A Game for Counting Logic Formula Size and an Application to Linear Orders

Grégoire Fournier
University of Illinois at Chicago, IL, USA

György Turán
University of Illinois at Chicago, IL, USA
HUN-REN-SZTE Research Group on AI, Szeged, Hungary

Abstract

Ehrenfeucht–Fraïssé (EF) games are a basic tool in finite model theory for proving definability lower bounds, with many applications in complexity theory and related areas. They have been applied to study various logics, giving insights on quantifier rank and other logical complexity measures. In this paper, we present an EF game to capture formula size in counting logic with a bounded number of variables. The game combines games introduced previously for counting logic quantifier rank due to Immerman and Lander, and for first-order formula size due to Adler and Immerman, and Hella and Väänänen. The game is used to prove an extension of a formula size lower bound of Grohe and Schweikardt for distinguishing linear orders, from 3-variable first-order logic to 3-variable counting logic.

2012 ACM Subject Classification Theory of computation → Finite Model Theory; Theory of computation → Models of learning; Theory of computation → Complexity theory and logic

Keywords and phrases Finite Model Theory, Logical Aspects of Computational Complexity

Digital Object Identifier 10.4230/LIPIcs.CSL.2026.36

Related Version *Full Version*: `https://arxiv.org/abs/2505.16185`

Funding Support from grants NSF 2217023 and NSF 2240532 is acknowledged. Support from Project 2024-1.2.3-HU-RIZONT-2024-00017 is acknowledged, financed by the Ministry of Culture and Innovation of Hungary from the National Research, Development and Innovation Fund, under the a 2024-1.2.3-HU-RIZONT funding scheme.

1 Introduction

Ehrenfeucht–Fraïssé (EF) games [7, 11] are a basic tool of finite model theory for proving definability lower bounds [6, 26]. Combined with logical characterizations of complexity classes, they provide a logic-based approach to problems in complexity theory. The original form of EF games gives bounds for quantifier rank in first-order logic. The games have been extended and modified for many logics and formula complexity measures. An EF game for *formula size* in first-order logic (FO) is given by Hella and Väänänen [19], building on Adler and Immerman [1]. In what follows, we refer to this game as the HV-game.

Counting logic extends first-order logic by adding the counting quantifier $\exists^{\geq k}$, and is frequently used in complexity theory and combinatorics. Counting logic turns out to be relevant for understanding the computational power of graph neural networks (GNN) as well [12]. An EF game for distinguishing graphs in counting logic with a bounded number of variables is formulated by Immerman and Lander [22]. It extends the basic EF setup by an additional phase in each round involving the choice of subsets of the same cardinality in the two structures.

In this paper, we formulate a game for capturing formula-size complexity for counting logic. The game is a combination of the Immerman-Lander and Hella-Väänänen games. Restricted versions characterize formula size for fragments of counting logic where the number

of variables and the counting rank are bounded. The game for formulas with a bounded number of variables could be a useful tool for exploring the implications of the logical characterization of GNNs (Barceló et al. [2]).

While the standard EF game is played on two structures, the HV game is played on two *sets of structures*, referred to in this paper as *families*. Grohe and Schweikardt [14] prove a formula size lower bound for linear orders for the 3-variable fragment of first-order logic, using this technique implicitly. They show that every first-order 3-variable formula that distinguishes a linear order of size n from a larger one has size $\Omega(\sqrt{n})$. Their proof is based on the notion of a *separator* and an involved weighting technique, which allows for a refined analysis of the syntax tree of a formula, by gauging the progress made in subformulas of a distinguishing formula. The proof is a detailed case analysis, with numerous subcases required to deal with quantifiers.

The formula size characterization is given in Theorem 10. The main result of the paper is that every 3-variable counting logic formula with counting rank t distinguishing a linear order of size n from a larger one has size at least $\sqrt{n}/t$ (Theorem 16). This result extends the result of [14] from first-order logic to counting logic. The theorem is proved using the game characterization. There is a simple distinguishing formula of size n/t (Proposition 15). For the case $t = 1$, i.e., for 3-variable FO, our result improves the formula size lower bound of [14] from $\sqrt{n}/2$ to $\sqrt{n}$.

The proof of Theorem 16 adapts the framework of [14]. The main difference is in the most technical part, the quantifier case. We present the main lemma dealing with this case, with proofs of lemmas used in this proof given in the appendix. Other lemmas used in the lower bound are stated without proof due to lack of space.

The paper is structured as follows. After reviewing related work in Section 2, we describe the counting and HV games in Section 3. Section 4 describes the game and Section 5 gives the correspondence between the game and counting logic formula size. Section 6 contains the application on linear orders, with the proof of the lower bound in Section 7. The proof of the main lemma on counting quantifiers is presented separately in Section 7.3 and additional details are given in Appendix C. Section 8 contains further remarks.

2 Related Work

Counting logic has been discussed in several different forms. Grohe [12] defines counting logic $\mathcal{C}$ as first-order logic (FO) extended by counting quantifiers of the form $\exists^{\geq k} x$, and $\mathcal{C}_m$ as its fragment using at most m variables. This is the counting logic we consider in this paper. Previous work using this kind of counting logic includes Immerman and Lander [22] and Cai et al. [3]. Grohe [13], on the other hand, considers a more powerful counting logic, where formulae can include arithmetic operations on the number of elements satisfying a formula (see also Kuske and Schweikardt [25]).

EF games using sets of structures, capturing the number of quantifiers as opposed to quantifier rank, have been proposed by Immerman [20]. These *multi-structural (MS)* games receive increasing current attention (Fagin et al. [10], Carmosino et al. [4], Vinall-Smeeth [32]). HV games are essentially extensions of MS games, also modelling Boolean connectives in the formulae.

A graph neural network (GNN) is a variant of neural networks for machine learning problems involving graphs [31]. Such a network allows the use of deep learning techniques to classify graphs (graph classification), or to classify the nodes of a large graph (node classification). The computational power of GNN is closely related to the Weisfeiler–Leman

(WL) graph isomorphism algorithm (Morris et al. [27], Xu et al. [33]). The connection of the WL algorithm to counting logic with a bounded number of variables [3, 12] brings these logics into the GNN picture as well. Barceló et al. [2] gave logical characterizations in terms of counting logic with a bounded number of variables using results established in modal logic (Otto [29]). The complexity aspects of the characterizations are not discussed in [2], and studying this aspect (also pointed out in Grohe [12]) has been a motivation for the topic of this paper (a brief further discussion is given at the end of the paper).

The GNN characterizations of Grohe [13] establish a connection of GNN to threshold circuits, a Boolean circuit model of neural networks. The computational power of such circuits corresponds to counting logic with an arbitrary built-in predicate. Proving superpolynomial lower bounds for threshold circuits is an open problem. Hajnal et al. [15, 16] prove an exponential lower bound for depth-2 circuits with polynomial weights. The same papers prove a quantifier rank lower bound for counting logic with successor as the built-in relation. Similar results are also given in Etessami [8, 9]. Karchmer and Wigderson [24] formulate an approach, related to HV games, to proving monotone formula depth lower bounds. They also prove a depth version of the Krapchenko formula size lower bound. Krapchenko's Theorem is proved in [19] as an application of HV games.

General background for the topic of this paper is given in Immerman [21], Ebbinghaus and Flum [6], Libkin [26], Otto [28] and Hamilton [17].

3 Background

In this section we introduce basic notation used in the paper and review EF and HV games.

3.1 Basic Definitions

3.1.1 Logics

We consider relational structures over a fixed vocabulary. Counting logic $\mathcal{C}$ is obtained by extending first-order logic with counting quantifiers $\exists^{\geq k} x\, \phi(x)$ and $\forall^{\geq k} x\, \phi(x)$. Here $\exists^{\geq k} x \phi(x)$ means that there are at least k distinct assignments to the variable x that satisfy ϕ. Thus $\exists^{\geq k}\, \phi(x)$ is logically equivalent to $\exists x_1 \ldots \exists x_k (\bigwedge_i \phi(x_i) \wedge \bigwedge_{i,j} x_i \neq x_j)$. The quantifier $\forall^{\geq k} x \phi(x)$ stands for $\neg \exists^{\geq k} x \neg \phi(x)$. In $\exists^{\geq k} x \phi(x)$ and $\forall^{\geq k} x \phi(x)$, k is referred to as the *counting rank* of the quantifier. The counting rank of a formula is the maximum counting rank of its quantifiers.

As a counting quantifier can be replaced by standard quantifiers, adding counting quantifiers does not change the expressivity of first-order logic. It does, however, impact the succinctness, the minimum size of formulae expressing a property. In applications to finite model theory one usually considers a sequence $(\mathcal{A}_n, \mathcal{B}_n)$ of pairs of structures. Complexity bounds to be proven are also functions of n. Note that $\forall^{\geq k} x\, \varphi(x)$ is equivalent to $\exists^{\geq n-k+1} x\, \varphi(x)$ for an n-element structure. The transformation increases counting rank and thus it cannot be used in formula size bounds for the bounded counting rank case.

Parameters to be considered are the bounds m on the number of variables, t on the counting rank, and w on formula size. The fragment of $\mathcal{C}$ containing at most m variables is denoted by $\mathcal{C}_m$, and by $\mathcal{C}_m^t$ if, in addition, counting rank is at most t.

3.1.2 Structures and Families

The universe of a structure $\mathcal{A}$ is denoted by $U^{\mathcal{A}}$. We use x_j, $j \in \mathbb{N}$, to denote variables. A variable assignment for a structure $\mathcal{A}$ is a finite partial mapping $\alpha : \mathrm{Var} \to U^{\mathcal{A}}$, where Var is the set of variables. The finite domain of α is denoted by $dom(\alpha)$.

An *interpretation* is a pair $(\mathcal{A}, \alpha)$. For a formula ϕ, $(\mathcal{A}, \alpha) \models \phi$ means that the assignment α satisfies the formula ϕ in the structure $\mathcal{A}$, with $dom(\alpha)$ containing all the j for which the variable x_j is free in ϕ. A formula ϕ *distinguishes* interpretations $(\mathcal{A}, \alpha)$ and $(\mathcal{B}, \beta)$, denoted by $((\mathcal{A}, \alpha), (\mathcal{B}, \beta)) \models \phi$, if $(\mathcal{A}, \alpha) \models \phi$ and $(\mathcal{B}, \beta) \models \neg\phi$.

A *family* $A_{\mathcal{A},D}$ is a set of interpretations $\{(\mathcal{A}, \alpha_i) : i \in \Gamma\}$ where $\mathcal{A}$ and $D = dom(\alpha_i)$ for every i and Γ is some set. When the context is clear, we drop the subscript. We write $(A, B) \models \phi$ to express that for all $(\mathcal{A}, \alpha) \in A$ $(\mathcal{B}, \beta) \in B$ it holds that $((\mathcal{A}, \alpha), (\mathcal{B}, \beta)) \models \phi$, and we say that ϕ distinguishes (A, B). For a structure $\mathcal{A}$, we denote by A_0 the family $\{(\mathcal{A}, \emptyset)\}$, containing a single interpretation with the empty assignment.

3.1.3 Operations

If α is an assignment on $\mathcal{A}$, $a \in \mathcal{U}^{\mathcal{A}}$ and $j \in \mathbb{N}$, then $\alpha(a/j)$ is the assignment that maps x_j to a and agrees with α otherwise.

