Electronic Colloquium on Computational Complexity, Report No. 65 (2013)

On Limitations of the Ehrenfeucht-Fraïssé-method in Descriptive Complexity

Yijia Chen
Department of Computer Science
Shanghai Jiaotong University
yijia.chen@cs.sjtu.edu.cn

Jörg Flum
Mathematisches Institut
Albert-Ludwigs-Universität Freiburg
joerg.flum@math.uni-freiburg.de

Abstract

Ehrenfeucht-Fraïssé games and their generalizations have been quite successful in finite model theory and yield various inexpressibility results. However, for key problems such as P $\neq$ NP or NP $\neq$ co-NP no progress has been achieved using the games. We show that for these problems it is already hard to get the board for the corresponding Ehrenfeucht-Fraïssé game. We obtain similar results for the so-called Ajtai-Fagin games and for a variant where the structures are obtained randomly.

1. Introduction

Originally Ehrenfeucht-Fraïssé games were used to show that two structures satisfy the same sentences of first-order logic FO or at least, the same such sentences of quantifier rank at most m for a given natural number m. In classical model theory the games have widely been applied, e.g., to show the completeness of theories. While Fraïssé [7] introduced this method in more algebraic terms, Ehrenfeucht [5] phrased it in an appealing game-theoretic form. Generalizations of the games to fragments and to extensions of first-order logic were introduced and put to good use.

In finite model theory and in descriptive complexity theory Ehrenfeucht-Fraïssé games for FO are mainly used to obtain *inexpressibility results*, that is, they are used to show that given properties are not expressible in FO (the standard tool for this purpose in classical model theory, the compactness theorem, does not survive when restricting to finite structures). Concerning generalizations one developed games for further logics, mainly for logics relevant in descriptive complexity theory like deterministic transitive closure logic DTC, least fixed-point logic LFP, and finite variable logics.

To show via the Ehrenfeucht-Fraïssé method that, for example, for (finite) graphs [1] "connectivity" is not expressible in first-order logic one exhibits, for every natural number m, a connected graph G_m and a graph H_m, which is not connected, such that $G_m \equiv_{\mathrm{FO}_m} H_m$, that is, G_m and H_m satisfy the same sentences of first-order logic of quantifier rank at most m. The latter property is shown by analyzing the Ehrenfeucht-Fraïssé game with board (G_m, H_m).

We realized that essentially in all successful applications of the Ehrenfeucht-Fraïssé method showing that a property Q of structures is not expressible in a logic L, the boards for the game can be constructed in a reasonable time and the equivalence of the corresponding structures can be realized efficiently; i.e., more or less, the two following conditions are fulfilled (we identify Q with the class of structures having the property Q):

(i) (*"The boards may be constructed in a reasonable time."*) There is an algorithm that on input m yields a board $(\mathcal{A}_m, \mathcal{B}_m)$ in time polynomial in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$ [2] with $\mathcal{A}_m \in Q$, $\mathcal{B}_m \notin Q$, and $\mathcal{A}_m \equiv_{L_m} \mathcal{B}_m$ (that is, $\mathcal{A}_m$ and $\mathcal{B}_m$ satisfy the same sentences of the logic L of "quantifier rank" at most m).

(ii) (*"The property $\mathcal{A}_m \equiv_{L_m} \mathcal{B}_m$ may be verified in a reasonable time."*) There is an algorithm that

[1] In the following all structures will assumed to be finite.
[2] $\|\mathcal{A}\|$ denotes the size of the structure $\mathcal{A}$.

ISSN 1433-8092

verifies that $\mathcal{A}_m \equiv_{L_m} \mathcal{B}_m$ in time $f(m) \cdot (\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

In finite model theory, Ehrenfeucht-Fraïssé games were also used to get a second type of results, namely, *hierarchy results*. A hierarchy result states that a certain hierarchy is strict. E.g., for $m \in \mathbb{N}$ consider the fragment FO^m consisting of those formulas of FO which contain at most m variables. The hierarchy $(\mathrm{FO}^m)_{m\in\mathbb{N}}$ is strict, that means, that for every $m \in \mathbb{N}$ there is an FO^{m+1}-sentence not equivalent to any FO^m-sentence.

Let us mention some further examples of inexpressibility results (see (I1)–(I3)) and hierarchy results (see (H1)–(H3)) obtained by the Ehrenfeucht-Fraïssé method using the classical Ehrenfeucht-Fraïssé games or variants such as the so called Ajtai-Fagin games or probabilistic generalizations of them.

(I1) There is a polynomial time property of structures not expressible in fixed-point logic with counting [4].

(I2) Reachability in directed graphs is not expressible in monadic Σ^1_1 [1].

(I3) For ordered graphs connectivity is not expressible in monadic Σ^1_1 [15].

(H1) The finite variable hierarchy for FO on ordered structures is strict [14].

(H2) The arity hierarchies are strict for DTC and LFP [8].

(H3) For every $k \in \mathbb{N}$ the hierarchy whose m-th member consists of formulas with at most m nested k-ary fixed-point operators is strict for DTC and LFP [10].

Also in the corresponding derivations of these hierarchy results (analogues of) the properties (i) and (ii) are fulfilled.

In contrast to some methods of complexity theory such as the method of proofs merely based on diagonalization [3] or the method of natural proofs [13], there exist proofs by the Ehrenfeucht-Fraïssé method of statements such as P $\neq$ NP or NP $\neq$ co-NP assuming the statements are true. Indeed, P $\neq$ NP if and only if there is 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 would be shown by the Ehrenfeucht-Fraïssé game for LFP. And in [6], 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 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 $\Sigma^1_1 = \Pi^1_1$ question has not so far led to any progress on either of these questions.*
>
> *One way of attacking these difficult questions is to restrict the classes under consideration... The hope is that the restriction to the monadic classes will yield more tractable questions and will serve as a training ground for attacking the problems in their full generality.*

Definitely, the authors are right with their observation that "playing" second-order Ehrenfeucht-Fraïssé games is very difficult.[3] However, the results of this paper show that there is a further fundamental difference between, on the one hand, monadic classes and other cases handled with success so far and partly mentioned above and, on the other hand, key problems such as P $\neq$ NP and NP $\neq$ co-NP. In fact, there is *a difference already in the complexity of constructing the boards for the Ehrenfeucht-Fraïssé game.* So far in successful applications of the Ehrenfeucht-Fraïssé method the boards could be constructed in a reasonable time (see (i)). However, a sequence of boards $(G_m, H_m)_{m\in\mathbb{N}}$ as mentioned above to show that the 3-colorability problem is not in P (and hence, P $\neq$ NP) can *not* be constructed in time polynomial in $\|G_m\| + \|H_m\|$. Even more, it is open whether we can get such a sequence of boards by an algorithm more efficient than brute force.

[3] But note that to derive the results of (H2) and (H3) the corresponding authors successfully apply games for logics containing nonmonadic second-order quantifiers.

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" results even stronger with respect to the existence of positive applications of the Ehrenfeucht-Fraïssé method for sufficiently rich logics.

For simplicity, we speak not only in the title but also in the whole paper of the Ehrenfeucht-Fraïssé method and in some sections even of Ehrenfeucht-Fraïssé games instead of using the cumbersome formulation "the selection of boards for the Ehrenfeucht-Fraïssé game."

The content of the different sections is the following. After fixing our notation in Section 2, we recall the Ehrenfeucht-Fraïssé method in Section 3. In particular, we see that a property Q of structures is not expressible in a logic L if there is a sequence of boards $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ such that $\mathcal{A}_m \in Q$, $\mathcal{B}_m \notin Q$, and such that every L_m-sentence holding in $\mathcal{A}_m$ holds in $\mathcal{B}_m$ (where L_m consists of all L-sentences of "quantifier rank" at most m). We call such a sequence a (Q, L)-sequence. Among others, in Section 4 we show that if the logic L captures one of the complexity classes LOGSPACE, P, or PSPACE, then we can't get a (Q, L)-sequence by an algorithm which satisfies the resource bound in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$ characteristic for the corresponding complexity class (e.g., not in space log $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)$ for LOGSPACE and not in time polynomial in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$ for P). We realized that in the proof we do not need any specific properties of Q. Thus we can show that within the same resource bound we can't get an L-sequence, that is, a sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ such that $\mathcal{A}_m \not\cong \mathcal{B}_m$ and every L_m-sentence holding in $\mathcal{A}_m$ holds in $\mathcal{B}_m$.

As it is hard to construct $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$, can we say anything on $\min\{\|\mathcal{A}_m\|, \|\mathcal{B}_m\|\}$ and $\max\{\|\mathcal{A}_m\|, \|\mathcal{B}_m\|\}$? In Section 5, for least fixed-point logic LFP, we study the minimal size of structures in LFP-sequences and in (Q, LFP)-sequences (taking as LFP_m the set of LFP-formulas of length at most m). It turns out that $2^{\varepsilon\cdot m}$, for some $\varepsilon > 0$, is a lower bound for the size of the structures of the mth member of such sequences; furthermore, one gets an upper bound exponentially related to m for such sequences of structures of minimum size, at least if Q has no circuit size in $2^{o(n)}$ (see Theorem 5.1 for the precise statements).

We then turn to generalizations of the Ehrenfeucht-Fraïssé method: We see that limitations similar to those we obtained in Section 4 for the classical Ehrenfeucht-Fraïssé method hold for the Ajtai-Fagin variant of it (see Section 6) and for a variant where the structures are obtained randomly (see Section 7).

Finally, in the Appendix, we verify that the successful applications of the Ehrenfeucht-Fraïssé method mentioned in (I3) and (H2) satisfy the conditions (i) and (ii), since this is not obvious for these two applications.

2. Preliminaries

By $n^{O(1)}$ we denote the class of all polynomially bounded functions on $\mathbb{N}$, the set of natural numbers. We let Σ be the alphabet $\{0, 1\}$ and denote the length of a string $x \in \Sigma^*$ by $|x|$. We identify decision problems with subsets of Σ^*.

Structures. A *vocabulary* τ is a finite set of relation symbols. Each relation symbol has an *arity*. A *structure* $\mathcal{A}$ of vocabulary τ, or *τ-structure*, consists of a nonempty set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^r$ of each r-ary relation symbol $R \in \tau$. In this paper all structures are assumed to have a finite universe.

If τ contains a binary relation symbol $<$ and in the structure $\mathcal{A}$ the relation $<^{\mathcal{A}}$ is a (total) ordering of the universe, then $\mathcal{A}$ is an *ordered* structure. We denote by $\|\mathcal{A}\|$ the length of a binary string encoding $\mathcal{A}$, up to isomorphism, in a natural way. Thereby two ordered isomorphic structures should be encoded by the same string. A *problem (or, property) of τ-structures* is a class of τ-structures closed under isomorphism.

We say that a deterministic or nondeterministic algorithm is an *algorithm for (ordered) τ-structures* if the class of accepted (ordered) τ-structures is closed under isomorphism. By our convention on the encoding of ordered structures every deterministic or nondeterministic algorithm is an algorithm for ordered τ-structures. By default, algorithms are deterministic.

Often we consider graphs or ordered graphs as structures. Then we also use the graph-theoretic notation for graphs, That is, we denote them by $G = (V(G), E(G))$ where $V(G)$ is the vertex set and $E(G)$ the

edge set of the graph G.

Logics. If L is any logic, we denote its set of formulas again by L. For a vocabulary τ and any set F of formulas of L we let $F[\tau]$ be the set of formulas of F of vocabulary τ. We write $\mathcal{A} \equiv_F \mathcal{B}$ for τ-structures $\mathcal{A}$ and $\mathcal{B}$, if for all $F[\tau]$-sentences φ we have ($\mathcal{A} \models \varphi \iff \mathcal{B} \models \varphi$). Here $\mathcal{A} \models \varphi$ means that φ holds in $\mathcal{A}$. And we write $\mathcal{A} \equiv\rangle_F \mathcal{B}$, if for all $F[\tau]$-sentences φ we have that $\mathcal{A} \models \varphi$ implies $\mathcal{B} \models \varphi$.

Logics and complexity classes. Let L be a logic. A property Q of τ-structures is *expressible in* L (or, *axiomatizable in* L) if there is an $L[\tau]$-sentence such that Q is its class of models. Thereby we identify Q with the class of structures having the property Q. For a complexity class C we write $\mathrm{C} \leq L$ *(on ordered structures)* and say that C-*properties of (ordered) structures are expressible in* L if every problem of (ordered) structures in C is expressible in L.

The logic L *captures the complexity class* C *(on ordered structures)* if for every vocabulary τ and every property Q of (ordered) τ-structures we have:

$$Q \text{ is expressible in } L \iff Q \in \mathrm{C},$$

that is, if $\mathrm{C} \leq L$ (on ordered structures) and every problem of (ordered) structures expressible in L is in C.

We assume familiarity with basic notions of first-order logic FO and of standard logics relevant in descriptive complexity as deterministic transitive closure logic DTC, transitive closure logic TC, least fixed-point logic LFP, the fragment Σ^1_1 of second-order logic, and partial fixed-point logic PFP. Concerning these logics essentially we only use the following well-known facts (further properties of LFP are needed in Section 5):

Theorem 2.1. *(a) On ordered structures,* DTC*,* TC*,* LFP*, and* PFP *capture* LOGSPACE*,* NLOGSPACE*,* P*, and* PSPACE*, respectively.*

(b) Σ^1_1 *captures* NP.

3. The Ehrenfeucht-Fraïssé method

As already mentioned, in finite model theory the Ehrenfeucht-Fraïssé method is often applied to show that a given problem Q of τ-structures is not expressible in a logic L. Thereby, one uses a *filtering* of L, that is, a sequence $(L_m)_{m\in\mathbb{N}}$ satisfying (f1) and (f2).

