The Ehrenfeucht-Fraïssé Method and the Planted Clique Conjecture

Yijia Chen[1] and Jörg Flum[2]

[1] Fudan University, Shanghai, China
[2] Albert-Ludwigs-Universität Freiburg, Germany

Abstract. The Ehrenfeucht-Fraïssé method for first-order logic and further logics relevant in descriptive complexity has been quite successful. However, for key problems such as P $\neq$ NP or NP $\neq$ co-NP no progress has been achieved using it. We show that for these problems we can not get the board for the corresponding Ehrenfeucht-Fraïssé game in polynomial output time, even if we allow probabilistic methods to obtain the board. In order to get this result in the probabilistic case, we need an additional hypothesis, namely that there is an algorithm, the verifier, verifying in a reasonable time that the two structures of the board satisfy the same properties expressible in a suitable fragment of the logic. The (non)existence of such a verifier is related to a logic version of the planted clique conjecture.

1. Introduction

In finite model theory and in descriptive complexity theory the Ehrenfeucht-Fraïssé method for first-order logic FO is mainly used to obtain *inexpressibility results* and *hierarchy results*. While Fraïssé [9] introduced this method in more algebraic terms, Ehrenfeucht [6] phrased it in an appealing game-theoretic form. Concerning generalizations, games were developed for further logics, mainly for logics relevant in descriptive complexity theory such as least fixed-point logic LFP, (monadic) existential second-order logic (monadic) Σ^1_1, and finite variable logics.

An inexpressibility result for a logic L shows that a given property is not definable (or expressible) in L. A hierarchy result states that a certain increasing sequence $\mathrm{H}_1 \subseteq \mathrm{H}_2 \subseteq \ldots$ of classes H_m of sentences of a given logic is strict; that is, that for every $m \in \mathbb{N}$ there is a property of finite structures expressible by some sentence of H_{m+1} but by no sentence of H_m. Often, to obtain such an inexpressibility result, Ehrenfeucht-Fraïssé games have been used. The finite variable hierarchy $(\mathrm{FO}^m)_{m\in\mathbb{N}}$ is an example of a strict hierarchy. Here FO^m consists of those FO-formulas which contain at most m variables.

Suppose we want to show, using the Ehrenfeucht-Fraïssé method, that for (finite) ordered graphs "eveness" of the cardinality of the vertex set is not expressible in FO, or equivalently, that for every $m \in \mathbb{N}$ "eveness" is not expressible by an FO_m-sentence. Here FO_m denotes the set of sentences of first-order logic of quantifier rank at most m. One chooses ordered graphs G_m and H_m that are paths of length $2^m + 1$ and 2^m, respectively, and shows that $G_m \equiv_{\mathrm{FO}_m} H_m$, that is, that G_m and H_m satisfy the same sentences of FO_m. The latter property is shown by playing, more precisely, by analyzing the Ehrenfeucht-Fraïssé game (for first-order logic) with board (G_m, H_m). It is not hard to show that the size of the board (G_m, H_m) must be exponential in m.

Let us mention some further results obtained by the Ehrenfeucht-Fraïssé method (or by a probabilistic generalization of it):

– Reachability in directed graphs is not expressible in monadic Σ^1_1 [1].
– For ordered graphs connectivity is not expressible in monadic Σ^1_1 [20].
– The finite variable hierarchy for FO on ordered structures is strict [18, 12].
– The arity hierarchy is strict for LFP [10].
– For every $k \in \mathbb{N}$ the hierarchy whose mth member consists of formulas with at most m nested k-ary fixed-point operators is strict for LFP [15].

We know (see Theorem 1) that P $\neq$ NP if and only if for every m there are a 3-colorable ordered graph G_m and an ordered graph H_m, which is not 3-colorable, such that G_m and H_m are indistinguishable by sentences of LFP of "quantifier rank" or length at most m; this last property, denoted by $G_m \equiv_{\mathrm{LFP}_m} H_m$, would be shown by the Ehrenfeucht-Fraïssé game for LFP. Let us call such a sequence $(G_m, H_m)_{m\in\mathbb{N}}$ a (3-COL, LFP)-sequence. Furthermore, NP $\neq$ co-NP if and only if there is a (3-COL, Σ^1_1)-sequence, where a (3-COL, Σ^1_1)-sequence is defined in a similar way. In [8], the authors remark:

> *It is known that $\Sigma^1_1 \neq \Pi^1_1$ if and only if such a separation can be proven via a second-order Ehrenfeucht-Fraïssé games. Unfortunately, "playing" second-order Ehrenfeucht-Fraïssé games is very difficult, and the above promise is still largely unfulfilled; for example, the equivalence between the* NP = co-NP *question and the Σ^1_1 = Π^1_1 question has not so far led to any progress on either of these questions.*

And Kolaitis remarks in [7, page 56]:

> *Although . . . Ehrenfeucht-Fraïssé games yield a sound and complete method for studying ESO-definability [that is, Σ^1_1-definability] (and thus potentially leading to the separation of* NP *and* co-NP*), so far this approach has had rather limited success. The reason is that formidable combinatorial difficulties arise in implementing this method . . . when dealing with ESO-formulas in which at least one of the existentially quantified second-order variables has an arity bigger than 1.*

Definitely the authors are right with their observation that "playing" second-order Ehrenfeucht-Fraïssé games is very difficult. However, in order to derive the last two hierarchy results mentioned above, the corresponding authors successfully apply games for logics containing nonmonadic second-order quantifiers.

In the example of "eveness" we already observed that the size of a board (G_m, H_m) of ordered graphs has to be exponential in m. On the other hand, analyzing most of the successful applications of the Ehrenfeucht-Fraïssé method obtained so far, we realized that the boards $(G_m, H_m)_{m\in\mathbb{N}}$ could be constructed in polynomial output time, that is, in time $(|V(G_m)|+|V(H_m)|)^{O(1)}$. However, by a simple and standard diagonal argument we show:

(A) *No* (3-COL, LFP)*-sequence can be generated in polynomial output time.*

Even more, to the best of our knowledge, it is open whether we can get such a sequence of boards by an algorithm more efficient than brute force.

Mostly in successful applications of the Ehrenfeucht-Fraïssé method the main task consisted in constructing boards such that one can find an argument showing, via Ehrenfeucht-Fraïssé games for the given logic, that the corresponding structures are indistinguishable to a certain extent. As mentioned, for a proof of P $\neq$ NP via the Ehrenfeucht-Fraïssé method, already the presumably easier step of merely constructing the sequence

of boards (and forgetting about the concrete verification of their indistinguishability) is hard. This makes our "negative" result even stronger with respect to the existence of positive applications of the Ehrenfeucht-Fraïssé method for sufficiently rich logics. It is an interesting challenge, though: how can we use the Ehrenfeucht-Fraïssé method to prove P $\neq$ NP if we must necessarily work with non-constructive boards?

What happens if we allow probabilistic algorithms[3] to yield the boards for the Ehrenfeucht-Fraïssé method? Such random constructions have been used for two of the applications mentioned above, namely to show that reachability in directed graphs is not definable in monadic second-order logic and in the proof of Rossman [18] that the finite variable hierarchy for first-order logic on ordered graphs is strict. It turns out that in order to derive a probabilistic generalization of (A) of the type "No (3-COL, LFP)-sequence can be generated by a probabilistic algorithm in polynomial output time" we need a further assumption,[4] namely that there is a *verifier*, that is, an algorithm that in a reasonable time verifies that with high probability the board (G_m, H_m) satisfies

$$G_m \in \text{3-COL}, H_m \notin \text{3-COL}, \text{ and } G_m \equiv_{\text{LFP}_m} H_m.$$

So we get:

(B) *Assume that there is a pseudorandom generator. No* (3-COL, LFP)*-sequence having a verifier can be generated by a probabilistic algorithm in polynomial output time.*

Is the assumption of the existence of a verifier necessary? The question is related to the *planted clique conjecture*. This conjecture claims that there is no polynomial time algorithm that detects a clique of size $4 \cdot \log n$, which has been planted uniformly at random in a random graph with n vertices and edge probability $1/2$. In this article we introduce a stronger conjecture, a logic version LPCC of the planted clique conjecture. It is not hard to show:

(C) *If* LPCC *holds, then a* (3-COL, LFP)*-sequence can be generated by a probabilistic algorithm in polynomial output time.*

As already the planted clique conjecture implies P $\neq$ NP, so does LPCC. Can we refute LPCC? We show that this is the case for some strengthening of LPCC.