Given a family A, a *choice function* is of the form $F : A \to \mathcal{U}^{\mathcal{A}}$. The set of all choice functions on the family A is denoted by F_A. We define two operations on families [19].

- *Change*: Given a family A, a choice function $F \in F_A$ and $j \in \mathbb{N}$, the change operation on A with F for the variable x_j produces the family

$$A(F/j) := \{(\mathcal{A}, \alpha(F(\mathcal{A}, \alpha)/j)) : (\mathcal{A}, \alpha) \in A\}.$$

  In the new family, the assignment to x_j is changed based on the choice function F. If $j \notin D$ then x_j is a new variable, and D is updated to $D \cup \{j\}$. Thus a change operation may either leave the domain unchanged or add a new element to it.
- *Multiplication*: Given a family A and $j \in \mathbb{N}$, the multiplication operation A for the variable x_j produces the family

$$A(*/j) := \{(\mathcal{A}, \alpha(a/j)) : (\mathcal{A}, \alpha) \in A, a \in \mathcal{U}^{\mathcal{A}}\}.$$

  The new family consists of interpretations with x_j assigned to all possible values in $\mathcal{U}^{\mathcal{A}}$. Here, again, the domain is either unchanged or a new element is added to it.

3.1.4 Formula Complexity

The size of a formula is defined inductively: if ϕ is an atomic formula, $|\phi| = 1$; and for formulae ϕ, ψ, $|\neg\phi| = |\phi| + 1$; $|\phi \vee \psi| = |\phi \wedge \psi| = |\phi| + |\psi|$; $|\exists^{\geq k} x_j \phi| = |\forall^{\geq k} x_j \phi| = 1 + |\phi|$.

3.2 Review of Games

In this section we review the counting logic game [22] and the first-order logic formula size game [19], referred to as the HV game.

▶ **Definition 1** (r-round IL m-pebbling game)**.** *The game* $\boldsymbol{IL}(\mathcal{A}, \mathcal{B})$ *is played on two relational structures* $\mathcal{A}$ *and* $\mathcal{B}$*. There are two players, Spoiler and Duplicator, and* m *pairs of pebbles* (a_i, b_i) *for* $i \in [m]$*. It goes as follows:*

- *For* r *rounds:*
  - *Spoiler picks a set of elements* S_1 *of* $\mathcal{A}$ *or* $\mathcal{B}$ *and a number* $i \in [m]$*. Duplicator selects a set* S_2 *of elements of the same cardinality in the other structure.*
  - *Spoiler picks an element in* S_2 *and Duplicator selects an element in* S_1*. The pebble* a_i *(resp.,* b_i*) holds the value of the element picked in* $\mathcal{A}$ *(resp.,* $\mathcal{B}$*).*

- *The r-round game ends in the position $\vec{a} = (a_1, \ldots, a_m)$, $\vec{b} = (b_1, \ldots, b_m)$. Duplicator wins if the mapping from $((\vec{a}, \vec{c}^{\mathcal{A}})$ to $(\vec{b}, \vec{c}^{\mathcal{B}}))$ is a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$, where $\vec{c}$ denotes the constants of the language.*

▶ **Theorem 2.** *The following are equivalent:*

- *$\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of $\mathcal{C}_m$ of quantifier rank at most r.*
- *Duplicator has a winning strategy in the r-round m-pebbling game.*

A similar result holds for the bounded counting rank fragment $\mathcal{C}_m^t$, by restricting the cardinality of the sets picked to be at most t.

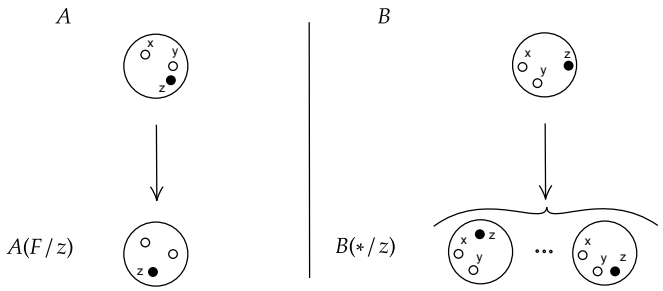

Figure 1 Two families, $A = \{(\mathcal{A}, \alpha)\}$ and $B = \{(\mathcal{B}, \beta)\}$, for the $\exists z$ move in the HV-game.

▶ **Definition 3** (HV game for formula size on first-order logic)**.** *The game $\boldsymbol{HV}_w(A, B)$ is played on two families, A and B, by Spoiler and Duplicator, for a positive integer w. The initial position is (w, A, B). Spoiler chooses one of five possibilities for the continuation of the game:*

- *$\neg$-move: the game continues from the position $(w-1, B, A)$.*
- *$\bigvee$-move: Spoiler chooses $1 \leq u, v < w$ such that $u + v = w$ and partitions A to get $C \cup D$. Duplicator sets the next position as (u, C, B) or (v, D, B), from which the game goes on.*
- *$\bigwedge$-move: similar but played on B.*
- *$\exists$-move: Spoiler chooses $j \in \mathbb{N}$ and a choice function F from F_A. Then the game continues from $(w-1, A(F/j), B(*/j))$.*
- *$\forall$-move: similar but Spoiler chooses on B.*

Spoiler wins if the game reaches a position (w, A, B) for $w \geq 1$ and there is an atomic formula that distinguishes A and B. Duplicator wins if the game reaches a position $(1, A, B)$ and Spoiler does not win.

▶ **Theorem 4.** *Let (A, B) be a pair of families, and let w be a positive integer. Then the following are equivalent:*

1. *Spoiler has a winning strategy in the game $\boldsymbol{HV}_w(A, B)$.*
2. *There is a formula ϕ of FO of size $|\phi| \leq w$ such that $(A, B) \models \phi$.*

Note that for any $m \in \mathbb{N}$, one can define the variant of the game $\mathbf{HV}_w^m(A, B)$, for which the moves $\exists$ and $\forall$ are restricted by $j \in [m]$. This game then characterizes FO_m, the fragment of FO logic, with m variables. An illustration of the $\exists$-move is given in Fig. 1.

In Table 1, we summarize the game characterizations discussed above and the game we are about to introduce.

Table 1 Complexity characterizing games.

	FO_m	$\mathcal{C}_m$
Quantifier Rank	[11], [7]	$\mathbf{IL}_m$ [22]
Size	$\mathbf{HV}_m$ [18]	$\mathbf{CS}_m$ (new)

4 The Counting Logic Formula Size Game

In this section we define the game for counting logic formula size. We start by extending the operations of the HV game to this setting.

4.1 Extended Operations

▶ **Definition 5** (k-choice function). *Given a family A, a* k-choice function *is of the form $F^k = (F_1^k, F_2^k, \ldots, F_k^k)$, where each F_i^k is a choice function, and for every $(\mathcal{A}, \alpha) \in A$ the elements $F_i^k(\mathcal{A}, \alpha)$ are pairwise distinct. The set of all the k-choice functions on the family A is denoted by F_A^k.*

▶ **Definition 6** (selection). *A mapping $S : F_A^k \to F_A$ is a selection if $S(F^k)(\mathcal{A}, \alpha) \in \{F_1^k(\mathcal{A}, \alpha), \ldots, F_k^k(\mathcal{A}, \alpha)\}$ for every $(\mathcal{A}, \alpha) \in A$. The set of all the selection functions of F_A^k is denoted by S_A^k.*

The extended set of operations is the following:

- *k-Change*: Given a family A and $j, k \in \mathbb{N}$, the k-change operation associated to $F^k \in F_A^k$ produces the family

$$A(F^k/j) := \{(\mathcal{A}, \alpha(F_i^k(\mathcal{A}, \alpha)/j)) : (\mathcal{A}, \alpha) \in A, 1 \leq i \leq k\}.$$

  Thus each interpretation gives rise to k interpretations composing the new family, where the assignments to x_j are changed based on the k-choice function F^k.
- *k-Multiplication*: Given a family A and $j, k \in \mathbb{N}$, the k-multiplication operation associated to the selection $S \in S_A^k$ produces the family

$$A(*_S^k/j) := \{(\mathcal{A}, \alpha(S(F^k)(\mathcal{A}, \alpha)/j)) : (\mathcal{A}, \alpha) \in A, F^k \in F_A^k\}.$$

  To avoid overloading the notation, we will keep the S implicit with the notation $A(*^k/j)$, and use the term "selects" when referring to the action of S.
  In plain language, for every $F^k \in F_A^k$ and $(\mathcal{A}, \alpha) \in A$, one interpretation is selected from $\{(\mathcal{A}, \alpha(F_i^k(\mathcal{A}, \alpha)/j)) : 1 \leq i \leq k\}$ to be part of $A(*^k/j)$. Thus each interpretation generates $card(F_A^k)$ interpretations that compose the family $A(*^k/j)$.

4.2 The CS Game

We now define the game for counting logic formula size (referred to as the **CS** game for "counting formula size"). It is presented in the $\mathcal{C}_m$ version, as this will be used in the rest of the paper. The new $\exists^{\geq k}$-move is illustrated in Fig. 2.

▶ **Definition 7** (CS game for formula size on counting logic). *The game* $\boldsymbol{CS}_w^m(A, B)$ *has two players, Spoiler and Duplicator,* m *is the number of variables.* A, B *are two families with* $dom(A) = dom(B)$ *of size at most* m*. Suppose after* p *moves we reach the position* (w, A, B)*. Depending on Spoiler's choice, the game continues as follows:*

- $\neg$*-move: the game continues from the position* $(w-1, B, A)$*.*
- $\bigvee$*-move: Spoiler first chooses numbers* u *and* v *such that* $1 \leq u, v < w$ *and* $u + v = w$*. Then Spoiler partitions* A *into a pair of families* C *and* D*. The game continues either from the position* $(u,\ C,\ B)$ *or from the position* $(v,\ D,\ B)$ *according to Duplicator's choice.*
- $\bigwedge$*-move: Spoiler first chooses numbers* u *and* v *such that* $1 \leq u, v < w$ *and* $u + v = w$*. Then Spoiler partitions* B *into a pair of families* C *and* D*. The game continues either from the position* $(u,\ A,\ C)$ *or from the position* $(v,\ A,\ D)$ *according to Duplicator's choice.*
- $\exists^{\geq k}$*-move: Spoiler chooses* $j \in [m], k \in \mathbb{N}$ *and a* k*-choice function* F^k *on* A*. For every* k*-choice function* G^k *on* B*, Spoiler selects* [1] G *from* G^k*. The union of the* $B(G/j)$ *over* $G^k \in F_B^k$ *forms* $B(*^k/j)$*. Then the game continues from the position* $(w-1,\ A(F^k/j),\ B(*^k/j))$*.*
- $\forall^{\geq k}$*-move: Spoiler chooses* $j \in [m], k \in \mathbb{N}$ *and a* k*-choice function* F^k *on* B*. For every* k*-choice function* G^k *on* A*, Spoiler selects* G *from* G^k*. The union of the* $A(G/j)$ *over* $A^k \in F_A^k$ *forms* $A(*^k/j)$*. Then the game continues from the position* $(w-1,\ A(*^k/j),\ B(F^k/j))$*.*