(f1) $L_0 \subseteq L_1 \subseteq \ldots \subseteq L_m \subseteq \ldots \quad$ and $\quad L = \bigcup_{m\in\mathbb{N}} L_m$.

(f2) For every vocabulary τ there is 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 τ-structures $\mathcal{A}$ and $\mathcal{B}$ and all $m \in \mathbb{N}$ we have:

$$\text{if } \mathcal{A} \equiv\rangle_{L_{s(m)}} \mathcal{B} \text{ and } \mathcal{A} \ncong \mathcal{B}, \text{ then } \|\mathcal{A}\|, \|\mathcal{B}\| > m.$$

The condition (f2) is based on the fact that all logics we consider extend first-order logic and therefore, for every structure they contain a sentence characterizing this structure up to isomorphism. In applications, L_m may consist of the L-formulas of "quantifier rank" at most m or of the L-formulas of length at most m. Unless stated otherwise explicitly, *our results refer to an arbitrary but fixed filtering of the given logic which we do not mention in the statements of our results*.

Most filterings used when applying the Ehrenfeucht-Fraïssé method are finitary in the sense of the following definition.

Definition 3.1. A filtering $(L_m)_{m\in\mathbb{N}}$ of a logic L is *finitary* if for every vocabulary τ each $L_m[\tau]$ contains, up to logical equivalence, only finitely many sentences.

The Ehrenfeucht-Fraïssé method relies on the implication (i) $\Rightarrow$ (ii) in the following well-known result.

Proposition 3.2. *Let L be a logic and Q a problem of τ-structures. Then (i) implies (ii) where:*

(i) For all $m \in \mathbb{N}$ there are τ-structures $\mathcal{A}_m$ and $\mathcal{B}_m$ with

$$\mathcal{A}_m \in Q, \quad \mathcal{B}_m \notin Q, \qquad \textit{and} \qquad \mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m. \tag{1}$$

(ii) Q is not expressible in L.

Furthermore:

(a) If L is closed under conjunction and disjunction and the filtering is finitary, then (i) and (ii) are equivalent.

(b) All previous statements (including (a)) remain valid if L is closed under negation and if in (i) we replace (1) *by*

$$(\mathcal{A}_m \in Q \iff \mathcal{B}_m \notin Q) \qquad \textit{and} \qquad \mathcal{A}_m \equiv_{L_m} \mathcal{B}_m,$$

and all statements remain valid if L is closed under negation and finitary and if in (i) we replace (1) *by*

$$\mathcal{A}_m \in Q, \quad \mathcal{B}_m \notin Q, \qquad \textit{and} \qquad \mathcal{B}_m \equiv\rangle_{L_m} \mathcal{A}_m.$$

Proof: (i) $\Rightarrow$ (ii): By contradiction, assume that Q is the class of models of an $L[\tau]$-sentence φ. We choose m such that $\varphi \in L_m[\tau]$. With (i) we get $\mathcal{A}_m$ and $\mathcal{B}_m$ satisfying (1). As $\mathcal{A}_m \in Q$, we have $\mathcal{A}_m \models \varphi$ and thus, $\mathcal{B}_m \models \varphi$ by $\mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m$; therefore, $\mathcal{B} \in Q$, a contradiction.

(ii) $\Rightarrow$ (i) (under the additional hypotheses on L mentioned in (a)): By contradiction, assume that (i) fails for some $m \in \mathbb{N}$, that is, for all τ-structures $\mathcal{A}$ and $\mathcal{B}$ we have

$$\mathcal{A} \in Q \text{ and } \mathcal{A} \equiv\rangle_{L_m} \mathcal{B} \text{ imply } \mathcal{B} \in Q. \tag{2}$$

For every τ-structure $\mathcal{A}$, by our additional assumptions on L, we can view

$$\varphi^m_{\mathcal{A}} := \bigwedge \{\psi \mid \psi \text{ in } L_m[\tau] \text{ with } \mathcal{A} \models \psi\}$$

as an $L[\tau]$-sentence and can assume that $\{\varphi^m_{\mathcal{B}} \mid \mathcal{B} \text{ a } \tau\text{-structure}\}$ is finite. Thus

$$\varphi^m := \bigvee \{\varphi^m_{\mathcal{A}} \mid \mathcal{A} \in Q\}$$

is an $L[\tau]$-sentence, too. Clearly, for every τ-structure we have

$$\mathcal{B} \models \varphi^m_{\mathcal{A}} \iff \mathcal{A} \equiv\rangle_{L_m} \mathcal{B}.$$

With this equivalence and (2) one easily gets that Q is the class of models of φ^m.

To verify the first statement in (b), observe that for every L-sentence φ there is a natural number m such that L_m contains φ and its negation "$\neg\varphi$." Hence, if in addition, the filtering is finitary, then for every $m \in \mathbb{N}$ there is a $k \geq m$ such that, up to equivalence, $\{\neg\varphi \mid \varphi \in L_m\} \subseteq L_k$; therefore, $\mathcal{B} \equiv\rangle_k \mathcal{A}$ implies $\mathcal{A} \equiv\rangle_m \mathcal{B}$. This yields the second statement in (b). □

So, in order to show by the Ehrenfeucht-Fraïssé method that a problem Q of τ-structures is not expressible in the logic L, it suffices to exhibit a (Q, L)-sequence in the sense of the following definition.

Definition 3.3. Let L be a logic and Q a problem of τ-structures. A sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is a (Q, L)-*sequence* if EF(Q) and EF(L) hold.

EF(Q): $\mathcal{A}_m \in Q$ and $\mathcal{B}_m \notin Q$;

EF(L): $\mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m$.

We speak of a *strong* (Q, L)*-sequence* if instead of EF(L) the property s-EF(L) holds.

s-EF(L): $\mathcal{A}_m \equiv_{L_m} \mathcal{B}_m$.

Clearly, every strong (Q, L)-sequence is a (Q, L)-sequence. By the property (f1) of a filtering every infinite subsequence of a (strong) (Q, L)-sequence is a (strong) (Q, L)-sequence, too.

Using the terminology just introduced, we can restate (part of) Proposition 3.2 as follows:

Corollary 3.4. *(a) If there is a (Q, L)-sequence, then Q is not expressible in L.*

(b) If L is closed under conjunction and disjunction and the filtering is finitary, then there is a (Q, L)-sequence if and only if Q is not expressible in L.

(c) If in addition to the assumptions in (b) the logic L is closed under negation, then there is a strong (Q, L)-sequence if and only if Q is not expressible in L.

We often will consider a logic L and a complexity class C with C $\leq L$ only on *ordered* structures. Then, in order to show by the Ehrenfeucht-Fraïssé method that a problem of ordered τ-structures is not expressible in L, and thus, is not in C, of course, one has to choose *ordered* τ-structures $\mathcal{A}_m$ and $\mathcal{B}_m$. We then speak of an *ordered sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$.

4. Limitations of the Ehrenfeucht-Fraïssé method

Let C denote one of the complexity classes LOGSPACE, NLOGSPACE, P, NP, or PSPACE. A simple diagonalization argument shows:

Proposition 4.1. *Let L be a logic and assume that C $\leq L$ (on ordered structures), that is, that C-properties of (ordered) structures are expressible in L. Let Q be a problem of (ordered) τ-structures and let $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ be an (ordered) (Q, L)-sequence. Then there is no algorithm $\mathbb{A}$ of type C for τ-structures [4] that accepts all $\mathcal{A}_m$ and rejects all $\mathcal{B}_m$ (that is, that decides Q on $\{\mathcal{A}_m, \mathcal{B}_m \mid m \in \mathbb{N}\}$).*

Furthermore, for every infinite subset I of $\mathbb{N}$ the previous statement remains valid if we replace "accepts all $\mathcal{A}_m$ and rejects all $\mathcal{B}_m$" by "accepts $\mathcal{A}_m$ for $m \in I$ and rejects $\mathcal{B}_m$ for $m \in I$."

Proof: By contradiction assume that $\mathbb{A}$ is an algorithm of type C such that for all $m \in \mathbb{N}$,

$$\mathbb{A} \text{ accepts } \mathcal{A}_m \quad \text{and} \quad \mathbb{A} \text{ rejects } \mathcal{B}_m. \tag{3}$$

As $\mathbb{A}$ is an algorithm for (ordered) τ-structures, the class of (ordered) τ-structures accepted by $\mathbb{A}$ is closed under isomorphism. Therefore, since C $\leq L$ (on ordered structures), there is an $L[\tau]$-sentence φ such that for all (ordered) τ-structures $\mathcal{D}$ we have

$$\mathcal{D} \models \varphi \iff \mathbb{A} \text{ accepts } \mathcal{D}.$$

We choose $m \in \mathbb{N}$ such that $\varphi \in L_m$. Then, by (3), $\mathcal{A}_m \models \varphi$ and $\mathcal{B}_m \not\models \varphi$; but this contradicts EF(L).

The last claim of the proposition follows as we already saw that infinite subsequences of (Q, L)-sequences are themselves (Q, L)-sequences. □

Already this simple proposition reflects some limitations of the Ehrenfeucht-Fraïssé method. In fact, to show that P $\neq$ NP, one could, for example, try to find ordered graphs G_m and H_m such that G_m but not H_m is 3-colorable and such that every LFP-sentence of "quantifier rank" at most m holding in G_m holds in H_m. One hopes to get graphs such that it is evident that the first one but not the second one is 3-colorable. The previous proposition shows that "evident" may not be replaced by "solvable in polynomial time."

Further note that in the previous proposition the problem Q could be *any* problem of (ordered) τ-structures. That is, we need no specific property of Q. This reflects the fact that it is already hard to get nonisomorphic structures $\mathcal{A}_m$ and $\mathcal{B}_m$ with $\mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m$. Proposition 4.3 contains the corresponding precise statement. The reader will easily prove it along the lines of the proof of Proposition 4.1.

Definition 4.2. Let L be a logic. A sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ of structures, all of the same vocabulary, is an *L-sequence* if $\mathcal{A}_m \ncong \mathcal{B}_m$ and $\mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m$ for all $m \in \mathbb{N}$ $\big($that is, if $\mathcal{A}_m \ncong \mathcal{B}_m$ for all $m \in \mathbb{N}$ and if EF(L) holds.$\big)$.

[4] In particular, for C = NLOGSPACE and C = NP the algorithm $\mathbb{A}$ is nondeterministic.

Clearly, every (Q, L)-sequence for any problem Q of τ-structures, is an L-sequence.

Proposition 4.3. *Let* $\mathrm{C} \leq L$ *(on ordered structures). For every (ordered)* L*-sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *of* τ*-structures there is no algorithm* $\mathbb{A}$ *of type* C *for* τ*-structures that accepts all* $\mathcal{A}_m$ *and rejects all* $\mathcal{B}_m$*.*

Assume the logic L (and its filtering) have the property that we can decide whether $\mathcal{A} \equiv\rangle_{L_m} \mathcal{B}$ (given m, $\mathcal{A}$, and $\mathcal{B}$). Then, if there is a (Q, L)-sequence for some decidable property Q of structures or if there is an L-sequence, we can get such sequences algorithmically; for example, by a brute force algorithm that for every $m \in \mathbb{N}$ systematically checks for all pairs $(\mathcal{A}, \mathcal{B})$ whether $\mathcal{A} \equiv\rangle_{L_m} \mathcal{B}$. Are there efficient algorithms?

Let C be a deterministic complexity class with $C \leq L$ on ordered structures. Our next result shows that then we can't get the structures $\mathcal{A}_m$ and $\mathcal{B}_m$ of an ordered L-sequence by an algorithm of type C, which satisfies the resource bound in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$. This reveals a further essential limitation of the Ehrenfeucht-Fraïssé method.

In the following we say that an *algorithm* $\mathbb{A}$ *generates the sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *in time* $g(m)$ $\big($*in space* $h(m)\big)$ if $\mathbb{A}$ on input $m \in \mathbb{N}$ outputs $\mathcal{A}_m$ and $\mathcal{B}_m$ in time $g(m)$ $\big($in space $h(m)\big)$.

Theorem 4.4. *Let* L *be a logic.*

(a) If $\textsc{Logspace} \leq L$ *on ordered structures, then there is no algorithm that generates an ordered* L*-sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *in space* $O(\log(\|\mathcal{A}\| + \|\mathcal{B}\|))$*.*

(b) If $\mathrm{P} \leq L$ *on ordered structures, then there is no algorithm that generates an ordered* L*-sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *in time* $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$*.*

(c) If $\textsc{Pspace} \leq L$ *on ordered structures, then there is no algorithm that generates an ordered* L*-sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *in space* $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$*.*

Thus, by Theorem 2.1, we get:

Corollary 4.5. *Let* Q *be a problem of ordered* τ*-structures.*

(a) No ordered (Q, DTC)*-sequence (and hence, no ordered* (Q, TC)*-sequence) can be generated in space* $O(\log(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|))$*.*

(b) No ordered (Q, LFP)*-sequence (and hence, no ordered* (Q, Σ^1_1)*-sequence) can be generated in time* $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$*.*

(c) No ordered (Q, PFP)*-sequence can be generated in space* $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$*.*

Proof of Theorem 4.4. We prove (b); the other proofs are obtained by the obvious modifications. The essential idea is the following: Using an algorithm generating an ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$ one can define a polynomial time algorithm accepting $\mathcal{A}_m$ and rejecting $\mathcal{B}_m$ for infinitely many $m \in \mathbb{N}$. This contradicts Proposition 4.3 (as subsequences of L-sequences are L-sequences).

We call a sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ *monotone* if for all $m \in \mathbb{N}$,

$$\max\big\{\|\mathcal{A}_m\|, \|\mathcal{B}_m\|\big\} < \min\big\{\|\mathcal{A}_{m+1}\|, \|\mathcal{B}_{m+1}\|\big\}. \tag{4}$$

Claim. If there is an algorithm generating an ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$, then there is an algorithm generating a monotone and ordered L-sequence (even a subsequence of $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$) within the same time bound.