The content of the different sections is the following. After fixing some notation (in Section 2), we recall the Ehrenfeucht-Fraïssé method in Section 3. In Section 4, first we study the minimum size of the board (G_m, H_m) of a (3-COL, LFP)-sequence and then we prove statement (A). Section 5 is devoted to a proof of the probabilistic generalization of this result, stated as (B) above. In Section 6 we introduce the logic version LPCC of the planted clique conjecture and derive statement (C) in Section 7. In Section 8 we show that some strengthened versions of LPCC are refutable. Finally, in the last section we mention extensions of our results and some further results related to the topic of this article. Moreover, we state some conjectures and open questions.

[3] At least here we should mention that there exist successful applications of the Ehrenfeucht-Fraïssé method, where the boards are not defined by a (probabilistic) *algorithm*; for example, in [21] random graphs with edge probability $n^{-\alpha}$ are considered, where n is the cardinality of the vertex set and α is *irrational*.

[4] Besides the assumption of the existence of a pseudorandom generator.

2. Preliminaries

For a natural number n we set $[n] := \{1, \ldots, n\}$. For a graph G we denote by $V(G)$ and $E(G)$ its vertex set and its edge set, respectively. We speak of an ordered graph G if G comes with an ordering of its vertex set. As already mentioned, in this article graph always means finite graph. A *problem* (or, *property*) Q of ordered graphs is a class of ordered graphs closed under isomorphism.

We assume familiarity with basic notions of first-order logic FO and of least fixed-point logic LFP. Concerning LFP, till Section 8 essentially we only need the Immerman-Vardi Theorem, which we recall in the next section.

Let L be a logic. A property Q of ordered graphs is *definable in* L (or, *expressible in* L) if there is a sentence of L such that Q is its class of models.

3. The Ehrenfeucht-Fraïssé-method

Let us denote by FO_m the set of sentences of first-order logic of quantifier rank (= maximum number of nested quantifiers) at most m and by LFP_m the set of LFP-sentences φ of length $|\varphi| \leq m$. Here $|\varphi|$ denotes the number of *symbols* in φ (that is, the number of connectives, quantifiers, LFP-operators, variables, ...; however, two occurrences, say, of the same variable in φ count as two symbols).

Let L be one of the logics FO or LFP and denote by L_m the corresponding set FO_m or LFP_m. The Ehrenfeucht-Fraïssé method relies on the following result.

Theorem 1. *For $L \in \{\mathrm{FO}, \mathrm{LFP}\}$ and a problem Q of ordered graphs the following are equivalent:*

(i) For all $m \in \mathbb{N}$ there are ordered graphs G_m and H_m with

$$G_m \in Q, \quad H_m \notin Q, \qquad \text{and} \qquad G_m \equiv_{L_m} H_m. \tag{1}$$

(ii) Q is not definable in L.

So, in order to show that the problem Q is not definable in the logic $L \in \{\mathrm{FO}, \mathrm{LFP}\}$, it suffices to exhibit a (Q, L)-sequence in the sense of the following definition.

Definition 2. Assume $L \in \{\mathrm{FO}, \mathrm{LFP}\}$ and let Q be a problem of ordered graphs. A sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of ordered graphs is a (Q, L)-*sequence* if for all $m \in \mathbb{N}$

$$G_m \in Q, \;\; H_m \notin Q, \;\; \text{and} \;\; G_m \equiv_{L_m} H_m.$$

In many concrete applications of Theorem 1, Ehrenfeucht-Fraïssé-games are applied to show that $G_m \equiv_{L_m} H_m$. We recall the Ehrenfeucht-Fraïssé-game for FO (see [4, 10, 15] for the Ehrenfeucht-Fraïssé-game for LFP and other extensions of FO by fixed-point operators). Let G and H be ordered graphs and $m \in \mathbb{N}$. The Ehrenfeucht-Fraïssé-game $\mathrm{G}_m(G, H)$ (with *boards* G and H) is played by two players called Spoiler and Duplicator. The game consists of a sequence of m rounds. In round i of the game, first Spoiler picks a graph (either G or H) and a vertex of his choice in that graph. Duplicator then replies by picking a vertex of his choice in the other graph. Thus, after m rounds, vertices $u_1, \ldots, u_m$ in $V(G)$ and $v_1, \ldots, v_m$ in $V(H)$ have been selected, u_i and v_i being the vertices chosen in round i. Duplicator *wins* if the induced ordered subgraphs $G[\{u_1, \ldots, u_m\}]$ and $H[\{v_1, \ldots, v_m\}]$ (induced by G on $\{u_1, \ldots, u_m\}$ and by H on $\{v_1, \ldots, v_m\}$, respectively) are isomorphic via the mapping $f(u_i) := v_i$ for $i \in [m]$. It should be clear what it means that Duplicator has a winning strategy for the game $\mathrm{G}_m(G, H)$.

Theorem 3 (Ehrenfeucht-Fraïssé-Theorem). *Let G and H be ordered graphs and $m \in \mathbb{N}$. Then Duplicator has a winning strategy for the game* $\mathrm{G}_m(G, H)$ *if and only if* $G \equiv_{\mathrm{FO}_m} H$.

The following simple application of the Ehrenfeucht-Fraïssé-game shows that the class EVEN of ordered graphs with vertex set of even cardinality is not definable in FO: For $m \in \mathbb{N}$ let the ordered graphs G_m and H_m be paths of length $2^m + 1$ and 2^m, respectively. Then Duplicator has a winning strategy for the game $\mathrm{G}_m(G_m, H_m)$. In fact, in the ith round he picks his vertex, u_i or v_i, such that for all $j \in [i-1]$,

$$d^{G_m}(u_i, u_j) = d^{H_m}(v_i, v_j) \quad \text{or} \quad \big(d^{G_m}(u_i, u_j) > 2^{m-i} \text{ and } d^{H_m}(v_i, v_j) > 2^{m-i}\big).$$

Here $d^G(u, u')$ denotes the distance of the vertices u and u' in the graph G. Thus, $G_m \equiv_{\mathrm{FO}_m} H_m$ and hence, $(G_m, H_m)_{m\in\mathbb{N}}$ is an (EVEN, FO)-sequence.

The graphs G_m and H_m just constructed have size exponential in m. We can't do it better: the sizes of the graphs of every (Q, FO)-sequence for any problem Q of ordered graphs must be exponential in m. This follows from the following result, which can easily been derived.

Proposition 4. *Let $m \in \mathbb{N}$. If G and H are nonisomorphic ordered graphs, then*

$$G \equiv_{\mathrm{FO}_{m+3}} H \text{ implies } |V(G)|, |V(H)| > 2^m.$$

4. A logical reformulation of P $\neq$ NP

Immerman and Vardi have proven that least fixed-point logic LFP captures the complexity class P in the following sense.

Theorem 5 (Immerman-Vardi-Theorem). *A problem of ordered graphs is decidable in polynomial time if and only if it can be defined in least fixed-point logic* LFP.

As the problem 3-COL, the 3-colorability problem of ordered graphs, is NP-complete, we get:

Corollary 6. P $\neq$ NP *if and only if* 3-COL *is not definable in* LFP.

We defined $\mathrm{LFP}_m = \{\varphi \mid \varphi \text{ LFP}_m\text{-sentence with } |\varphi| \leq m\}$. The previous corollary together with Theorem 1 yield:

Corollary 7. P $\neq$ NP *if and only if there is a* (3-COL, LFP)*-sequence, that is, a sequence* $(G_m, H_m)_{m\in\mathbb{N}}$ *of ordered graphs such that for all* m,

$$G_m \in \text{3-COL}, \quad H_m \notin \text{3-COL}, \quad \textit{and} \quad G_m \equiv_{\mathrm{LFP}_m} H_m.$$

Assume P $\neq$ NP. What can we say about the minimum size of the graphs of a (3-COL, LFP)-sequence and what about the running time of an algorithm generating a (3-COL, LFP)-sequence? We set

$$\text{SIZE(3-COL)}(m) := \min\{\max\{|V(G)|,|V(H)|\} \mid G \text{ and } H \text{ are ordered graphs with } G \in \text{3-COL}, H \notin \text{3-COL}, \text{and } G \equiv_{\mathrm{LFP}_m} H\}.$$

Recall that a problem Q has circuit size c, where $c : \mathbb{N} \to \mathbb{N}$, if for $n \in \mathbb{N}$, $c(n)$ is the least $d \in \mathbb{N}$ such there exists a (Boolean) circuit C with n input variables of size $\leq d$ such that for every x with $|x| = n$,

$$x \in Q \quad \Longleftrightarrow \quad C(x) = 1 \text{ (i.e., } C \text{ accepts } x).$$

In [5] we derived the following lower and upper bound for SIZE(3-COL)(m).