(Atomic) The game ends in a position (w, A, B) *if either there is an atomic formula* ϕ *such that* $(A, B) \models \phi$*, in which case Spoiler wins, or, otherwise if* $w = 1$ *Duplicator wins if there is no such* ϕ*.*

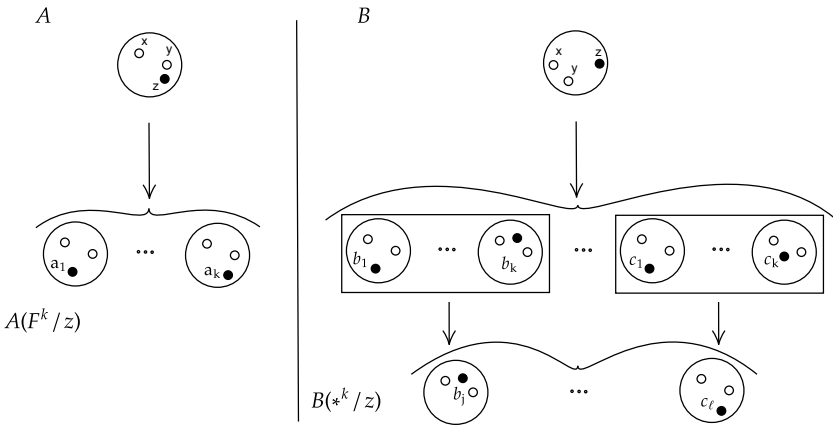

Figure 2 The $\exists^{\geq k}$-move in the CS game. On the left the k-Change operation: k different elements are chosen by Spoiler. On the right the two steps of the k-Multiplication operation. The first step is to form all k-choice functions (each denoted by a box). In the second step Spoiler picks one interpretation in each box to compose the new family.

[1] Note that formally a selection S chosen by Spoiler is considered here as in Definition 6.

In the application to linear orders we use the version of the game for bounded counting rank.

▶ **Definition 8** (CS game for $\mathcal{C}^t_m$-formula size). *The* $\boldsymbol{CS}^{m,t}_w(A,B)$ *game is the version of the* $\boldsymbol{CS}^m_w(A,B)$ *game where* $k \leq t$ *for the* $\exists^{\geq k}$ *and* $\forall^{\geq k}$ *moves.*

4.3 An Illustration of the CS Game

In this section we present a small example illustrating the game.

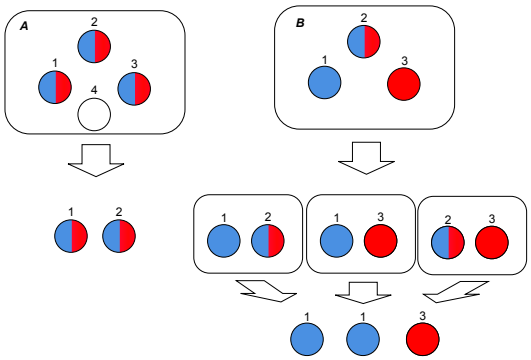

Figure 3 Structures $\mathcal{A}$ and $\mathcal{B}$ with relations *red* and *blue*, and a possible $\exists^{\geq 2}x$ move.

▶ **Example 9.** Consider playing the game $\mathbf{CS}^{1,2}_{w_0}$ on structures $\mathcal{A}$ and $\mathcal{B}$ of 4 and 3 elements respectively with unary relations *blue* and *red*, as illustrated in Fig. 3. Formulas of size $w_0 = 2$ cannot distinguish the structures, as those can only contain a single atomic formula and a quantifier. The sentence $\exists^{\geq 2}x\,(blue(x) \wedge red(x))$ distinguishes $\mathcal{A}$ and $\mathcal{B}$ and is of size $w_0 = 3$. We show that Spoiler has a winning strategy when $w_0 = 3$ and illustrate an $w_0 = 2$ strategy.

Consider the starting position with no assignment $\mathbf{CS}^{1,2}_3(A_0, B_0)$. Spoiler starts with an $\exists^{\geq 2}x$-move and picks the 2-choice function $\{1,2\}$, corresponding to two copies of $\mathcal{A}$, with $\alpha(x) = 1$ and $\alpha(x) = 2$. This is shown on the left of Fig. 3 by showing only the assignments in the two copies.

The 2-choice functions on B are $\{1,2\},\{1,3\},\{2,3\}$(the three boxes on the right of Fig. 2). Spoiler then selects one of the two possible choices in each box. For example, picking the selection function $(1,1,3)$ (picking 1 from the first and second boxes and 3 from the third box) is a valid choice for Spoiler. To win at this step (bottom of the Fig.), and make $w_0 = 2$ sufficient for the initial game, Spoiler must distinguish all elements on the left from the element of the right with *red* or *blue* in one move, since this is not possible no matter what Spoiler had picked from the three boxes, the game has to go on for Spoiler to be able to win.

From $w_0 = 3$, in the position shown in the bottom of Fig. 3, $w = 2$, and families $\{1,2\}$, $\{1,3\}$ need to be distinguished. Spoiler can make a $\wedge$ move with $u = v = 1$ to get to position $(1,\{1,2\},\{1\})$ on one side, and position $(1,\{1,2\},\{3\})$ on the other side. On the first side, Spoiler wins in $u = 1$ with *red* and on the other side wins in $v = 1$ with *blue*.

5 Game Characterization Theorem

In this section, we state the game characterization theorem and formulate some corollaries. The proof of the characterization theorem is given in Appendix A.

▶ **Theorem 10** (Characterization Theorem). *Let A, B be families and w be a positive integer. Then the following are equivalent:*

1. *Spoiler has a winning strategy in the game $\boldsymbol{CS}_w^m(A, B)$.*
2. *There is a $\mathcal{C}_m$ formula ϕ of size $|\phi| \leq w$ such that $(A, B) \models \phi$.*

A corollary for structures is as follows:

▶ **Corollary 11.** *Let $\mathcal{A}$ and $\mathcal{B}$ be structures and let w be a positive integer. Then the following conditions are equivalent:*

1. *Spoiler has a winning strategy in the game $\boldsymbol{CS}_w^m(A_0, B_0)$.*
2. *There is a $\mathcal{C}_m$-sentence ϕ of size $|\phi| \leq w$ such that $(\mathcal{A}, \mathcal{B}) \models \phi$.*

The game for $\mathcal{C}_m^t$ can be used to prove analogous results for bounded counting rank.

▶ **Corollary 12.** *Suppose A and B are families and let w be a positive integer. Then the following are equivalent:*

1. *Spoiler has a winning strategy in the game $\boldsymbol{CS}_w^{m,t}(A, B)$.*
2. *There is a $\mathcal{C}_m^t$-formula ϕ of size $|\phi| \leq w$ such that $(A, B) \models \phi$.*

The proof is similar to the **CS** game characterization of $\mathcal{C}_m$ formula size presented in Appendix A, but with the aforementioned parameter k bounded, and is omitted.

We derive a variation of the counting game for guarded counting logic [12], where the guarding binary relation is denoted by E. The resulting game is denoted by E-$\mathbf{CS}_w^2(A, B)$. By the equivalence between the expressivity of GNNs and guarded counting logic established in [2, Theorem 4.2], we obtain the following corollary linking the distinguishing power of GNNs and the guarded counting game:

▶ **Corollary 13.** *Let A and B be families and let w be a positive integer. Then the following are equivalent:*

1. *Spoiler has a winning strategy in the game E-$\boldsymbol{CS}_w^2(A, B)$.*
2. *There is a GNN that expresses an FO formula of size w that is capable of distinguishing A from B.*

6 Distinguishing Linear Orders with Counting Logics

In the rest of the paper we deal with counting logic definability on linear orders and apply **CS** games in this context. A linear order is defined over the signature $\{min, max, <, succ\}$, where $<$ is a linear ordering. $\mathcal{A}_n$ denotes the structure $(\{0, \ldots, n\}\ , <)$, where $<$ is the standard linear ordering. For $a, b \in \mathbb{N}$, let $\mathrm{d}(a, b) := |a - b|$ and $<$-type(a, b) be $=, <$ or $>$ reflecting the order between a and b.

The first result gives a lower bound for the quantifier rank needed to distinguish linear orders in counting logic with bounded counting rank. It extends the standard lower bound for first-order logic (see, e.g., [26]). The proof uses the IL pebble game of Definition 1 with bounded counting rank [22] and is presented in Appendix B. The argument provides intuition for the framework developed for Theorem 16.

▶ **Theorem 14.** *Let $t, k > 0$, and let $n, m \geq (t+1)^k$. Then $\mathcal{A}_n$ and $\mathcal{A}_m$ cannot be distinguished by sentences of counting rank at most t and quantifier rank at most k.*

We now turn to the problem of distinguishing two linear orders using counting logic sentences with a bounded number of variables and bounded counting rank, starting with an upper bound using two variables.

▶ **Proposition 15.** *There is a $\mathcal{C}_2^t$-sentence of size $O(n/t)$ distinguishing $\mathcal{A}_n$ and $\mathcal{A}_m$, where $n < m$.*

Proof. Consider the formulae:

- $\phi_0(x) = (x = x)$.
- $\phi_{t(l+1)}(x) = \exists^{\geq t} y((y < x) \wedge \phi_{tl}(y))$.
- For $n = tl + p$ with $p < t$: $\phi_{tl+p}(x) = \exists^{\geq p} y((y < x) \wedge \phi_{tl}(y))$

Let $\phi = \neg\exists x\ \phi_{n+1}(x)$ Then $\mathcal{A}_m$ satisfies ϕ iff $n < m$. The size of ϕ is $O(n/t)$. ◀

As discussed earlier, the following lower bound extends [14] to counting logic. The proof is presented in the remainder of the paper and in Appendix C. For $t = 1$, i.e. for FO_3, this theorem improves on lower bound of [14] from $\sqrt{n}/2$ to $\sqrt{n}$.

▶ **Theorem 16.** *If ϕ is a $\mathcal{C}_3^t$-sentence distinguishing $\mathcal{A}_n$ and $\mathcal{A}_m$, where $n < m$, then $|\phi| \geq \frac{\sqrt{n}}{t}$.*

7 Proof of Theorem 16

In this section we first introduce the framework of [14]. Extended syntax trees are adapted to counting logic. The definitions of separators and the weighting scheme are unchanged. The lower bound for formula size in terms of separator weights and the proof of Theorem 16 based on it are given in Section 7.2. The main technical result is Lemma 22, Part 3, which is proved in Section 7.3. Proofs of lemmas used in this section are given in Appendix C.

7.1 Description of the Framework

We extend the framework for FO of [14] to counting logic. In the following, we write x, y, z for the three variables used. We start by describing the main concepts: the counting game derived extended syntax trees, separators, and weighting scheme.

7.1.1 Extended Syntax Tree

Given two families A, B, the extended syntax tree represents a winning strategy for Spoiler in the $\mathbf{CS}_w^{3,t}$ game on (A, B). It assigns to each tree node v a pair of families $il(v)$ ("interpretation label"), along with Spoiler's move $sl(v)$ ("syntax label") in this position such that:

1. A node v_1 is a child of node v if position $il(v_1)$ can be obtained in one move from $il(v)$,
2. A node v is a leaf if $il(v)$ satisfies the atomic win condition.

