On the Number of Quantifiers Needed to Define Boolean Functions

Marco Carmosino
MIT-IBM Watson AI Lab, Cambridge, MA, USA

Ronald Fagin
IBM Research-Almaden, San Jose, CA, USA

Neil Immerman
University of Massachusetts, Amherst, MA, USA

Phokion G. Kolaitis
University of California Santa Cruz, CA, USA
IBM Research-Almaden, San Jose, CA, USA

Jonathan Lenchner[1]
IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

Rik Sengupta
MIT-IBM Watson AI Lab, Cambridge, MA, USA

Abstract

The number of quantifiers needed to express first-order (FO) properties is captured by two-player combinatorial games called *multi-structural* games. We analyze these games on binary strings with an ordering relation, using a technique we call *parallel play*, which significantly reduces the number of quantifiers needed in many cases. Ordered structures such as strings have historically been notoriously difficult to analyze in the context of these and similar games. Nevertheless, in this paper, we provide essentially tight bounds on the number of quantifiers needed to characterize different-sized subsets of strings. The results immediately give bounds on the number of quantifiers necessary to define several different classes of Boolean functions. One of our results is analogous to Lupanov's upper bounds on circuit size and formula size in propositional logic: we show that every Boolean function on n-bit inputs can be defined by a FO sentence having $(1+\varepsilon)\frac{n}{\log(n)} + O(1)$ quantifiers, and that this is essentially tight. We reduce this number to $(1+\varepsilon)\log(n) + O(1)$ when the Boolean function in question is sparse.

2012 ACM Subject Classification Theory of computation → Complexity theory and logic; Theory of computation → Computational complexity and cryptography

Keywords and phrases logic, combinatorial games, Boolean functions, quantifier number

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.34

Related Version *Full Version*: `https://arxiv.org/abs/2407.00688` [2]

Funding *Rik Sengupta*: Supported by NSF CCF-1934846.

Acknowledgements The authors acknowledge Ryan Williams for numerous helpful discussions and conversations, Sebastian Pfau for an observation that improved the statement of the Parallel Play Lemma, and the anonymous reviewers for comments and suggestions that improved the quality of this manuscript.

[1] Corresponding author

1 Introduction

In 1981, Immerman [11] introduced *quantifier number* (QN) as a measure of the complexity of first-order (FO) sentences. For a function $g\colon \mathbb{N} \to \mathbb{N}$, he defined $\mathrm{QN}[g(n)]$ as the class of properties on n-element structures describable by a uniform sequence of FO sentences with $O(g(n))$ quantifiers. He then showed that on *ordered* structures, for $f(n) \geq \log n$, one has:

$$\mathrm{NSPACE}[f(n)] \subseteq \mathrm{QN}[(f(n))^2/\log n] \subseteq \mathrm{DSPACE}[(f(n))^2], \tag{1}$$

thereby establishing an important connection between QN and space complexity and so directly linking a logical object to classical complexity classes.

The same paper [11] described a two-player combinatorial game (which Immerman called the *separability game*), that captures quantifier number in the same way that the more well-known Ehrenfeucht-Fraïssé (EF) game [3,7] captures quantifier rank (QR). The paper additionally showed that any property whatsoever of n-element *ordered* structures can be described with a sentence having a QR of $\log n + 3$. Since a QR of $\log n + 1$ is required just to distinguish a linear order of size n from smaller linear orders [17], QR has limited power to distinguish properties over ordered structures. QN is potentially a more fine-grained and powerful measure for this purpose. However, owing to the inherent difficulties of the analysis of Immerman's separability game, the study of the game and of QN in general lay dormant for forty years, until the game was rediscovered and renamed the *multi-structural* (MS) game in [4]. In that paper the authors made initial inroads into understanding how to analyze the game, leading to several follow-up works [1,5,18]. Other related games to study the number of quantifiers were recently introduced in [9], and close cousins of MS games were used to study formula size in [8,10]. In [10] the authors study a related problem to ours – they examine the (existential) sentences of minimum size needed to express a particular set of string properties. However, even without the existential restriction, the connection between the minimum size of a sentence and its minimum number of quantifiers is not obvious. It is possible for a property to be expressible only by a much longer sentence with fewer quantifiers than one with more quantifiers.