Proposition 8. *Assume* P $\neq$ NP. *Then:*
(a) There is an $\varepsilon > 0$ *such that for all* $m \in \mathbb{N}$ *we have* $2^{\varepsilon \cdot m} \leq \text{SIZE}(3\text{-COL})(m)$.
(b) If the circuit size of 3-COL *is not in* $2^{o(n)}$*, then for all* $\varepsilon > 0$ *and infinitely many* m,

$$\text{SIZE}(3\text{-COL})(m) \leq 2^{(1+\varepsilon)\cdot m \cdot \log m}.$$

Definition 9. An algorithm $\mathbb{A}$ *generates* the sequence $(G_m, H_m)_{m\in\mathbb{N}}$ if $\mathbb{A}$ on input $m \in \mathbb{N}$ outputs (G_m, H_m).

By systematically testing, for $\ell = 1, 2, \ldots$, all graphs G and H with vertex sets of cardinality $\leq \ell$ whether they satisfy

$$G \in 3\text{-COL}, \quad H \notin 3\text{-COL}, \quad \text{and} \quad G \equiv_{\text{LFP}_m} H,$$

we obtain from the previous result an upper bound for the time needed to get the graphs of a (3-COL, LFP)-sequence, even of a sequence with boards of minimum size:

Proposition 10 ([5]). *If* P $\neq$ NP*, then there is an algorithm that generates a* (3-COL, LFP)*-sequence in time* $2^{O(\text{SIZE}(3\text{-COL})(m)^2)}$*. The sequence* $(G_m, H_m)_{m\in\mathbb{N}}$ *generated by the algorithm satisfies* $\text{SIZE}(3\text{-COL})(m) = \max\{|V(G_m)|, |V(H_m)|\}$.

By Proposition 4, the boards of all (Q, FO)-sequences for any problem Q of ordered graphs must have size exponential in m. However we could construct the graphs G_m and H_m of an (EVEN, FO)-sequence in *polynomial output time*, that is, in time $(|V(G_m)| + |V(H_m)|)^{O(1)}$. In fact, we realized that in most successful applications of the Ehrenfeucht-Fraïssé method showing that a property is not definable in a given logic, the boards for the corresponding game can be constructed in polynomial output time. So we ask, is it possible to construct a (3-COL, LFP)-sequence in polynomial output time? By a standard diagonalization argument we show that this is not possible:

Theorem 11. *No* (3-COL, LFP)*-sequence can be constructed in polynomial output time.*

Proof. We sketch the main steps of a proof (for more details see [5]). Assume for a contradiction that the algorithm $\mathbb{A}$ generates a (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ in polynomial output time. By passing to a suitable subsequence (cf. the proof of Lemma 16), we can assume that $(G_m, H_m)_{m\in\mathbb{N}}$ is *monotone*, that is, that it satisfies

$$\max\{|V(G_m)|, |V(H_m)|\} < \min\{|V(G_{m+1})|, |V(H_{m+1})|\}.$$

Furthermore, we can assume (again by passing to a suitable subsequence) that $|V(G_m)| \geq |V(H_m)|$ for all $m \in \mathbb{N}$ or that $|V(G_m)| \leq |V(H_m)|$ for all $m \in \mathbb{N}$. Then we can transform $\mathbb{A}$ into an algorithm $\mathbb{B}$ running in polynomial time such that for all $m \in \mathbb{N}$,

$$\mathbb{B} \text{ accepts } G_m \quad \text{and} \quad \mathbb{B} \text{ rejects } H_m.$$

By the Immerman-Vardi Theorem there is an LFP-sentence $\varphi_{\mathbb{B}}$, say $\varphi_{\mathbb{B}} \in \text{LFP}_{m_0}$, such that for all ordered graphs G,

$$G \models \varphi_{\mathbb{B}} \iff \mathbb{B} \text{ accepts } G.$$

In particular, for all $m \in \mathbb{N}$,

$$G_m \models \varphi_{\mathbb{B}} \quad \text{and} \quad H_m \not\models \varphi_{\mathbb{B}}.$$

For $m \geq m_0$, this equivalence contradicts $G_m \equiv_{\text{LFP}_m} H_m$. □

The same proof works for every property Q of ordered graphs (instead of 3-COL), even more: By definition, an LFP-*sequence* is a sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of ordered graphs G_m and H_m with

$$G_m \not\cong H_m \ \text{ (G_m and H_m are not isomorphic) and } \ G_m \equiv_{\text{LFP}_m} H_m.$$

Clearly every (Q, LFP)-sequence for any property Q of ordered graphs is an LFP-sequence. We state the following result, which can be derived similarly to Theorem 11.

Theorem 12 ([5]). *No* LFP*-sequence can be generated in polynomial output time.*

We should mention that also for first-order logic there are problems Q such that no (Q, FO)-sequence can be generated in polynomial output time:

Example 13. Let $B \subseteq \{0,1\}^*$ be a P-bi-immune set; that is, neither B nor $\{0,1\}^* \setminus B$ contains an infinite subset decidable in polynomial time. For $x \in B$, $x = x_1 \dots x_s$ with $x_i \in \{0,1\}$, let $G(x)$ be the ordered graph with vertex set $[s+1]$, with the natural ordering on $[s+1]$, and with edge set $\{\{i, i+1\} \mid i \in [s] \text{ and } x_i = 1\}$. Let $Q(B)$ be the smallest class of ordered graphs containing all $G(x)$ with $x \in B$ and closed under isomorphism. No $(Q(B), \text{FO})$-sequence can be generated in polynomial output time. For a contradiction assume that $(G_m, H_m)_{m\in\mathbb{N}}$ is a $(Q(B), \text{FO})$-sequence generated in polynomial output time. As above we can assume that the sequence is monotone and that $|V(G_m)| \geq |V(H_m)|$ for all $m \in \mathbb{N}$ or that $|V(G_m)| \leq |V(H_m)|$ for all $m \in \mathbb{N}$. In the first case, B contains an infinite subset in P and in the second case $\{0,1\}^* \setminus B$.

5. On random (3-COL, LFP)-sequences

We have seen that we cannot construct a (3-COL, LFP)-sequence in polynomial output time. What happens if we consider random sequences? There are successful applications of the Ehrenfeucht-Fraïssé-method where the graphs of the corresponding sequences are constructed randomly. For example, in this way it has been shown that reachability in directed graphs is not definable in monadic second-order logic (see [1]) and that the finite variable hierarchy for first-order logic on ordered graphs is strict (see [18]).

We aim at a result showing limitations of the probabilistic Ehrenfeucht-Fraïssé-method similar to Theorem 11. For this purpose we have to take into consideration a further property of such sequences $(G_m, H_m)_{m\in\mathbb{N}}$ satisfied in most successful applications of the Ehrenfeucht-Fraïssé-method obtained so far. For (3-COL, LFP)-sequences $(G_m, H_m)_{m\in\mathbb{N}}$ this property ensures that we can verify that $G_m \in$ 3-COL, $H_m \notin$ 3-COL, and that $G_m \equiv_{\text{LFP}_m} H_m$ in a reasonable time. Condition (r2) of the following definition of random (3-COL, LFP)-sequence contains the precise formulation.

Definition 14. A probabilistic algorithm $\mathbb{P}$ *generates a random* (3-COL, LFP)*-sequence* $(G_m, H_m)_{m\in\mathbb{N}}$ if (r1) and (r2) are satisfied.

(r1) For every $m \in \mathbb{N}$ the algorithm $\mathbb{P}$, on input m, first *deterministically* computes the vertex sets $V(G_m)$ and $V(H_m)$, and then it constructs the ordered graphs G_m and H_m probabilistically.

(r2) There is an algorithm $\mathbb{V}$, the *verifier*, such that (a)–(c) hold.

(a) For all ordered graphs G and H and all $m \in \mathbb{N}$,

$$\text{if } \mathbb{V} \text{ accepts } (G, H, m), \text{ then } G \equiv_{\text{LFP}_m} H, \quad G \in \text{3-COL, and } H \notin \text{3-COL}.$$

(b) For sufficiently large $m \in \mathbb{N}$ and all $m' \geq m$,

$$\Pr\left[\mathbb{V} \text{ accepts } (G_{m'}, H_{m'}, m)\right] \geq \frac{1}{\left(|V(G_{m'})| + |V(H_{m'})|\right)^{O(1)}}.$$

(c) The running time of $\mathbb{V}$ on input (G, H, m) is bounded by $f(m) \cdot (|V(G)| + |V(H)|)^{O(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

In this section we show:

Theorem 15. *Assume that there is a $2^{\lceil \ell/c \rceil}$-pseudorandom generator*[5] *for some natural number $c \geq 1$. Then there is no probabilistic algorithm that generates a random* (3-COL, LFP)*-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ in polynomial output time.*

The following lemmas will finally yield a proof of Theorem 15 along the following lines: For a contradiction we assume that there exists a probabilistic algorithm $\mathbb{P}$ generating a random (3-COL, LFP)-sequence in polynomial output time. Essentially we use the pseudorandom generator to derandomize the algorithm $\mathbb{P}$. In this way we obtain a deterministic algorithm which generates a (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ in polynomial output time. This contradicts Theorem 11.

