Stratified Datalog.

▶ **Definition 4.** Existential fixed-point logics $\mathrm{EFP} := \bigcup_{\ell \geq 0} \mathrm{EFP}_\ell$ generalises EFP_0 as follows. The stratum $\mathrm{EFP}_{\ell+1}$ is the closure under disjunction, conjunction and existential quantification of formulas of form $[\mathbf{lfp}\ R\overline{x}.\exists\overline{y}\varphi(R, \overline{x}, \overline{y})](\overline{x})$ where $\varphi(R, \overline{x}, \overline{y})$ is obtained from a quantifier-free formula, that may contain positive and negative occurrences of additional relations $S_1, \dots, S_m$, by substituting these relations by formulas from EFP_ℓ.

Notice that first-order logic FO is contained in EFP, but not in any bounded level EFP_ℓ, because every quantifier alternation in FO must be simulated by an additional level of stratified negation. For the same reason EFP, but none of its levels EFP_ℓ, is closed under first-order operations. As a consequence of Theorem 1 we can infer

▶ **Theorem 5.** *On finite structures,* $\mathrm{EFP} \leq \mathrm{FO}(3\text{-Res})$.

Proof. Theorem 1 directly establishes this for EFP_0. So assume that the claim is established for EFP_ℓ. Every formula in $\mathrm{EFP}_{\ell+1}$ can be written as an EFP_0-formula over predicates that are EFP_ℓ-definable. Hence, by applying Theorem 1 once more, it can be rewritten as an $\mathrm{FO}(3\text{-Res})$-formula over predicates that are themselves definable in $\mathrm{FO}(3\text{-Res})$. Since Lindström extensions of FO are closed under nesting of generalised quantifiers, it follows that also $\mathrm{EFP}_{\ell+1} \leq \mathrm{FO}(3\text{-Res})$. ◀

We require clauses of width 3 for translating EFP-formulas into Horn formulas. In fact, if we restrict to clauses of width 2, then we obtain the power of first-order logic with a transitive closure operator $\mathrm{FO}(\mathrm{TC})$. This immediately follows from the fact that satisfiability of 2-CNF formulas reduces to graph reachability, and from the reduction of graph reachability to the non-satisfiability problem for a 2-CNF formula that we described in the introduction.

▶ **Theorem 6.** *It holds that* $\mathrm{FO}(2\text{-Res}) = \mathrm{FO}(\mathrm{TC})$.

Simulating bounded width resolution in EFP To describe resolution of width k, for any fixed $k \geq 1$, in EFP, we shall use the representation of CNF-formula ψ by structures $\mathfrak{A}(\psi) = (A, C, L, \neg, \in)$ where C is the set of clauses and L is the set of literals, and the universe is $A = C \cup L \cup \{0\}$. Further we shall describe the set of all derivable clauses of size at most k as a k-ary relation $D \subseteq (L \cup \{0\})^k$, that contains those k-tuples $(x_1, \dots, x_k)$ for which $\{x_i : i \leq k, x_i \neq 0\}$ is a clause that is derivable from ψ. This relation D is defined by a fixed-point formula $[\mathbf{lfp}\, D\overline{x}\,.\,\varphi(D, \overline{x})](\overline{x})$ where $\varphi(D, \overline{x})$ expresses the following. Either

(1) there exists a clause $c \in C$ such that $c = \{x_1, \dots, x_k\} \setminus \{0\}$, or
(2) there exist tuples $\overline{y}, \overline{z} \in D$ such that, for some i, j, the literal z_j is the negated literal to y_i, and $(\{y_1, \dots, y_k\} \cup \{z_1, \dots, z_k\}) \setminus \{y_i, z_j, 0\} = \{x_1, \dots, x_k\} \setminus \{0\}$.

When spelling out these equations in first-order logic, we can express $\varphi(\overline{x}, D)$ by an existential FO-formula $\exists \overline{y}\, \alpha(\overline{x}, \overline{y}, D, Q)$ where Q is FO-definable by a formula (with quantifier prefix $\exists^*\forall$) that does not depend on D. This yields a formula in EFP_1. Since EFP is closed under FO-operations, this proves

▶ **Theorem 7.** *On finite structures,* $\mathrm{FO}(k\text{-}\textsc{Res}) \leq \mathrm{EFP}$ *for all* $k \in \mathbb{N}$.

4 The Monomial-PC and Fixed-Point Logic with Counting

We turn our attention to the *monomial-PC* (MON-PC). The monomial-PC is a propositional proof system that is based on algebraic reasoning techniques and which lies between the *Nullstellensatz* proof system and the *polynomial calculus* (PC). It was introduced by Berkholz and Grohe in [9] in order to characterise the power of an important graph isomorphism test, the Weisfeiler-Lehman method, in terms of propositional proof complexity. We show in this section that the monomial-PC has precisely the same expressive power as fixed-point logic with counting (FPC), which is a natural and powerful logic of great importance in the area of descriptive complexity theory.

4.1 The Monomial-PC

We start with background on the polynomial calculus and its variant, the monomial-PC. Both systems refute the solvability of a given set of *(multivariate) polynomial equations* over some field $\mathbb{F}$ using proof rules that manipulate such equations. In this paper, $\mathbb{F}$ will always be the field of rationals $\mathbb{Q}$. We denote by $\mathbb{Q}[\vec{X}]$ the ring of polynomials in variables X_j, $j \in J$, for some (unordered) index set J and with coefficients in $\mathbb{Q}$. For a multi-index $\alpha : J \to \mathbb{N}$ we let the *monomial* X^α be defined as $X^\alpha = \Pi_{j \in J} X_j^{\alpha(j)}$. Then polynomials $f \in \mathbb{Q}[\vec{X}]$ can be written as $f = \sum_\alpha f_\alpha \cdot X^\alpha$ where the $f_\alpha \in \mathbb{Q}$ are coefficients from the field $\mathbb{Q}$ and such that $f_\alpha \neq 0$ for finitely many α only. The *degree* $\deg(X^\alpha)$ of a monomial X^α is defined as $|\alpha| = \sum_{j \in J} \alpha(j)$, and the degree of a polynomial is defined as the maximal degree of its monomials. A *polynomial equation* is an equation of the form $f = 0$ for a polynomial $f \in \mathbb{Q}[\vec{X}]$. For better readability, we usually omit the equality "$= 0$" when we specify polynomial equations, that is we identify polynomials $f \in \mathbb{Q}[\vec{X}]$ with the corresponding normalised polynomial equations $f = 0$. A *system of polynomial equations* is a set $\mathcal{P} = \{f_i : i \in I\}$ consisting of polynomials $f_i \in \mathbb{Q}[\vec{X}]$ for all $i \in I$ where I is an (unordered) index set. A *solution* of $\mathcal{P}$ is a common zero $\overline{a} \in \mathbb{Q}^J$ of all polynomials in $\mathcal{P}$. In what follows, we only consider systems $\mathcal{P} = \{f_i : i \in I\}$ which contain for every variable $X = X_j$, $j \in J$, the polynomial equation $(X^2 - X) = 0$. The axioms $(X^2 - X) = 0$ enforce that each variable $X = X_j$, $j \in J$, can only take values 0 or 1.