The MS game is played by two players, Spoiler (**S**, he/him) and Duplicator (**D**, she/her), on two *sets* $\mathcal{A}, \mathcal{B}$ of structures. Essentially, **S** tries to break all partial isomorphisms between all pairs of structures (one from $\mathcal{A}$ and the other from $\mathcal{B}$) over a prescribed number of rounds, whereas **D** tries to maintain a partial isomorphism between *some* pair of structures. Unlike in EF games, **D** has more power in MS games, since she can make arbitrarily many copies of structures before her moves, enabling her to play all possible responses to **S**'s moves. The fundamental theorem for MS games [4,11] (see Theorem 1) states that **S** has a winning strategy for the r-round MS game on $(\mathcal{A}, \mathcal{B})$ if and only if there is a FO sentence φ with at most r quantifiers that is true for every structure in $\mathcal{A}$ but false for every structure in $\mathcal{B}$. We call such a φ a *separating sentence* for $(\mathcal{A}, \mathcal{B})$. In general, our eventual objective will be to separate a set $\mathcal{A}$ of n-bit strings from all other n-bit strings (i.e., from its complement $\mathcal{A}^{\mathrm{C}}$). This is a particularly interesting question because of its intimate connection to the complexity of *Boolean functions*.

Boolean Functions. Any Boolean function on n-bit strings is specified by two complementary sets, $\mathcal{A}, \mathcal{A}^{\mathrm{C}} \subseteq \{0,1\}^n$, representing the input strings that get mapped to 1 and 0 respectively. For such a function $f\colon \{0,1\}^n \to \{0,1\}$, we say that a FO sentence φ in the vocabulary of strings *defines* the function f if φ is a separating sentence for $(f^{-1}(1), f^{-1}(0))$. Hence, the key results of this paper can be thought of as giving sharp bounds on the number of quantifiers needed to define Boolean functions. Our main results about the definability of Boolean functions are Theorems A and B below.

▶ **Theorem A.** *Given an arbitrary $\varepsilon > 0$, every Boolean function on n-bit strings can be defined by a FO sentence having $(1+\varepsilon)\frac{n}{\log(n)} + O_\varepsilon(1)$ quantifiers, where the $O_\varepsilon(1)$ additive term depends only on ε and not n. Moreover, there are Boolean functions on n-bit strings that require $\frac{n}{\log(n)} + O(1)$ quantifiers to define.*

Say that a family, $\mathscr{F} = \{f_n\}_{n=1}^{\infty}$, of Boolean functions on n-bit strings, is *sparse* if the cardinality of the set of strings mapping to 1 under each f_n is polynomial in n. For example, if $\mathcal{L}$ is a sparse language, then the family of Boolean functions, defined for each n, by the characteristic function of $\mathcal{L}$ restricted to n-bit inputs, is sparse [6, 15].

▶ **Theorem B.** *Given an arbitrary $\varepsilon > 0$, and a sparse family, $\mathscr{F} = \{f_n\}_{n=1}^{\infty}$, of Boolean functions on n-bit strings, each function $f_n \in \mathscr{F}$ can be defined by a FO sentence having $(1+\varepsilon)\log(n) + O_\varepsilon(1)$ quantifiers, where the $O_\varepsilon(1)$ additive term depends only on ε and not n. Moreover, there are sparse families of Boolean functions on n-bit strings, the functions of which require $\log(n)$ quantifiers to define.*

Theorem A follows from Theorems 18 and 19 (in Section 5), whereas Theorem B follows from Theorem 16 and Proposition 14 (in Section 5). Theorem 18 can be viewed as a first-order logic analog of the upper bounds obtained by Lupanov for minimum circuit size [13] and minimum propositional formula size [14] to capture an arbitrary Boolean function. Note that *any* property whatsoever of n-bit strings can be captured trivially by a sentence with n existential quantifiers. Similar to Lupanov's bounds, our result shows that we can shave off a factor of $\log(n)$ from this trivial upper bound. Furthermore, Theorem 19 establishes via a counting argument that there are functions with a QN lower bound that essentially matches our worst-case upper bound – a result that can be viewed as a first-order logic analog of the Riordan-Shannon lower bound [16] for propositional formula size.

Parallel Play. A key technical contribution we make in this paper is the Spoiler strategy of *parallel play*, which widens the scope of winning strategies for **S** compared to previous work. The essential idea is for **S** to partition the sets $\mathcal{A}$ and $\mathcal{B}$ into subsets $\mathcal{A}_1 \sqcup \ldots \sqcup \mathcal{A}_k$ and $\mathcal{B}_1 \sqcup \ldots \sqcup \mathcal{B}_k$, and then play k MS "sub-games" in parallel on $(\mathcal{A}_i, \mathcal{B}_i)$. In certain circumstances, **S** can then combine his strategies for each of those sub-games into a strategy for the entire game, and thereby save many superfluous moves. Applying the fundamental theorem, this results in a very small number of quantifiers in the corresponding separating sentence.