For each node some additional information is provided, including the family distinguished by the interpretation label of the node. The root is associated to the starting position of the game $\mathbf{CS}_w^{3,t}$, and we consider the nodes associated to positions reached throughout a winning strategy.

▶ **Definition 17.** *Let ψ be an $\mathcal{C}_3^t$-formula and A and B be families such that $(A, B) \models \psi$. The extended syntax tree $T_\psi^{\langle A,B\rangle}$ is defined as follows:*

- *If ψ is an atomic formula, then $T_\psi^{\langle A,B\rangle}$ is a single node v with syntax label $sl(v) := \psi$ and interpretation label $il(v) := \langle A, B\rangle$.* Comment*: $(A, B) \models \psi$.*
- *If ψ is of the form $\neg\psi_1$, then $T_\psi^{\langle A,B\rangle}$ has root v with $sl(v) := \neg$ and $il(v) := \langle A, B\rangle$. The unique child of v is the root of $T_{\psi_1}^{\langle B,A\rangle}$.* Comment*: $(B, A) \models \psi_1$.*

- *If ψ is of the form $\psi_1 \vee \psi_2$, then $T_\psi^{\langle A,B\rangle}$ has root v with $sl(v) := \vee$ and $il(v) := \langle A, B\rangle$. The first child of v is the root of $T_{\psi_1}^{\langle A_1,B\rangle}$ and the second child of v is the root of $T_{\psi_2}^{\langle A_2,B\rangle}$, where $A_i = \{(\mathcal{A},\alpha) \in A : (\mathcal{A},\alpha) \models \psi_i\}$ for $i \in \{1,2\}$.* Comment*: $A = A_1 \cup A_2$ and $(A_i, B) \models \psi_i$.*
- *If ψ is of the form $\psi_1 \wedge \psi_2$, then $T_\psi^{\langle A,B\rangle}$ has root v with $sl(v) := \wedge$ and $il(v) := \langle A, B\rangle$. The first child of v is the root of $T_{\psi_1}^{\langle A,B_1\rangle}$ and the second child of v is the root of $T_{\psi_2}^{\langle A,B_2\rangle}$, where $B_i = \{(\mathcal{B},\beta) \in B : (\mathcal{B},\beta) \models \neg\psi_i\}$ for $i \in \{1,2\}$.* Comment*: $B = B_1 \cup B_2$ and $(A, B_i) \models \psi_i$.*
- *If ψ is of the form $\exists^{\geq k} u\psi_1$, for a variable $u \in \{x, y, z\}$ and $k \leq t$, then $T_\psi^{\langle A,B\rangle}$ has root v with $sl(v) := \exists^{\geq k} u$ and $il(v) := \langle A, B\rangle$. The unique child of v is the root of $T_{\psi_1}^{\langle A(F^k/u),B(*^k/u)\rangle}$, where $F^k \in F_A^k$ is chosen so that $A(F^k/u) \models \psi_1$, and for every $G^k \in F_B^k$, G is selected from G^k so that $B(G/u) \models \neg\psi_1$.* Comment*: $(A(F^k/u), B(*^k/u)) \models \psi_1$, as $B(*^k/u) = \bigcup_{G^k \in F_B^k} B(G/u)$.*
- *If ψ is of the form $\forall^{\geq k} u\psi_1$, for a variable $u \in \{x, y, z\}$ and $k \leq t$, then $T_\psi^{\langle A,B\rangle}$ has a root node v with $sl(v) := \forall^{\geq k} u$ and $il(v) := \langle A, B\rangle$. The unique child of v is the root of $T_{\psi_1}^{\langle A(*^k/u),B(F^k/j)\rangle}$, where $F^k \in F_B^k$ is chosen so that $B(F^k/u) \models \neg\psi_1$, and for every $G^k \in F_A^k$, G is selected from G^k so that $A(G/u) \models \psi_1$.* Comment*: $(A(*^k/u), B(F^k/u)) \models \psi_1$, as $A(*^k/u) = \bigcup_{G^k \in F_A^k} A(G/u)$.*

We define $|T_\psi^{\langle A,B\rangle}|$ to be the number of nodes in the tree, and clearly $|T_\psi^{\langle A,B\rangle}| = |\psi|$. Note that the definition of the tree corresponds to the formula to game direction of Theorem 10, detailed in Appendix A.

7.1.2 Separators and Weighting Scheme

Separators are the key concept in [14] to study succinctness of formulas with a bounded number of variables. They provide a different "yardstick" for every pair of variables and constants for distances to be represented exactly. In contrast to the standard approach (as in Theorem 14), this "context-dependent" approach allows for a detailed representation of the progress made in the game.

▶ **Definition 18** (separator). *A separator for families $\langle A, B\rangle$ is a mapping $\delta : \mathcal{P}_2(\{\min, \max, x, y, z\}) \to \mathbb{N}$ such that for every $I := (\mathcal{A},\alpha) \in A$ and $J := (\mathcal{B},\beta) \in B$, there are $u, u' \in \{\min, \max, x, y, z\}$ with $u \neq u'$ one of the following hold:*

1. *$<$-type$(\alpha(u), \alpha(u')) \neq <$-type$(\beta(u), \beta(u'))$*
2. *both of the following two conditions hold:*
   - *MIN$[d(\alpha(u), \alpha(u')), d(\beta(u), \beta(u'))] \leq \delta(\{u, u'\})$*
   - *$d(\alpha(u), \alpha(u')) \neq d(\beta(u), \beta(u'))$.*

In this case the separator *distinguishes* the two families. We will also use the notion of the type of an element, in the following specific form.

▶ **Definition 19.** *(type of an element) Let $(\mathcal{A},\alpha)$ be an interpretation and $a \in \mathcal{U}^{\mathcal{A}}$. Then $<$-type$(a) = (<$-type(a, min_A), $<$-type$(a, \alpha(x))$, $<$-type$(a, \alpha(y))$, $<$-type$(a, max_A))$.*

The definition of the weight of a separator is motivated by the quantities appearing in the computations [14].

▶ **Definition 20** (weight of a separator). *Let δ be a separator, we define its:*

1. *border-distance* $b(\delta) := MAX\ \{\delta(\{min, max\}),$
   $\delta(\{min, u\}) + \delta(\{u', max\}) : u, u' \in \{x, y, z\}\}$
2. *center-distance* $c(\delta) := MAX\ \{\delta(p) + \delta(q) : p, q \in \mathcal{P}_2(\{x, y, z\}), p \neq q\}$
3. *weight* $w(\delta) := \sqrt{c(\delta)^2 + b(\delta)}$.

The weight is a measure of the distinguishing power of separators. A separator is *minimal* if it has minimal weight.

7.2 Proof of Theorem 16

The proof is based Theorem 21, which gives a formula size lower bound in terms of minimal separator weight. Theorem 21, in turn, is based on two lemmas, Lemma 22 and 23. The proofs of these lemmas are similar for FO and counting logic and are omitted due to lack of space, with the exception of quantifiers, handled in Lemma 22 Part 3. This is the most technical part of [14] and the problem we consider is to prove a version of this part for counting quantifiers. This is the point where counting rank t enters the bounds. Other than this difference, the rest of the argument, like handling propositional connectives, involves minor changes in the computation to account for t. The proof Lemma 22 Part 3 is based on Lemma 24. This lemma is proved in Section 7.3 and Appendix C. The derivation of Lemma 22 from Lemma 24 is again similar to the FO case and is omitted.

▶ **Theorem 21.** *Suppose $(A, B) \models \psi$ and δ is a minimal separator for $\langle A, B\rangle$, then $|\psi| \geq \frac{w(\delta)}{t}$.*

The following lemma links separator weights to the structure of extended syntax trees.

▶ **Lemma 22.** *Suppose $(A, B) \models \psi$ and let T be the extended syntax tree $T_\psi^{\langle A,B\rangle}$, and δ be a minimal separator for $il(v)$. For every node v of T the following is true:*

1. *If v is a leaf, then $w(\delta) \leq 1$.*
2. *If v has 2 children v_1 and v_2, and δ_i is a minimal separator for $il(v_i)$, for $i \in \{1, 2\}$, then $w(\delta) \leq w(\delta_1) + w(\delta_2)$.*
3. *If v has exactly one child v_1, and δ_1 is a minimal separator for $il(v_1)$, then $w(\delta) \leq w(\delta_1)+t$.*

The bound of part 3. extends the corresponding bound of [14] to counting logic. In fact, for counting rank 1, i.e., for first-order logic without counting, it improves the bound from $w(\delta) \leq w(\delta_1) + 2$ to $w(\delta) \leq w(\delta_1) + 1$. This leads to the slight improvement of the lower bound of [14] mentioned after Theorem 16.

▶ **Lemma 23.** *Let T be a finite binary tree where each node v is equipped with a weight $w(v) > 0$ such that the following is true:*

- *If v is a leaf, then $w(v) \leq 1$.*
- *If v has 2 children v_1 and v_2, then $w(v) \leq w(v_1) + w(v_2)$.*
- *If v has exactly one child v_1, then $w(v) \leq w(v_1) + t$.*

Then, $|T| \geq \frac{w(r)}{t}$, where r is the root of T and $|T|$ is the number of nodes of T.

Theorem 21 follows directly from the previous lemmas.

Proof of Theorem 21. Consider $T_\psi^{\langle A,B\rangle}$ the syntax tree for the pair $\langle A, B\rangle$, ψ.
We associate to each node v of $T_\psi^{\langle A,B\rangle}$ a weight $w(v) := w(\delta_v)$, where δ_v is a minimal separator for $il(v)$. From , we get that $|\psi| = |T_\psi^{\langle A,B\rangle}| \geq \frac{w(\delta)}{t}$, where δ is a minimal separator of $il(r) = \langle A, B\rangle$. ◀

Finally, as a consequence, we get Theorem 16.

Proof of Theorem 16. Suppose ψ is an $\mathcal{C}_3^t$-sentence such that $\mathcal{A}_n \models \psi$ and $\mathcal{A}_m \models \neg\psi$. Let α be the assignment that assigns x, y and z to 0. A mapping δ is a separator for the two structures iff $\delta(\{u, max\}) \geq n$ for some $u \in \{min, x, y\}$. The separator defined as $\delta(\{min, max\}) = n$ and 0 elsewhere is one of the minimal weight separator available, and $w(\delta) = \sqrt{n}$. Finally, Theorem 16 follows from Theorem 21. ◀

7.3 Proof of Lemma 22

The proof of Lemma 22 Part 3. relies on the following key lemma, which gives a bound on how much a counting quantifier can increase the distinguishing power of separators.

▶ **Lemma 24.** *Let v be a node of T that has syntax-label $sl(v) = Q^{\geq k}u$ for $Q \in \{\exists, \forall\}$, $k \leq t$ and $u \in \{x, y, z\}$. Let δ_1 be a separator for $il(v_1)$, where v_1 is the unique child of v in T.*

Let δ be the separator defined via:

- $\delta(\{u, u'\}) := 0$ *, for all* $u' \in \{min, max, x, y, z\} \setminus \{u\}$ *and,*
- $\delta(\{u', u''\}) := MAX\{\delta_1(\{u', u''\}), \delta_1(\{u', u\}) + \delta_1(\{u, u''\}) + k - 1\}$ *for all* $(u', u'') \in \mathcal{P}_2(\{min, max, x, y, z\} \setminus \{u\})$.