Proof of Claim. Assume that $\mathbb{S}$ is an algorithm that generates an ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time polynomial in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$. Let $s : \mathbb{N} \to \mathbb{N}$ be the function according to the property (f2) of a filtering for the common vocabulary of the structures of this sequence. We define a function $h : \mathbb{N} \to \mathbb{N}$ inductively:

$$h(m) := \begin{cases} s(0), & \text{if } m = 0, \\ s\big(\max\{\|\mathcal{A}_{h(m-1)}\|, \|\mathcal{B}_{h(m-1)}\|\}\big), & \text{if } m > 0. \end{cases} \tag{5}$$

As $\mathcal{A}_{h(m)} \equiv\rangle_{h(m)} \mathcal{B}_{h(m)}$, that is, $\mathcal{A}_{h(m)} \equiv\rangle_{s\left(\max\{\|\mathcal{A}_{h(m-1)}\|, \|\mathcal{B}_{h(m-1)}\|\}\right)} \mathcal{B}_{h(m)}$, we have, by (f2), $\|\mathcal{A}_{h(m)}\|$, $\|\mathcal{B}_{h(m)}\| > \max\{\|\mathcal{A}_{h(m-1)}\|, \|\mathcal{B}_{h(m-1)}\|\}$. Moreover, by property (f1), $\mathcal{A}_{h(m)} \equiv\rangle_{h(m)} \mathcal{B}_{h(m)}$ implies $\mathcal{A}_{h(m)} \equiv\rangle_{m} \mathcal{B}_{h(m)}$. Therefore, it is routine to show that the algorithm which on input m first computes $h(m)$ and then simulates $\mathbb{S}$ generates the monotone and ordered L-sequence $(\mathcal{A}_{h(m)}, \mathcal{B}_{h(m)})_{m\in\mathbb{N}}$ in the desired time bound. $\dashv$

We continue the proof of Theorem 4.4 and assume, by contradiction, that there is an algorithm $\mathbb{S}$ which on input $m \in \mathbb{N}$ outputs structures $\mathcal{A}_m$ and $\mathcal{B}_m$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^d$ (with $d \in \mathbb{N}$) such that $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is an ordered L-sequence. By the Claim we may assume that the sequence is monotone. Furthermore, we can assume that $\|\mathcal{A}_m\| \geq \|\mathcal{B}_m\|$ for all $m \in \mathbb{N}$ or that $\|\mathcal{B}_m\| \geq \|\mathcal{A}_m\|$ for all $m \in \mathbb{N}$ (otherwise, one passes to a subsequence which still can be generated in the required time bound). W.l.o.g. we assume that $\|\mathcal{A}_m\| \geq \|\mathcal{B}_m\|$ for all $m \in \mathbb{N}$.

We show that the following polynomial time algorithm $\mathbb{A}$ accepts all $\mathcal{A}_m$ and rejects all $\mathcal{B}_m$, which contradicts Proposition 4.3.

```
𝔸      // 𝒞 an ordered τ-structure
1.  for i = 0 to ‖𝒞‖ do
2.        simulate 𝕊 on i for (2 · ‖𝒞‖)^d steps
3.              if the simulation halts with output (𝒜_i, ℬ_i) and 𝒞 ≅ 𝒜_i
4.                    then accept
5.  reject.
```

As $(\mathcal{A}_m, \mathcal{B}_m)$ is a monotone sequence, we know that $\mathcal{A}_i \ncong \mathcal{B}_j$ for all $i \neq j$ and that $m \leq \|\mathcal{A}_m\|$. Thus, the algorithm $\mathbb{A}$ will reject all $\mathcal{B}_m$. Now we consider an input $\mathcal{C} = \mathcal{A}_m$. Then, $m \leq \|\mathcal{C}\|$. The simulation of $\mathbb{S}$ on m in Line 2 will halt since, by $\|\mathcal{C}\| = \|\mathcal{A}_m\| \geq \|\mathcal{B}_m\|$,

$$(2 \cdot \|\mathcal{C}\|)^d \geq (\|\mathcal{A}_m\|\| + \|\mathcal{B}_m\|)^d.$$

Therefore, $\mathbb{A}$ will accept $\mathcal{A}_m$. The algorithm $\mathbb{A}$ runs in polynomial time as for any two ordered structures we can test in polynomial time whether they are isomorphic. $\square$

The last sentence makes clear why Theorem 4.4 (b) may be wrong if we allow arbitrary and not only ordered structures. But if the (graph) isomorphism problem GI happens to be in P and L is a logic with $\mathrm{P} \leq L$, then there is no algorithm generating an L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$. Note that $\mathrm{P} \leq \Sigma^1_1$ (by Theorem 2.1 (b)). Hence, if GI $\in$ P, then this result applies to Σ^1_1. Without the assumption GI $\in$ P we only are able to show:

Theorem 4.6. *If $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is a Σ^1_1-sequence, then there is no pair $(\mathbb{S}, \mathbb{T})$ of algorithms such that on input $m \in \mathbb{N}$ the algorithms $\mathbb{S}$ and $\mathbb{T}$ output $\mathcal{A}_m$ and $\mathcal{B}_m$, respectively, and $\mathbb{S}$ does this in time $\|\mathcal{A}_m\|^{O(1)}$.*

As NP $\leq \Sigma^1_1$, this result is a special case of:

Theorem 4.7. *Let L be a logic with* NP $\leq L$. *If $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is an L-sequence, then there is no pair $(\mathbb{S}, \mathbb{T})$ of algorithms such that on input $m \in \mathbb{N}$ the algorithms $\mathbb{S}$ and $\mathbb{T}$ output $\mathcal{A}_m$ and $\mathcal{B}_m$, respectively, and $\mathbb{S}$ does this in time $\|\mathcal{A}_m\|^{O(1)}$.*

Proof: The proof is similar to that of Theorem 4.4. By contradiction we assume the existence of a pair $(\mathbb{S}, \mathbb{T})$ of algorithms with the properties mentioned in the theorem; in particular, let $d \in \mathbb{N}$ be such that $\mathbb{S}$ on input m yields $\mathcal{A}_m$ in time $\|\mathcal{A}_m\|^d$. Again we may assume that $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is monotone, that is, that it satisfies (4). For this we argue as in the proof of the Claim in Theorem 4.4, however, in (5) we have to replace the term $\|\mathcal{B}_{h(m-1)}\|$ by $t_{\mathbb{B}(h(m-1))}$, where $t_{\mathbb{B}(h(m-1))}$ is the number of steps of the computation of $\mathbb{B}$ on input $h(m-1)$; in particular, $t_{\mathbb{B}(h(m-1))} \geq \|\mathcal{B}_{h(m-1)}\|$.

On input $m \in \mathbb{N}$ the algorithm $\mathbb{S}$ outputs the encoding of the structure $\mathcal{A}_m$; this encodings yields an ordering $<^{\mathcal{A}_m}$ of $\mathcal{A}_m$. The following polynomial time nondeterministic algorithm $\mathbb{A}$ accepts the class

$$K := \{\mathcal{C} \mid \text{there is an } m \in \mathbb{N} \text{ such that } \mathcal{C} \cong \mathcal{A}_m\}.$$

$\mathbb{A}$ // $\mathcal{C}$ a τ-structure

1. guess an ordering $<$ on C
2. **for** $i = 0$ **to** $\|\mathcal{C}\|$ **do**
3. simulate $\mathbb{S}$ on i for $\|\mathcal{C}\|^d$ steps
4. **if** the simulation halts with output $\mathcal{A}_i$ and $(\mathcal{C}, <) \cong (\mathcal{A}_i, <^{\mathcal{A}_i})$
5. **then** accept
6. reject.

As K contains all $\mathcal{A}_m$ and, by monotonicity of the sequence $(\mathcal{A}_m, \mathcal{B}_m)$, no $\mathcal{B}_m$, this contradicts Proposition 4.3. □

It is not clear to us whether a similar result holds for the $\mathcal{B}_m$s or whether one can show that such a sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ cannot be generated in time polynomial in $\|\mathcal{A}_m\| + \|\mathcal{B}_m\|$. However, both results hold if GI is in co-NP.

Finally, let us remark that for different classes K of structures containing NP-complete problems it is known that, when restricting to K, there is a logic capturing P on K. Thus, one might wonder whether this may help proving P $\neq$ NP via the Ehrenfeucht-Fraïssé method. However, for all known such Ks the isomorphism problem is solvable in polynomial time on K. But then Theorem 4.4 (b) remains true when restricting to structures in K.

4.1. A further result for LOGSPACE. For a given $Q \notin$ LOGSPACE, part (a) of Corollary 4.5 does not rule out the existence of an ordered (Q, DTC)-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ computable in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$. However, we show that if such a sequence exists, then we do not only separate LOGSPACE and P (by Corollary 4.5 (b)) but also the complexity classes LINSPACE and E and at the same time, from such a sequence we would get an explicit problem in E $\setminus$ LINSPACE. Recall that

$$\text{LINSPACE} := \bigcup_{c\in\mathbb{N}} \text{DSPACE}[c \cdot n] \qquad \text{and} \qquad \text{E} := \text{DTIME}(2^{O(n)}).$$

Again we state our result on the level of nonisomorphic structures.

Theorem 4.8. *Let L be a logic with* LOGSPACE $\leq L$ *on ordered structures. If there is an algorithm generating an ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$, then* LINSPACE $\neq$ E.

Proof: Let the sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ be as in the statement of the result. Again, we may assume that the sequence is monotone. Furthermore, by passing to a suitable subsequence, we may assume that $\|\mathcal{A}_m\| \geq \|\mathcal{B}_m\|$ for all m or that $\|\mathcal{B}_m\| \geq \|\mathcal{A}_m\|$ for all m. W.l.o.g. let us assume the second one is the case. Then there is an algorithm $\mathbb{S}$ that on input m computes $\mathcal{B}_m$ in time $\|\mathcal{B}_m\|^d$ for some $d \in \mathbb{N}$.

We consider the following problem:

$$P := \big\{(m, n, i, b) \;\big|\; m, n, i \in \mathbb{N} \text{ in binary, } \|\mathcal{B}_m\| = n \text{ and the } i\text{th bit of the encoding of } \mathcal{B}_m \text{ is } b\big\}.$$

Claim 1. P is in E.

Proof of Claim 1. It is easy to verify that the following algorithm $\mathbb{A}$ decides P in exponential time (note that m, n, i are all in binary).

$\mathbb{A}$ // $m, n, i \in \mathbb{N}$ in binary and $b \in \{0, 1\}$.

1. simulate $\mathbb{S}$ on m for at most n^d steps
2. **if** the simulation does not halt **then** reject
3. **if** the simulation halts **then** let $\mathcal{C}$ be its output
4. **if** $\|\mathcal{C}\| = n$ and the ith bit of the encoding of $\mathcal{C}$ is b **then** accept
5. **else** reject.

⊣

Claim 2. P is not in LINSPACE.

Proof of Claim 2. Assume that there is an algorithm $\mathbb{A}$ that decides P in linear space. We use $\mathbb{A}$ to get an algorithm $\mathbb{B}$ that in logarithmic space accepts all $\mathcal{A}_m$ and rejects all $\mathcal{B}_m$, which contradicts Proposition 4.3.

Let $\mathbb{B}$ the following algorithm:

```
B      // C an ordered structure.
 1.  n ← ||C||
 2.  ℓ ← 1
 3.  while ℓ ≤ n do
 4.        diff ← FALSE
 5.        i ← 1
 6.        while diff = FALSE and i ≤ n do
 7.              b ← the ith bit of the encoding of C
 8.              if A rejects (ℓ, n, i, b) then diff ← TRUE
 9.                    else i ← i + 1
10.        if diff = FALSE then reject and halt
11.              else ℓ ← ℓ + 1
12.  accept.
```

Let $\mathcal{C}$ be an ordered structure and $n := \|\mathcal{C}\|$. First assume $\mathcal{C} = \mathcal{B}_m$ for some $m \in \mathbb{N}$; by monotonicity, then m is uniquely determined and $m \leq \|\mathcal{B}_m\| = \|\mathcal{C}\| = n$. Therefore, eventually the variable ℓ reaches the value m. Then, the while loop between Line 6 to Line 9 detects that $\mathcal{C} = \mathcal{B}_m$ and the algorithm $\mathbb{B}$ rejects. If $\mathcal{C}$ is not (isomorphic to) any of the $\mathcal{B}_m$, then clearly the algorithm $\mathbb{B}$ accepts. As the sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ is monotone, we know that $\mathcal{A}_i \not\cong \mathcal{B}_j$ for all $i, j \in \mathbb{N}$ and thus $\mathbb{B}$ accepts all $\mathcal{A}_m$.

The algorithm $\mathbb{B}$ requires space in order

- to store the variables $n, \ell, \mathtt{diff}, i, b$;
- to simulate $\mathbb{A}$ on (ℓ, n, i, b).

As $\mathbb{A}$ runs in linear space (for inputs ℓ, n and i given in binary) and n denotes the size of the input of the algorithm $\mathbb{B}$, this algorithm runs in logarithmic space. ⊣

The result follows from Claim 1 and Claim 2. □

5. On the size of the structures of an EF-sequence: a case study

Theorem 4.4 (b) shows for the least fixed-point logic LFP, a logic that captures P on ordered structures, that it is hard to construct ordered LFP-sequences. Here we analyze how big the structures of such a sequence have to be at least and how the size increases if we consider (Q, LFP)-sequences for a given Q. We carry out our analysis for a fixed filtering and a fixed vocabulary.