Outline of the Paper. This paper is organized as follows. In Section 2, we set up some preliminaries. In Section 3, we precisely formulate what we call the *Parallel Play Lemma* (Lemma 5) and the *Generalized Parallel Play Lemma* (Lemma 6). In Section 4, we develop results on linear orders that are similar to but more nuanced than those in [4, 5], with the extra nuance being critical for our subsequent string separation results. In Section 5, we present our results on separating disjoint sets of strings. In Section 6, we wrap up with some conclusions and open problems.

Owing to space constraints, in some places we provide proof sketches, and refer the reader to the full proofs in the appendix of the full version of the paper [2].

2 Preliminaries

Fix a vocabulary τ with finitely many relation and constant symbols. We typically designate structures in boldface ($\mathbf{A}$), their universes in capital letters (A), and sets of structures in calligraphic typeface ($\mathcal{A}$). This last convention includes sets of pebbled structures (see below).

We always use $\log(\cdot)$ to designate the base-2 logarithm. Furthermore, in several results in Section 5, we have an $O(1)$ additive term. This term will always be independent of n. Any additional dependence will be stated in the form of a subscript on the O, e.g., $O_t(1)$ would denote a term independent of n, but dependent on the choice of some parameter t.

Pebbled Structures and Matching Pairs. Consider a palette $\mathcal{C} = \{\mathrm{r}, \mathrm{b}, \mathrm{g}, \ldots\}$ of *pebble colors*, with infinitely many pebbles of each color available. A τ-structure $\mathbf{A}$ is *pebbled* if some of its elements $a_1, a_2, \ldots \in A$ have pebbles on them. There can be at most one pebble of each color on a pebbled structure. There can be multiple pebbles (of different colors) on the same element $a_i \in A$. Occasionally, when the context is clear, we will use the term *board* synonymously with "pebbled structure".

If $\mathbf{A}$ is a τ-structure, and the first few pebbles are placed on elements $a_1, a_2, a_3 \ldots \in A$, we designate the resulting pebbled τ-structure as $\langle \mathbf{A} \mid a_1, a_2, a_3, \ldots \rangle$. Note that $\mathbf{A}$ can be viewed as a pebbled structure $\langle \mathbf{A} \mid \rangle$ with the empty set of pebbles.

By convention, we use r, b, and g for the first three pebbles we play (in that order), as a visual aid in our proofs. Hence, the pebbled structure $\langle \mathbf{A} \mid a_1, a_2, a_3 \rangle$ has pebbles r on $a_1 \in A$, b on $a_2 \in A$, and g on $a_3 \in A$. Note that a_1, a_2, and a_3 need not be distinct.

We say that the pebbled structures $\langle \mathbf{A} \mid a_1, \ldots, a_k \rangle$ and $\langle \mathbf{B} \mid b_1, \ldots, b_k \rangle$ are a *matching pair* if the map $f\colon A \to B$ defined by:

- $f(a_i) = b_i$ for all $1 \le i \le k$
- $f(c^{\mathbf{A}}) = c^{\mathbf{B}}$ for all constants c in τ

is an isomorphism on the induced substructures. Note that $\langle \mathbf{A} \mid a_1, \ldots, a_k \rangle$ and $\langle \mathbf{B} \mid b_1, \ldots, b_k \rangle$ can form a matching pair even when $\mathbf{A} \ncong \mathbf{B}$.

Multi-Structural Games. Assume $r \in \mathbb{N}$, and let $\mathcal{A}$ and $\mathcal{B}$ be two sets of pebbled structures, each pebbled with the *same* set $\{x_1, \ldots, x_k\} \subseteq \mathcal{C}$ of pebble colors. The *r-round multi-structural (MS) game on* $(\mathcal{A}, \mathcal{B})$ is defined as the following two-player game, played by two players, **Spoiler** (**S**, he/him) and **Duplicator** (**D**, she/her). In each round i for $1 \le i \le r$, **S** chooses either $\mathcal{A}$ or $\mathcal{B}$, and an **unused** color $y_i \in \mathcal{C}$; he then places ("plays") a pebble of color y_i on an element of *every* board in the chosen set ("side"). In response, **D** makes as many copies as she wants of each board on the other side, and plays a pebble of color y_i on an element of each of those boards. **D** wins the game if at the end of round r, there is a board in $\mathcal{A}$ and a board in $\mathcal{B}$ forming a matching pair. Otherwise, **S** wins. For readability, we always call the two sets $\mathcal{A}$ and $\mathcal{B}$, even though the structures change over the course of a game in two ways:

- $\mathcal{A}$ or $\mathcal{B}$ can increase in size over the r rounds, as **D** can make copies of the boards.
- The number of pebbles on each of the boards in $\mathcal{A}$ and $\mathcal{B}$ increases by 1 in each round.