Then, δ is a separator for $il(v)$.

7.3.1 Proof of Lemma 24

Due to symmetry, we only consider the case $\exists^{\geq k} z$. It has to be shown that δ is a separator for $\langle A, B\rangle = il(v)$. By definition, $il(v_1) = \langle A(F^k/z), B(*^k/z)\rangle$. Consider $I = (\mathcal{A}, \alpha) \in A$ and $J = (\mathcal{B}, \beta) \in B$. We need to show that δ separates $\langle I, J\rangle$. Let:

- $I_{F^k} := \{(\mathcal{A}, \alpha(F_i^k(\mathcal{A}, \alpha)/z)\}$ be the set of interpretations generated from I in v_1, and let $a_i = F_i^k(\mathcal{A}, \alpha)$ and $I^i_{F^k} = (\mathcal{A}, \alpha(a_i/z))$, $1 \leq i \leq k$.
- For any k-choice function G^k on B, let G be the choice function selected from G^k, and let J_G be the interpretation selected from $J_{G^k} = \{(\mathcal{B}, \beta(G_i^k(\mathcal{B}, \beta)/z))\}$. We define $b_i := G_i^k(\mathcal{B}, \beta)$; note that J_G corresponds to some b_j, $1 \leq j \leq k$.

Note that for all $u \in \{min, max, x, y\}$, and for all $i \in [k]$, $\alpha(F_i^k(\mathcal{A}, \alpha)/z)(u) = \alpha(u)$. So we will omit subscripts and just write $\alpha(x)$ for the assignment of x and min_A for the minimum in the k copies of $(\mathcal{A}, \alpha)$ (and similarly with y, max). Similarly, for all $u \in \{min, max, x, y\}$, for all $i \in [k]$, and for all $G^k \in F_B^k$, $\beta(G_i^k(\mathcal{B}, \beta)/z)(u) = \beta(u)$, so we will also write $\beta(x)$ for the assignment of x and min_B for the minimum in the k copies of $(\mathcal{B}, \beta)$ (and similarly with y, max). We will say that $\delta(\{u, u'\})$ separates $\langle I, J\rangle$ to signify that $\{u, u'\}$ witnesses the separation property of δ on $\langle I, J\rangle$.

We formulate several indistinguishability properties for the choice functions and the interpretations. Lemma 30 shows that if each property holds then δ_1 does not separate $il(v_1)$ as, using the indistinguishabilities, a k-choice function G^k can be constructed for which I_{F^k} and J_G are not separated by δ_1. Thus some of the assumptions fail. Lemmas 26-29 combined show that if any of the properties fail, then δ separates $\langle I, J\rangle$.

The first property is about the role of min, x, y, max in separation.

Property 1.

Part a) For every G^k, none of the $I^i_{F^k}$ is separated from J_G by $\delta_1(\{u, u'\})$ for $\{u, u'\} \in \{min, x, y, max\}$.

Part b) If there are $u, u' \in \{x, y, min, max\}$ such that MIN [d$(\alpha(u), \alpha(u'))$,d$(\beta(u), \beta(u'))] \leq \delta_1(\{z, u\})$ then d$(\alpha(u), \alpha(u'))$ =d$(\beta(u), \beta(u'))$.

We may assume *w.l.o.g.* that $\alpha(x) \leq \alpha(y)$. **Property 1** implies that $\beta(x) \leq \beta(y)$ holds as well.

The other properties refer to"closeness" of elements a_i to $\{\alpha(x), \alpha(y)y, min_A, max_A\}$ in $(\mathcal{A}, \alpha)$, gauged by $\delta_1(\{z, u\})$ for $u \in \{x, y, min, max\}$, depending on whether $\mathrm{d}(a_i, \alpha(u)) \leq \delta_1(\{z, u\})$ or $\mathrm{d}(a_i, \alpha(u)) > \delta_1(\{z, u\})$. The information about the location of a_i is completed by its type $<$-type(a_i) (Definition 19).

The next two properties deal with implications of closeness to the existence of a "twin" in the other structure.

Property 2. For every a_i and $u \in \{x, y, min, max\}$ such that $\mathrm{d}(a_i, \alpha(u)) \leq \delta_1(\{z, u\})$ there is a $b \in \mathcal{U}^{\mathcal{B}}$ such that $\mathrm{d}(a_i, \alpha(u)) = \mathrm{d}(b, \beta(u))$ and $<$-type$(a_i) = <$-type(b).

Let $S_i := \{u \in \{x, y, min, max\} : \mathrm{d}(a_i, \alpha(u)) \leq \delta_1(\{z, u\})\}$ be the set of variables to which a_i is close.

Property 3. For every a_i such that $S_i \neq \emptyset$, there is a $b \in \mathcal{U}^{\mathcal{B}}$ such that $<$-type$(a_i) = <$-type(b) and for every $u \in \{x, y, min, max\}$ it holds that

a) $\mathrm{d}(a_i, \alpha(u)) = \mathrm{d}(b, \beta(u))$ if $u \in S_i$

b) $\mathrm{d}(b, \beta(u)) > \delta(\{z, u\})$ otherwise.

The last property is about elements which are not in any closeness relation. We define quantities, called gaps, to characterize those elements.

▶ **Definition 25** (Gap). *For* $(\mathcal{U}, \gamma) \in \{(\mathcal{A}, \alpha), (\mathcal{B}, \beta)\}$ *and*

$$I_{\{min,x\}} = [min_U, \gamma(x)], \quad I_{\{x,y\}} = [\gamma(x), \gamma(y)], \quad I_{\{y,max\}} = [\gamma(y), max_U],$$

define: $Gap_{(\mathcal{U},\gamma)}(S) := \{\, a \in I_S : d(a, \gamma(u)) > \delta_1(\{u, z\}) \ \forall u \in \{min, x, y, max\} \,\}$.

A separator cannot distinguish using a pair of variables in $\{\{min, z\}, \{x, z\}, \{y, z\}, \{max, z\}\}$ between an element in $Gap_{(\mathcal{U},\gamma)}(S)$ and $Gap_{(\mathcal{U}',\gamma')}(S)$. This is the point where the additive term $k - 1$ enters the bounds.

Property 4. If $(u, u') \in \{(min, x), (x, y), (y, max)\}$ and there are p choices of F^k in $Gap_{(\mathcal{A},\alpha)}(\{u, u'\})$ then $|Gap_{(\mathcal{B},\beta)}(\{u, u'\})| \geq p$.

Now we formulate the lemmas which conclude the separator property of δ in case some subsets of the properties fail. The first lemma notes that a separating pair in δ_1 not involving z also separated in δ.

▶ **Lemma 26.** *If Property 1 fails then* δ *separates* $\langle I, J \rangle$.

In view of Lemma 26, the next two lemmas say that if Property 1 holds, then the non-existence of a twin implies separation.

▶ **Lemma 27.** *If Properties 1 or 2 fails then* δ *separates* $\langle I, J \rangle$.

▶ **Lemma 28.** *If Properties 1, 2 or 3 fails then* δ *separates* $\langle I, J \rangle$.

The last lemma of this kind concludes separation from the sizes of gaps.

▶ **Lemma 29.** *If Properties 1 or 4 fails then* δ *separates* $\langle I, J \rangle$.

The final Lemma 30 provides the step needed to complete the proof of Lemma 24. It shows that, on the other hand, the conjunction of all the indistinguishabilities contradicts the property that δ_1 is a separator,

▶ **Lemma 30.** *If Properties 1,2,3,4 hold, then there is a G^k such that δ_1 does not separate $\langle I_{F_j^k}, J_G\rangle$.*

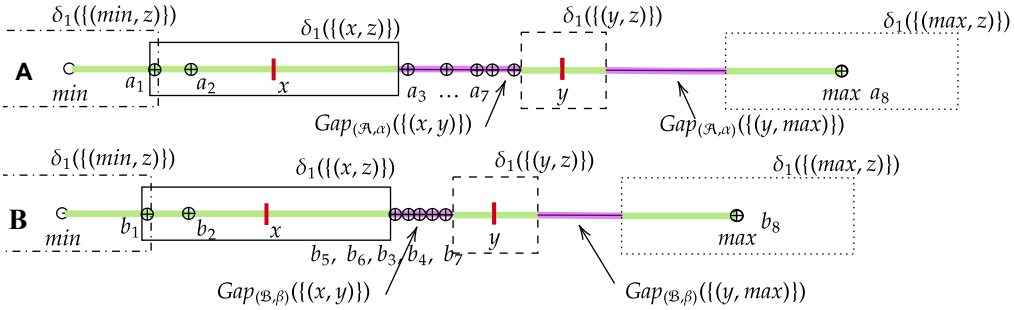

Figure 4 Illustration of why $\langle I_{F^k}, J_G\rangle$ cannot be separated by δ_1 in Lemma 30 of the proof of Lemma 24. For each $u \in \{min, x, y, max\}$ we draw on $\mathcal{A}$ (resp. $\mathcal{B}$) a box centered at $\alpha(u)$ (resp. $\beta(u)$) of half-length $\delta_1(\{u,z\})$, representing the separating power of the pair $\{u,z\}$. The portions of the linear orders not covered by any such interval form the *gaps* (shown in purple). Crossed dots on $\mathcal{A}$ indicate the choices coming from F^k, while crossed dots on $\mathcal{B}$ represent the best matching replies for indistinguishability. In Lemma 30, every choice on $\mathcal{A}$ admits a distinct δ_1-indistinguishable counterpart on $\mathcal{B}$, preventing separation by δ_1.

Proof. As **Property 3** holds, for every a_i such that $S_i \neq \emptyset$, we have that there exists a unique $b_i \in \mathcal{U}^{\mathcal{B}}$ such that $<$-type$(a_i) = <$-type(b_i) and $\mathrm{d}(b_i, \beta(u)) = \mathrm{d}(a_i, \alpha(u))$ for all $u \in S_i$, and $\mathrm{d}(b_i, \beta(u)) > \delta_1(\{z,u\})$ for all $u \in \{x, y, min, max\} \setminus S_i$. Additionally, since **Property 4** holds, for every a_i in $Gap_{(\mathcal{A},\alpha)}(\{u,u'\})$ for $(u,u') \in \{(min,x),(x,y),(y,max)\}$, there exists a distinct $b_i \in Gap_{(\mathcal{B},\beta)}(\{u,u'\})$.
Consider the choice function G^k consisting of the k b_i's matching the k a_i's as described above. Suppose index j from the choice function G^k is selected to create J_G.

- If $S_j \neq \emptyset$, i.e., a_j is not in a gap, then b_j satisfies the conditions described by **Property 3**, so $\langle I_{F_j^k}, J_G\rangle$ cannot be separated by $\delta_1(z,u)$ for $u \in \{x, y, min, max\}$.
- If a_j is in $Gap_{(\mathcal{A},\alpha)}(\{u,u'\})$, then by construction b_j is in $Gap_{(\mathcal{B},\beta)}(\{u,u'\})$, and $\langle I_{F_j^k}, J_G\rangle$ cannot be distinguished by $\delta_1(z,u)$ for $u \in \{x, y, min, max\}$.