The polynomial calculus is based on the following result from algebra which is known as *Hilbert's Nullstellensatz*. It says that the non-solvability of the system $\mathcal{P} = \{f_i : i \in I\}$ is equivalent to the existence of polynomials $g_i \in \mathbb{Q}[\vec{X}], i \in I$, such that $\sum_{i \in I} g_i \cdot f_i = 1$. The polynomials g_i are called a *Nullstellensatz refutation* for the system $\mathcal{P}$. The idea of the polynomial calculus is to search for such polynomials g_i in a sequential way.

▶ **Definition 8.** The axioms and inference rules of *polynomial calculus* (PC) are as follows.

$$\frac{p \in \mathcal{P}}{p}\ \text{(Axioms)} \qquad \frac{f}{Xf}\ \text{(Multiplication)} \qquad \frac{g \quad f}{ag + bf}\ \text{(Linear combinations)}$$

The *monomial-PC* (MON-PC) is the restriction that permits the use of the multiplication rule only in the cases where f is either a monomial or the product of a monomial and an axiom. A polynomial equation system system $\mathcal{P}$ has a *refutation* of degree $k \geq 1$ in PC (or MON-PC) if the polynomial $1 \in \mathbb{Q}[\vec{X}]$ can be derived from $\mathcal{P}$ using the above rules using polynomials of degree at most k.

The polynomial calculus, and also the monomial-PC, are sound and, by Hilbert's Nullstellensatz, complete proof systems. However, this only holds if we do not restrict the degree of the polynomials which are allowed to occur in a refutation. In fact, the "degree of polynomials" for the PC (MON-PC) is a complexity measure which has similar properties as the "width of clauses" measure that we studied for the resolution proof system. If we restrict the PC (MON-PC) to polynomials of degree at most k, for some fixed $k \geq 1$, then the systems become incomplete, but admit proof search in polynomial time. In what follows, whenever we speak of the monomial-PC, we implicitly refer to the variant where we restricted the degree of polynomials to some constant $k \geq 1$. If we want to make this constant precise, then we denote the corresponding proof system by MON-PC$_k$. Another fact which we use throughout this section is that the axioms $(X^2 - X)$ guarantee that in (monomial-)PC proofs we can restrict ourselves to *multilinear* polynomials. To see this, say that we were able to derive the polynomial $p = X^2Y + Z$ within some (monomial)-PC proof. Of course, p is not multilinear. However, we can use the axiom $(X^2 - X)$ together with the "linear combination"-rule to reduce this polynomial to the corresponding multilinear polynomial $p' = XY + Z$. Indeed, $p' = p - Y(X^2 - X)$. Hence, restricting to multlinear polynomials, and modifying the multiplication rule accordingly with implicit linearisation, does not change the power of the corresponding proof systems. For a polynomial $p \in \mathbb{Q}[\vec{X}]$ we denote its *multilinearisation* by MultLin(p).

4.2 Monomial-PC in Fixed-Point Logic with Counting

We show that FPC can simulate MON-PC$_k$. For this, we have to agree on an encoding of a set $\mathcal{P}$ of rational, multilinear polynomials as relational structures. Similar to our representation of CNF-formulas described in Section 2, a natural encoding can be based on a many-sorted structure $\mathfrak{A}_\mathcal{P}$ whose universe is partitioned into sets of polynomials, (multilinear) monomials, variables, and rational coefficients that occur in $\mathcal{P}$. As usual, we represent rationals as fractions of integers using binary encoding. Hence, $\mathfrak{A}_\mathcal{P}$ should also provide a linear order of sufficient length to encode these binary strings. Again, the exact technical details are not important, as long as the encoding has some natural properties, such as FO-definability of the class of valid encodings. By a slight abuse of notation, we also denote by MON-PC$_k$ the class of structures $\mathfrak{A}_\mathcal{P}$ which encode a system $\mathcal{P}$ which can be refuted in MON-PC$_k$.

▶ **Theorem 9.** *For every $k \geq 1$,* MON-PC$_k \in$ FPC.

Given a set of multilinear polynomials $\mathcal{P}$ of degree at most k, we consider the set $V_\mathcal{P}$ of multilinear polynomials which can be derived from $\mathcal{P}$ within MON-PC$_k$. The first observation is that $V_\mathcal{P}$ is a $\mathbb{Q}$-linear space. Since $V_\mathcal{P}$ only contains multilinear polynomials of degree at most k, we can naturally associate polynomials $p \in V_\mathcal{P}$ with vectors $p \in \mathbb{Q}^{M_k}$ where the index set M_k denotes the set of all multilinear monomials of degree at most k. For fixed $k \geq 1$, this set M_k is of polynomial size.

To prove Theorem 9 we express in FPC an inductive algorithm (based on a similar algorithm for the full polynomial calculus in [13]) for computing a generating set of the space $V_\mathcal{P}$. Then, in order to see whether MON-PC$_k$ can refute the system $\mathcal{P}$, we simply have to check whether the constant polynomial 1 is contained in $V_\mathcal{P}$, see Figure 1.

During the run of the algorithm we have that $\langle \mathcal{B} \rangle \leq V_{\mathcal{P}}$. Moreover, after termination it holds that $\langle \mathcal{B} \rangle = V_{\mathcal{P}}$. We further observe that after the initialisation step we only add *monomials* to the set $\mathcal{B}$. Since there are only polynomially many different monomials of degree at most k, for a fixed k, this means that the algorithm is guaranteed to terminate after a polynomial number of iterations.