We usually refer to $\mathcal{A}$ as the *left* side, and $\mathcal{B}$ as the *right* side.

Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of pebbled structures, with each pebbled structure containing pebbles colored with $\{x_1, \ldots, x_k\} \subseteq \mathcal{C}$. Let $\varphi(x_1, \ldots, x_k)$ be a FO formula with free variables $\{x_1, \ldots, x_k\}$. We say φ is a *separating formula* for $(\mathcal{A}, \mathcal{B})$ (or φ *separates* $\mathcal{A}$ and $\mathcal{B}$) if:

- for every $\langle \mathbf{A} \mid a_1, \ldots, a_k \rangle \in \mathcal{A}$ we have $\mathbf{A}[a_1/x_1, \ldots, a_k/x_k] \models \varphi$,
- for every $\langle \mathbf{B} \mid b_1, \ldots, b_k \rangle \in \mathcal{B}$ we have $\mathbf{B}[b_1/x_1, \ldots, b_k/x_k] \models \neg\varphi$.

The following key theorem [4,11], stated here without proof, relates the logical characterization of a separating formula with the combinatorial property of a game strategy.

▶ **Theorem 1** (Fundamental Theorem of MS Games, [4, 11]). **S** *has a winning strategy in the r-round MS game on $(\mathcal{A}, \mathcal{B})$ iff there is a formula with $\le r$ quantifiers separating $\mathcal{A}$ and $\mathcal{B}$.*

In the theorem above, if $\mathcal{A}$ and $\mathcal{B}$ are sets of *unpebbled* structures, and φ is a sentence, we call φ a *separating sentence* for $(\mathcal{A}, \mathcal{B})$.

We note that **D** has a clear optimal strategy in the MS game, called the *oblivious* strategy: for each of **S**'s moves, **D** can make enough copies of each pebbled structure on the other side to play all possible responses at the same time. If **D** has a winning strategy, then the oblivious strategy is winning. For this reason, the MS game is essentially a single-player game, where **S** can simulate **D**'s responses himself.

We make an easy observation here without proof, that will help us *discard* some boards during gameplay; we can remove them without affecting the result of the game. This will help us in the analysis of several results in the paper.

▶ **Observation 2.** *During gameplay in any instance of the MS game, consider a board $\langle \mathbf{A} \mid a_1, \ldots, a_k \rangle$ such that there is no board on the other side forming a matching pair with it. Then, $\langle \mathbf{A} \mid a_1, \ldots, a_k \rangle$ can be removed from the game without affecting the result.*

Linear Orders. Let $\tau_{\mathsf{ord}} = \langle <; \ \mathsf{min}, \mathsf{max} \rangle$ be the vocabulary of orders, where $<$ is a binary predicate, and min and max are constant symbols. For every $\ell \geq 1$, we shall use L_ℓ to refer to a structure of type τ_{ord}, which interprets $<$ as a total linear order on $\ell + 1$ elements, and min and max as the first and last elements in that total order respectively. Note that there is only one linear order for any fixed value of ℓ. When unambiguous, we may suppress the subscript and refer to the linear order as simply L.

We define the *length* of a linear order L as the size of its universe minus one (equivalently, as the number of edges if the linear order were represented as a path graph). Hence, the length of L_ℓ is ℓ. Since we only consider $\ell \geq 1$, the length is always positive, and min and max are necessarily distinct. Our convention is different from [4] and [5], where the length of a linear order was the number of elements, and the vocabulary had no built-in constants. Note that having min and max is purely for convenience; each can be defined and reused at the cost of two quantifiers.

Let L be a linear order with elements $a < b$. The linear order $L[a, b]$ is the induced linear order on all elements from a to b, both inclusive. If the variables x and y have been interpreted by L so that $x^L = a$ and $y^L = b$, then we shall use $L[x, y]$ and $L[a, b]$ interchangeably; we adopt a similar convention for constants. If pebbles r and b have been placed on L on a and b respectively, we use $L[\mathrm{r}, \mathrm{b}]$ to mean $L[a, b]$.

We will frequently need to consider sets of linear orders. For $\ell \geq 1$, we will use the notation $L_{\leq \ell}$ to denote the set of linear orders of length at most ℓ, and $L_{>\ell}$ to denote the set of linear orders of length greater than ℓ.