As in the deterministic case we say that a probabilistic algorithm $\mathbb{P}$ generates a random *monotone* (3-COL, LFP)-sequence if it generates a random (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$, which in addition to (r1) and (r2) satisfies (r3), where

(r3) for all $m \in \mathbb{N}$, $\max\{|V(G_m)|, |V(H_m)|\} < \min\{|V(G_{m+1})|, |V(H_{m+1})|\}$.

If furthermore (r4) and (r5) hold, where

(r4) $\lceil \log\left(|V(G_m)| + |V(H_m)|\right)\rceil < \lceil \log\left(|V(G_{m+1})| + |V(H_{m+1})|\right)\rceil$

(r5) $f(m) \leq \max\{|V(G_m)|, |V(H_m)|\}$ (where f is the computable function of (r2)(c) used to bound the running time of the verifier $\mathbb{V}$),

then we speak of a *strongly monotone* (3-COL, LFP)*-sequence*.

For our proof of Theorem 15 we need to show that we can restrict ourselves to strongly monotone (3-COL, LFP)-sequences.

Lemma 16. *If there is a probabilistic algorithm generating a random* (3-COL, LFP)*-sequence in polynomial output time, then there is a probabilistic algorithm that generates a strongly monotone random* (3-COL, LFP)*-sequence in polynomial output time.*

Proof. Similar to Proposition 4 one gets an increasing function $s : \mathbb{N} \to \mathbb{N}$ such that $s(m)$ is computable in space $O(\log\ m)$ and such that for all ordered graphs G and H and all $m \in \mathbb{N}$,

$$\text{if } G \equiv_{\text{LFP}_{s(m)}} H \text{ and } G \not\cong H, \text{ then } |V(G)|, |V(H)| > m.$$

Assume that the (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ is generated by the probabilistic algorithm $\mathbb{P}$ in polynomial output time. Recall that the universes of G_m and H_m are obtained deterministically. We define a function $h : \mathbb{N} \to \mathbb{N}$ inductively by

$$h(m) := \begin{cases} s(0), & \text{if } m = 0, \\ s\big(\max\{|V(G_{h(m-1)})|, |V(H_{h(m-1)})|\}\big), & \text{if } m > 0. \end{cases}$$

[5] We recall the notion of a pseudorandom generator in Definition 17.

As $G_{h(m)} \equiv_{\mathrm{LFP}_{h(m)}} H_{h(m)}$, that is, $G_{h(m)} \equiv_{\mathrm{LFP}_{s\left(\max\{|V(G_{h(m-1)})|,|V(H_{h(m-1)})|\}\right)}} H_{h(m)}$, we have

$$|V(G_{h(m)})|, |V(H_{h(m)})| > \max\{|V(G_{h(m-1)})|, |V(H_{h(m-1)})|\}.$$

As $G_{h(m)} \equiv_{\mathrm{LFP}_{h(m)}} H_{h(m)}$, we have $G_{h(m)} \equiv_{\mathrm{LFP}_m} H_{h(m)}$. Therefore, it is routine to show that the probabilistic algorithm, which on input m first computes $h(m)$ and then simulates $\mathbb{P}$ on $h(m)$, generates a random monotone (3-COL, LFP)-sequence in polynomial output time.

So we may assume that the (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ generated by $\mathbb{P}$ is monotone. We will get the sequence satisfying (r4) and (r5) as a subsequence of $(G_m, H_m)_{m\in\mathbb{N}}$, therefore it will be itself monotone. We may assume that the function $f : \mathbb{N} \to \mathbb{N}$ mentioned in (r2) is time constructible. We define $g : \mathbb{N} \to \mathbb{N}$ by

$$g(k) := \begin{cases} \text{the least } m \text{ such that } f(0) \le \max\{|V(G_m)|, |V(H_m)|\}, & \text{if } k = 0, \\ \text{the least } m \text{ such that } f(k) \le \max\{|V(G_m)|, |V(H_m)|\} \text{ and} \\ \lceil \log\left(|V(G_{g(k-1)})| + |V(H_{g(k-1)})|\right)\rceil < \lceil \log\left(|V(G_m)| + |V(H_m)|\right)\rceil, & \text{if } k > 0. \end{cases}$$

Again it is routine to show that the probabilistic algorithm, which on input m first computes $g(m)$ and then simulates $\mathbb{P}$ on $g(m)$, generates a random and strongly monotone (3-COL, LFP)-sequence in polynomial output time. □

Before turning to the main step of the proof of Theorem 15, for the reader's convenience we recall the definition of pseudorandom generator (following [3, Definition 20.2]).

Definition 17. Let $c \in \mathbb{N}$. An algorithm $\mathbb{G}$ is a $2^{\lceil \ell/c\rceil}$-pseudorandom generator if it satisfies (g1) and (g2).

(g1) On every input $s \in \{0,1\}^*$ the algorithm $\mathbb{G}$ computes a string $\mathbb{G}(s) \in \{0,1\}^*$ with $|\mathbb{G}(s)| = 2^{\lceil |s|/c\rceil}$ in time $2^{|s|}$.

(g2) For every $\ell \in \mathbb{N}$ and every circuit C of size at most t^3, where $t := 2^{\lceil \ell/c\rceil}$, we have

$$\left| \Pr_{s\in\{0,1\}^\ell}\left[C(\mathbb{G}(s)) = 1\right] - \Pr_{r\in\{0,1\}^t}\left[C(r) = 1\right]\right| < 1/10.$$

In the left term we consider the uniform probability space on $\{0,1\}^\ell$, in the right term the uniform probability space on $\{0,1\}^t$.

Lemma 18. *Assume*

- *there is a $2^{\lceil \ell/c\rceil}$-pseudorandom generator $\mathbb{G}$ for some $c \in \mathbb{N}$;*
- *there is a probabilistic algorithm $\mathbb{P}$ that generates a strongly monotone random (3-COL, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ in polynomial output time.*

Then there is a deterministic *algorithm $\mathbb{A}$ such that for every $m \in \mathbb{N}$ the algorithm $\mathbb{A}$ on input m computes a sequence of pairs*

$$(G_m^1, H_m^1), \ldots, (G_m^{t_m}, H_m^{t_m})$$

of ordered graphs, where all G_m^i have $V(G_m)$ as vertex set, and all H_m^i have $V(H_m)$ as vertex set (recall that $V(G_m)$ and $V(H_m)$ are the vertex sets deterministically computed by $\mathbb{P}$ on input m). Moreover, the following conditions (a1)–(a3) hold:

(a1) *The algorithm $\mathbb{A}$ runs in time $(|V(G_m)|+|V(H_m|)^{O(1)}$; in particular, $t_m = (|V(G_m)|+|V(H_m|)^{O(1)}$.*

(a2) *For sufficiently large $m \in \mathbb{N}$,*

$$\Pr_{p\in[t_m]}\big[G^p_m \equiv_{\mathrm{LFP}_m} H^p_m,\ G^p_m \in \text{3-COL and } H^p_m \notin \text{3-COL}\big] \geq \Pr_{p\in[t_m]}\big[\mathbb{V} \text{ accepts } (G^p_m, H^p_m, m)\big] > 1/2,$$

where $\mathbb{V}$, the verifier, is the algorithm associated with $\mathbb{P}$ and mentioned in condition (r2) of Definition 14. Note that the first inequality holds by this condition.

(a3) *For every $m \in \mathbb{N}$ we have*

- $\max\{|V(G_m)|, |V(H_m)|\} < \min\{|V(G_{m+1})|, |V(H_{m+1})|\}$
- $\lceil \log (|V(G_m)| + |V(H_m)|\rceil < \lceil \log (|V(G_{m+1}| + |V(H_{m+1})|)\rceil$*;*
- $f(m) \leq \max\{|V(G_m)|, |V(H_m)|\}$ *(where f is the function mentioned in (r2)(c)).*

Proof. For the probabilistic algorithm $\mathbb{P}$ we choose the verifier $\mathbb{V}$ according to (r2). By (r5) we know that $\mathbb{V}$ on input (G_m, H_m, m) runs in time polynomial in $(|V(G_m)| + |V(H_m)|)$. We can assume that $\mathbb{P}$ satisfies (r2)(b′) instead of (r2)(b), where

(r2)(b′) for sufficiently large $m \in \mathbb{N}$, $\Pr\big[\mathbb{V} \text{ accepts } (G_m, H_m, m)\big] \geq 4/5$.

This is achieved by the standard amplification method. More precisely, by repeating the algorithm $\mathbb{P}$, on input m, polynomial many times, that is, polynomial in $(|V(G_m)| + |V(H_m)|)$ many times, and each time checking whether $\mathbb{V}$ accepts (G_m, H_m, m), where (G_m, H_m) is the output of $\mathbb{P}$.