Input: Set of multilinear polynomials $\mathcal{P} \subseteq \mathbb{Q}^{M_k}$
Output: $\mathcal{B} \subseteq \mathbb{Q}^{M_k}$ such that $\langle \mathcal{B} \rangle = V_{\mathcal{P}}$.
Initialisation (lift all axioms in $\mathcal{P}$)
$\mathcal{B} := \{\text{MultLin}(m \cdot p) \mid p \in \mathcal{P}, m \text{ a monomial s.th. } \deg(\text{MultLin}(m \cdot p)) \leq k\}$
repeat
  for all monomials $m \in \langle \mathcal{B} \rangle$, $\deg(m) < k$ **do**
    $\mathcal{B} := \mathcal{B} \cup \{\text{MultLin}(X \cdot m) : \text{for some variable } X\}$
  end for
until $\mathcal{B}$ remains unchanged
return $\mathcal{B}$

Figure 1 FPC-procedure to define generating set for $V_{\mathcal{P}}$

It is not obvious how to express this algorithm in FPC. Most steps, such as the representation of the set $\mathcal{B}$ and the multilinearisation of polynomials, are easy to formalise, but there is a severe obstacle hidden in the condition for the main loop. Here, we want to iterate, in parallel, through all monomials $m \in \langle \mathcal{B} \rangle$. This condition "$m \in \langle \mathcal{B} \rangle$" translates to solving a linear equation system over $\mathbb{Q}$. Although it is provably impossible to express the method of Gaussian elimination in FPC, since it requires arbitrary choices during its computation, and although FPC cannot define the solvability of linear equation systems over finite fields [3], it is known [17] that FPC can indeed express solvability of linear equation systems over the rationals.

▶ **Theorem 10** ([17]). *The solvability of linear equation systems over $\mathbb{Q}$ is definable in* FPC.

Using this result we can express the algorithm in FPC. In order to complete our proof of Theorem 9 we recall that MON-PC$_k$ can refute $\mathcal{P}$ if, and only if, $1 \in \langle \mathcal{B} \rangle = V_{\mathcal{P}}$. This last assertion, again, reduces to a linear equation system over $\mathbb{Q}$ and can thus be defined in FPC.

4.3 Monomial-PC captures Fixed-Point Logic with Counting

Next we show that the monomial-PC can simulate fixed-point logic with counting. We first observe, however, that the logic FO(MON-PC$_k$) does *not* suffice for this purpose. This is due to the fact that FPC has access to the second numeric sort, on which it can perform arbitrary polynomial time computations, whereas FO(MON-PC$_k$) is evaluated over standard single sorted input structures. To overcome this mismatch we have to extend the logic FO(MON-PC$_k$) to the second-sorted framework as well. We denote this extension of FO(MON-PC$_k$) by FO$^+$(MON-PC$_k$). As in the case of FPC, this means that formulas are evaluated over extensions $\mathfrak{A}^+$ of relational structures $\mathfrak{A}$ by a second numeric sort, as defined in Section 2. In particular, interpretations for the Lindström quantifiers can make use of the second numeric sort, and we require this capability in the proof of our following result.

▶ **Theorem 11.** *For every $k \geq 2$,* FPC $\leq$ FO$^+$(MON-PC$_k$).

An elegant way to prove Theorem 11 is to use a game-theoretic characterisation of FPC which was recently established in [20]. It is based on the notion of so called *threshold games*.

A *threshold game* is a two-player game played on a directed graph $G = (V, E)$ that is equipped with a *threshold function* $\vartheta \colon V \to \mathbb{N}$. This function satisfies that $\vartheta(v) \leq \delta(v) + 1$ for all $v \in V$, where $\delta(v)$ denotes the out-degree of v in G. Moreover, there is a designated vertex $s \in V$ at which each play starts. A play is a sequence of G-nodes that arises

according to the following rules. At the current position $v \in V$, Player 0 first selects a set $X \subseteq vE = \{w : (v, w) \in E\}$ with $|X| \geq \vartheta(v)$. Then Player 1 chooses a node $w \in X$ and the play moves on to w. A player who cannot move loses. Hence Player 0 wins at all nodes in $T_0 := \{v \in V \mid \vartheta(v) = 0\}$ and Player 1 at all nodes in $T_1 := \{v \in V \mid \delta(v) < \vartheta(v)\}$.

In [20] it is shown that threshold games provide appropriate model-checking games $\mathcal{T}(\mathfrak{A}, \varphi)$ for any finite structure $\mathfrak{A}$ and any formula $\varphi \in \mathrm{FPC}$. Since fixed-point evaluations on finite structures can be uniformly unraveled to first-order evaluations, we can in fact assume that the game graphs of these threshold games are acyclic. For any fixed FPC-formula φ, these model checking games are polynomially bounded in the size of the input structure and can, in fact, be interpreted in (two-sorted) input structures using a first-order interpretation. This is related to the transformation of FPC-formulas into uniform families of polynomial-size threshold circuits, as used for instance in [23] and [1].

▶ **Theorem 12** ([20]). *For every* FPC*-formula φ there is a first-order interpretation I_φ which, for every finite structure $\mathfrak{A}$, interprets in $\mathfrak{A}^+$ an acyclic threshold game $\mathcal{G}(\mathfrak{A}, \varphi)$ such that $\mathfrak{A} \models \varphi$ if, and only if, Player 0 has a winning strategy for $\mathcal{G}(\mathfrak{A}, \varphi)$.*

It remains to show that the monomial-PC can define winning regions in acyclic threshold games. Given an acyclic threshold game $\mathcal{G} = (G = (V, E), \vartheta)$, we construct an axiom system $\mathcal{P}(\mathcal{G})$ which consists of polynomial equations of degree at most two. For every node $v \in V$ in the threshold game $\mathcal{G}$, the system $\mathcal{P}(\mathcal{G})$ contains a variable X_v. Let us denote by $W_\sigma^{\mathcal{G}}$ the winning region of Player σ in $\mathcal{G}$. Then $\mathcal{P}(\mathcal{G})$ satisfies the following:

- if $v \in W_0^{\mathcal{G}}$, then $X_v = 1$ is derivable from $\mathcal{P}(\mathcal{G})$ in MON-PC$_2$;
- if $v \in W_1^{\mathcal{G}}$, then $X_v = 0$ is derivable from $\mathcal{P}(\mathcal{G})$ in MON-PC$_2$;
- $\mathcal{P}(\mathcal{G})$ is consistent; in particular, either $X_v = 1$ or $X_v = 0$ is derivable for every $v \in V$;

If we can construct such a system $\mathcal{P}(\mathcal{G})$ via an FO-interpretation in $\mathcal{G}$, then this completes our proof of Theorem 11. In fact, it then follows that FO$^+$(MON-PC$_2$) can define winning regions in acyclic threshold games: a node $v \in V$ is in the winning region of Player 0 if, and only if, the system $\mathcal{P}(\mathcal{G}) \cup \{X_v = 0\}$ can be refuted in MON-PC$_2$.