Strings. Let $\tau_{\mathsf{string}} = \langle <, \ S \ ; \ \mathsf{min}, \mathsf{max} \rangle$ be the vocabulary of binary strings, where $<$ is a binary predicate, S is a unary predicate, and min and max are constant symbols. We encode a string $w = (w_1, \ldots, w_n) \in \{0, 1\}^n$ by the τ_{string}-structure $\mathbf{B}_w$ having universe $B_w = \{1, \ldots, n\}$, relation $<$ interpreted by the linear order on $\{1, \ldots, n\}$, relation $S = \{i \mid w_i = 1\}$, and min and max interpreted as 1 and n respectively.

For an n-bit string w, and i, j such that $1 \leq i \leq j \leq n$, denote by $w[i, j]$ the substring $w_i \ldots w_j$ of w. Note that $w[i, j]$ corresponds to the induced substructure of $\mathbf{B}_w$ on $\{i, \ldots, j\}$. We will often interchangeably talk about the string w and the τ_{string}-structure $\mathbf{B}_w$, when the context is clear. As in τ_{ord}, having min and max in the vocabulary is purely for convenience.

3 Parallel Play

In this section, we prove our key lemma, that shows how, in certain cases, **S** can combine his winning strategies in two sub-games, playing them in parallel in a single game that requires no more rounds than the longer of the two sub-games.

To understand why this is helpful, note that in general, if a formula φ is of the form $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \vee \varphi_2$, the number of quantifiers in φ is the sum of the number of quantifiers in φ_1 and φ_2, even if the two subformulas have the same quantifier structure. We will see that playing parallel sub-games roughly corresponds to taking a φ of the form $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \vee \varphi_2$ where the subformulas have the same quantifier prefix, and writing φ with the same quantifier prefix as φ_1 or φ_2, saving half the quantifiers we normally require.

Suppose **S** has a winning strategy for an instance $(\mathcal{A}, \mathcal{B})$ of the r-round MS game. In principle, the choice of which side **S** plays on could depend on **D**'s previous responses. However, note that any strategy $\mathcal{S}$ used by **S** that wins against the oblivious strategy also wins against any other strategy that **D** plays. Therefore, we may WLOG restrict ourselves to strategies used by **S** against **D**'s oblivious strategy. It follows that the choice of which side to play on in every round is completely determined by the instance $(\mathcal{A}, \mathcal{B})$, and independent of any of **D**'s responses. Let $\mathcal{S}$ be such a winning strategy for **S**. We now define the *pattern* of $\mathcal{S}$, which specifies which side **S** plays on in each round, when following $\mathcal{S}$.

▶ **Definition 3.** *Suppose $\mathcal{A}$ and $\mathcal{B}$ are sets of pebbled structures, and assume that* **S** *has a winning strategy $\mathcal{S}$ for the r-round* MS *game on $(\mathcal{A}, \mathcal{B})$. The* pattern *of $\mathcal{S}$, denoted* $\mathsf{pat}(\mathcal{S})$*, is an r-tuple $(Q_1, \dots, Q_r) \in \{\exists, \forall\}^r$, where:*

$$Q_i = \begin{cases} \exists & \text{if } \mathbf{S} \text{ plays in } \mathcal{A} \text{ in round } i, \\ \forall & \text{if } \mathbf{S} \text{ plays in } \mathcal{B} \text{ in round } i. \end{cases}$$

We say that **S** wins the game with pattern $(Q_1, \dots, Q_r)$ *if* **S** *has a winning strategy $\mathcal{S}$ for the game in which* $\mathsf{pat}(\mathcal{S}) = (Q_1, \dots, Q_r)$*.*

The following lemma is implicit in the proof of Theorem 1.

▶ **Lemma 4.** *For any two sets $\mathcal{A}$ and $\mathcal{B}$ of pebbled τ-structures, the following are equivalent:*
1. **S** *wins the r-round* MS *game on $(\mathcal{A}, \mathcal{B})$ with pattern $(Q_1, \dots, Q_r)$.*
2. *$(\mathcal{A}, \mathcal{B})$ has a separating formula with r quantifiers and quantifier prefix $(Q_1, \dots, Q_r)$.*

Note that Lemma 4 implies that, as long as there is a separating formula φ for $(\mathcal{A}, \mathcal{B})$ with r quantifiers, **S** has a winning strategy for the r-round MS game on $(\mathcal{A}, \mathcal{B})$ that "follows" φ; namely, if $\varphi = Q_1 \dots Q_r \psi$, then in round i, **S** plays in $\mathcal{A}$ if $Q_i = \exists$, and in $\mathcal{B}$ if $Q_i = \forall$. Hence, for the rest of the paper, we will refer to **S** moves in $\mathcal{A}$ and $\mathcal{B}$ as *existential* and *universal* moves respectively. We are now ready to state our main lemma from this section.