We consider the following finitary filtering $(\mathrm{LFP}_m)_{m\in\mathbb{N}}$ of least fixed-point logic,

$$\mathrm{LFP}_m := \big\{\varphi, \neg\varphi \;\big|\; \varphi \text{ an LFP-sentence with } \|\varphi\| \leq m\big\}.$$

Here $\|\varphi\|$ denotes the number of *symbols* (that is, of connectives, quantifiers, LFP-operators, variables, ...) in φ.

We fix the vocabulary $\tau := \{E, <\}$ for ordered graphs. We assume that an ordered τ-structure has, for some $n \geq 1$, the set $[n] := \{1, \ldots, n\}$ as universe and that $<$ is interpreted by the natural ordering on $[n]$. We therefore can use the graph-theoretic notation $G = (V(G), E(G))$ for ordered τ-structures.

For our analysis we define the function $I : \mathbb{N} \to \mathbb{N}$ by

$$I(m) := \min\big\{\max\{\|G\|, \|H\|\} \;\big|\; G \text{ and } H \text{ are ordered } \tau\text{-structures such that } G \ncong H \text{ and } G \equiv\rangle_{\mathrm{LFP}_m} H\big\}.$$

And for every problem Q of ordered graphs not solvable in polynomial time we introduce a function (which we denote by Q, too, as from the context it will be always clear whether we mean the problem or the function) $Q : \mathbb{N} \to \mathbb{N}$ by

$$Q(m) := \min\big\{\max\{\|G\|, \|H\|\} \;\big|\; G \text{ and } H \text{ are ordered graphs such that } G \in Q,\ H \notin Q, \text{ and } G \equiv\rangle_{\mathrm{LFP}_m} H,\ \big\}.$$

Clearly, $I(m) \leq Q(m)$. If $(G_m, H_m)_{m\in\mathbb{N}}$ is an ordered LFP-sequence, then $I(m) \leq \max\{\|G_m\|, \|H_m\|\}$; and $Q(m) \leq \max\{\|G_m\|, \|H_m\|\}$ for any (Q, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of ordered graphs.

In this section we aim to show the following theorem which contains lower and upper bounds for $I(m)$ and $Q(m)$. The lower bound we derive for $I(m)$ turns out to be also a lower bound for $\min\{\|G\|, \|H\|\}$ (instead of $\max\{\|G\|, \|H\|\}$ in the definition of $I(m)$), cf. Corollary 5.6 for the precise statement. In parts of the theorem we assume that the circuit size of the given problem Q is not in $2^{o(n)}$. Recall that the *circuit size of* Q is the function $\mathscr{H}_Q : \mathbb{N} \to \mathbb{N}$ defined by

$$\mathscr{H}_Q(n) := \min\Big\{d \in \mathbb{N} \;\Big|\; \text{there exists a circuit } D \text{ with } n \text{ input variables and with } |D| \leq d \text{ such that for every ordered graph } G \text{ with } \|G\| = n \ \big(G \in Q \iff D(G) = 1\big)\Big\}.$$

Theorem 5.1. *Let Q be a problem of ordered graphs not solvable in polynomial time.*

(a) For some $\varepsilon > 0$ and all sufficiently large $m \in \mathbb{N}$,

$$2^{\varepsilon \cdot m} \leq I(m) \leq Q(m).$$

(b) Let $\varepsilon > 0$. Then for all sufficiently large $m \in \mathbb{N}$,

$$I(m) \leq 2^{(1+\varepsilon)\cdot m \cdot \log m}.$$

(c) Let $\varepsilon > 0$. If the circuit size $\mathscr{H}_Q(n)$ of Q is not in $2^{o(n)}$, then for infinitely *many $m \in \mathbb{N}$,*

$$Q(m) \leq 2^{(1+\varepsilon)\cdot m \cdot \log m}.$$

The inequality $I(m) \leq Q(m)$ is clear from the definitions of the functions. The following considerations will lead to a proof of the remaining claims.

We set

$$\mathrm{LFP}^0_m := \big\{\varphi \;\big|\; \varphi \text{ an LFP-sentence with } \|\varphi\| \leq m\big\}.$$

We often tacitly use the fact that the relations $\equiv\rangle_{\mathrm{LFP}_m}$, $\equiv_{\mathrm{LFP}^0_m}$, and $\equiv_{\mathrm{LFP}_m}$ coincide. Hence, every (Q, LFP)-sequence is strong.

By a tedious induction one can show:

Lemma 5.2. *Let $m \in \mathbb{N}$. Then $\mathrm{LFP}^0_m[\tau]$ contains at most m^m sentences up to logical equivalence.*

Lemma 5.3 ([16])**.** *There is an algorithm that for every ordered τ-structure G, all $m \in \mathbb{N}$, and all $\varphi \in \mathrm{LFP}_m$ decides whether $G \models \varphi$ in time*

$$O\left(m \cdot \|G\|^{2\cdot m}\right).$$

The next lemma could be reformulated in order to give an explicit function s in (f2) for our concrete case.

Lemma 5.4. *There is a $c \in \mathbb{N}$ such that for any nonisomorphic ordered τ-structures G and H and every $m \in \mathbb{N}$,*

$$\text{if } G \equiv_{\mathrm{LFP}_m} H, \text{ then } |V(G)| > 2^{(m-c)/c} - 1 \text{ and } |V(H)| > 2^{(m-c)/c} - 1.$$

Proof: As LFP captures P on ordered structures, there are LFP-formulas $\varphi_{+1}(x,y)$ and $\varphi_{\times 2}(x,y)$ such that for all ordered τ-structures G (recall that by convention, $V(G) = \big[|V(G)|\big]$) and every $u, v \in V(G)$ we have

- $G \models \varphi_{+1}(u,v) \iff u+1 = v$;
- $G \models \varphi_{\times 2}(u,v) \iff u \times 2 = v$.

Let $i \geq 1$ be a natural number with binary representation $b_1 \dots b_{k(i)}$, where $b_1 = 1$; we write

$$[i]_2 = b_1 \dots b_{k(i)}.$$

For $j \in [k(i)]$ we define inductively the formula $\varphi_{i,j}$ such that for every ordered τ-structure G and any $v \in V(G)$

$$G \models \varphi_{i,j}(v) \quad \iff \quad [v]_2 = b_1 \dots b_j.$$

In particular, then for the formula $\mathit{vertex}_i(x) := \varphi_{i,k(i)}(x)$, we get

$$G \models \mathit{vertex}_i(v) \quad \iff \quad v = i.$$

We set

$$\varphi_{i,1}(x) := \forall y\ x \leq y,$$

$$\varphi_{i,j+1}(x) := \begin{cases} \exists y\big(\varphi_{i,j}(y) \wedge \varphi_{\times 2}(y,x)\big), & \text{if } b_{i+1} = 0, \\ \exists y \exists z\big(\varphi_{i,j}(y) \wedge \varphi_{\times 2}(y,z) \wedge \varphi_{+1}(z,x)\big), & \text{if } b_{i+1} = 1. \end{cases}$$

Note that $\|\mathit{vertex}_i\| = O(k(i)) = O(\log i)$. For every $i, j \in \mathbb{N}$ let

$$\mathit{edge}_{i,j}(x,y) := \exists x \exists y\big(\mathit{vertex}_i(x) \wedge \mathit{vertex}_j(y) \wedge Exy\big).$$

It follows that

- for every ordered τ-structure G we have

$$G \models \mathit{edge}_{i,j} \iff |V(G)| \geq \max\{i,j\} \text{ and } (i,j) \in E(G);$$

- there is a $c \in \mathbb{N}$ such that $\|\mathit{edge}_{i,j}\| \leq c \cdot \log \max(i,j) + c$; in particular, then

$$\big\|\exists x\ \mathit{vertex}_i(x)\big\| \leq c \cdot \log i + c - 1. \tag{6}$$

Now assume $m \in \mathbb{N}$ and let G and H be ordered τ-structures with $G \equiv_{\mathrm{LFP}_m} H$ and with

$$n := |V(G)| \leq 2^{(m-c)/c} - 1.$$

Then, $G \models \neg\exists x\ \mathit{vertex}_{n+1}(x)$. By (6)

$$\big\|\neg\exists x\ \mathit{vertex}_{n+1}(x)\big\| \leq c \cdot \log\,(n+1) + c \leq m.$$

Therefore, $H \models \neg\exists x\ \mathit{vertex}_{n+1}(x)$, i.e., $|V(H)| \leq n$. By symmetry, we conclude $|V(G)| = |V(H)| = n$. For every $i, j \in [n]$ we have

$$\|\mathit{edge}_{i,j}\| \leq c \cdot \log n + c \leq m.$$

Therefore,

$$G \models \mathit{edge}_{i,j} \iff H \models \mathit{edge}_{i,j}.$$

Thus, G and H are isomorphic. □

Lemma 5.5. *There is an $\varepsilon > 0$ such that for all $m \in \mathbb{N}$,*

$$2^{\varepsilon \cdot m} \leq I(m).$$

Proof: We choose $c \in \mathbb{N}$ such that the conclusion of Lemma 5.4 holds and set $\varepsilon = 1/(2c)$. Let $m \geq 2c$. By definition of the function I, there are ordered τ-structures G_m and H_m with

(i) $G_m \ncong H_m$ and $G_m \equiv_{\text{LFP}_m} H_m$;

(ii) $\|G_m\| \leq I(m)$ and $\|H_m\| \leq I(m)$.

By (i) and Lemma 5.4,

$$|V(G_m)|, |V(H_m)| > 2^{(m-c)/c} - 1 \geq 2^{\varepsilon \cdot m} - 1.$$

Then (ii) implies that

$$I(m) \geq 2^{\varepsilon \cdot m}. \qquad \square$$

The preceding proof shows that the inequalities in Theorem 5.1 (a) even hold if we replace in the definitions of the functions I and Q the max-operation by the min-operation; that is:

Corollary 5.6. *There is an $\varepsilon > 0$ such that for all ordered* LFP*-sequences (G_m, H_m) and all sufficiently large $m \in \mathbb{N}$,*

$$2^{\varepsilon \cdot m} \leq \|G_m\|, \|H_m\|.$$

We derive an upper bound for Q (and hence, for I).

Lemma 5.7. *Let Q be a problem of ordered graphs with $\mathscr{H}_Q(n) \notin 2^{o(n)}$. Then for every $\varepsilon > 0$ and* infinitely *many $m \in \mathbb{N}$,*

$$Q(m) \leq 2^{(1+\varepsilon) \cdot m \cdot \log m}.$$

Proof: Let $m \in \mathbb{N}$ and $k_m := Q(m) - 1$. Thus, for every pair of ordered graphs G and H,

$$\text{if } \|G\| \leq k_m, \|H\| \leq k_m \text{ and } G \equiv_{\text{LFP}_m} H, \text{ then both, } G \text{ and } H, \text{ are in } Q \text{ or none is.} \tag{7}$$

We use this fact to define an algorithm $\mathbb{A}^m$ deciding Q on ordered graphs of size at most k_m $(= Q(m) - 1)$. Then we will see that the statement "$Q(m) > 2^{(1+\varepsilon) \cdot m \cdot \log m}$ for all sufficiently large $m \in \mathbb{N}$" implies that $\mathscr{H}_Q(n) \in 2^{o(n)}$.

We come to the details. For every ordered graph G the following algorithm $\mathbb{A}^m_G$, on input an ordered graph H, decides whether $G \equiv_{\text{LFP}_m} H$:

$\mathbb{A}^m_G$ // H an ordered graph.
1. **for all** sentences $\psi \in \text{LFP}_m$ with $G \models \psi$ **do**
2. **if** $H \not\models \psi$ **then** reject
3. accept.

Since G is a fixed graph, the algorithm $\mathbb{A}^m_G$ actually uses a list of all sentences $\psi \in \text{LFP}_m$ with $G \models \psi$. Therefore, by Lemma 5.2,

$$\|\mathbb{A}^m_G\| = O\left(m \cdot m^m\right). \tag{8}$$

By Lemma 5.3 the running time of $\mathbb{A}^m_G$ on input H can be bounded by

$$O\left(m^m \cdot m \cdot \|H\|^{2 \cdot m}\right). \tag{9}$$

Clearly, $\equiv\rangle_{\text{LFP}_m}$ $(= \equiv_{\text{LFP}^0_m})$ is an equivalence relation on all ordered graphs with at most 2^{m^m} equivalence classes (by Lemma 5.2). For each equivalence class containing a graph $G \in Q$ with $\|G\| \leq k_m$ we choose such a G as a representative. Then the following algorithm $\mathbb{A}^m$ accepts an ordered graph H if and only if it is LFP_m-equivalent to a graph $G \in Q$ with $\|G\| \leq k_m$.

$\mathbb{A}^m$ // H an ordered graph.

1. **for all** $G \in Q$ with $\|G\| \le k_m$ which is a representative **do**
2. simulate $\mathbb{A}^m_G$ on H
3. **if** the simulation accepts **then** accept
4. reject.