Recall that $vE = \{w \in V : (v, w) \in E\}$, for $v \in V$, denotes the set of successors of v. Further, we let $\mathrm{s}(v)$ denote the number of successors of v, and we let $\mathrm{ws}(v)$ denote the number of successors of v which are in the winning region of Player 0, that is $\mathrm{s}(v) = |vE|$ and $\mathrm{ws}(v) = |vE \cap W_0^{\mathcal{G}}|$. We denote the set of non-terminal positions by $\mathrm{NonTerm} = \{v \in V : \mathrm{s}(v) > 0\}$. The system $\mathcal{P}(\mathcal{G})$ uses the following set of variables:

- a variable X_v, for every $v \in V$,
- a variable Y_v^m for every $v \in \mathrm{NonTerm}$, and $0 \leq m \leq \mathrm{s}(v)$,
- a variable $Z_v^m[u \mapsto j]$ for every $v \in \mathrm{NonTerm}$, $1 \leq m \leq \mathrm{s}(v)$, $1 \leq j \leq m$, $u \in vE$.

The intuition is that the variables X_v encode the winning regions of both players, as described above. Moreover, the variables Y_v^m should indicate whether $\mathrm{ws}(v) = m$, in the following way: if $\mathrm{ws}(v) \neq m$, then $Y_v^m = 0$ is derivable, and if $\mathrm{ws}(v) = m$, then $Y_v^m = 1$ is derivable. The variables $Z_v^m[u \mapsto j]$ are auxiliary variables used to encode this last condition, cf. [9]. The

system $\mathcal{P}(\mathcal{G})$ consists of the following axioms:

(T) For $v \in T_0 : X_v = 1$ and for $v \in T_1 : X_v = 0$

(C) For $v \in \text{NonTerm}, 1 \leq m \leq \text{s}(v), u \in vE$: $\sum_{j=1}^{m} Z_v^m[u \mapsto j] - Y_v^m = 0$

For $v \in \text{NonTerm}, 1 \leq m \leq \text{s}(v), 1 \leq j \leq m$: $\sum_{u \in vE} X_u Z_v^m[u \mapsto j] - Y_v^m = 0$

For $v \in \text{NonTerm}$: $\sum_{u \in vE} X_u \cdot Y_v^0 = 0$

(E) For $v \in V : (1 - X_v) - \sum_{m=0}^{\vartheta(v)-1} Y_v^m = 0$ and $X_v - \sum_{m=\vartheta(v)}^{\text{s}(v)} Y_v^m = 0$

We also add for each variable $X = X_v$ a dual variable $\overline{X}$ with the axiom (N) $1 - X = \overline{X}$.

▶ **Lemma 13.** *The system $\mathcal{P}(\mathcal{G})$ is consistent.*

Proof. We define an intended model of $\mathcal{P}(\mathcal{G})$. For X-variables, we set $X_v := 1$, if $v \in W_0^{\mathcal{G}}$, and $X_v := 0$, if $v \in W_1^{\mathcal{G}}$. For Y-variables, we set $Y_v^m := 1$, if $\text{ws}(v) = m$, and $Y_v^m := 0$ if $m \neq \text{ws}(v)$. For Z-variables, we set $Z_v^m[u \mapsto j] := 0$ for all non-terminal positions $v \in V$, $u \in vE$, and $j \in \{1, \ldots, m\}$, if $m \neq \text{ws}(v)$. For $m = \text{ws}(v) > 0$, we let $vE \cap W_0^{\mathcal{G}} = \{u_1, \ldots, u_m\}$. We then set $Z_v^m[u_i \mapsto j] = 1$ if $j = i$, and $Z_v^m[u_i \mapsto j] = 0$ for $j \neq i$. Moreover, for $u \in vE \setminus W_0^{\mathcal{G}}$, we set $Z_v^m[u \mapsto 1] = 1$, and $Z_v^m[u \mapsto j] = 0$ for $j \in \{2, \ldots, m\}$. ◀

▶ **Lemma 14.** *If $v \in W_0^{\mathcal{G}}$, then we can derive $X_v = 1$ from $\mathcal{P}(\mathcal{G})$ in* MON-PC$_2$*; and if $v \in W_1^{\mathcal{G}}$, then $X_v = 0$ is derivable from $\mathcal{P}(\mathcal{G})$ in* MON-PC$_2$.

Proof. We start with a small remark. Assume that we can derive $(1 - X)$ for a variable $X = X_v$, $v \in V$. We show how to derive $V(1 - X)$ for every variable V. This is clearly possible in the full polynomial calculus. In the monomial-PC, however, we cannot multiply $(1 - X)$ by V, since $(1 - X)$ is neither a monomial nor an axiom. Instead, we use our negation axioms: We obtain $\overline{X} = 0$ by subtracting $X = 1$ from (N). Since (N) is an axiom, we can multiply it by V; also, $\overline{X}$ is a monomial and it can be multiplied by V. Thus, $V(1 - X - \overline{X}) + V\overline{X} = V(1 - X)$ can be derived. We make use of this in what follows.

The proof is by induction on the height of the subgame rooted at $v \in V$ (recall that $\mathcal{G}$ is acyclic). For terminal positions $v \in V$, the claim is obvious. Assume $v \in V$ is a non-terminal position. Let $W_0(v) = vE \cap W_0^{\mathcal{G}}$ and $W_1(v) = vE \cap W_1^{\mathcal{G}}$. By the induction hypothesis we know that we can derive in MON-PC$_2$ for every $u \in W_0(v)$ the equation $X_u = 1$ and for every $u \in W_1(v)$ the equation $X_u = 0$.