Property 1 states that pairs $\delta_1(u,u')$ with $u \in \{x, y, min, max\}$ cannot distinguish $\langle I_{F_j^k}, J_G\rangle$ either. Thus δ_1 does not distinguish $\langle I_{F_j^k}, J_G\rangle$. ◀

We complete the proof of Lemma 24 using Lemmas 26-30.

Proof. If Lemma 26 cannot be applied to show that δ separates $\langle I, J\rangle$ then Property 1 holds. Then, if Lemma 27 cannot be applied to show that δ separates $\langle I, J\rangle$, Property 2 holds as well. It follows similarly from Lemmas 28 and 29 that Properties 3-4 hold. Lemma 30 then implies that δ_1 does not separate $il(v_1)$, a contradiction. ◀

8 Conclusion and Perspectives

In this paper an EF game is formulated for counting logic formula size. It is used to prove a $\sqrt{n}/t$ lower bound for the size of 3-variable counting logic formulae with counting rank t, distinguishing a linear order of size n from a larger one. The lower bound extends a $\Omega(\sqrt{n})$ lower bound of [14] for FO_3. The proof is based on the approach of [14], with a different

argument for the central case of handling counting quantifiers. Closing the gap between the lower bound and the upper bound of size $O(n/t)$ is an open problem. This is open even in the FO case where, as far as we know, no improvement is known of the linear upper bound.

Comparing the succinctness of various knowledge representation formalisms is studied in detail in knowledge compilation [5], Boolean complexity theory [23] and other areas, but, as noted in [14], perhaps less so for predicate logic. Comparing m-variable counting logic formula sizes for $m = 2, 3, 4$ seems to be an interesting problem.

We conclude with a brief description of the connection between GNN and counting logic formula size. A GNN works on a graph with feature vectors assigned to the nodes. In each round these are updated by applying a *combination* function to the previous vector and an *aggregate* of the feature vectors of the neighbors [17]. A *logical classifier* computes a unary query on graphs (e.g., assigning to every graph the set of red vertices with all blue neighbors). Barceló et al. [2] showed that an FO logical classifier is computable by a GNN iff it is definable in 2-variable guarded counting logic. See also [12, 13], also noting that this is a "uniform" model. Every such formula has a GNN simulation with complexity (number of features and rounds) depending on the complexity of the formula. One of the many questions raised by this connection is: how is the complexity of the formula related to the complexity of learning the GNN? A related question, recently considered in [30], is whether the underlying formula can be extracted from a learned GNN using explainability techniques.

References

1 Micah Adler and Neil Immerman. An *n!* lower bound on formula size. *ACM Trans. Comput. Log.*, 4(3):296–314, 2003. doi:10.1145/772062.772064.

2 Pablo Barceló, Egor V. Kostylev, Mikaël Monet, Jorge Pérez, Juan L. Reutter, and Juan Pablo Silva. The logical expressiveness of graph neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. URL: https://openreview.net/forum?id=r1lZ7AEKvB.

3 Jin-yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Comb.*, 12(4):389–410, 1992. doi:10.1007/BF01305232.

4 Marco Carmosino, Ronald Fagin, Neil Immerman, Phokion G. Kolaitis, Jonathan Lenchner, and Rik Sengupta. A finer analysis of multi-structural games and beyond. *CoRR*, abs/2301.13329, 2023. doi:10.48550/arXiv.2301.13329.

5 Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *J. Artif. Intell. Res.*, 17:229–264, 2002. doi:10.1613/jair.989.

6 Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995.

7 Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 1961.

8 Kousha Etessami. Counting quantifiers, successor relations, and logarithmic space. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference, Minneapolis, Minnesota, USA, June 19-22, 1995*, pages 2–11. IEEE Computer Society, 1995. doi:10.1109/SCT.1995.514723.

9 Kousha Etessami. Counting quantifiers, successor relations, and logarithmic space. *J. Comput. Syst. Sci.*, 54(3):400–411, 1997. doi:10.1006/jcss.1997.1485.

10 Ronald Fagin, Jonathan Lenchner, Nikhil Vyas, and R. Ryan Williams. On the number of quantifiers as a complexity measure. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, Vienna, Austria, August 22-26, 2022*, volume 241 of *LIPIcs*, pages 48:1–48:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.MFCS.2022.48.

11 Roland Fraisse. Sur quelques classifications des systemes de relations, 1954.

12 Martin Grohe. The logic of graph neural networks. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–17. IEEE, 2021. doi:10.1109/LICS52264.2021.9470677.

13 Martin Grohe. The descriptive complexity of graph neural networks. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–14. IEEE, 2023. doi:10.1109/LICS56636.2023.10175735.

14 Martin Grohe and Nicole Schweikardt. The succinctness of first-order logic on linear orders. *Log. Methods Comput. Sci.*, 1(1), 2005. doi:10.2168/LMCS-1(1:6)2005.

15 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 99–110. IEEE Computer Society, 1987. doi:10.1109/SFCS.1987.59.

16 András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *J. Comput. Syst. Sci.*, 46(2):129–154, 1993. doi:10.1016/0022-0000(93)90001-D.

17 William L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2020. doi:10.2200/S01045ED1V01Y202009AIM046.

18 Lauri Hella. Logical hierarchies in PTIME. *Inf. Comput.*, 129(1):1–19, 1996. doi:10.1006/inco.1996.0070.

19 Lauri Hella and Jouko Väänänen. The size of a formula as a measure of complexity. In Åsa Hirvonen, Juha Kontinen, Roman Kossak, and Andrés Villaveces, editors, *Logic Without Borders - Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics*, volume 5. De Gruyter, 2015. doi:10.1515/9781614516873.193.

20 Neil Immerman. Number of quantifiers is better than number of tape cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981. doi:10.1016/0022-0000(81)90039-8.

21 Neil Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999. doi:10.1007/978-1-4612-0539-5.

22 Neil Immerman and Eric Lander. Describing graphs: A first-order approach to graph canonization, 1990.

23 Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

24 Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require superlogarithmic depth. *SIAM J. Discret. Math.*, 3(2):255–265, 1990. doi:10.1137/0403021.

25 Dietrich Kuske and Nicole Schweikardt. First-order logic with counting: At least, weak hanf normal forms always exist and can be computed! In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, volume abs/1703.01122, pages 1–12. IEEE, 2017. doi:10.48550/arXiv.1703.01122.

26 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004. doi:10.1007/978-3-662-07003-1.

27 Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 4602–4609. AAAI Press, 2019. doi:10.1609/aaai.v33i01.33014602.

28 Martin Otto. *Bounded Variable Logics and Counting: A Study in Finite Models*, volume 9 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi:10.1017/9781316716878.

29 Martin Otto. Graded modal logic and counting bisimulation. *CoRR*, abs/1910.00039, 2019. doi:10.48550/arXiv.1910.00039.

30 Alexander Pluska, Pascal Welke, Thomas Gärtner, and Sagar Malhotra. Logical distillation of graph neural networks. In Pierre Marquis, Magdalena Ortiz, and Maurice Pagnucco, editors, *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*, 2024. doi:10.24963/kr.2024/86.

31 Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009. doi:10.1109/TNN.2008.2005605.

32 Harry Vinall-Smeeth. From quantifier depth to quantifier number: Separating structures with k variables, 2024. doi:10.48550/arXiv.2311.15885.

33 Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL: https://openreview.net/forum?id=ryGs6iA5Km.

A Proof of Theorem 10

Proof. We use induction on w. If $w = 1$, Spoiler wins in $\mathbf{CS}_1^m(A, B)$ only if there is an atomic formula ϕ, so verifying $|\phi| = 1$, such that $(A, B) \models \phi$. Reciprocally, if $|\phi| = 1$ and $(A, B) \models \phi$, then ϕ is an atomic formula and spoiler wins in $\mathbf{CS}_1^m(A, B)$.
We now assume the equivalence for $v < w$.
To prove the forward direction of the equivalence, suppose Spoiler has a winning strategy in the game $\mathbf{CS}_w^m(A, B)$ starting with:

- $\neg$-move; Spoiler has a winning strategy in $\mathbf{CS}_{w-1}^m(B, A)$, so by induction hypothesis there is a formula ψ verifying $|\psi| \leq w - 1$ and $(B, A) \models \psi$. Finally $(A, B) \models \neg\psi$ and $|\neg\psi| = 1 + |\psi| \leq w$ so we get (2).
- $\bigvee$-move, choosing u, v, C and D such that $1 \leq u, v < w$ with $u + v = w$, splitting A into C and D. Spoiler has a winning strategy in $\mathbf{CS}_u^m(C, B)$ and $\mathbf{CS}_u^m(D, B)$, so by the induction hypothesis, there are formulae ψ and θ such that $|\psi| \leq u$, $|\theta| \leq v$, $(C, B) \models \psi$ and $(D, B) \models \theta$.
  We have that $C \models \psi$ and $D \models \theta$, so $A \models \psi \vee \theta$. Conversely, $B \models \neg\psi$ and $B \models \neg\theta$, and consequently $B \models \neg(\psi \vee \theta)$.
  It follows that $(A, B) \models \psi \vee \theta$, and since $|\psi \vee \theta| = |\psi| + |\theta| \leq u + v = w$, we get (2).
- $\bigwedge$-move; and by a symmetrical argument to the $\bigvee$-move we get (2).
- $\exists^{\geq k}$-move; choosing $j \in [m], k \in \mathbb{N}$, a k-choice function F^k on A and for every $G^k \in F_B^k$, selecting G from G^k to form $B(*^k/u) = \bigcup_{G^k \in F_B^k} B(G/u)$. Spoiler has a winning strategy in $\mathbf{CS}_{w-1}^m(A(F^k/j), B(*^k/j))$, so by the induction hypothesis, there is a formula ψ such that $|\psi| \leq w - 1$ and $(A(F^k/j), B(*^k/j)) \models \psi$.
  Define $\phi := \exists^{\geq k} x_j \psi$. We have that $A(F^k/j) \models \psi$, and so for all $1 \leq i \leq k$, $(\mathcal{A}, \alpha(F_i^k/j)) \models \psi$ and therefore $A \models \phi$.
  Now suppose that there exists a k-choice function G^k on B and a $(\mathcal{B}, \beta) \in B$ such that $\{(\mathcal{B}, \beta(G_i^k(\mathcal{B}, \beta)/j)) : 1 \leq i \leq k\} \models \psi$. Then there is an $i \in [k]$ such that $(\mathcal{B}, \beta(G_i^k(\mathcal{B}, \beta)/j))$ is part of the family $B(*^k/j)$ through the selected G, but $B(*^k/j) \models \neg\psi$, which is a contradiction. Therefore $B \models \neg\phi$.
  Finally $(A, B) \models \phi$ and $|\phi| = |\psi| + 1 \leq w$, so we get (2).
- $\forall^{\geq k}$-move; a symmetrical argument to the $\bigvee$-move yields (2).

To prove the backward direction, assume there is a formula ϕ of size $w > 1$ such that $(A, B) \models \phi$. We show that Spoiler has a winning strategy in $\mathbf{CS}_w^m(A, B)$. Let us consider the form of the formula ϕ:

- $\phi = \neg\psi$: Spoiler plays the $\neg$-move and gets in the position $\mathbf{CS}_{w-1}^m(B, A)$. Since $|\psi| < w$ and $(B, A) \models \phi$, Spoiler has a winning strategy in $\mathbf{CS}_{w-1}^m(B, A)$ by induction hypothesis, and therefore Spoiler has also a winning strategy in $\mathbf{CS}_w^m(A, B)$.