Together with (7) we get:

$$\text{for every ordered graph } H \text{ with } \|H\| \le k_m\text{:} \quad \big(H \in Q \iff \mathbb{A}^m \text{ accepts } H\big). \tag{10}$$

By (8), we have:

$$\|\mathbb{A}^m\| = O(2^{m^m} \cdot m \cdot m^m) = 2^{O(m^m)}. \tag{11}$$

Furthermore, (9) implies that the running time of $\mathbb{A}^m$ on any ordered input graph H is bounded by

$$O\left(2^{m^m} \cdot m^m \cdot m \cdot \|H\|^{2\cdot m}\right) = 2^{O(m^m + m\cdot \log \|H\|)}. \tag{12}$$

Now let $n \in \mathbb{N}$ be arbitrary and choose an $m \in \mathbb{N}$ such that

$$Q(m) - 1 = k_m < n \le k_{m+1}. \tag{13}$$

Since every algorithm $\mathbb{A}$ of running time $t(n)$ can be simulated by a family of circuits of size $O(\|\mathbb{A}\| \cdot t^2(n))$ (see [12]), we get (by (10)–(12))

$$\mathscr{H}_Q(n) = 2^{O((m+1)^{m+1})} \cdot 2^{O((m+1)^{m+1} + (m+1)\cdot \log\, n)} = 2^{O((m+1)^{m+1} + (m+1)\cdot \log\, n)}. \tag{14}$$

Let $\varepsilon > 0$ and, by contradiction, assume that $Q(m) > 2^{(1+\varepsilon)\cdot m \cdot \log\, m}$ for all sufficiently large $m \in \mathbb{N}$; thus, for $\varepsilon' := \varepsilon/2$ and for sufficiently large m,

$$Q(m) > 2^{(1+\varepsilon')\cdot (m+1)\cdot \log\, m}.$$

Then, for $n \in \mathbb{N}$ as in (13),

$$n \ge 2^{(1+\varepsilon')\cdot (m+1)\cdot \log\, m}.$$

Therefore,

$$n' := n^{1/(1+\varepsilon')} \ge 2^{(m+1)\cdot \log\, m}. \tag{15}$$

The mapping

$$x \mapsto \frac{\log x}{\log \log x}$$

is defined for $x \ge 4$ and it is increasing. Hence, by (15),

$$\frac{\log n'}{\log \log n'} \ge \frac{(m+1)\cdot \log m}{\log (m+1) + \log \log m} = \frac{m+1}{\log (m+1)/\log m + \log \log m/\log m} = \frac{m+1}{\iota(n)}$$

for an appropriate nonincreasing function $\iota : \mathbb{N} \to \mathbb{R}$ with $\lim_{n\to\infty} \iota(n) = 1$. Or equivalently,

$$m + 1 \le \frac{\iota(n) \cdot \log n'}{\log \log n'}.$$

Then, by (14),

$$\begin{aligned}
\mathscr{H}_Q(n) &\le 2^{O\left((\iota(n)\cdot \log n'/\log \log n')^{\iota(n)\cdot \log n'/\log \log n'} + (\iota(n)\cdot \log n'/\log \log n')\cdot \log n\right)} \\
&\le 2^{(\log n')^{\iota(n)\cdot \log n'/\log \log n'}} \cdot 2^{o(n)} \\
&\qquad\qquad \big(\text{as } n' = n^{1/(1+\varepsilon')} \text{ is sufficiently large and } \lim_{n\to\infty} \iota(n) = 1\big) \\
&= 2^{n^{\iota(n)/(1+\varepsilon')}}.
\end{aligned}$$

As $\varepsilon' > 0$, altogether we get

$$\mathscr{H}_Q(n) = 2^{o(n)},$$

which contradicts our assumption $\mathscr{H}_Q(n) \notin 2^{o(n)}$. □

Remark 5.8. Let us assume that for the problem Q of ordered graphs we have $\mathscr{H}_Q(n) \notin 2^{o(n)}$ *infinitely often*. By this we mean that for every function $h : \mathbb{N} \to \mathbb{N}$ in $o(n)$ and all sufficiently large n we have $\mathscr{H}_Q(n) > 2^{h(n)}$. Then a slight modification of the above proof yields that for every $\varepsilon > 0$ we have

$$Q(m) \leq 2^{(1+\varepsilon)\cdot m \cdot \log\, m}$$

for all sufficiently large $m \in \mathbb{N}$. Together with Lemma 5.5 we thus see that $Q(m)$ and m are exponentially related.

A simple counting argument shows that for every n there exists a Boolean function $\{0,1\}^n \to \{0,1\}$ which cannot be computed by any circuit of size

$$\frac{2^n}{2 \cdot n}.$$

The mapping sending n to the term just displayed is not in $2^{o(n)}$ infinitely often. We can turn these Boolean functions into a class Q of ordered graphs with $\mathscr{H}_Q(n) \notin 2^{o(n)}$ infinitely often.

Proof of Theorem 5.1. Part (a) follows from Lemma 5.5 and part (c) from Lemma 5.7. By the previous remark we have seen that there are problems Q of ordered graphs such that

$$Q(m) \leq 2^{(1+\varepsilon)\cdot m \cdot \log\, m}$$

for all $\varepsilon > 0$ and sufficiently large $m \in \mathbb{N}$. As $I(m) \leq Q(m)$, now part (b) follows from (c). □

5.1. Upper bounds on the time to generate EF-sequences. In Section 4 we have seen that we can't generate an ordered LFP-sequence (G_m, H_m) in time polynomial in $\|G_m\| + \|H_m\|$. The following result yields an upper bound for the generating time of a corresponding "brute-force" algorithm in terms of $I(m)$ and hence, using the results of Theorem 5.1 (b), in terms of m. We derive the corresponding result for (Q, LFP)-sequences, where first we take the 3-colorability problem for ordered graphs as Q.

Proposition 5.9. *(a) There is an algorithm that generates an ordered* LFP*-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of τ-structures with*

$$I(m) = \max\{\|G_m\|, \|H_m\|\} \quad \textit{in time} \quad 2^{O\left(I(m)^2\right)}.$$

(b) Let Q be the 3-colorability problem for ordered graphs. If P $\neq$ NP*, then there is an algorithm that generates an (Q, LFP)-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ of ordered graphs with*

$$Q(m) = \max\{\|G_m\|, \|H_m\|\} \quad \textit{in time} \quad 2^{O\left(Q(m)^2\right)}.$$

Proof: We prove (a) and leave (b) to the reader. It should be obvious that the following algorithm $\mathbb{A}$ generates an ordered LFP-sequence $(G_m, H_m)_{m\in\mathbb{N}}$ with $I(m) = \max\{\|G_m\|, \|H_m\|\}$.

```
𝔸    // m ∈ ℕ.
1.  n ← 1
2.  for all ordered τ-structures G with ‖G‖ = n do
3.      for all i = 1 to n do
4.          for all ordered τ-structures H with ‖H‖ = i do
5.              if G ≇ H and G ≡_{LFP_m} H
6.                  then output (G, H) and halt
7.  n ← n + 1
8.  goto 2.
```

For every n there are at most

$$2^{n^2} \cdot n \cdot 2^{n^2} = 2^{2n^2+\log n}$$

pairs (G, H) reaching Line 5. For each such pair checking the condition in Line 5 takes time at most

$$O\left(m^m \cdot m \cdot n^{2 \cdot m} + n\right) = 2^{O(m \cdot \log m + m \cdot \log n)}$$

by Lemma 5.2 and Lemma 5.3. For $n = I(m)$ the algorithm $\mathbb{A}$ will halt in Line 6 for some pair (G, H). Therefore, the running time of $\mathbb{A}$ on input m is bounded by

$$O\left(I(m) \cdot 2^{2I(m)^2+\log I(m)} \cdot 2^{O(m \cdot \log m + m \cdot \log I(m))}\right).$$

By Lemma 5.4, we know that $m \leq I(m)$ for sufficiently large m. Thus, we immediately get that the running time of $\mathbb{A}$ on input m is bounded by

$$2^{O\left(I(m)^2\right)}. \qquad \square$$

As the preceding proof shows, the function $I(m)$ may be computed in time $2^{O\left(I(m)^2\right)}$ (and similarly, if Q is the 3-colorability problem the function $Q(m)$ may be computed in time $2^{O\left(Q(m)^2\right)}$).

Suppose that instead of the 3-colorability problem we consider any problem Q of ordered graphs not solvable in polynomial time. If $G \in Q$ is solvable in time $t(\|G\|)$, then the obvious changes in the preceding proof yields a (Q, LFP)-sequence $(G_m, H_m)_{m \in \mathbb{N}}$ of ordered graphs with $Q(m) = \max\{\|G_m\|, \|H_m\|\}$ in time $2^{O\left(Q(m)^2+\log t(Q(m))\right)}$. In particular, for every $Q \in \text{NP} \setminus \text{P}$, we have the time bound $2^{Q(m)^{O(1)}}$.

6. The Ajtai-Fagin variant of the Ehrenfeucht-Fraïssé-method

We fix again an arbitrary logic L with a filtering and a problem Q of τ-structures. We reinterpret the Ehrenfeucht-Fraïssé-method by a game. In the (Q, L)-game there are two players, called *Spoiler* and *Duplicator*. The rules of the game are as follows:

$\big((Q, L)\text{-}1\big)$: Spoiler selects an $m \in \mathbb{N}$.

$\big((Q, L)\text{-}2\big)$: Duplicator selects τ-structures $\mathcal{A}$ and $\mathcal{B}$.

Duplicator wins if

$$\mathcal{A} \in Q, \quad \mathcal{B} \notin Q, \quad \text{and} \quad \mathcal{A} \equiv\rangle_{L_m} \mathcal{B}. \tag{16}$$

Otherwise, Spoiler wins. Spoiler or Duplicator *has a winning strategy* if he can guarantee that he will win, no matter how the other player plays. Thus, we can identify (Q, L)-sequences with winning strategies for Duplicator.

By Corollary 3.4 we know that if Duplicator has a winning strategy, then Q is not expressible in L. Furthermore, this condition is also necessary if L is closed under Boolean conjunction and disjunction and the filtering is finitary.

The Ajtai-Fagin game is a variant of this game tailored for pseudo-elementary classes of L which simplifies for Duplicator the verification that Q is not pseudo-elementary. The main known applications of this game are for $L = \text{FO}$ and thus we will restrict ourselves to this case here. Then the pseudo-elementary classes are those axiomatizable by a Σ^1_1-sentence, that is, by a sentence of the form

$$\exists X_1 \ldots \exists X_\ell \, \psi,$$

where $\psi = \psi(X_1, \ldots, X_\ell)$ is a first-order formula containing the second-order variables $X_1, \ldots, X_\ell$ of any arity. It is easy to see that every such sentence is logically equivalent to one of the form

$$\exists X \chi$$

with first-order χ. Recall that Σ^1_1 captures NP (cf. Theorem 2.1 (b)).

We assume that a finitary filtering $(\mathrm{FO}_m)_{m\in\mathbb{N}}$ of first-order logic has been fixed. As already mentioned in the proof of Proposition 3.2, then for every τ-structure $\mathcal{A}$ and $m \in \mathbb{N}$ the first-order (!) sentence

$$\rho^m_{\mathcal{A}} := \bigwedge \{\psi \mid \psi \in \mathrm{FO}_m[\tau],\ \mathcal{A} \models \psi\} \tag{17}$$

has the property that for every τ-structure $\mathcal{B}$ we have

$$\mathcal{B} \models \rho^m_{\mathcal{A}} \iff \mathcal{A} \equiv\rangle_{\mathrm{FO}_m} \mathcal{B}.$$

Furthermore, for notational simplicity, we assume that the property (f2) of a filtering has the simple form

$$\text{if } \mathcal{A} \equiv\rangle_{\mathrm{FO}_m} \mathcal{B} \text{ and } \mathcal{A} \ncong \mathcal{B}, \text{ then } |A|, |B| \geq m.$$

The Q-AF-*game*, the *Ajtai-Fagin game for* Q, [5] again is played by *Spoiler* and *Duplicator*. The rules are as follows:

($Q1$) Spoiler selects $r, m \in \mathbb{N}$.

($Q2$) Duplicator selects a τ-structure $\mathcal{A} \in Q$.

($Q3$) Spoiler selects an r-ary relation R on A, that is, $R \subseteq A^r$.

($Q4$) Duplicator selects a τ-structure $\mathcal{B} \notin Q$ and an r-ary relation S on B.

Duplicator wins if

$$(\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R).$$ [6]

Theorem 6.1 ([1])**.** *The problem Q is not expressible in the logic Σ^1_1-sentence or, equivalently, is not in* NP *if and only if Duplicator has a winning strategy in the Q-AF-game.*

Proof: First assume that Q is axiomatized by a sentence $\exists X \chi$ with X of arity r_0 and with $\neg\chi \in \mathrm{FO}_{m_0}$. Then it is easy to see that the following strategy is a a winning strategy for Spoiler: He chooses r_0, m_0 in the first step and after the selection of a structure $\mathcal{A}$ in Q by Duplicator, he chooses an $R \subseteq A^r$ such that $(\mathcal{A}, R) \models \chi$.

For the converse, note first that for every τ-structure $\mathcal{A}$ and $m \in \mathbb{N}$ the first-order (!) sentence

$$\rho^m_{\mathcal{A}} := \bigwedge \{\psi \mid \psi \in \mathrm{FO}_m[\tau],\ \mathcal{A} \models \psi\}$$

has the property that for every τ-structure $\mathcal{B}$ we have

$$\mathcal{B} \models \rho^m_{\mathcal{A}} \iff \mathcal{A} \equiv\rangle_{\mathrm{FO}_m} \mathcal{B}.$$

Now assume that Spoiler has a winning strategy and that he chooses $r, m \in \mathbb{N}$ in the first step if he plays according to it. Then it is easy to verify, using (17), that Q is the class of models of

$$\exists X \bigwedge \left\{\neg\rho^m_{(\mathcal{B},S)} \;\middle|\; \mathcal{B} \notin Q \text{ and } S \subseteq B^r\right\},$$

where the relation variable X is r-ary. □

Definition 6.2. A (doubly indexed) sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ of τ-structures is a Q-AF-*sequence* if Duplicator can win the Q-AF-game by choosing $\mathcal{A}_{r,m}$ in ($Q2$) assuming that Spoiler selected r, m in ($Q1$).

[5] As the logic is always FO, we do not mention it explicitly.

[6] In view of the winning condition (16) for the Ehrenfeucht-Fraïssé-game, perhaps the reader would also expect here the condition $(\mathcal{A}, R) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{B}, S)$. But note that by the second statement of Proposition 3.2 (b), in (16) we could replace $(\mathcal{A}, R) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{B}, S)$ by $(\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R)$.