▶ **Lemma 5** (Parallel Play Lemma)**.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of pebbled structures, and let $r \in \mathbb{N}$. Suppose that $\mathcal{A}$ and $\mathcal{B}$ can be partitioned as $\mathcal{A} = \mathcal{A}_1 \sqcup \mathcal{A}_2$ and $\mathcal{B} = \mathcal{B}_1 \sqcup \mathcal{B}_2$ respectively, such that for $1 \le i \le 2$,* **S** *has a winning strategy $\mathcal{S}_i$ for the r-round MS game on $(\mathcal{A}_i, \mathcal{B}_i)$, satisfying the following conditions:*
1. *Both $\mathcal{S}_i$'s have the same pattern $P = \mathsf{pat}(\mathcal{S}_1) = \mathsf{pat}(\mathcal{S}_2)$.*
2. *At the end of the sub-games, both of the following are true:*
   - *There does not exist a board in $\mathcal{A}_1$ and a board in $\mathcal{B}_2$ forming a matching pair.*
   - *There does not exist a board in $\mathcal{A}_2$ and a board in $\mathcal{B}_1$ forming a matching pair.*

Then **S** *wins the r-round MS game on $(\mathcal{A}, \mathcal{B})$ with pattern P.*

Proof. **S** plays the r-round MS game on $(\mathcal{A}, \mathcal{B})$ by playing his winning strategy $\mathcal{S}_1$ on $(\mathcal{A}_1, \mathcal{B}_1)$, and his winning strategy $\mathcal{S}_2$ on $(\mathcal{A}_2, \mathcal{B}_2)$, *simultaneously* in parallel. This is a well-defined strategy, since every $\mathcal{S}_i$ has the same pattern P. At the end of the game:

- for $i = j$, no board from $\mathcal{A}_i$ forms a matching pair with a board from $\mathcal{B}_j$, since **S** wins the sub-game $(\mathcal{A}_i, \mathcal{B}_i)$.
- for $i \neq j$, no board from $\mathcal{A}_i$ forms a matching pair with a board from $\mathcal{B}_j$, by assumption.

Therefore, no matching pair remains after round r, and so, **S** wins the game. The pattern for this strategy is P by construction. ◀

We observe that Lemma 5 can be generalized in two ways. Firstly, we could split into k sub-games instead of two. Secondly, we can weaken assumption 1 in the statement of the lemma, so that each of the patterns is a subsequence of some r-tuple $P = \{\exists, \forall\}^r$. This is because **S** can simply extend the strategy $\mathcal{S}_i$ with pattern P_i to a strategy $\mathcal{S}'_i$ with pattern P, where for every "missing" entry in the tuple P, **S** makes a dummy move on the corresponding side. We state a generalized version below without a proof.

▶ **Lemma 6** (Generalized Parallel Play Lemma). *Let $\mathcal{A}$ and $\mathcal{B}$ be two sets of pebbled structures, and let $r \in \mathbb{N}$. Let $P \in \{\exists, \forall\}^r$ be a sequence of quantifiers of length r. Suppose that $\mathcal{A}$ and $\mathcal{B}$ can be partitioned as $\mathcal{A} = \mathcal{A}_1 \sqcup \ldots \sqcup \mathcal{A}_k$ and $\mathcal{B} = \mathcal{B}_1 \sqcup \ldots \sqcup \mathcal{B}_k$ respectively, such that for all $1 \leq i \leq k$, **S** has a winning strategy $\mathcal{S}_i$ for the r_i-round MS game on $(\mathcal{A}_i, \mathcal{B}_i)$ (where $r_i \leq r$), satisfying the following conditions:*

1. *For all i, $\mathsf{pat}(\mathcal{S}_i)$ is a subsequence of P.*
2. *At the end of the sub-games, for $i \neq j$, there does not exist a board in $\mathcal{A}_i$ and a board in $\mathcal{B}_j$ forming a matching pair.*

*Then **S** wins the r-round MS game on $(\mathcal{A}, \mathcal{B})$ with pattern P.*

Note that Lemmas 5 and 6 can be applied in conjunction with Observation 2 as long as there is at least one structure remaining on either side, since a winning strategy (and therefore its corresponding pattern) is unaffected if some of the pebbled structures in the instance are deleted. Furthermore, in many cases, we can provide a strategy for **S** where condition 2 in Lemmas 5 and Lemma 6 will be automatically met after the first move, and therefore will continue to be satisfied at the end of the game. We shall use these two facts implicitly in the proofs that follow.