By the properties of $\mathbb{P}$, we know that for some $d \in \mathbb{N}$ with $d \geq 10$:

- The running time of $\mathbb{P}$ on m is bounded by $(|V(G_m)| + |V(H_m)|)^d$.
- The running time of the algorithms $\mathbb{V}$ on inputs (G, H, m) with $f(m) \leq \max\{|V(G)|, |V(H)|\}$ is bounded by $(|V(G)| + |V(H)|)^d$.

We let $\mathbb{A}$ be the following deterministic algorithm:

```
𝔸 //      m ∈ ℕ in unary
1. simulate the (deterministic) part of the computation of ℙ
2.                  on input m yielding the universes V(G_m) and V(H_m)
3. n ← |V(G_m)| + |V(H_m)|
4. ℓ ← c · ⌈d · log n⌉
5. for all s ∈ {0,1}^ℓ do
6.       compute 𝔾(s)
7.       simulate ℙ on input m where in the simulation
8.             the internal coin tosses of ℙ are replaced according to 𝔾(s)
9.       output (G^s_m, H^s_m), the output of this simulation of ℙ.
```

Then (a1) holds as $2^\ell = (|V(G_m)|+|V(H_m)|)^{O(1)}$. Since $\mathbb{P}$ generates strongly monotone sequences, also (a3) holds. It remains to establish (a2). For a contradiction assume that

$$\text{for infinitely many } m \in \mathbb{N}\text{:}\quad \Pr_{p\in[t_m]}\big[\mathbb{V} \text{ accepts } (G^p_m, H^p_m, m)\big] \leq 1/2. \qquad (2)$$

For every $m \in \mathbb{N}$ we let

$$n_m := |V(G_m)| + |V(H_m)|.$$

Clearly there is an algorithm that decides in time $O(n^{d+1})$ whether a given $n \in \mathbb{N}$ is equal to n_m for some $m \in \mathbb{N}$, and if so, outputs m $\big($which is unique by (a3)$\big)$. We consider the following algorithm $\mathbb{D}$:

$\mathbb{D}$ // $r \in \{0,1\}^*$

1. compute an m with $|r| = 2^{\lceil d \cdot \log n_m \rceil}$
2. **if** no such m exists **then** reject
3. compute the output (G_m, H_m) of $\mathbb{P}$ on input m if
4. the internal coin tosses of $\mathbb{P}$ are replaced according to r
5. simulate $\mathbb{V}$ on (G_m, H_m, m)
6. **if** the simulation rejects **then** reject
7. accept.

By (r2)(b′), for sufficiently large $m \in \mathbb{N}$, and hence sufficiently large $n^* := 2^{\lceil d \cdot \log n_m \rceil}$,

$$\Pr_{r \in \{0,1\}^{n^*}} \big[\mathbb{D} \text{ accepts } r\big] = \Pr_{p \in [t_m]} \big[\mathbb{V} \text{ accepts } (G_m^p, H_m^p, m)\big] \geq 4/5. \tag{3}$$

Furthermore note that by (2),

$$\text{for infinitely many } m \text{ and } \ell := c \cdot \lceil d \cdot \log n_m \rceil\text{:} \quad \Pr_{s \in \{0,1\}^\ell} \big[\mathbb{D}(\mathbb{G}(s)) = 1\big] \leq 1/2. \tag{4}$$

Moreover, as $f(m) \leq \max\{|V(G_m)|, |V(H_m)|\}$ (by the strong monotonicity of the random (3-COL, LFP)-sequence computed by $\mathbb{P}$), we see that the running time of $\mathbb{D}$ is bounded by $O(|r|^{1+1/d}) \leq O(|r|^{1.1})$. Using the Cook-Levin's reduction, from the algorithm $\mathbb{D}$ we can construct, for every $m \in \mathbb{N}$ and $n^* := 2^{\lceil d \cdot \log n_m \rceil}$, a circuit C_{n^*} such that for every $r \in \{0,1\}^{n^*}$,

$$C_{n^*}(r) = 1 \iff \mathbb{D} \text{ accepts } r \tag{5}$$

and such that for the size $|C_{n^*}|$ of the circuit C_{n^*} we have

$$|C_{n^*}| = O\big((n^*)^{2.2}\big). \tag{6}$$

By (3) and (5), for sufficiently large $m \in \mathbb{N}$, and hence sufficiently large $n^* = 2^{\lceil d \cdot \log n_m \rceil}$,

$$\Pr_{r \in \{0,1\}^{n^*}} \big[C_{n^*}(r) = 1\big] = \Pr_{p \in [t_m]} \big[\mathbb{V} \text{ accepts } (G_m^p, H_m^p, m)\big] \geq 4/5.$$

By (4) and (5), we know that for infinitely many $m \in \mathbb{N}$ and $\ell := c \cdot \lceil d \cdot \log n_m \rceil$ we have for $n^* = 2^{\lceil d \cdot \log n_m \rceil}$,

$$\Pr_{s \in \{0,1\}^\ell} \big[C_{n^*}(\mathbb{G}(s)) = 1\big] \leq 1/2.$$

Together with the previous inequality, for such an m and the corresponding ℓ and n^*,

$$\Big| \Pr_{r \in \{0,1\}^{n^*}} \big[C_{n^*}(r) = 1\big] - \Pr_{s \in \{0,1\}^\ell} \big[C_{n^*}(\mathbb{G}(s)) = 1\big] \Big| \geq 4/5 - 1/2 > 1/10,$$

which, by (6), contradicts (g2) in Definition 17. □

Proof of Theorem 15: Assume that there is a probabilistic algorithm that generates a random ordered (3-COL, LFP)-sequence in polynomial output time. We show that there is a deterministic algorithm which generates a (3-COL, LFP)-sequence in polynomial output time. This contradicts Theorem 11.

By Lemma 16 and Lemma 18 there is an algorithm $\mathbb{A}$ with the properties stated in Lemma 18. We show that the following algorithm $\mathbb{S}$ generates a (3-COL, LFP)-sequence $(G'_m, H'_m)_{m\in\mathbb{N}}$ in polynomial output time.

$\mathbb{S}$ // $m \in \mathbb{N}$

1. simulate $\mathbb{A}$ on input m to compute $(G_m^1, H_m^1), \ldots, (G_m^{t_m}, H_m^{t_m})$
2. **for all** $i \in [t_m]$ **do**
3. simulate $\mathbb{V}$ on (G_m^i, H_m^i, m)
4. **if** the simulation accepts **then** output (G_m^i, H_m^i) as (G'_m, H'_m) and halt

By (a2) of Lemma 18, the algorithm $\mathbb{S}$ will halt on input m and yield the desired (G'_m, H'_m). By (a3) of Lemma 18, the algorithm $\mathbb{V}$ is applied to inputs (G, H, m) with $f(m) \leq \max\{|V(G)|, |V(H)|\}$; on such inputs its running time is bounded by $(|V(G)| + |V(H)|)^{O(1)}$. Together with (a1), this shows that $\mathbb{S}$ runs in polynomial output time. □

In contrast to deterministic algorithms generating "standard" (3-COL, LFP)-sequences we require of randomized (3-COL, LFP)-sequences $(G_m, H_m)_{m\in\mathbb{N}}$ that the property

$$G_m \equiv_{\text{LFP}_m} H_m, \quad G_m \in \text{3-COL}, \text{ and } H_m \notin \text{3-COL}$$

can be checked in a reasonable time (the existence of the verifier, see property (r2) in Definition 14). What happens if we drop this requirement? The following sections address this problem.

6. The Planted Clique Conjecture

In the standard planted clique problem, we are given a graph G whose edges are generated by starting with a random graph with universe $[n]$, then "planting" (adding edges to make) a random clique on k vertices; the problem asks for efficient algorithms finding such a clique of size k. The problem was addressed in [13, 16, 2], the authors of the last paper mention that it was suggested by M. Saks. It has applications in cryptography [14], algorithmic game theory [11, 17], and classical complexity [19]. Here we study some consequences for the Ehrenfeucht-Fraïssé method of a "logic reformulation" of the planted clique problem.

The Erdős-Rényi probability space ER$(n, 1/2)$ is obtained as follows. We start with the set $[n]$ of vertices. Then we choose every $e \in \binom{[n]}{2}$ $\left(:= \{X \subseteq [n] \mid |X| = 2\}\right)$ as an edge with probability $1/2$, independently of the choices of other edges.