Thus, a Q-AF-sequence is part of a winning strategy for Duplicator. As a (Q, L)-sequence it contains the answer to the first selection of Spoiler. The following result shows that it is already hard to get this part of a winning strategy.

Theorem 6.3. *Let Q be a class of τ-structures. Then there is no algorithm that generates a Q-*AF*-sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ in time $\|\mathcal{A}_{r,m}\|^{O(1)}$.*

In the proof of this theorem, we need the following consequence of standard proofs of Fagin's Theorem, i.e., of Theorem 2.1 (b).

Theorem 6.4. *Let K be a class of τ-structures closed under isomorphism. Assume that there is a* nondeterministic *algorithm accepting K in time $O(n^d)$. Then, there is a Σ^1_1-sentence*

$$\varphi = \exists X \psi$$

with X of arity $2(d+4)$ and a first-order sentence ψ such that for all sufficiently large $\mathcal{A}$*,*

$$\mathcal{A} \in K \quad \Longleftrightarrow \quad \mathcal{A} \models \varphi.$$

Furthermore, we observe:

Lemma 6.5. *If $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ is a Q-*AF*-sequence, then $|A_{r,m}| \geq m$ for all $r, m \in \mathbb{N}$.*

Proof: Let $r, m \in \mathbb{N}$ and assume that in a Q-AF-game, which Duplicator plays according to a winning strategy based on $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$, Spoiler first chooses r and m. So, in ($Q2$) Duplicator chooses $\mathcal{A}_{r,m}$. Let $R \subseteq A^r_{r,m}$ be Spoiler's selection in ($Q3$) and $\mathcal{B} \notin Q$ and $S \subseteq B^r$ Duplicator's selection in ($Q4$). Then $(\mathcal{B}_{r,m}, S) \equiv\rangle_{\mathrm{FO}_m}(\mathcal{A}, R)$ and thus, $|A_{r,m}| \geq m$. □

Proof of Theorem 6.3: By contradiction suppose that there is an algorithm $\mathbb{S}$ generating a Q-AF-sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ in time $\|\mathcal{A}_{r,m}\|^{O(1)}$. We may assume that

$$|A_{r,m}| < |A_{r,m+1}| \text{ for all } r, m \in \mathbb{N}. \tag{18}$$

Otherwise we pass to the Q-AF-sequence $(\mathcal{A}^*_{r,m})_{r,m\in\mathbb{N}}$ given by

$$\begin{array}{rcl} \mathcal{A}^*_{r,0} & := & \mathcal{A}_{r,0} \\ \mathcal{A}^*_{r,m+1} & := & \mathcal{A}_{r,|A^*_{r,m}|+1}. \end{array}$$

Then Lemma 6.5 ensures (18).

The algorithm $\mathbb{S}$, on input $r, m \in \mathbb{N}$, outputs a string encoding the structure $\mathcal{A}_{r,m}$; this encoding yields an ordering $<_{r,m}$ of $A_{r,m}$. There is a $d \in \mathbb{N}$ such that from $\mathbb{S}$ we obtain, for $r \in \mathbb{N}$, a nondeterministic algorithm $\mathbb{A}_r$ (defined similarly as the nondeterministic algorithm $\mathbb{A}$ of page 9) that accepts the class

$$K_r := \big\{\mathcal{A} \;\big|\; \mathcal{A} \cong \mathcal{A}_{r,k} \text{ for some } k \in \mathbb{N}\big\}$$

and has running time $O(n^d)$. The constant hidden in the big-O notation may depend on r but note that d is independent of r.

As K_r is closed under isomorphism, by Theorem 6.4 there is a Σ^1_1-sentence

$$\varphi_r = \exists X \psi_r$$

with X of arity $2(d+4)$ and first-order ψ_r such that for all *sufficiently large* $\mathcal{A}$,

$$\mathcal{A} \in K_r \iff \mathcal{A} \models \varphi_r. \tag{19}$$

We set $r_0 := 2(d+4)$ and choose m_0 so large that $\neg\psi_{r_0} \in \mathrm{FO}_{m_0}$ and that for $r = r_0$ the equivalence (19) holds for structures $\mathcal{A}$ with $|A| \geq m_0$.

As $\mathcal{A}_{r_0,m_0} \in K_{r_0}$ and $|A_{r_0,m_0}| \geq m_0$,

$$\mathcal{A}_{r_0,m_0} \models \exists X \psi_{r_0},$$

i.e., there exists an r_0-ary relation $R \subseteq A_{r_0,m_0}^{r_0}$ with

$$\big(\mathcal{A}_{r_0,m_0}, R\big) \models \psi_{r_0}. \tag{20}$$

In $(Q1)$ of an AF-game let Spoiler choose r_0 and m_0. Then Duplicator can win the game by choosing $\mathcal{A}_{r_0,m_0}$ in $(Q2)$. Let Spoiler choose R in $(Q3)$. Then in $(Q4)$ Duplicator answers with a structure $\mathcal{B} \notin Q$ together with an r_0-ary relation S on B such that

$$\big(\mathcal{B}, S\big) \equiv\rangle_{\mathrm{FO}_{m_0}} \big(\mathcal{A}_{r_0,m_0}, R\big).$$

In particular, $|B| \geq m_0$. Furthermore, $(\mathcal{B}, S) \not\models \neg\psi_{r_0}$, as otherwise, $(\mathcal{A}_{r_0,m_0}, R) \models \neg\psi_{r_0}$ contradicting (20). Thus, $(\mathcal{B}, S) \models \psi_{r_0}$, and therefore,

$$\mathcal{B} \models \exists X \psi_{r_0}.$$

Hence, $\mathcal{B} \in K_{r_0}$ by (19). Thus, $\mathcal{B} \in Q$, a contradiction. □

6.1. The monadic case. Originally, Ajtai and Fagin used (a variant of) AF games to prove inexpressibility results for mon-Σ_1^1, the monadic part of Σ_1^1. It consists of all formulas of the form

$$\exists X_1 \ldots \exists X_\ell \, \psi,$$

where $\psi = \psi(X_1, \ldots, X_\ell)$ is a first-order formula and the second-order variables $X_1, \ldots, X_\ell$ are monadic, that is, of arity one. They realized:

> *A problem Q is not axiomatizable by a sentence of* mon-Σ_1^1 *if and only if Duplicator has a winning strategy in the Q-mAF-game,*

where the rules of the Q-mAF-game are as follows:

(m$Q1$) Spoiler selects $r, m \in \mathbb{N}$.

(m$Q2$) Duplicator selects a τ-structure $\mathcal{A} \in Q$.

(m$Q3$) Spoiler selects unary relations $R_1, \ldots, R_r$ on A.

(m$Q4$) Duplicator selects a τ-structure $\mathcal{B} \notin Q$ and unary relations $S_1, \ldots, S_r$ on B.

And Duplicator wins if

$$(\mathcal{B}, S_1, \ldots, S_r) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R_1, \ldots, R_r).$$

If in Definition 6.2 we replace Q-AF-game by Q-mAF-game and $(Q2)$ by (m$Q2$), we get the notion of Q-mAF-sequence.

However, Theorem 6.3 does not survive in the monadic context, not even for ordered structures. In fact, let Q be the connectivity property of ordered graphs. Then, in [15], Schwentick (implicitly) shows:

> *There is an algorithm which generates a Q-*mAF*-sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ in time $\|\mathcal{A}_{r,m}\|^{O(1)}$.*

6.2. The nonisomorphic case. The reader will have noticed that essentially, in the proof of Theorem 6.3, again we did not use any specific property of Q. And again it turns out that it is already hard to find a sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ for Duplicator which may serve him as part of a winning strategy in the "Ajtai-Fagin game for isomorphism." Here we introduce the corresponding framework, state the result, and sketch the proof (which, however, is more involved than that of Theorem 6.3). Again we fix a vocabulary τ.

The $\cong$-AF-game, the *Ajtai-Fagin game for isomorphism*, consists of the following steps:

($\cong$1) Spoiler selects $r, m \in \mathbb{N}$.

($\cong$2) Duplicator selects a τ-structure $\mathcal{A}$.

($\cong$3) Spoiler selects an r-ary relation R on A, that is, $R \subseteq A^r$.

($\cong$4) Duplicator selects a τ-structure $\mathcal{B}$ and an r-ary relation S on B.

Duplicator wins if

$$\mathcal{A} \ncong \mathcal{B} \quad \text{and} \quad (\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R).$$

Definition 6.6. A sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ of τ-structures is an $\cong$-AF-*sequence* if Duplicator can win the $\cong$-AF-game by choosing $\mathbb{A}_{r,m}$ in ($\cong$2) supposing that Spoiler selected r, m in ($\cong$1).

Clearly, every Q-AF-sequence for any class Q of τ-structures is an $\cong$-AF-sequence. Hence, the following result generalizes Theorem 6.3.

Theorem 6.7. *There is no algorithm that generates an* $\cong$-AF-*sequence* $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ *in time* $\|\mathcal{A}_{r,m}\|^{O(1)}$.

Proof: Let $r, m \in \mathbb{N}$. We say that a τ-structure $\mathcal{A}$ is (r, m)-*suitable* if for every $R \subseteq A^r$ there is a τ-structure $\mathcal{B}$ and $S \subseteq B^r$ such that

$$\mathcal{A} \ncong \mathcal{B} \quad \text{and} \quad (\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R).$$

The next two claims follow immediately from the definition of $\cong$-AF-sequence.

Claim 1: If $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ is an $\cong$-AF-sequence, then $\mathcal{A}_{r,m}$ is (r, m)-suitable for every $r, m \in \mathbb{N}$. $\dashv$

Claim 2: Let $\mathcal{A}$ be a τ-structure and $r, m \in \mathbb{N}$. Then $\mathcal{A}$ is (r, m)-suitable if and only if the following algorithm $\mathbb{C}$ halts on input $(\mathcal{A}, r, m)$:

```
ℂ     // 𝒜 a τ-structure and r, m ∈ ℕ
  1. for all R ⊆ A^r do
  2.         exhaustively search for a τ-structure ℬ and S ⊆ B^r
  3.                 such that 𝒜 ≇ ℬ and (ℬ, S) ≡⟩_{FO_m}(𝒜, R).
```

Moreover, if $\mathcal{A}$ is (r, m)-suitable, then for every $R \subseteq A^r$ there is a τ-structure $\mathcal{B}$ and $S \subseteq B^r$ such that $(\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}, R)$, and

$$m \leq |A|, |B| \leq t_{\mathbb{C}}(\mathcal{A}, r, m),$$

where $t_{\mathbb{C}}(\mathcal{A}, r, m)$ denotes the running time of the algorithm $\mathbb{C}$ on input $(\mathcal{A}, r, m)$. $\dashv$

Claim 3: Assume that there is an algorithm that generates an $\cong$-AF-sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ in time $\|\mathcal{A}_{r,m}\|^{O(1)}$. Then there is an algorithm which generates a monotone $\cong$-AF-sequence $(\mathcal{A}'_{r,m})_{r,m\in\mathbb{N}}$ in time $\|\mathcal{A}'_{r,m}\|^{O(1)}$, where here *monotone* means that for every $r, m \in \mathbb{N}$:

(i) $|A'_{r,m}| < |A'_{r,m+1}|$;

(ii) for every $R \subseteq A'^r_{r,m}$ there is a τ-structure $\mathcal{B}$ and $S \subseteq B^r$ such that $(\mathcal{B}, S) \equiv\rangle_{\mathrm{FO}_m} (\mathcal{A}'_{r,m}, R)$, and

$$|A'_{r,m}| \leq B < |A'_{r,m+1}|.$$

Proof of Claim 3: We set

$$\mathcal{A}'_{r,m} := \mathcal{A}_{r,\pi_r(m)},$$

where $\pi_r : \mathbb{N} \to \mathbb{N}$ is defined inductively by:

$$\begin{aligned} \pi_r(0) &:= 0 \\ \pi_r(m+1) &:= 1 + t_{\mathbb{C}}(\mathcal{A}_{r,\pi_r(m)}, r, \pi_r(m)). \end{aligned}$$

We leave it to the reader to verify that $(\mathcal{A}'_{r,m})_{r,m\in\mathbb{N}}$ is a *monotone* $\cong$-AF-sequence. $\dashv$

Now the proof essentially runs along the lines of that of Theorem 6.3. By contradiction, we assume that there is an algorithm $\mathbb{S}$ generating an $\cong$-AF-sequence $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ in time

$$\|\mathcal{A}_{r,m}\|^{O(1)}.$$

We can assume that the sequence is monotone. Again we denote by $<_{r,m}$ the ordering induced by the encoding of the structure $\mathcal{A}_{r,m}$ as output of $\mathbb{S}$.

We can turn $\mathbb{S}$ into a class of nondeterministic algorithms $(\mathbb{A}_r)_{r\in\mathbb{N}}$ such that for some $d \in \mathbb{N}$:

– For every $r \in \mathbb{N}$ the nondeterministic algorithm $\mathbb{A}_r$ accepts the class

$$K_r := \big\{\mathcal{A} \;\big|\; \mathcal{A} \cong \mathcal{A}_{r,m} \text{ for some } m \in \mathbb{N}\big\}.$$

– The running time of $\mathbb{A}_r$ is bounded by $O(n^d)$.

By Theorem 6.4, there are Σ^1_1-sentences

$$\varphi_r = \exists X \psi_r$$

with X of arity $2(d+4)$ and first-order ψ_r such that for all *sufficiently large* $\mathcal{A}$

$$\mathcal{A} \in K_r \iff \mathcal{A} \models \varphi_r. \tag{21}$$

We set $r_0 := 2(d+4)$ and choose m_0 so large that $\neg\psi_{r_0} \in \mathrm{FO}_{m_0}$ and that for $r = r_0$ the equivalence (21) holds for structures $\mathcal{A}$ with $|A| \geq m_0$.