4 Linear Orders

As noted in Section 1, the results in this section are similar to those in [4,5], but somewhat more nuanced, leading ultimately to the *quantifier alternation theorems* (Theorems 12 and 13). Instead of the unwieldy function $g(\cdot)$ studied in those papers, we study the simpler function $q(\cdot)$, which, given an integer ℓ, returns the minimum number of quantifiers needed to separate $L_{\leq \ell}$ from $L_{> \ell}$. A key result, not appreciated in [4,5], is that the number of quantifiers needed to separate two linear orders of different sizes never exceeds the quantifier rank needed by more than one (Theorem 11).

Let $r(\ell)$ (resp. $q(\ell)$) be the minimum QR (resp. QN) needed to separate $L_{\leq \ell}$ and $L_{> \ell}$. Let $q_\forall(\ell)$ (resp. $q_\exists(\ell)$) be the minimum number of quantifiers needed to separate $L_{\leq \ell}$ and $L_{> \ell}$ with a sentence whose prenex normal form starts with $\forall$ (resp. $\exists$). Note that $q(\ell) = \min(q_\forall(\ell), q_\exists(\ell))$. The values of $r(\ell)$ are well understood [17]:

▶ **Theorem 7** (Quantifier Rank, [17]). *For $\ell \geq 1$, we have $r(\ell) = 1 + \lfloor \log(\ell) \rfloor$.*

Since QR lower bounds QN, we have $r(\ell) \leq q(\ell)$ for all ℓ. On the other hand, for each $\ell > 0$, we will show that **S** can always separate $L_{\leq \ell}$ from $L_{>\ell}$ in a multi-structural game of at most $r(\ell) + 1$ rounds, which shows that $q(\ell) \leq r(\ell) + 1$.

For notational convenience, we denote by $\mathrm{MSL}_{\exists,r}(\ell)$ an r-round MS game on $(L_{\leq \ell}, L_{>\ell})$, in which **S** *must* play an existential first round move. We use $\mathrm{MSL}_{\forall,r}(\ell)$ analogously, where the first round move *must* be universal. Observe that, *a priori*, any such game may be winnable by either **S** or **D**. Since we are primarily interested in upper bounds, we restrict our attention only to **S**-winnable games. We call such games simply *winnable*.

4.1 The Closest-to-Midpoint with Alternation Strategy

In this section, we describe a divide-and-conquer recursive strategy for **S** to play winnable game instances $\mathrm{MSL}_{Q,r}(\ell)$. This strategy will give us upper bounds on $q_\exists(\ell)$ and $q_\forall(\ell)$, which we will then relate to $r(\ell)$.

We define the *closest-to-midpoint* of a linear order $L[x, y]$ as the element halfway between the elements corresponding to x and y if $L[x, y]$ has even length, or the element just left of center if $L[x, y]$ has odd length.

The **S**-winning strategy is called *Closest-to-Midpoint with Alternation* (CMA). The pattern for this strategy will alternate between $\exists$ and $\forall$, splitting each game recursively into two smaller sub-games that can be played in parallel using Lemma 5. In these sub-games, placed pebbles will take on the roles of min and max. **S** continues in this way until the sub-games are on linear orders of length 2 or less, at which point he can win them easily.

The idea is for **S** to obey the following two rules throughout, except possibly the last three rounds:

- **S** starts on his designated side (determined by Q), and then alternates in every round;
- on every board, **S** plays on the closest-to-midpoint of a linear order $L[x, y]$, chosen carefully to ensure he essentially "halves" the length of the instance every round.

Note that one consequence of the second point above is that **S** will *never* play on max.

Before getting to a formal description of the strategy, let us illustrate the main idea through a worked example. Consider the (winnable) game $\mathrm{MSL}_{\exists,4}(5)$. In round 1, **S** plays on the closest-to-midpoint of all boards in $L_{\leq 5}$ (by the two conditions in the CMA strategy). Before **D**'s response, we reach the position shown in Figure 1.