For $G \in$ ER$(n, 1/2)$ the expected size of a maximum clique is approximately $2 \cdot \log\, n$. Clearly, the probability that $G \in$ ER$(n, 1/2)$ contains a clique of size k is bounded by

$$\binom{n}{k} \cdot 2^{-\binom{k}{2}}.$$

For $k = 4 \cdot \log n$ we have

$$\binom{n}{k} \cdot 2^{-\binom{k}{2}} \leq n^{4 \cdot \log n} \cdot 2^{-\binom{k}{2}} = 2^{4 \cdot \log^2 n} \cdot 2^{2 \cdot \log n - 8 \cdot \log^2 n} \leq 2^{-2 \cdot \log^2 n} = n^{-2 \cdot \log n}.$$

Thus

Proposition 19. $\Pr_{G \in \mathrm{ER}(n,1/2)} \big[G \text{ contains a clique of size } 4 \cdot \log n\big] = \frac{1}{n^{\Omega(\log n)}}$.

For any graph G with vertex set $[n]$ and $A \subseteq [n]$ we denote by $G + K(A)$ the graph obtained from G by adding edges such that the subgraph induced on A is a clique. For $n \in \mathbb{N}$ and $k \in [n]$ we consider a second distribution $\mathrm{ER}(n, 1/2, k)$: pick a random (ordered) graph $G \in \mathrm{ER}(n, 1/2)$ and a uniformly random subset A of $[n]$ of size k and plant in a clique on A in G, thus getting $G + K(A)$. [6] We view G and $G + K(A)$ as *ordered* graphs equipped with the natural ordering on $[n]$.

The following decision version PCC(δ) of the planted clique conjecture states that no polynomial time algorithm distinguishes between the distributions $\mathrm{ER}(n, 1/2)$ and $\mathrm{ER}(n, 1/2,\ 4 \cdot \log n)$ more than $\delta(n)$.

Conjecture 20 (The planted clique conjecture PCC(δ)). Let $\delta : \mathbb{N} \to \mathbb{R}$ with $0 < \delta(n) < 1$ for all $n \in \mathbb{N}$. For every polynomial time algorithm $\mathbb{A}$ there is an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,

$$\left| \Pr_{G \in \mathrm{ER}(n,1/2)} \big[\mathbb{A} \text{ accepts } G\big] - \Pr_{G+K(A) \in \mathrm{ER}(n,1/2,\ 4 \cdot \log n)} \big[\mathbb{A} \text{ accepts } G + K(A)\big] \right| \leq \delta(n).$$

Clearly, if $\delta(n) \leq \delta'(n)$ for all $n \in \mathbb{N}$, then PCC(δ) implies PCC(δ'). In [14] the assumption PCC($1 - 1/q$) for some $q \in \mathbb{N}[X]$, that is, for some polynomial q with natural numbers as coefficients, has been put to good use.

Proposition 21. *For $q \in \mathbb{N}[X]$, the statement* PCC($1 - 1/q$) *implies* P $\neq$ NP.

Proof. By Proposition 19 we know that for sufficiently large n,

$$\Pr_{G \in \mathrm{ER}(n,1/2)} \big[G \text{ contains a clique of size } 4 \cdot \log n\big] < 1/q(n). \tag{7}$$

If P = NP, then there is a (deterministic) polynomial time algorithm $\mathbb{A}$ deciding whether a graph contains a clique of size $4 \cdot \log n$. For such an $\mathbb{A}$ we have by (7),

$$\Pr_{G+K(A) \in \mathrm{ER}(n,1/2,\ 4 \cdot \log n)} \big[\mathbb{A} \text{ accepts } G + K(A)\big] - \Pr_{G \in \mathrm{ER}(n,1/2)} \big[\mathbb{A} \text{ accepts } G\big] > 1 - \frac{1}{q(n)}.$$

This contradicts to PCC($1 - 1/q$). □

By the Immerman-Vardi Theorem, on ordered graphs polynomial time algorithms correspond to LFP-sentences. Therefore, PCC(δ) just says that for every LFP-sentence φ and all sufficiently large n,

$$\left| \Pr_{G \in \mathrm{ER}(n,1/2)} \big[G \models \varphi\big] - \Pr_{G+K(A) \in \mathrm{ER}(n,1/2,\ 4 \cdot \log n)} \big[G + K(A) \models \varphi\big] \right| \leq \delta(n).$$

[6] In the following the notation $G + K(A) \in \mathrm{ER}(n, 1/2, k)$ should give the information that the random graph was G and that the random subset of $[n]$ of size k was A.

This holds if

$$\Pr_{G+K(A)\in \mathrm{ER}(n,1/2,\ 4\cdot\log\ n)} \big[G \models \varphi \iff G + K(A) \models \varphi\big] \geq 1 - \delta(n). \tag{8}$$

For our intended application to the Ehrenfeucht-Fraïssé-method we need an even stronger assumption, namely that for every $m \in \mathbb{N}$ and all sufficiently large n,

$$\Pr_{G+K(A)\in \mathrm{ER}(n,1/2,\ 4\cdot\log\ n)} \big[\text{for all } \varphi \in \mathrm{LFP}_m\colon\ \big(G \models \varphi \iff G+K(A) \models \varphi\big)\big] \geq 1-\delta(n),$$

or more succinctly,

$$\Pr_{G+K(A)\in \mathrm{ER}(n,1/2,\ 4\cdot\log\ n)} \big[G \equiv_{\mathrm{LFP}_m} G + K(A)\big)\big] \geq 1 - \delta(n).$$

We shall need an effective bound for the rate of convergence. So we introduce the following logic version LPCC(ε) of the planted clique conjecture.

Conjecture 22 (LPCC(ε)). Let $\varepsilon : \mathbb{N} \to \mathbb{R}$ with $0 < \varepsilon(n) < 1$ for all $n \in \mathbb{N}$. There is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for every $m \in \mathbb{N}$ and all $n \geq f(m)$,

$$\Pr_{G+K(A)\in \mathrm{ER}(n,1/2,\ 4\cdot\log\ n)} \big[G \equiv_{\mathrm{LFP}_m} G + K(A)\big] \geq \varepsilon(n).$$

The previous remarks show:

Proposition 23. *Let $\varepsilon : \mathbb{N} \to \mathbb{R}$ with $0 < \varepsilon(n) < 1$ for all $n \in \mathbb{N}$. Then* LPCC(ε) *implies* PCC($1 - \varepsilon$).

By this proposition and Proposition 21, we get

Corollary 24. *For $q \in \mathbb{N}[X]$,* LPCC($1/q$) *implies* P $\neq$ NP.

Assume that LPCC(ε) holds. By taking a natural number m such that LFP_m contains a sentence expressing that the number of edges is even, we see that $\lim_{n\in\mathbb{N}} \varepsilon(n) \leq 1/2$. In Proposition 26 we generalize this and show that $\lim_{n\to\infty} \varepsilon(n)$ must be 0.

7. The planted clique conjecture and (3-COL, LFP)-sequences

The following result shows that, assuming LPCC($1/q$), there is a probabilistic algorithm yielding a random sequence $(G_m, H_m)_{m\in\mathbb{N}}$ such that

$$G_m \equiv_{\mathrm{LFP}_m} H_m,\ \ G_m \in \text{3-COL},\ \text{ and }\ H_m \notin \text{3-COL} \tag{9}$$

holds with high probability. By Theorem 15 we cannot have a verifier for this algorithm, that is an efficient algorithm that verifies the properties stated in (9) (assuming the existence of a pseudorandom generator).

Theorem 25. *Assume that* LPCC($1/q$) *holds for some polynomial $q \in \mathbb{N}[X]$. Then there is a probabilistic algorithm $\mathbb{P}$ which on input $m \in \mathbb{N}$ generates a pair (G_m, H_m) of ordered graphs in time $(|V(G_m)| + |V(H_m)|)^{O(1)}$ such that*

$$\Pr\big[G_m \equiv_{\mathrm{LFP}_m} H_m,\ G_m \in \text{3-COL},\ \textit{and}\ H_m \notin \text{3-COL}\big] \geq \frac{1}{\big(|V(G_m)| + |V(H_m)|\big)^{O(1)}}.$$

Moreover, $\mathbb{P}$ on input $m \in \mathbb{N}$ first deterministically computes the vertex sets of the graphs G_m and H_m.

Proof. Consider the problem

CLIQUE($4 \cdot \log$)
Instance: An $n \in \mathbb{N}$ and an ordered graph G with $|V(G)| = n$.
Problem: Does G have a clique of size $4 \cdot \log n$?