Let $m > 0$. Consider an equation of the form $\sum_{u \in vE} X_u Z_v^m[u \mapsto j] - Y_v^m = 0$ for $j \in \{1, \ldots, m\}$ of type (C). We have $vE = W_0(v) \uplus W_1(v)$. For every Z-variable and for every $u \in W_0(v)$ we can derive $ZX_u = Z$ in MON-PC$_2$, and for every $u \in W_1(v)$ we can derive $ZX_u = 0$ in MON-PC$_2$. Hence, we can simplify these equations of type (C) as $\sum_{u \in W_0(v)} Z_v^m[u \mapsto j] - Y_v^m = 0$ for $j \in \{1, \ldots, m\}$ in MON-PC$_2$.

Next, we consider for every $u \in W_0(v)$ the equations $\sum_{j=1}^{m} Z_v^m[u \mapsto j] - Y_v^m = 0$, again of type (C). We combine these two sets of equations as follows:

$$\sum_{j \in \{1,\ldots,m\}} \left(\sum_{u \in W_0(v)} Z_v^m[u \mapsto j] - Y_v^m \right) - \sum_{u \in W_0(v)} \left(\sum_{j=1}^{m} Z_v^m[u \mapsto j] - Y_v^m \right) = (m - \text{ws}(v)) Y_v^m.$$

Hence, for every $m > 0$, $m \neq \mathrm{ws}(v)$, we can derive $Y_v^m = 0$ in MON-PC$_2$. Indeed, also in the case where $m = 0 < \mathrm{ws}(v)$ we can derive $Y_v^m = 0$. In this case we just use the equation $\sum_{u \in vE} X_u Y_v^0 = 0$. Using the same arguments as above, this equation simplifies to $\mathrm{ws}(v) \cdot Y_v^0 = 0$. Hence, if $\mathrm{ws}(v) > 0$, we can also derive $Y_v^0 = 0$. Note that the two equations of type (E) can be combined to the equation $\sum_{m=0}^{s(v)} Y_v^m = 1$. Hence, altogether we showed the following. For all $0 \leq m \leq s(v)$ it holds that:

- if $m = \mathrm{ws}(v)$, then we can derive $Y_v^m = 1$ in MON-PC$_2$; and
- if $m \neq \mathrm{ws}(v)$, then we can derive $Y_v^m = 0$ in MON-PC$_2$.

Having this, the claim follows immediately by using the equations of type (E). Finally, the system $\mathcal{P}(\mathcal{G})$ can easily be obtained from the game $\mathcal{G}$ by means of an FO-interpretation. ◀

In summary, we have seen that defining the winning regions in acyclic threshold games is an FPC-complete problem, with respect to FO$^+$-reductions, and that the winning regions in such games can be defined in FO$^+$(MON-PC$_2$). This completes the proof of Theorem 11 and, together with Theorem 9, establishes the main theorem of this section.

▶ **Theorem 15.** *For every $k \geq 2$,* FPC = FO$^+$(MON-PC$_k$).

5 Applications

5.1 Lower Bounds on Resolution Width and Size

Our characterisation of bounded width resolution in terms of EFP-definability reproves many lower bounds on the resolution size and width for families of propositional formulas.

We denote the finite-variable fragment of *infinitary logic* by $L^\omega_{\infty\omega}$, that is $L^\omega_{\infty\omega}$ is the finite-variable fragment of the extension of first-order logic by infinite conjunctions and disjunctions. Further, we denote by $L^\ell_{\infty\omega}$ the ℓ-variable fragment of $L^\omega_{\infty\omega}$; we have $L^\omega_{\infty\omega} = \bigcup_{\ell \geq 1} L^\ell_{\infty\omega}$. We write $\mathfrak{A} \equiv^\ell \mathfrak{B}$ if two structures $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by any sentence of the ℓ-variable fragment $L^\ell_{\infty\omega}$ of $L^\omega_{\infty\omega}$. It is well-known that EFP, and even LFP, can be embedded into the logic $L^\omega_{\infty\omega}$. More precisely, if $\varphi \in$ LFP is a sentence with ℓ (first-order) variables, then we can find an equivalent sentence φ^* in $L^\ell_{\infty\omega}$. We formulate the following definition and result with respect to $L^\ell_{\infty\omega}$-interpretations, instead of FO-interpretations.

▶ **Definition 16.** Let $k \geq 1$ and let $(\Phi_n) = (\Phi_n)_{n \geq 1}$ be a family of k-CNF formulas. Moreover, let $\mathcal{I}$ be an $L^\ell_{\infty\omega}$-interpretation which transforms τ-structures $\mathfrak{A}$ into k-CNF formulas $\mathcal{I}(\mathfrak{A})$, and let $(\mathfrak{A}_n) = (\mathfrak{A}_n)_{n \geq 1}$ be a family of τ-structures. We say that $\mathcal{I}$ *interprets* (Φ_n) *in* $(\mathfrak{A}_n)$ if for every $n \geq 1$, $\Phi_n = \mathcal{I}(\mathfrak{A}_n)$.

▶ **Theorem 17.** *Let $k \geq 1$, let (Φ_n) and (Ψ_n) be two families of k-CNF formulas, let $\mathcal{I}$ be an $L^\ell_{\infty\omega}$-interpretation that maps τ-structures to k-CNF formulas, and let $(\mathfrak{A}_n)$ and $(\mathfrak{B}_n)$ be two families of τ-structures such that:*

- *(Φ_n) consists of satisfiable formulas, and $\mathcal{I}$ interprets (Φ_n) in $(\mathfrak{A}_n)$;*
- *(Ψ_n) consists of unsatisfiable formulas, and $\mathcal{I}$ interprets (Ψ_n) in $(\mathfrak{B}_n)$;*
- *$\mathfrak{A}_n \equiv^{\Omega(n)} \mathfrak{B}_n$.*

(a) Let RES-WIDTH(n) *denote the resolution width required to refute the formula Ψ_n. Then* RES-WIDTH$(n) \in \Omega(n)$.