- $\phi = \psi \vee \theta$: Define $C = \{(\mathcal{A}, \alpha) \in A | (\mathcal{A}, \alpha) \models \psi\}$, $D = \{(\mathcal{A}, \alpha) \in A | (\mathcal{A}, \alpha) \models \theta\}$, u and v such that $w = u + v$, $|\psi| \leq u$ and $|\theta| \leq v$. Spoiler plays the $\bigvee$-move associated to u, v, C, D and gets to $\mathbf{CS}^m_u(C, B)$ or $\mathbf{CS}^m_v(D, B)$ according to Duplicator's choice. We have that $(C, B) \models \psi$, $(D, B) \models \theta$ and therefore by induction hypothesis Spoiler has a winning strategy in $\mathbf{CS}^m_u(C, B)$ or $\mathbf{CS}^m_v(D, B)$, and therefore Spoiler has also a winning strategy in $\mathbf{CS}^m_w(A, B)$.
- $\phi = \psi \wedge \theta$: a symmetrical argument shows that Spoiler has a winning strategy in $\mathbf{CS}^m_w(A, B)$.
- $\phi = \exists^{\geq k} x_j \psi$: Since $A \models \phi$, there is a k-choice function F^k on A such that $(\mathcal{A}, \alpha(F^k_i((\mathcal{A}, \alpha)/j)) \models \psi$, for every $1 \leq i \leq k$ and for all $(\mathcal{A}, \alpha) \in A$. Thus, $A(F^k/j) \models \psi$. On the other hand, $B \models \neg\phi$, thus from every G^k on B we can select G such that $B(G/j) \models \neg\psi$ and therefore, $B(*^k/j) \models \neg\psi$.
  Finally we have that $(A(F^k/j), B(*^k/j)) \models \psi$ and $|\psi| = |\phi| - 1 < w$, so by induction hypothesis, Spoiler has a winning strategy in the game $\mathbf{CS}^m_{w-1}(A(F^k/j), B(*^k/j))$ and therefore Spoiler has also a winning strategy in $\mathbf{CS}^m_w(A, B)$.
- $\phi = \forall^{\geq k} x_j \psi$: a symmetrical argument shows that Spoiler has a winning strategy in $\mathbf{CS}^m_w(A, B)$. ◀

B Proof of Theorem 14

We want to show that Duplicator can win a k-round EF game on $\mathcal{A}_n$ and $\mathcal{A}_m$ taken to be of length at least $(t+1)^k$. The key idea behind the proof is that, from a given assignment, Spoiler can "only look up to a certain distance" in the two directions on the line.

Proof. Suppose $\mathcal{A}_n$ and $\mathcal{A}_m$ are linear orders of length at least $(t+1)^k$, on which the EF game will be played.
After i moves, we denote by a the "position": a is a tuple consisting of $min_{\mathcal{A}_n}, max_{\mathcal{A}_n}$ concatenated to the i moves played on $\mathcal{A}_n$: $a = (a_{-1}, a_0, a_1, \ldots, a_i)$, $a_{-1} = min_{\mathcal{A}_n}$, $a_0 = max_{\mathcal{A}_n}$). Similarly, we define the tuple b of moves played on $\mathcal{A}_m$.
For $-1 \leq j, l \leq i$, we prove that regardless of Spoiler's choices, Duplicator can maintain the following inequalities:

1. if $d(a_j, a_l) \leq (t+1)^{k-i}$, then $d(b_j, b_l) = d(a_j, a_l)$.
2. if $d(a_j, a_l) > (t+1)^{k-i}$, then $d(b_j, b_l) > (t+1)^{k-i}$.
3. $a_j \leq a_l$ iff $b_j \leq b_l$.

Using those inequalities for $i = k$ moves, Property 3. yields that Duplicator win the k-round EF game on $\mathcal{A}_n$ and $\mathcal{A}_m$. Since this happens no matter what Spoiler plays, the game characterization theorem for bounded counting rank implies that $\mathcal{A}_n$ and $\mathcal{A}_m$ verify the same $\mathcal{C}^t[k]$ sentences.
We now prove by induction on the moves (on i) that the inequalities can be maintained. The base case of $i = 0$ is immediate.
For the induction step, assume the inequalities hold for i moves, and suppose without loss of generality that Spoiler makes his $(i+1)^{\text{th}}$ move on $\mathcal{A}_n$. Spoiler plays a set M of cardinality at most t in $\mathcal{A}_n$. We describe Duplicator's response, a set N on $\mathcal{A}_m$. In the second part of the move, if Spoiler picks N_q on $\mathcal{A}_m$ for $q \in [t]$ (i.e. $b_{i+1} = N_q$), Duplicator will respond with M_q on $\mathcal{A}_n$ (i.e. $a_{i+1} = M_q$).
If $a_j = M_q$ for $j \leq i$ and $q \in [t]$, Duplicator sets N_q to be b_j. Suppose there is an element $M_q \in M$ which has not been selected earlier. We define $j, l \leq i$ such that $a_j < M_q < a_l$ and there is no other previously played moves on $\mathcal{A}_n$ inside this interval. By Property 3., the interval between b_j and b_l contains no other elements of b. Then we have two cases regarding the length of the interval:

- $d(a_j, a_l) \leq (t+1)^{k-i}$. Then by Property 1. and the inductive hypothesis, $d(b_j, b_l) = d(a_j, a_l)$, and the intervals $[a_j, a_l]$ and $[b_j, b_l]$ are isomorphic. Duplicator picks N_q so that $d(a_j, M_q) = d(b_j, N_q)$ and $d(M_q, a_l) = d(M_q, b_l)$, which ensures that the three properties hold for $i+1$ moves.
- $d(a_j, a_l) > (t+1)^{k-i}$. In this case by Property 2., $d(b_j, b_l) > (t+1)^{k-i}$. We have three possibilities:
  - $d(a_j, M_q) \leq (t+1)^{k-(i+1)}$. Then $d(M_q, a_l) > (t+1)^{k-(i+1)}$, and Duplicator picks N_q on $[b_j, b_l]$ so that $d(b_j, N_q) = d(a_j, M_q)$ maintaining Properties 1 and 3. Since $d(M_q, b_l) > (t+1)^{k-(i+1)}$ this maintains $d(N_q, b_l) > (t+1)^{k-(i+1)}$.
  - $d(M_q, a_l) \leq (t+1)^{k-(i+1)}$, in which case a similar reasoning applies.
  - Otherwise, Spoiler has picked M_q such that both: $d(a_j, M_q) > (t+1)^{k-(i+1)}$ and $d(M_q, a_l) > (t+1)^{k-(i+1)}$. There can be at most t such M_q as $|M| \leq t$.
    Since $d(b_j, b_l) > (t+1)^{k-i}$, there are at least t distinct elements in $\mathcal{A}_m$ between $b_j + (t+1)^{k-(i+1)}$ and $b_l - (t+1)^{k-(i+1)}$. Duplicator picks the first N_q on that interval that is not already in N. This ensures that, $d(b_j, N_q) > (t+1)^{k-(i+1)}$ and $d(N_q, b_l) > (t+1)^{k-(i+1)}$, therefore satisfying all three properties.

Thus, in all the cases, the induction is preserved. ◀

C Proof of the lemmas of Section 7

C.1 Proof of Lemma 26

If **Part a)** fails then there is G^k for which one of the $I^i_{F^k}$ is separated from J_G by $\delta_1(\{u, u'\})$ for $\{u, u'\} \in \{min, x, y, max\}$. As $\delta_1 \leq \delta$ for these pairs, then $\delta(\{u, u'\})$ separates $\langle I, J\rangle$. ◀
If **Part b)** fails then there are $u, u' \in \{x, y, min, max\}$ such that
MIN [d$(\alpha(u), \alpha(u'))$,d$(\beta(u), \beta(u'))] \leq \delta_1(\{z, u\})$ and d$(\alpha(u), \alpha(u')) \neq$d$(\beta(u), \beta(u'))$.
Since $\delta_1(\{z, u\}) \leq \delta(\{u, u'\})$ we have that: d$(\alpha(u), \alpha(u')) \neq$d$(\beta(u), \beta(u'))$ and
MIN[d$(\alpha(u), \alpha(u'))$,d$(\beta(u), \beta(u'))] \leq \delta(\{u, u'\})$, i.e., $\delta(\{u, u'\})$ separates $\langle I, J\rangle$. ◀

C.2 Proof of Lemma 27

If **Property 1** fails then we already know from the previous lemma that δ separates $\langle I, J\rangle$. So assume **Property 1** holds and **Property 2** fails. Then there is an a_i and a $u \in \{x, y, min, max\}$ such that d$(a_i, \alpha(u)) \leq \delta_1(\{z, u\})$ and there is no $b \in \mathcal{U}^{\mathcal{B}}$ that satisfies d$(a_i, \alpha(u)) =$ d$(b, \beta(u))$ and $<$-type$(a_i) = <$-type(b).

Note that there is at most one $b \in \mathcal{U}^{\mathcal{B}}$ such that d$(a_i, \alpha(u)) =$ d$(b, \beta(u))$ and $<$-type$(a_i, \alpha(u)) = <$-type$(b, \beta(u))$. We distinguish two cases depending on whether such an element exists or not.

Case 1. There are a_i and $u \in \{x, y, min, max\}$ such that d$(a_i, \alpha(u)) \leq \delta_1(\{z, u\})$ and there is no $b \in \mathcal{U}^{\mathcal{B}}$ such that d$(a_i, \alpha(u)) =$ d$(b, \beta(u))$ and $<$-type$(a_i, \alpha(u)) = <$-type$(b, \beta(u))$.

There are four cases, we consider min and x (the other two follow by symmetry).