The proof relies on the following two facts (we leave the details to the reader):

– "LPCC($1/q$) for some $q \in \mathbb{N}[X]$" essentially states that there is a probabilistic algorithm $\mathbb{P}$ which generates a $\big(\text{CLIQUE}(4 \cdot \log\ ), \text{LFP}\big)$-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of ordered graphs in polynomial output time such that

$$\Pr\big[G_m \equiv_{\text{LFP}_m} H_m,\ \ G_m \in \text{CLIQUE}(4 \cdot \log\ ),\ \text{and}\ \ H_m \notin \text{CLIQUE}(4 \cdot \log\ )\big] \geq \frac{1}{\big(|V(G_m)| + |V(H_m)|\big)^{O(1)}}.$$

– As CLIQUE($4 \cdot \log$) is in NP and 3-COL is NP-complete and has a padding function, we can transform the $\big(\text{CLIQUE}(4 \cdot \log\ ), \text{LFP}\big)$-sequence into a (3-COL, LFP)-sequence. □

8. Some remarks on the logic version of the planted clique conjecture

In this section we show (see Lemma 27) that with positive asymptotic probability we can distinguish the LFP_m-theory of the graphs G and $G + K(A)$ by modulo counting their edges (see Lemma 27 for the precise statement). Using this fact, we refute LPCC(ε) unless $\lim_{n\in\mathbb{N}} \varepsilon(n) = 0$.

Proposition 26. *Let* $\varepsilon : \mathbb{N} \to \mathbb{R}^+$. *If* LPCC($\varepsilon$) *holds, then* $\lim_{n\in\mathbb{N}} \varepsilon(n) = 0$

Proof. It suffices to show that for every positive $\delta \in \mathbb{R}$ there is an $m \in \mathbb{N}$ such that

$$\lim_{n\to\infty} \Pr_{G+K(A)\in \text{ER}(n,1/2,\ 4\cdot\log\ n)} \big[G \equiv_{\text{LFP}_m} G + K(A)\big] \leq \delta.$$

This is an immediate consequence of the following lemma as there are LFP-sentences expressing in an ordered graph that the number of edges is congruent i modulo ℓ (for $\ell \in \mathbb{N}$ and $i \in \{0, \ldots, \ell - 1\}$). □

Lemma 27. *Let* $\ell \in \mathbb{N}$ *and* $i \in \{0, \ldots, \ell - 1\}$. *Then for every nondecreasing and unbounded function* $h : \mathbb{N} \to \mathbb{N}$,

$$\lim_{n\to\infty} \Pr_{G+K(A)\in \text{ER}(n,1/2,\ h(n))} \big[\ |E(G + K(A))| - |E(G)| \equiv i \mod \ell\big] = \frac{1}{\ell}.$$

Proof. Let $n \in \mathbb{N}$ and $k \in [n]$. Then, for every graph G with vertex set $[n]$, every subset A of $[n]$ of size k, and every $i \in \{0, 1, \ldots, \ell - 1\}$, we have

$$|E(G+K(A))| - |E(G)| \equiv i \mod \ell \iff |E(G) \cap E(K(A))| \equiv \binom{k}{2} - i \mod \ell. \quad (10)$$

Here, $E(K(A))$ denotes the set of edges of the clique on A. We set $s(k) := \binom{k}{2}$. Then $|E(K(A))| = s(k)$. For every $r \in \{0, 1, \ldots, \ell - 1\}$, we let $a_r(k)$ be the number of those subsets of $E(K(A))$, whose cardinality is equivalent to r modulo ℓ; thus

$$a_r(k) = \sum_{0 \le j \le s}^{j \equiv r \mod \ell} \binom{s(k)}{j}.$$

Note that $a_r(k)$ does not depend on n (and in particular, not on the chosen subset A of $[n]$ of size k). By (10), we get for all $n \ge k$, all subsets A of $[n]$ of size k, and all $i \in \{0, 1, \ldots, \ell - 1\}$,

$$\Pr_{G \in \mathrm{ER}(n,1/2)} \Big[|E(G + K(A))| - |E(G)| \equiv i \mod \ell \Big] = \frac{a_{s(k)-i}}{2^{s(k)}}. \tag{11}$$

Claim 1. Let $r \in \{0, 1, \ldots, \ell - 1\}$. Then (here $a_\ell(k) := a_0(k)$),

$$\lim_{k \to \infty} \frac{|a_{r+1}(k) - a_r(k)|}{2^{s(k)}} = 0.$$

Proof of Claim 1: First we show that there is a positive $\iota \in \mathbb{R}$ such for all sufficiently small positive $\delta \in \mathbb{R}$ and all $n \in \mathbb{N}$ with $(1/2 - \delta) \cdot n \in \mathbb{N}$,

$$\binom{n}{(1/2 - \delta) \cdot n} = O\left(\frac{2^{(1 - \iota\delta^2) \cdot n}}{\sqrt{n}} \right). \tag{12}$$

In fact, using Stirling's formula

$$\sqrt{2\pi n} \cdot \left(\frac{n}{e}\right)^n \le n! \le e \cdot \sqrt{n} \cdot \left(\frac{n}{e}\right)^n,$$

we get for $n \in \mathbb{N}$ and $\varepsilon \in \mathbb{R}$ with $\varepsilon \cdot n \in \mathbb{N}$,

$$\binom{n}{\varepsilon \cdot n} \le \frac{e \cdot 2^{H(\varepsilon) \cdot n}}{2\pi \cdot \sqrt{\varepsilon \cdot (1 - \varepsilon) \cdot n}}. \tag{13}$$

Here $H : (0, 1) \to \mathbb{R}$ denotes the *binary entropy* function defined by

$$H(\varepsilon) = -\varepsilon \cdot \log \varepsilon - (1 - \varepsilon) \cdot \log (1 - \varepsilon).$$

Recall that H attains 1, its maximum value, at $\varepsilon = 1/2$. We want to bound the values of H in the neighborhood of $1/2$. Let $\delta \in \mathbb{R}$ with $0 \le \delta < 1/2$. Then

$$H(1/2 - \delta) = -(1/2 - \delta) \cdot \log (1/2 - \delta) - (1/2 + \delta) \cdot \log (1/2 + \delta).$$

Using the Taylor series for $\log x$, we get from this equality that there is an $\iota \in \mathbb{R}$ with $\iota > 0$ such that for sufficiently small $\delta \in \mathbb{R}$ with $\delta \ge 0$,

$$H(1/2 - \delta) \le 1 - \iota \cdot \delta^2. \tag{14}$$

Hence, assuming in addition that $\delta < 1/\sqrt{8}$ and $(1/2-\delta)\cdot n \in \mathbb{N}$,

$$\begin{aligned}\binom{n}{(1/2-\delta)\cdot n} &\le \frac{e\cdot 2^{(1-\iota\cdot\delta^2)\cdot n}}{2\pi\cdot\sqrt{(1/4-\delta^2)\cdot n}} && \text{(by (13) and (14)}\\ &= O\left(\frac{2^{(1-\iota\cdot\delta^2)\cdot n}}{\sqrt{n}}\right) && \text{(as } \delta^2 < 1/8),\end{aligned}$$

which is the desired equality.

Now let $j, s \in \mathbb{N}$ satisfy $0 \le j < s$. Note that

$$\binom{s}{j+1} - \binom{s}{j} = \frac{s-2j-1}{j+1}\cdot\binom{s}{j}. \tag{15}$$

We distinguish two cases.

Case $j \le s/2 - \sqrt[3]{s^2}$: Then $j \le (1/2-\delta)\cdot s$ for $\delta \in (s^{-2/3}, s^{-1/3})$. If $(1/2-\delta)\cdot s \in \mathbb{N}$, we get by (12)

$$\begin{aligned}\binom{s}{j+1} - \binom{s}{j} &\le s\cdot\binom{s}{(1/2-\delta)\cdot s} \le s\cdot O\left(\frac{2^{(1-\iota\cdot\delta^2)\cdot s}}{\sqrt{s}}\right) && \text{(by (15) and (12))}\\ &= O\left(\frac{s\cdot 2^s}{\sqrt{s}\cdot 2^{\iota\cdot\sqrt[3]{s}}}\right) = O\left(\frac{\sqrt{s}\cdot 2^s}{2^{\iota\cdot\sqrt[3]{s}}}\right).\end{aligned}$$

Case $s/2 - \sqrt[3]{s^2} < j < s/2$: Then

$$\begin{aligned}\binom{s}{j+1} - \binom{s}{j} &\le \frac{2\sqrt[3]{s^2}}{s/2-\sqrt[3]{s^2}+1}\cdot\binom{s}{s/2} && \text{(by (15)}\\ &= O\left(\frac{2^s}{s^{-2/3+3/3+1/2}}\right) = O\left(\frac{2^s}{s^{5/6}}\right).\end{aligned}$$