(b) Let T-RES-SIZE(n) *denote the size of a tree-like resolution refutation for Ψ_n. Then* T-RES-SIZE(n) *is bounded below by a function in $2^{\Omega(n)}$.*

(c) Let RES-SIZE(n) *denote the size of resolution refutation for Ψ_n. If the formulas Ψ_n only contain $\mathcal{O}(n)$ many variables, then* RES-SIZE(n) *is bounded below by a function in $2^{\Omega(n)}$.*

Proof. First, assume that RES-WIDTH$(n) \notin \Omega(n)$. For $r \geq 1$, we choose an EFP-formula ϑ_r, according to Theorem 7, which expresses that a k-CNF formula has a resolution refutation of width r. We can choose ϑ_r in such a way that it only contains $\mathcal{O}(r)$ many variables. By the embedding of EFP into $L^\omega_{\infty\omega}$ we can actually assume that ϑ_r is an $L^\omega_{\infty\omega}$-formula with at most $\mathcal{O}(r)$ many variables. We now translate the formulas ϑ_r back, via $\mathcal{I}$, to τ-structures, that is we obtain $L^\omega_{\infty\omega}$-formulas $\vartheta^{\mathcal{I}}_r$ such that for every τ-structure $\mathfrak{A}$ it holds that $\mathfrak{A} \models \vartheta^{\mathcal{I}}_r$ if, and only if, $\mathcal{I}(\mathfrak{A})$ has a resolution refutation of width r. Since $\mathcal{I}$ is fixed and only contains formulas with $\ell \geq 1$ many variables, the number of variables in the formulas $\vartheta^{\mathcal{I}}_r$ is still bounded by $\mathcal{O}(r)$. For concreteness, assume that $\vartheta^{\mathcal{I}}_r$ contains at most $c \cdot r$ many variables for some constant $c \geq 1$ and all large enough $r \geq 1$. Further, we choose $d > 0$, such that $\mathfrak{A}_n \equiv^{d \cdot n} \mathfrak{B}_n$ for all large enough $n \geq 1$. By our initial assumption we can now choose, for $e := (1/c) \cdot d$, a still larger $n \geq 1$, such that RES-WIDTH$(n) < e \cdot n$. This, however, means that $\mathfrak{A}_n \not\models \vartheta^{\mathcal{I}}_{e \cdot n}$ and $\mathfrak{B}_n \models \vartheta^{\mathcal{I}}_{e \cdot n}$, which implies that $\mathfrak{A}_n \not\equiv^{e \cdot n \cdot c} \mathfrak{B}_n$. This, however, is a contradiction, as $e \cdot n \cdot c = d \cdot n$.

Hence, we know that RES-WIDTH$(n) \in \Omega(n)$. The remaining statements follow from well-known size-width relations for the resolution proof system. In fact, recall from [8], that the size of a smallest tree-like resolution refutation for a k-CNF formula Ψ is bounded below by 2^{w-k} where w denotes the minimal width required to refute Ψ. Since k is a constant, we get a lower bound on T-RES-SIZE(n) as $2^{\Omega(n)}$. Moreover, it is also shown in [8] that the size of a smallest resolution refutation for a k-CNF formula Ψ with m variables is bounded below by $2^{\Omega((w-k)^2/m)}$. Hence, if we additionally have that the number of variables in Ψ_n is at most $\mathcal{O}(n)$, then this gives us a lower bound of $2^{\Omega(n)}$ for RES-SIZE(n). ◀

We remark that the assumptions of the theorem can be generalised in several ways. For instance, one could formulate a similar theorem for the degree of Monomial-PC proofs by replacing logical equivalence in $L^\omega_{\infty\omega}$ by equivalence in counting logic. This would allow us to generalise some of our lower bounds below on resolution width and size to lower bounds on Monomial-PC degree and size. We defer details to a full version of the paper.

Pigeonhole Principle For $n, m \geq 1$ the pigeonhole principle formulas PHP^m_n express that there is an injective function from a set of size n to a set of size m. The usual definition leads to CNF-formulas of unbounded width. However, one can also formulate the pigeonhole principle using 3-CNF formulas and auxiliary variables, so called extension variables. More precisely, the 3-CNF formula EPHP^m_n contains variables X_{ij}, for $i \in \{1, \dots, n\}, j \in \{1, \dots, m\}$, saying that pigeon i sits in hole j, and auxiliary variables Y_{ij} for every $i \in \{1, \dots, n\}$ and $j \in \{0, \dots, m\}$. The formula EPHP^m_n is built-up using the following formulas:

$$\neg Y_{i0} \wedge \bigwedge_{j=1}^{m} (Y_{ij-1} \vee X_{ij} \vee \neg Y_{ij}) \wedge Y_{im} \qquad \text{for all } i \in \{1, \dots, n\}$$

$$(\neg X_{ij} \vee \neg X_{i'j}) \qquad \text{for all } i, i' \in \{1, \dots, n\}, j \in \{1, \dots, m\}, i \neq i'$$

We consider relational structures $\mathfrak{C}^m_n = ([n] \uplus [m], P = [n], Q = [m], <)$ where $<$ is a linear order on $Q = [m]$. It is straightforward to construct a first-order interpretation $\mathcal{I}$ which interprets the formula EPHP^m_n in the structure $\mathfrak{C}^m_n$ for all $n, m \geq 1$. However, note that we really need the linear order on the set $Q = [m]$ for this. Let $\Phi_n = \mathrm{EPHP}^n_n$ and $\Psi_n = \mathrm{EPHP}^n_{n+1}$, $\mathfrak{A}_n = \mathfrak{C}^n_n$ and $\mathfrak{B}_n = \mathfrak{C}^n_{n+1}$. Then $\mathfrak{A}_n \equiv^n \mathfrak{B}_n$. Hence, all preconditions of Theorem 17 are satisfied (however, the formulas Ψ_n contain more than n^2 many variables).

▶ **Corollary 18.** *The resolution width required to refute the formulas* EPHP^n_{n+1} *is linear in* n*. This implies that the size of treelike resolution refutations is exponential in* n*.*

Three colourability Given a graph $G = (V, E)$ we can write down a propositional formula $\Theta[G]$ saying that the graph G is 3-colourable. For each vertex $v \in V$ of G, and each colour $i \in \{0, 1, 2\}$ we consider a variable X_v^i indicating that the vertex v is coloured with colour i. Then $\Theta[G]$ consists of the following clauses:

$$(X_v^0 \vee X_v^1 \vee X_v^2) \qquad \text{for every } v \in V$$
$$(\neg X_v^i \vee \neg X_w^i) \qquad \text{for each edge } (v, w) \in E \text{ and colour } i \in \{0, 1, 2\}.$$

Note that $\Theta[G]$ is a 3-CNF formula and the number of variables is linear in $|V|$. Again, it is easy to construct a first-order interpretation $\mathcal{I}$ such that for every graph G it holds that $\mathcal{I}(G) = \Theta[G]$. We make use of the following fact from finite model theory, see Appendix B.