In particular, as $\mathcal{A}_{r_0,m_0} \in K_{r_0}$ and $|A_{r_0,m_0}| \geq m_0$,

$$\mathcal{A}_{r_0,m_0} \models \exists X \psi_{r_0},$$

i.e., there exists an r_0-ary relation $R \subseteq A^{r_0}_{r_0,m_0}$ with

$$\big(\mathcal{A}_{r_0,m_0}, R\big) \models \psi_{r_0}. \tag{22}$$

By the property (ii) of $(\mathcal{A}_{r,m})_{r,m\in\mathbb{N}}$ (mentioned in Claim 3) we can choose a structure $\mathcal{B}$, $\mathcal{A}_{r_0,m_0} \ncong \mathcal{B}$, together with $S \subseteq B^{r_0}$ such that

$$\big(\mathcal{B}, S\big) \equiv\rangle_{\mathrm{FO}_{m_0}} \big(\mathcal{A}_{r_0,m_0}, R\big) \tag{23}$$

and

$$|A_{r_0,m_0}| \leq |B| < |A_{r_0,m_0+1}|.$$

Hence, $\mathcal{B} \notin K_{r_0}$. On the other hand, as $\neg\psi_{r_0} \in \mathrm{FO}_{m_0}$, we conclude, using (23) and (22), that $(\mathcal{B}, S) \models \psi_{r_0}$ and thus,

$$\mathcal{B} \models \exists X \psi_{r_0}.$$

Therefore, $\mathcal{B} \in K_{r_0}$ by (21), a contradiction. $\square$

7. Ehrenfeucht-Fraïssé games on random structures

We have seen (cf. Theorem 4.4 (b)) that for any logic L with $\mathrm{P} \leq L$ on ordered structures (that is, polynomial time properties of ordered structures are expressible in L), we cannot construct an ordered L-sequence efficiently. What happens if we consider random sequences? We deal with this question here.

So far, we already got our results without taking into consideration a further natural property we expect that an L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ fulfills if it should serve for the Ehrenfeucht-Fraïssé method; namely, we must be able to verify that $\mathcal{A}_m \equiv\rangle_{L_m} \mathcal{B}_m$ holds in a reasonable time (see (ii) in the Introduction). Condition (r3) in the following definition of random L-sequence takes care of this property. Throughout we fix a logic L with $\mathrm{P} \leq L$ on ordered structures and a vocabulary τ; all structures are assumed to be τ-structures.

Definition 7.1. We say that a probabilistic algorithm $\mathbb{P}$ generates a *random L-sequence* $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ if (r1)–(r3) are satisfied.

(r1) For every $m \in \mathbb{N}$ the algorithm $\mathbb{P}$ first *deterministically* computes the universes A_m and B_m, and then constructs the structures $\mathcal{A}_m$ and $\mathcal{B}_m$ probabilistically.

(r2) There is a polynomial time algorithm $\mathbb{I}$ such that:

- For all structures $\mathcal{A}$ and $\mathcal{B}$, if $\mathbb{I}$ accepts $(\mathcal{A}, \mathcal{B})$, then $\mathcal{A} \ncong \mathcal{B}$.
- For sufficiently large $m \in \mathbb{N}$, [7]

$$\Pr\left[\mathbb{I} \text{ accepts } (\mathcal{A}_m, \mathcal{B}_m)\right] \geq \frac{4}{5}.$$

(r3) There is an algorithm $\mathbb{E}$ such that:

- For all structures $\mathcal{A}$ and $\mathcal{B}$ and all $m \in \mathbb{N}$, if $\mathbb{E}$ accepts $(\mathcal{A}, \mathcal{B}, m)$, then $\mathcal{A} \equiv\rangle_{L_m} \mathcal{B}$.
- For sufficiently large $m \in \mathbb{N}$,

$$\Pr\left[\mathbb{E} \text{ accepts } (\mathcal{A}_m, \mathcal{B}_m, m)\right] \geq \frac{4}{5}.$$

- The running time of $\mathbb{E}$ on input $(\mathcal{A}, \mathcal{B}, m)$ is bounded by $f(m) \cdot (\|\mathcal{A}\| + \|\mathcal{B}\|)^{O(1)}$ for some computable function $f : \mathbb{N} \to \mathbb{N}$.

If we restrict to ordered structures, we can replace (r2) by the equivalent condition: For all sufficiently large $m \in \mathbb{N}$,

$$\Pr\left[\mathcal{A}_m \ncong \mathcal{B}_m\right] \geq \frac{4}{5}.$$

Furthermore, observe that the constant $4/5$ in (r2) and (r3) can be replaced by any constant ε with $0 < \varepsilon < 1$ using the standard amplification method by repetition. For example, for $\varepsilon = 1/2$, we repeat the algorithm $\mathbb{P}$ on input m three times thereby obtaining three pairs of structures. If one of them is accepted by both, $\mathbb{I}$ and $\mathbb{E}$, then output the first pair with this property; otherwise, choose an arbitrary one.

In this section we show:

Theorem 7.2. *Let L be a logic with $\mathrm{P} \leq L$ on ordered structures. Assume that there is a $2^{\lceil \ell/c \rceil}$-pseudorandom generator for some natural number $c \geq 1$. Then there is no probabilistic algorithm that generates a random ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$.*

The following is known: If, for the circuit size $\mathscr{H}_P$ of some problem $P \subseteq \{0,1\}^*$ with $P \in \mathrm{E}$, we have $\mathscr{H}_P(n) \notin 2^{o(n)}$ infinitely often (cf. Remark 5.8), then for some $c \geq 1$ we can construct a $2^{\lceil \ell/c \rceil}$-pseudorandom generator [11, 9]. For the reader's convenience we recall the definition of pseudorandom generator, following the presentation of [2].

Definition 7.3. 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

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

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

[7] Here and in the following we consider the probability space of the internal coin tosses of the algorithm $\mathbb{P}$ on input m with the uniform distribution.

The following lemmas will finally yield a proof of Theorem 7.2: Essentially we use the pseudorandom generator to derandomize the algorithm $\mathbb{P}$ in such a way that we obtain a deterministic algorithm which generates an ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$, which contradicts Theorem 4.4 (b).

We say that a random L-sequence is *strongly monotone* if for all $m \in \mathbb{N}$ we have:

- $\max\{|A_m|, |B_m|\} < \min\{|A_{m+1}|, |B_{m+1}|\}$
- $\lceil d \cdot \log(|A_m| + |B_m|)\rceil < \lceil d \cdot \log(|A_{m+1}| + |B_{m+1}|)\rceil$;
- $f(m) \leq \max\{|A_m|, |B_m|\}$ (where f is the computable function of (r3) used to bound the running time of $\mathbb{E}$).

Recall that the universes of the structures $\mathcal{A}_m$ and $\mathcal{B}_m$ of a random L-sequence are obtained deterministically. Therefore, arguing similarly as we obtained the Claim in the proof of Theorem 4.4, one gets:

Lemma 7.4. *If there is a probabilistic algorithm that generates a random ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$, then there is a probabilistic algorithm that generates a strongly monotone random ordered L-sequence $(\mathcal{A}'_m, \mathcal{B}'_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}'_m\| + \|\mathcal{B}'_m\|)^{O(1)}$.*

Lemma 7.5. *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 and random L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$.*

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

$$(\mathcal{A}_m^1, \mathcal{B}_m^1), \ldots, (\mathcal{A}_m^{t_m}, \mathcal{B}_m^{t_m})$$

of ordered structures, where all A_m^i have A_m as universe, that is, $A_m^i = A_m$ for $i \in [t_m]$ $(:= \{1, 2, \ldots, t_m\})$ and all B_m^i have B_m as universe (recall that A_m and B_m are the universes deterministically computed by $\mathbb{P}$ on input m). Moreover, the following conditions hold:

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

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

$$\Pr_{p\in[t_m]}\big[\mathcal{A}_m^p \ncong \mathcal{B}_m^p \text{ and } \mathcal{A}_m^p \equiv\rangle_{L_m} \mathcal{B}_m^p\big] \geq \Pr_{p\in[t_m]}\big[\mathbb{I} \text{ accepts } (\mathcal{A}_m^p, \mathcal{B}_m^p) \text{ and } \mathbb{E} \text{ accept } (\mathcal{A}_m^p, \mathcal{B}_m^p, m)\big] > \frac{1}{2},$$

where $\mathbb{I}$ and $\mathbb{E}$ are the algorithms associated with $\mathbb{P}$ and mentioned in (r2) and (r3) of Definition 7.1. Note that the first inequality is immediate by (r2) and (r3).

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

- $\max\{|A_m|, |B_m|\} < \min\{|A_{m+1}|, |B_{m+1}|\}$,
- $\lceil \log(|A_m| + |B_m|)\rceil < \lceil \log(|A_{m+1}| + |B_{m+1}|)\rceil$,
- $f(m) \leq \max\{|A_m|, |B_m|\}$ *(where f is the function of (r3)).*

Proof: For $\mathbb{P}$ choose algorithms $\mathbb{I}$ and $\mathbb{E}$ satisfying (r2) and (r3), respectively. By the assumptions, we know that there is $d \in \mathbb{N}$ such that:

- The running time of $\mathbb{P}$ on m is bounded by

$$\big(|A_m| + |B_m|\big)^d$$

(here we use that, for fixed vocabulary τ, we have $\|\mathcal{A}\| \leq |A|^{O(1)}$ for every τ-structure $\mathcal{A}$).

– The running time of $\mathbb{I}$ on structures $\mathcal{A}$ and $\mathcal{B}$ and the running time of $\mathbb{E}$ on inputs $(\mathcal{A}, \mathcal{B}, m)$ with $f(m) \leq \max\{|A|, |B|\}$ is bounded by

$$\big(|A| + |B|\big)^d.$$

By the strong monotonicity of $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ we already know that (a3) holds.

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

$\mathbb{A}$ // $m \in \mathbb{N}$ in unary

1. simulate the (deterministic) part of the computation of $\mathbb{P}$
2. on input m yielding the universes A_m and B_m
3. $n \leftarrow |A_m| + |B_m|$
4. $\ell \leftarrow c \cdot \lceil d \cdot \log n \rceil$
5. **for** all $s \in \{0,1\}^\ell$ **do**
6. compute $\mathbb{G}(s)$
7. simulate $\mathbb{P}$ on input m where in the simulation
8. the internal coin tosses of $\mathbb{P}$ are replaced according to $\mathbb{G}(s)$
9. output $(\mathcal{A}^s_m, \mathcal{B}^s_m)$, the output of this simulation of $\mathbb{P}$.

Of course, then (a1) holds and thus, it remains to establish (a2). So, by contradiction assume that

$$\text{for infinitely many } m \in \mathbb{N}\text{:} \quad \Pr_{p\in[t_m]}\big[\,\mathbb{I} \text{ accepts } (\mathcal{A}^p_m, \mathcal{B}^p_m) \text{ and } \mathbb{E} \text{ accepts } (\mathcal{A}^p_m, \mathcal{B}^p_m, m)\big] \leq \frac{1}{2}. \tag{24}$$

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

$$n_m := |A_m| + |B_m|.$$

Clearly there is an algorithm that, given $n \in \mathbb{N}$, decides in time $O(n^{2d})$ whether 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 $(\mathcal{A}_m, \mathcal{B}_m)$ of $\mathbb{P}$ on input m if
4. the internal coin tosses of $\mathbb{P}$ are replaced according to r
5. simulate $\mathbb{I}$ on $(\mathcal{A}_m, \mathcal{B}_m)$
6. **if** the simulation rejects **then** reject
7. simulate $\mathbb{E}$ on $(\mathcal{A}_m, \mathcal{B}_m, m)$
8. **if** the simulation rejects **then** reject
9. accept.

First note that by (24),

$$\text{for infinitely many } m \in \mathbb{N} \text{ and for } \ell := c \cdot \lceil d \cdot \log n_m\rceil\text{:} \quad \Pr_{s\in\{0,1\}^\ell}\big[\mathbb{D}(\mathbb{G}(s)) = 1\big] \leq \frac{1}{2}. \tag{25}$$

Moreover, as $f(m) \leq \max\{|A_m|, |B_m|\}$ (by the strong monotonicity of the random L-sequence computed by $\mathbb{S}$), we see that $\mathbb{D}$ is a polynomial time algorithm. Using the Cook-Levin's reduction, from the algorithm $\mathbb{D}$ we can construct, for every $m \in \mathbb{N}$ and $n^* := 2^{\lceil d\cdot\log\, n_m\rceil}$ $(\approx n^d_m)$, a circuit C_{n^*} such that for every

$r \in \{0,1\}^{n^*}$,

$$C_{n^*}(r) = 1 \quad \Longleftrightarrow \quad \mathbb{D}(r) = 1 \tag{26}$$

and

$$|C_{n^*}| = O\big((n^*)^2 + n_m^{2\cdot d}\big) = O\big((n^*)^2\big). \tag{27}$$

Thus, by (r2) and (r3) for sufficiently large $m \in \mathbb{N}$, and hence sufficiently large $n^* = 2^{\lceil d \cdot \log\, n_m \rceil}$,

$$\Pr_{r\in\{0,1\}^{n^*}}\big[C_{n^*}(r) = 1\big] = \Pr_{p\in[t_m]}\big[\mathbb{I} \text{ accepts } (\mathcal{A}_m^p, \mathcal{B}_m^p) \text{ and } \mathbb{E} \text{ accepts } (\mathcal{A}_m^p, \mathcal{B}_m^p, m)\big] \geq \frac{3}{5}. \tag{28}$$

By (25) and (26), we know that for infinitely many $m \in \mathbb{N}$ we have for $n^* = 2^{\lceil d\cdot\log\, n_m\rceil}$ and $\ell := c \cdot \lceil d \cdot \log\, n_m \rceil$,

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

Together with (28) we get for such n^* and ℓ,

$$\Big|\Pr_{r\in\{0,1\}^{n^*}}\big[C_{n^*}(r) = 1\big] - \Pr_{s\in\{0,1\}^{\ell}}\big[C_{n^*}(\mathbb{G}(s)) = 1\big]\Big| \geq \frac{3}{5} - \frac{1}{2} = \frac{1}{10},$$

which, by (27), contradicts (g2) in Definition 7.3. □

Proof of Theorem 7.2: Assume that there is a probabilistic algorithm that generates a random ordered L-sequence $(\mathcal{A}_m, \mathcal{B}_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}_m\| + \|\mathcal{B}_m\|)^{O(1)}$. We show that there is a deterministic algorithm which generates an ordered L-sequence $(\mathcal{A}'_m, \mathcal{B}'_m)_{m\in\mathbb{N}}$ in time $(\|\mathcal{A}'_m\| + \|\mathcal{B}'_m\|)^{O(1)}$. This contradicts Theorem 4.4 (b).