Putting all together we get the statement of Claim 1 as follows

$$\begin{aligned}a_{r+1}(k) - a_r(k) &= \sum_{0\le j\le s(k)}^{j\equiv r+1 \mod \ell}\binom{s(k)}{j} - \sum_{0\le j\le s(k)}^{j\equiv r \mod \ell}\binom{s(k)}{j}\\
&\le \sum_{0\le j<s(k)/2}^{j\equiv r \mod \ell}\left(\binom{s(k)}{j+1} - \binom{s(k)}{j}\right)\\
&= \sum_{0\le j\le s(k)/2-\sqrt[3]{s(k)^2}}^{j\equiv r \mod \ell}\left(\binom{s(k)}{j+1} - \binom{s(k)}{j}\right)\\
&\qquad + \sum_{s(k)/2-\sqrt[3]{s(k)^2}<j<s(k)/2}^{j\equiv r \mod \ell}\left(\binom{s(k)}{j+1} - \binom{s(k)}{j}\right)\\
&= O\left(\frac{s(k)\cdot\sqrt{s}\cdot 2^{s(k)}}{2^{\iota\cdot\sqrt[3]{s(k)}}}\right) + O\left(\frac{s(k)^{2/3}\cdot 2^{s(k)}}{s(k)^{5/6}}\right) \quad \text{(by the equalities derived above)}\\
&= o(2^{s(k)})\end{aligned}$$

Similarly we can show $a_r(k) - a_{r+1}(k) = o(2^{s(k)})$. ⊣

Claim 2. Let $\delta > 0$. If k is sufficiently large, then for all $n \geq k$, all subsets A of $[n]$ of size k, and all $i \in \{0, 1, \ldots, \ell - 1\}$, we have

$$\frac{1}{\ell} - \delta \leq \Pr_{G \in \mathrm{ER}(n,1/2)} \Big[|E(G + K(A))| - |E(G)| \equiv i \mod \ell \Big] \leq \frac{1}{\ell} + \delta.$$

Proof of Claim 2: For every $i \in \{0, 1, \ldots, \ell - 1\}$ let

$$p_i(k) := \frac{a_{s(k)-i}(k)}{2^{s(k)}}.$$

Claim 1 implies that for every $\iota > 0$ and all sufficiently large k,

$$|p_{i+1}(k) - p_i(k)| \leq \iota.$$

Thus,

$$p_0(k) - i \cdot \iota \leq p_i(k) \leq p_0(k) + i \cdot \iota. \tag{16}$$

As $\sum_{j=0}^{\ell-1} j = \ell \cdot (\ell - 1)/2$, we obtain

$$\ell \cdot p_0(k) - \frac{\ell \cdot (\ell - 1)}{2} \cdot \iota \leq \sum_{j=0}^{\ell-1} p_j(k) = 1 \leq \ell \cdot p_0(k) + \frac{\ell \cdot (\ell - 1)}{2} \cdot \iota$$

Hence,

$$\frac{1}{\ell} - \frac{(\ell - 1)}{2} \cdot \iota \leq p_0(k) \leq \frac{1}{\ell} + \frac{(\ell - 1)}{2} \cdot \iota. \tag{17}$$

Choosing ι small enough, (16) and (17) imply for all sufficiently large k and every $i \in \{0, 1, \ldots, \ell - 1\}$,

$$\frac{1}{\ell} - \delta \leq p_i(k) \leq \frac{1}{\ell} + \delta.$$

As for all $n \geq k$, all subsets A of $[n]$ of size k, and all $i \in \{0, 1, \ldots, \ell - 1\}$, we have (compare (11))

$$p_i(k) = \frac{a_{s(k)-i}}{2^{s(k)}} = \Pr_{G \in \mathrm{ER}(n,1/2)} \Big[|E(G + K(A))| - |E(G)| \equiv i \mod \ell \Big],$$

this yields our claim. ⊣

Clearly, Claim 2 immediately implies the statement of Lemma 27. □

9. Further results and open questions

In Section 4 we have seen that for no problem Q of ordered graphs there exists a (Q, LFP)-sequence, which can be generated in polynomial output time. Recall that LFP captures polynomial time on ordered graphs. More generally, let L be a logic capturing one of the complexity classes LOGSPACE, P, or PSPACE on (ordered) graphs: Then, for no problem Q of (ordered) graphs we can generate a (Q, L)-sequence (G_m, H_m) by an algorithm which satisfies the resource bound in $|V(G_m)| + |V(H_m)|$ characteristic for the corresponding complexity class, e.g., not in space $O(\log(|V(G_m)| + |V(H_m)|))$

for LOGSPACE. Furthermore there are extensions of these results to "nondeterministic classes" such as NLOGSPACE and NP and extensions for so-called Ajtai-Fagin games adequate for (monadic) Σ^1_1 (see [5] for most of these results).

We are far from understanding when an efficiently computable (Q, L)-sequence exists. Even for first-order logic we have no simple and informative characterization of the problems Q with a (Q, FO)-sequence computable in polynomial output time. Besides the "negative" Example 13, we have a positive result: If Q is NP-hard under FO-reductions (a property shared by many natural NP-complete problems), then a (Q, FO)-sequence can be generated in polynomial output time.

In Section 5 we have mentioned that in most applications of the Ehrenfeucht-Fraïssé-method the verification that G_m and H_m satisfy the same sentences of the corresponding logic of "quantifier rank" or length $\leq m$ was done by an algorithm running in time $f(m)\cdot(|V(G_m)|+|V(H_m)|)^{O(1)}$ for some computable function f. In the Appendix of [5], we have shown this explicitly for two (nontrivial) applications of the method. However, this is not always the case; for example, not for the highly nontrivial application of the Ehrenfeucht-Fraïssé-method in [21].

We have seen in Section 6 that LPCC($1/q$) for some $q \in \mathbb{N}[X]$ implies P $\neq$ NP. Can one refute the statement "there is a $q \in \mathbb{N}[X]$ with LPCC($1/q$)?" or are there results or insights which make the statement plausible?

Furthermore, we ask: Is it true that for every single LFP-sentence φ we have

$$\lim_{n\to\infty} \Pr_{G+K(A)\in \mathrm{ER}(n,1/2,\ 4\cdot\log\ n)} \big[G \models \varphi \iff G + K(A) \models \varphi\big] \geq 1/2.$$

We have shown that every algorithm realizable by AC^0 circuits almost surely can not distinguish G and $G + K(A)$ for $G + K(A) \in \mathrm{ER}(n, 1/2,\ 4 \cdot \log n)$.

References

1. Miklós Ajtai and Ronald Fagin. Reachability is harder for directed than for undirected finite graphs. *The Journal of Symbolic Logic*, 55(1):113–150, 1990.
2. Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Struct. Algorithms*, 13(3-4):457–466, 1998.
3. Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.
4. Uwe Bosse. An "Ehrenfeucht-Fraïssé game" for fixpoint logic and stratified fixpoint logic. In *Computer Science Logic, 6th Workshop, CSL '92, San Miniato, Italy, 1992*, volume 702 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 1993.
5. Yijia Chen and Jörg Flum. On limitations of the Ehrenfeucht-Fraïssé-method in descriptive complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:65, 2013.
6. Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.
7. Erich Grädel et. al. *Finite Model Theory and Its Applications*. Springer, 2006.
8. Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On monadic NP vs. monadic co-NP. *Information and Computation*, 120(1):78–92, 1995.
9. Roland Fraïssé. Sur quelques classifications des systèmes de relations. *Université d'Alger, Publications Scientifiques, Série A*, 1:35–182, 1954.
10. Martin Grohe. Arity hierarchies. *Annals of Pure and Applied Logic*, 82(2):103–163, 1996.
11. Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best Nash equilibrium? *SIAM Journal on Computing*, 40(1):79–91, 2011.
12. Yuguo He. k variables are needed to define k-clique in first-order logic. *CoRR*, abs/1501.04572, 2015.

13. Mark Jerrum. Large cliques elude the metropolis process. *Random Structures and Algorithms*, 3(4):347–360, 1992.
14. Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Designs, Codes and Cryptography*, 20(3):269–280, 2000.
15. Max Kubierschky. Yet another hierarchy theorem. *The Journal of Symbolic Logic*, 65(2):627–640, 2000.
16. Ludek Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.
17. Lorenz Minder and Dan Vilenchik. Small clique detection and approximate Nash equilibria. In *Proceedings of the 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2009, and the 13th International Workshop on Randomization and Computation, RANDOM 2009, Berkeley, CA, USA, August 21-23*, volume 5687 of *Lecture Notes in Computer Science*, pages 673–685, 2009.
18. Benjamin Rossman. Ehrenfeucht-Fraïssé games on random structures. In *Proceedings of the 16th International Workshop on Logic, Language, Information and Computation (WoLLIC'09)*, Lecture Notes in Computer Science, pages 350–364. Springer, 2009.
19. Rahul Santhanam. The complexity of explicit constructions. *Theory of Computing Systems*, 51(3):297–312, 2012.
20. Thomas Schwentick. Graph connectivity and monadic NP. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS'94)*, pages 614–622. IEEE Computer Society, 1994.
21. Saharon Shelah and Joel Spencer. Zero-one laws for sparse random graphs. *Journal of the American Mathematical Society*, 1:97–115, 1988.