▶ **Theorem 19** ([3, 11, 17, 27])**.** *For all $n \geq 1$, there are graphs G_n, H_n with $\mathcal{O}(n)$ many vertices, such that $G_n \equiv^{\Omega(n)} H_n$, and such that G_n is three-colourable and H_n is not three-colourable.*

Using this, we can apply Theorem 17: we set $\mathfrak{A}_n = G_n$, $\mathfrak{B}_n = H_n$, $\Phi_n = \Theta[G_n]$, $\Psi_n = \Theta[H_n]$. Also, note that the number of variables in Ψ_n is bounded by $\mathcal{O}(n)$. We get:

▶ **Corollary 20.** *Refuting the formulas $\Theta[H_n]$ for three-colourability requires resolution proofs of exponential size and linear width.*

Graph isomorphism problem In the introduction we mentioned the graph isomorphism problem. This problem is not known to be decidable in polynomial time, but there is strong evidence that it is not NP-complete. In [26] Toran showed that, with respect to a natural encoding of the graph isomorphism problem as a propositional formula, one cannot obtain an efficient graph isomorphism test based on resolution, not even for graphs with colour class size four. However, one could argue that Toran considered one *specific* encoding of the graph isomorphism problem as a propositional formula only. Maybe one could use a different encoding of the graph isomorphism problem, which may involve simple precomputations, that actually allows efficient resolution refutations? Our results indicate that this is not the case. In fact, Toran's result holds with respect to *every* LFP-definable encoding that uses a linear number of propositional variables only (which is a natural assumption in the context of graphs with constant colour class size), see Appendix A.

5.2 Solving Parity Games via Resolution

To some degree, the complexity-theoretic status of the problem of *computing winning regions in parity games* resembles the situation for graph isomorphism. Most importantly, we do not know whether winning regions in parity games can be computed in polynomial time, but we do not expect that this problem is NP-complete. In fact, for both problems there have been recent breakthroughs providing new algorithms that solve them in quasi-polynomial time [6, 12]. However, there also are notable differences. For example, computing winning regions in parity games is known to be P-hard, but we do not have such strong lower bounds for the graph isomorphism problem. Another remarkable difference between the graph isomorphism problem and the problem of computing winning regions in parity games follows from Theorem 2 and a result due to Dawar and the first author [16], showing that a polynomial-time algorithm for parity games would imply that their winning regions are LFP-definable (despite the fact that LFP is, in general, weaker than polynomial time).

▶ **Theorem 21.** *Parity games can be solved in polynomial time if, and only if, they can be solved using Horn resolution with respect to a first-order definable encoding.*

References

1 M. Anderson and A. Dawar. On symmetric circuits and fixed-point logics. In *STACS 2014*, pages 41–52, 2014.

2 A. Atserias. On sufficient conditions for unsatisfiability of random formulas. *J. ACM*, 51(2):281–311, 2004.

3 A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. *Theoretical Computer Science*, 410:1666–1683, 2009.

4 A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008.

5 A. Atserias and E. N. Maneva. Sherali-adams relaxations and indistinguishability in counting logics. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 367–379. ACM, 2012.

6 L. Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015. URL: `http://arxiv.org/abs/1512.03547`.

7 P. Beame and T. Pitassi. Propositional Proof Complexity: Past, Present, and Future. *Current Trends in TCS: Entering the 21st Century*, pages 42–70, 2001.

8 E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.

9 C. Berkholz and M. Grohe. Limitations of algebraic approaches to graph isomorphism testing. In *Proceedings of ICALP 2015*, pages 155–166, 2015.

10 C. Berkholz and M. Grohe. Linear diophantine equations, group csps, and graph isomorphism. In *Proceedings of SODA 2017*, pages 327–339, 2017.

11 J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

12 C. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. Deciding parity games in quasipolynomial time. In *STOC 2017*, 2017.

13 M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner Basis Algorithm to find Proofs of Unsatisfiability. In *STOC 1996*, pages 174–183, 1996.

14 S. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *J. Symbolic Logic*, 44:36–50, 1979.

15 A. Dawar. The nature and power of fixed-point logic with counting. *ACM SIGLOG News*, 2(1):8–21, 2015.

16 A. Dawar and E. Grädel. The descriptive complexity of parity games. In *Proceedings of CSL 2008*, pages 354–368, 2008.

17 A. Dawar, M. Grohe, B. Holm, and B. Laubner. Logics with rank operators. In *LICS 2009*, pages 113–122, 2009.

18 A. Dawar and P. Wang. Lasserre lower bounds and definability of semidefinite programming. *CoRR*, abs/1602.05409, 2016.

19 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

20 E. Grädel and S. Hegselmann. Counting in Team Semantics. In *CSL 2016*, 2016.

21 M. Grohe and M. Otto. Pebble games and linear equations. *J. Symb. Log.*, 80(3):797–844, 2015.

22 P. N. Malkin. Sherali-adams relaxations of graph isomorphism polytopes. *Discrete Optimization*, 12:73–97, 2014.

23 M. Otto. *Bounded Variable Logics and Counting*. Springer, 1997.

24 W. Pakusa. *Linear Equation Systems and the Search for a Logical Characterisation of Polynomial Time*. PhD thesis, RWTH Aachen University, 2016.

25 N. Segerlind. The Complexity of Propositional Proofs. *Bulletin of Symbolic Logic*, 13(04):417–481, 2007.

26 J. Torán. On the resolution complexity of graph non-isomorphism. In *Theory and Applications of Satisfiability Testing - SAT 2013*, volume 7962 of *LNCS*, pages 52–66, 2013.

27 O. Verbitsky. On the dynamic width of the 3-colorability problem. *CoRR*, abs/1312.5937, 2013.

A Details omitted from Section 5.1

A *graph with colour class size* $k \geq 1$ is a structure $G = (V, E, \preceq)$ where (V, E) is a graph and where $\preceq$ is a linear preorder on V such that every class of $\preceq$-incomparable vertices, that is every *colour class*, is of size at most k. In other words, one can think of the vertices of the graph G to be coloured, but we only allow that at most k many vertices get the same colour. We write $V = V_0 \preceq \cdots \preceq V_{n-1}$ to denote that V is linearly ordered by $\preceq$ into n colour classes V_i in the indicated way. We have that $|V_i| \leq k$ for every $i < n$.