By Lemma 7.4, we have an algorithm $\mathbb{A}$ with the properties stated in Lemma 7.5. The following algorithm $\mathbb{S}$ generates an ordered L-sequence $(\mathcal{A}'_m, \mathcal{B}'_m)_{m\in\mathbb{N}}$ in the claimed time; note that by (a3) in Lemma 7.5, the algorithm $\mathbb{E}$ is applied to inputs $(\mathcal{A}, \mathcal{B}, m)$ with $f(m) \leq \max\{|A|, |B|\}$; on such inputs its running time is bounded by $(\|\mathcal{A}\| + \|\mathcal{B}\|)^{O(1)}$.

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

1. simulate $\mathbb{A}$ on input m to compute $(\mathcal{A}_m^1, \mathcal{B}_m^1), \ldots, (\mathcal{A}_m^{t_m}, \mathcal{B}_m^{t_m})$
2. **for all** $i \in [t_m]$ **do**
3. simulate $\mathbb{I}$ on $(\mathcal{A}_m^i, \mathcal{B}_m^i)$ and $\mathbb{E}$ on $(\mathcal{A}_m^i, \mathcal{B}_m^i, m)$
4. **if** both simulations accept **then** output $(\mathcal{A}_m^i, \mathcal{B}_m^i)$ as $(\mathcal{A}'_m, \mathcal{B}'_m)$ and halt

By (a2) in Lemma 7.5, the algorithm $\mathbb{S}$ will halt on input m and yield the desired $(\mathcal{A}'_m, \mathcal{B}'_m)$. □

8. Appendix

In the Introduction we mentioned some concrete successful applications of the Ehrenfeucht-Fraïssé method, cf. (I1)–(I3) and (H1)–(H3). Often it is straightforward to verify the conditions (i) and (ii) mentioned on page 1. The purpose of this Appendix is to help the reader to verify them in the cases (I3) and (H2). We emphasize that we want to show the conditions for the concrete structures considered in the corresponding applications. Sometimes, one can achieve these conditions by a simple trick, namely, by artificially enlarging structures. We explain this trick at the end of the first example.

Graph connectivity. Let Q be the connectivity property of ordered graphs. In [15], Schwentick shows that Q is not expressible in mon-Σ^1_1. For this purpose he presents an ordered Q-mAF-sequence $(G_{r,m})_{r,m\in\mathbb{N}}$ [8]; thereby he considers the filtering of FO adequate for the classical Ehrenfeucht-Fraïssé-game. So all $G_{r,m}$ are connected ordered graphs. It is easy to verify that there is an algorithm generating this sequence in time $\|G_{r,m}\|^{O(1)}$. For every selection $R_1,\dots,R_r$ of unary relations on $G_{r,m}$, the author shows the existence of an ordered graph $H_{r,m}$, which is not connected and has the same vertex set as $G_{r,m}$ and the existence of unary relations $S_1,\dots,S_r$ such that $(G_{r,m},R_1,\dots,R_r)\equiv_{\mathrm{FO}_m}(H_{r,m},S_1,\dots,S_r)$. We sketch why the verification of this FO$_m$-equivalence is possible in time $f(m)\cdot(\|G_{r,m}\|+\|H_{r,m}\|)^{O(1)}$ (thus satisfying the property (ii) of the Introduction). Both, $G_{r,m}$ and $H_{r,m}$ consist of "columns" $C_1,\dots,C_s$, where

$$n:=|C_1|=\cdots=|C_s|<\log\log\|G_{r,m}\|=\log\log\|H_{r,m}\|.$$

If in both structures one replaces the ordering relation by its successor relation, then one gets structures, which have a path decomposition whose bags along the path consist of the vertices in the columns C_1,C_2,C_3, the vertices in the columns $C_1,C_3,C_4,\dots$, and the vertices in the columns C_1,C_{s-1},C_s. As the ordering is definable in monadic second-order logic from the successor relation, we obtain our claim by applying (the extension of) Courcelle's Theorem for monadic second-order logic $\big($to structures of treewidth $O(\log\log n)\big)$.

We sketch how one could satisfy (i) and (ii) once one knows that an ordered Q-mAF-sequence $(G_{r,m})_{r,m\in\mathbb{N}}$ exists. We can assume that there is an algorithm that generates $(G_{r,m})_{r,m\in\mathbb{N}}$ (by a brute force algorithm that takes into consideration all structures and all expansions by unary relations in a diagonal fashion). Now by adding a sufficiently large path "at the end of the ordering" to $G_{r,m}$, say a path consisting of $i(r,m)$ many points, one obtains an ordered graph $G'_{r,m}$ in time $\|G'_{r,m}\|^{O(1)}$ such that the treewidth of $G''_{r,m}$ is bounded by log log $\|G'_{r,m}\|$, where $G''_{r,m}$ is obtained from $G'_{r,m}$ by replacing the ordering by its successor relation. The number $i(r,m)$ should be so big that once Spoiler has selected unary relations $R_1,\dots,R_r$ on $G_{r,m}$, Duplicator can choose an ordered graph $H_{r,m}$ such that he wins the Q-mAF-game and such that by adding a path consisting of $i(r,m)$ many points at the end of the ordering to $H_{r,m}$ one gets an ordered graph $H'_{r,m}$, whose corresponding $H''_{r,m}$ has a treewidth bounded by log log $\|H'_{r,m}\|$. Now we can argue as above.

The arity hierarchy. In [8], Grohe shows, among others, that the arity hierarchy for the logics LFP, IFP, and PFP are strict, the arity of a formula being the maximal arity of a relation variable occurring in this formula. For this purpose he defines, for every $k\geq 2$, a query Q_k expressing that a certain k-tuple of vertices of a graph is not in the transitive closure of a relation definable by a quantifier-free formula. Let us write $\mathcal{A}\equiv_{\mathrm{PFP}_{k,m}}\mathcal{B}$ if $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of PFP of arity at most k and "quantifier rank" at most m. Grohe introduces a game of Ehrenfeucht-Fraïssé type which allows to verify that $\mathcal{A}\equiv_{\mathrm{PFP}_{k,m}}\mathcal{B}$. Now he constructs (essentially) graphs $G_{k,m}\in Q_k$ and $H_{k,m}\notin Q_k$ and verifies with this game that $G_{k,m}\equiv_{\mathrm{PFP}_{k-1,m}}H_{k,m}$. Here we want to sketch that these graphs can be constructed in time polynomial in their size. This is not obvious as they are obtained from bigger structures by a factorization process. As the graphs $G_{k,m}$ and $H_{k,m}$ only differ in a minor point, we concentrate on the construction of $G_{k,m}$. We address a reader familiar with the paper [8]

The starting point of Grohe's construction is a graph $A:=A(k,m)$ with universe $\{1,\dots,m\}\times\{-k,\dots,-1,1,\dots,k\}$; thus $|A|=2k\cdot m$. For every $I_1,\dots,I_{k-1}\in[m]$ and $a\in[k]$ a partial bijection $p(I_1,\dots,I_{k-1},a)$ of A is defined; thus in total $\ell:=k\cdot m^{k-1}$ many partial bijections $p_1,\dots,p_\ell$ are introduced. [9] Let $\Gamma:=(\mathbb{Z}_2,+)^\ell$ be the product of ℓ copies of the group $(\mathbb{Z}_2,+)$. Note that $|A\times\Gamma|=2k\cdot m\cdot 2^{k\cdot m^{k-1}}$. The elements of $A\times\Gamma$ are written in the form (I,a,γ) where $(I,a)\in A$ and $\gamma\in\Gamma$. Grohe defines an equivalence relation $\sim$ on $A\times\Gamma$. The vertex set $V(G_{k,m})$ of the graph $G_{k,m}$ consists of the equivalence classes of $\sim$ and indeed has a size decisively smaller than that of $A\times\Gamma$. Step by step one can prove the following claims.

[8] See Section 6.1 for the definition of Q-mAF-sequence

[9] Actually, the partial bijection $p(I_1,\dots,I_{k-1},a)$ only depends on a and the set $\{I_1,\dots,I_{k-1}\}$; but the following arguments are the same as for the notationally simpler case with $\ell=k\cdot m^{k-1}$ considered here.

Claim 1: For fixed k there is an algorithm deciding whether two tuples are equivalent in time polynomial in m. ⊣

We say that the bijection $p_j = p(I_1, \ldots, I_{k-1}, a)$ is *critical for* $I \in [m]$, if $|I - I_j| \leq 1$ for some $j \in [k-1]$. We set

$$\text{critic}(I) := \{j \in [\ell] \mid p_j \text{ is critical for } I\}.$$

Claim 2: Let $I \in [m]$. Then $|\text{critic}(I)| = \Theta\big(k^2 \cdot m^{k-2}\big)$. ⊣

Recall that in [8], for $\gamma \in \Gamma$ the *support* $\text{supp}(\gamma)$ *of* γ is defined by

$$\text{supp}(\gamma) := \big\{i \in [\ell] \;\big|\; \text{the } i\text{-th component of } \gamma \text{ is } 1\big\}.$$

Claim 3: Let $I \in [m]$, $a \in \{-k, \ldots, -1, 1, \ldots, k\}$, and $\gamma \in \Gamma$. Then for the unique $\delta \in \Gamma$ with

$$\text{supp}(\delta) = \text{supp}(\gamma) \cap \text{critic}(I)$$

we have

$$(I, a, \gamma) \sim (I, a, \delta).$$

Hence, $|V(G_{k,m})| \leq 2 \cdot k \cdot m \cdot 2^{\Theta(k^2 \cdot m^{k-2})}$. ⊣

Thus, by Claim 1 and Claim 3 we can construct $V(G_{k,m})$ in time polynomial in the expression on the right hand side of the last statement and thus in the desired time, once we have shown that we can replace $\leq$ by $=$ there. For this purpose we say that the bijection $p_j = p(I_1, \ldots, I_{k-1}, a)$ is *undefined for* $I \in [m]$ if $|I - I_j| = 1$ for some $j \in [k-1]$. We set

$$\text{undef}(I) := \{j \in [\ell] \mid p_j \text{ is undefined for } I\}.$$

Claim 4: For every $I \in [m]$ we have $\text{undef}(I)| = \Theta(k^2 \cdot m^{k-2})$. ⊣

For the last statement in the next claim we also need the fact shown in [8] that $(I, a, \sigma) \not\sim (J, b, \delta)$ if $I \neq J$.

Claim 5: Let $I \in [m]$ and $a, b \in \{-k, \ldots, -1, 1, \ldots, k\}$. Then for distinct $\gamma, \delta \in \Gamma$ with

$$\text{supp}(\gamma), \text{supp}(\delta) \subseteq \text{undef}(I)$$

we have

$$(I, a, \sigma) \not\sim (I, b, \delta).$$

Hence, $|V(G_{k,m}|)| = 2 \cdot k \cdot m \cdot 2^{\Theta(k^2 \cdot m^{k-2})}$. ⊣

It remains to be shown that the edge relation of the graph can be defined in time polynomial in the size of $V(G_{k,m})$. The reader will easily convince himself that this relation can be defined in this time using the following claim.

Claim 5: Assume $(I, a, \gamma) \sim (I, a', \xi)$ and $(J, b, \delta) \sim (J, b', \xi)$ with $\text{supp}(\gamma) \subseteq \text{critic}(I)$ and $\text{supp}(\delta) \subseteq \text{critic}(J)$. Then there exists a $\xi' \in \Gamma$ such that

$$(I, a, \gamma) \sim (I, a', \xi'), \;\; (J, b, \delta) \sim (J, b', \xi') \;\text{ and }\; \text{supp}(\xi) \subseteq \text{critic}(I) \cup \text{critic}(J).$$

⊣

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] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[3] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the P=?NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[4] Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identifications. *Combinatorica*, 12(4):389–410, 1992.

[5] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[6] Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On monadic NP vs. monadic co-NP. *Information and Computation*, 120(1):78–92, 1995.

[7] Roland Fraïssé. Sur quelques classifications des systèmes de relations. *Université d'Alger, Publications Scientifiques, Série A*, 1:35–182, 1954.

[8] Martin Grohe. Arity hierarchies. *Annals of Pure and Applied Logic*, 82(2):103–163, 1996.

[9] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC'97)*, pages 220–229, 1997.

[10] Max Kubierschky. Yet another hierarchy theorem. *The Journal of Symbolic Logic*, 65(2):627–640, 2000.

[11] Noam Nisan and Avi Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[12] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[13] Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[14] 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.

[15] 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.

[16] Moshe Y. Vardi. On the complexity of bounded-variable queries. In *Proceedings of the 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'95)*, pages 266–276, 1995.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de