- u is min: It holds that $\delta(\{min, max\}) \geq \delta_1(\{min, z\}) \geq$ d$(a_i, min_A) >$ d(max_B, min_B). Here the first inequality uses the definition of δ, the second is a property of this case and third holds as there is no such b. But d$(max_A, min_A) \geq$ d(a_i, min_A), so d$(max_B, min_B) \neq$ d(max_A, min_A). With d$(max_B, min_B) \leq \delta(\{min, max\})$ this implies that $\delta(\{min, max\})$ separates $\langle I, J\rangle$.
- u is x: The reasoning is similar. If $\alpha(x) < a_i$, $\delta(\{x, max\}) \geq \delta_1(\{x, z\}) \geq$ d$(a_i, \alpha(x)) >$ d$(max_B, \beta(x))$. So d$(max_A, \alpha(x)) \neq$ d$(max_B, \beta(x))$ and d$(max_B, \beta(x)) \leq \delta(\{x, max\})$. Then $\delta(\{x, max\})$ separates $\langle I, J\rangle$. If $\alpha(x) > a_i$, we reach a similar conclusion.
  Note that $\alpha(x) = a_i$ contradicts **Property 1**. ◀

Case 2. Case 1 does not hold, so there is a_i and $u \in \{x, y, min, max\}$ such that $\mathrm{d}(a_i, \alpha(u)) \leq \delta_1(\{z, u\})$ and there is $b \in \mathcal{U}^{\mathcal{B}}$ with $\mathrm{d}(a_i, \alpha(u)) = \mathrm{d}(b, \beta(u))$, $<$-type$(a_i, \alpha(u))$ $=$ $<$-type$(b, \beta(u))$, and $<$-type$(a_i) \neq <$-type(b). As noted above, given a_i, b is unique. Let $u' \in \{x, y, min, max\}$ be such that $<$-type$(a_i, \alpha(u')) \neq <$-type$(b, \beta(u'))$, which exists by $<$-type$(a_i) \neq <$-type(b). By **Property 1**, it holds that $<$-type$(\alpha(u), \alpha(u')) =$ $<$-type$(\beta(u), \beta(u'))$. As $\alpha(u')$ is on a different side of a_i than $\beta(u')$ of b, this implies that both: MIN$[\mathrm{d}(\alpha(u'), \alpha(u)), \mathrm{d}(\beta(u'), \beta(u))] \leq \mathrm{d}(a_i, \alpha(u))$ and $\mathrm{d}(\alpha(u'), \alpha(u)) \neq \mathrm{d}(\beta(u'), \beta(u))$. Since $\mathrm{d}(a_i, \alpha(u)) \leq \delta_1(\{z, u\}) \leq \delta(\{u', u\})$ we get that:
$\mathrm{d}(\alpha(u'), \alpha(u)) \neq \mathrm{d}(\beta(u'), \beta(u))$ and MIN$[\mathrm{d}(\alpha(u'), \alpha(u)), \mathrm{d}(\beta(u'), \beta(u))] \leq \delta(\{u', u\})$. So $\delta(\{u', u\})$ separates$\langle I, J\rangle$. ◀

C.3 Proof of Lemma 28

If **Property 1 or 2** fails, then we already know that δ separates $\langle I, J\rangle$. So assume **Properties 1, 2** hold and **Property 3** fails. For any a_i such that $S_i \neq \emptyset$, we pick $u \in S_i$. By **Property 2**, we can define b such that $\mathrm{d}(a_i, \alpha(u)) = \mathrm{d}(b, \beta(u))$ and $<$-type$(a_i) =$ $<$-type(b). We consider two cases.

Case 1 (Part a) fails): Suppose there is a_i such that $|S_i| \geq 2$ and b does not satisfy $\mathrm{d}(a_i, \alpha(u')) = \mathrm{d}(b, \beta(u'))$ for every $u' \in S_i$. Let $u' \in S_i$ be such that $\mathrm{d}(a_i, \alpha(u')) \neq \mathrm{d}(b, \beta(u'))$. Thus $\mathrm{d}(\alpha(u), \alpha(u')) \neq \mathrm{d}(\beta(u), \beta(u'))$.
Notice that $\mathrm{d}(\alpha(u), \alpha(u')) \leq \mathrm{d}(a_i, \alpha(u)) + \mathrm{d}(a_i, \alpha(u')) \leq \delta_1(\{z, u\}) + \delta_1(\{z, u'\}) \leq \delta(\{u, u'\})$. So $\mathrm{d}(\alpha(u), \alpha(u')) \neq \mathrm{d}(\beta(u), \beta(u'))$ and $\mathrm{d}(\alpha(u), \alpha(u')) \leq \delta(\{u, u'\})$. Therefore $\delta(\{u, u'\})$ separates $\langle I, J\rangle$.

Case 2 (Part b) fails): Suppose there are a_i and $u' \in \{x, y, min, max\}$ such that $\mathrm{d}(a_i, \alpha(u')) > \delta_1(\{z, u'\})$ and b does not satisfy $\mathrm{d}(b, \beta(u')) > \delta_1(\{z, u'\})$.
Then $\mathrm{d}(b, \beta(u')) \leq \delta_1(\{z, u'\}) < \mathrm{d}(a_i, \alpha(u'))$, so we must have that $\mathrm{d}(\beta(u'), \beta(u)) \neq \mathrm{d}(\alpha(u'), \alpha(u))$.
Notice that $\mathrm{d}(\beta(u'), \beta(u)) \leq \mathrm{d}(b, \beta(u)) + \mathrm{d}(b, \beta(u')) \leq \delta_1(\{z, u\}) + \delta_1(\{z, u'\}) \leq \delta(\{u, u'\})$. So $\mathrm{d}(\alpha(u'), \alpha(u)) \neq \mathrm{d}(\beta(u'), \beta(u))$ and $\mathrm{d}(\beta(u'), \beta(u)) \leq \delta(\{u, u'\})$. Therefore $\delta(\{u, u'\})$ separates $\langle I, J\rangle$. ◀

C.4 Proof of Lemma 29

We first define "gap boundaries", a pair from $\{x, y, min, max\}$ bounding a given gap.

▶ **Definition 31** (Gap boundaries). *Let $(\mathcal{U}, \gamma)$ be an interpretation, and let (u, u') be one of $\{(min, x), (x, y), (y, max)\}$. A pair (v, v') is the* gap boundaries *of $Gap_{(\mathcal{U},\gamma)}(\{u, u'\})$ if:*

1. *(v, v') belongs to the corresponding admissible set:*

$$\begin{aligned} (u, u') = (min, x): &\quad (v, v') \in \{(min, x), (min, y), (min, max)\}, \\ (u, u') = (x, y): &\quad (v, v') \in \{(x, y), (min, y), (x, max), (min, max)\}, \\ (u, u') = (y, max): &\quad (v, v') \in \{(y, max), (x, max), (min, max)\}, \end{aligned}$$

2. *the following coverage inequality holds:*

$$\delta_1(\{z, v\}) + |Gap_{(\mathcal{U},\gamma)}(\{u, u'\})| + \delta_1(\{z, v'\}) \geq \mathrm{d}(\gamma(v), \gamma(v')).$$

Among all such pairs, we choose the one with the smallest left component and the largest right component.

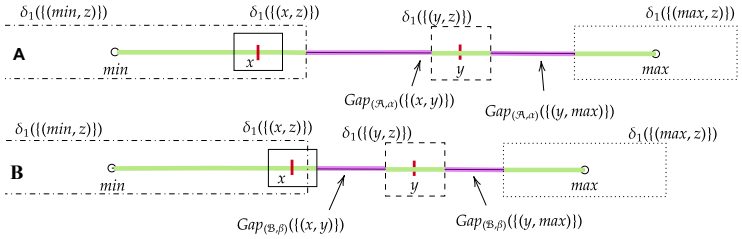

Figure 5 Gap construction for variables $u \in \{min, x, y, max\}$ on interpretations $\mathcal{A}$ (top) and $\mathcal{B}$ (bottom). Each variable u is represented by a box centered at $\alpha(u)$ (resp. $\beta(u)$) with half-length $\delta_1(\{u, z\})$, depicting the separating power of the pair $\{u, z\}$. Regions not covered by these intervals form the *gaps* (in purple). On $\mathcal{A}$, the $\{min, x\}$ gap is empty; the $\{x, y\}$ gap has boundaries (min, y); and the $\{y, max\}$ gap has boundaries (y, max). On $\mathcal{B}$, the $\{min, x\}$ gap is also empty; the $\{x, y\}$ gap boundaries are (x, y); and the $\{y, max\}$ gap boundaries remain (y, max).

The definition is illustrated in Fig. 5. The gap boundaries for gap $\{u, u'\}$ are always defined, and may differ from $\{u, u'\}$. For example, the gap $\{min, x\}$ might have gap boundaries $\{min, y\}$ when the $\delta_1(y, z)$ neighborhood of y contains the $\delta_1(x, z)$ neighborhood of x, hence covering a bigger chunk of the $\{min, x\}$ interval. If the gap is nonempty, the inequalities in Definition 31 become equalities, as the two segments corresponding to the gap boundaries are non-overlapping, and so the two δ_1 values added to the size of the gap add up to the length of the interval.

We derive a result for gap boundaries before proceeding to the next case.

▶ **Proposition 32.** *If* **Property 1** *holds, then the gap boundaries associated with* $Gap_{(\mathcal{A},\alpha)}(\{u, u'\})$ *and* $Gap_{(\mathcal{B},\beta)}(\{u, u'\})$ *are the same.*

Proof. Let (v, v') be the gap boundaries associated with $Gap_{(\mathcal{A},\alpha)}(\{u, u'\})$, and (w, w') be the gap boundaries associated with $Gap_{(\mathcal{B},\beta)}(\{u, u'\})$.
We will first reason from $(\mathcal{A}, \alpha)$ to $(\mathcal{B}, \beta)$, and suppose that $\alpha(v) < \alpha(w) \leq \alpha(u)$. By the definition of the gap on $(\mathcal{A}, \alpha)$, it must be that $\delta_1(\{z, v\}) \geq \mathrm{d}(\alpha(w), \alpha(v)) + \delta_1(\{z, w\})$. **Property 1** implies that $\mathrm{d}(\alpha(w), \alpha(u)) = \mathrm{d}(\beta(w), \beta(u))$ since $\delta\{v, w\} \geq \delta_1(\{z, v\}) \geq \mathrm{d}(\alpha(w), \alpha(v))$. So $\delta_1(\{z, v\}) \geq \mathrm{d}(\beta(w), \beta(v)) + \delta_1(\{z, w\})$. Therefore w cannot be a gap variable for $Gap_{(\mathcal{B},\beta)}(\{u, u'\})$, as otherwise v would have chosen on $(\mathcal{B}, \beta)$ instead of w.
If $\alpha(w) < \alpha(v)$, then by **Property 1** $\beta(w) < \beta(v)$, and the same reasoning applies on the gap on $(\mathcal{B}, \beta)$. By symmetry, the same analysis holds for u', v', w'. ◀

We now proceed to the proof of the lemma. Suppose again that **Property 1** holds and **Property 4** fails.
There is $(u, u') \in \{(min, x), (x, y), (y, max)\}$ such that there are p choices of F^k in $Gap_{(\mathcal{A},\alpha)}$ and $|Gap_{(\mathcal{B},\beta)}(\{u, u'\})| < p$.
By Proposition 32, there is a unique pair (v, v') of gap boundaries for both $Gap_{(\mathcal{A},\alpha)}(\{u, u'\})$ and $Gap_{(\mathcal{B},\beta)}(\{u, u'\})$. On one hand by the definition of gap boundaries on $(\mathcal{B}, \beta)$, we have

$$\begin{aligned}\mathrm{d}(\beta(v'), \beta(v)) &\leq \delta_1(\{z, v\}) + \delta_1(\{z, v'\}) + |Gap_{(\mathcal{B},\beta)}(\{u, u'\})| \\ &\leq \delta_1(\{z, v\}) + \delta_1(\{z, v'\}) + (p-1) \leq \\ &\leq \delta_1(\{z, v\}) + \delta_1(\{z, v'\}) + (k-1) \leq \delta(\{v, v'\})\end{aligned}$$

On the other hand by the definition of gap boundaries on $(\mathcal{A}, \alpha)$,

$$\begin{aligned}\mathrm{d}(\alpha(v'), \alpha(v)) &= \delta_1(\{z, v\}) + \delta_1(\{z, v'\}) + |Gap_{(\mathcal{A},\alpha)}(\{u, u'\})| \\ &\geq \delta_1(\{z, v\}) + \delta_1(\{z, v'\}) + p > \mathrm{d}(\beta(v'), \beta(v)).\end{aligned}$$

Thus $\delta(\{v, v'\})$ separates $\langle I, J\rangle$. ◀