We consider pairs of graphs $G = (V, E, \preceq_V)$ and $H = (W, F, \preceq_W)$ of colour class size $k \geq 1$ with the same number of colour classes, that is

$$V = V_0 \preceq_V V_1 \preceq_V \cdots \preceq_V V_{n-1}$$
$$W = W_0 \preceq_W W_1 \preceq_W \cdots \preceq_W W_{n-1}.$$

We consider the following propositional formula $\Theta(G, H)$ which encodes the graph isomorphism problem for G and H. The formula uses propositional variables $X[v \mapsto w]$ for $v \in V_i$, $w \in W_i$, $i < n$, indicating that the vertex $v \in V_i$ is mapped to the vertex $w \in W_i$ (we only consider colour-preserving mappings). The formula $\Theta(G, H)$ consists of the following set of clauses:

$$\bigvee_{w \in W_i} X[v \mapsto w] \qquad \text{for each } i < n, v \in V_i \qquad (1)$$

$$\bigvee_{v \in V_i} X[v \mapsto w] \qquad \text{for each } i < n, w \in W_i \qquad (2)$$

$$\neg(X[v_1 \mapsto w_1] \wedge X[v_2 \mapsto w_2]) \qquad \text{if } \{(v_1, w_1), (v_2, w_2)\} \text{ is not a local isomorphism.} \qquad (3)$$

This is basically the encoding that Toran used in [26]. It is important to observe that $\Theta(G, H)$ is a k-CNF formula where the number of propositional variables is linear in the number of vertices of the graph (we think of the colour class size k to be fixed). For this encoding, Toran proved, using the well-known construction of Cai, Fürer, and Immerman, that there is a family of pairs of non-isomorphic graphs (G_n, H_n), $n \geq 1$, of degree three and colour class size four for which the resolution refutations of $\Theta(G_n, H_n)$ are of exponential size (in the number of vertices of the graphs G_n, H_n). This is a special case of the following more general result.

▶ **Theorem 22.** *Let $k \geq 1$, and let $\mathcal{I}$ be an* LFP*-interpretation that maps pairs of n-vertex graphs (G, H) of degree three and colour class size four to a k-CNF formula $\mathcal{I}(G, H)$ such that*

- *the number of propositional variables in $\mathcal{I}(G, H)$ is $\mathcal{O}(n)$;*
- *$\mathcal{I}(G, H)$ is satisfiable if, and only if, G and H are isomorphic.*

Then the maximal size of a resolution refutation of the formula $\mathcal{I}(G, H)$ for pairs of non-isomorphic n-vertex graphs (G, H), as above, is bounded below by a function in $2^{\Omega(n)}$.

Proof. Another application of Theorem 17. We first fix for all $n \geq 1$ pairs of non-isomorphic graphs G_n, H_n with $\mathcal{O}(n)$ many vertices, of colour class size four, of degree three, and such that $G_n \equiv^n H_n$; the existence follows from [11]. It follows that $(G_n, G_n) \equiv^n (G_n, H_n)$. By the embedding of LFP into $L^{\omega}_{\infty\omega}$ we can view the LFP-interpretation $\mathcal{I}$ as an $L^{\ell}_{\infty\omega}$-interpretation for some fixed $\ell \geq 1$. Now we set $\mathfrak{A}_n = (G_n, G_n)$, $\mathfrak{B}_n = (G_n, H_n)$, $\Phi_n = \mathcal{I}(\mathfrak{A}_n)$, and $\Psi_n = \mathcal{I}(\mathfrak{B}_n)$. All preconditions of Theorem 17 are satisfied. In particular, the number of variables of the formulas Ψ_n is linear in n by our assumption. This shows that the size of a resolution refutation of Ψ_n is bounded below by a function in $2^{\Omega(n)}$ as claimed. ◀

B Proof of Theorem 19

▶ **Theorem 19.** *For all $n \geq 1$, there are graphs G_n, H_n with $\mathcal{O}(n)$ many vertices, such that $G_n \equiv^{\Omega(n)} H_n$, and such that G_n is three-colourable and H_n is not three-colourable.*

We remark that this result, and our proof sketch below, remain valid if we consider the stronger equivalence with respect to counting logic. Indeed, for this setting the result already appeared in [27] and, for the more general setting of constraint satisfaction problems with unbounded width, in [18]. For completeness, let us sketch a proof here which shows how to derive this result from [3, 11, 17].

Proof. Again, we start with the Cai-Fürer-Immerman construction [11]. For all $n \geq 1$ we obtain a pair of three-regular, non-isomorphic graphs (G_n, H_n) of colour class size four and with $\mathcal{O}(n)$ many vertices such that $G_n \equiv^n H_n$. We have $(G_n, G_n) \equiv^n (G_n, H_n)$.

Moreover, for pairs of CFI-graphs $(G^*, H^*) \in \{(G_n, H_n), (G_n, G_n) : n \geq 1\}$ it is known that the isomorphism problem reduces, via a first-order interpretation $\mathcal{I}$, to a linear equation system over the field with two elements, see e.g. [17], which can equivalently be formulated as the satisfiability problem of a propositional formula. It can be checked that the resulting propositional formula $\mathcal{I}(G^*, H^*)$ is indeed a 3-CNF formula, since we work with three-regular graphs. Also the number of propositional variables and clauses in $\mathcal{I}(G^*, H^*)$ is linear in the number of vertices of G^* (and H^*), which, in turn, is linear in n. We can now take a standard reduction from 3-SAT to 3-colourability from [19], which is first-order definable. The number of vertices of the resulting graphs is again linear in the number of clauses and variables from the 3-CNF formula. To sum up, there is a first-order interpretation $\mathcal{J}$ which maps pairs of CFI-graphs $(G^*, H^*) \in \{(G_n, H_n), (G_n, G_n)\}$ to graphs $\mathcal{J}(G^*, H^*)$ such that

- the number of vertices of the graphs $\mathcal{J}(G^*, H^*)$ is $\mathcal{O}(n)$;
- $\mathcal{J}(G_n, G_n)$ is three-colourable;
- $\mathcal{J}(G_n, H_n)$ is not three-colourable;
- $\mathcal{J}(G_n, G_n) \equiv^{\Omega(n)} \mathcal{J}(G_n, H_n)$.

This proves our claim. ◀