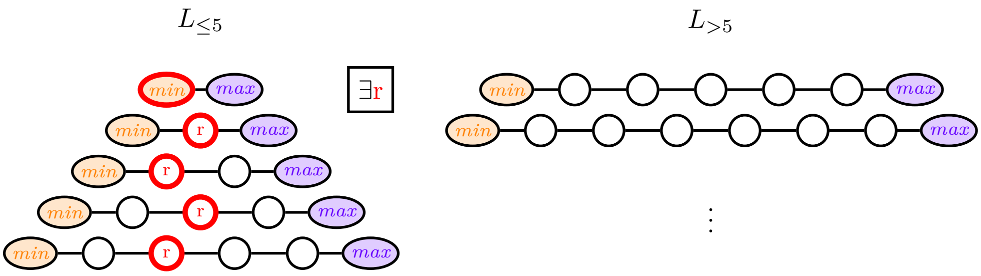

Figure 1 The position after **S**'s round 1 move in the game $\mathrm{MSL}_{\exists,4}(5)$. The pebble r is on the closest-to-midpoint of every board on the left.

Now assume **D** responds obliviously. We can first use Observation 2 to discard all boards on the right with r on max. By virtue of **S**'s first move, every board $\langle L \mid a_1 \rangle$ on the left satisfies *both* $L[\mathsf{min}, \mathrm{r}] \leq 2$, and $L[\mathrm{r}, \mathsf{max}] \leq 3$. Now consider any board $\langle L' \mid a'_1 \rangle$ on the right. Note that either $L'[\mathsf{min}, \mathrm{r}] > 2$, or $L'[\mathrm{r}, \mathsf{max}] > 3$. Partition the right side as $\mathcal{B}_1 \sqcup \mathcal{B}_2$, where every $\langle L' \mid a'_1 \rangle \in \mathcal{B}_1$ satisfies $L'[\mathsf{min}, \mathrm{r}] > 2$, and every $\langle L' \mid a'_1 \rangle \in \mathcal{B}_2$ satisfies $L'[\mathrm{r}, \mathsf{max}] > 3$.

In round 2, **S** makes a universal move (by the first condition in the CMA strategy). In all boards in $\mathcal{B}_1$, he plays pebble b on the closest-to-midpoint of $L'[\mathsf{min}, \mathrm{r}]$; similarly, in all boards in $\mathcal{B}_2$, he plays pebble b on the closest-to-midpoint of $L'[\mathrm{r}, \mathsf{max}]$. Note that in either case, **S** plays b on an element which is not on r, min, or max.

After **D** responds obliviously, we can use Observation 2 to discard all boards on the left where b is on min, max, or r. Since in particular this discards all boards on the left with r on min, we can again use Observation 2 to discard all boards from the right which have r on min. Every remaining board in $\mathcal{B}_1$ (resp. $\mathcal{B}_2$) corresponds to the isomorphism class $\mathsf{min} < \mathrm{b} < \mathrm{r} < \mathsf{max}$ (resp. $\mathsf{min} < \mathrm{r} < \mathrm{b} < \mathsf{max}$). The remaining boards on the left also correspond to exactly one of those classes. Partition the left side as $\mathcal{A}_1 \sqcup \mathcal{A}_2$ accordingly.

Now, because of this difference in isomorphism classes, we will never obtain a matching pair from $\mathcal{A}_1$ and $\mathcal{B}_2$ (or from $\mathcal{A}_2$ and $\mathcal{B}_1$). Furthermore, for the rest of the game, **S** will *only* play inside $L[\mathsf{min}, \mathrm{r}]$ on all boards in $\mathcal{A}_1$ and $\mathcal{B}_1$, and inside $L[\mathrm{r}, \mathsf{max}]$ on all boards in $\mathcal{A}_2$ and $\mathcal{B}_2$. Suppose, in response to such a move on $\mathcal{A}_1$, **D** plays outside the range $L[\mathsf{min}, \mathrm{r}]$ on a board from $\mathcal{B}_1$; the resulting board cannot form a partial match with any board from $\mathcal{A}_1$ (since there is a discrepancy with r), or with any board from $\mathcal{A}_2$ (as observed already). Therefore, this board from $\mathcal{B}_1$ can be discarded using Observation 2. A similar argument applies if **D** ever responds outside the corresponding range in $\mathcal{B}_2$, $\mathcal{A}_1$, or $\mathcal{A}_2$.

It follows that the sub-game $(\mathcal{A}_1, \mathcal{B}_1)$ (resp. $(\mathcal{A}_2, \mathcal{B}_2)$ corresponds *exactly* to the game $\mathrm{MSL}_{\forall,3}(2)$ (resp. $\mathrm{MSL}_{\forall,3}(3)$) where **S** has already made his first move using the CMA strategy by playing a universal move on the closest-to-midpoints of the (relevant) linear orders. Since **S** will alternate sides throughout, the patterns for both sub-game strategies will be the same.

We can now apply Lemma 5. Observe that the lengths of the instances in the sub-games have been roughly halved, at the cost of a single move. The game then proceeds as shown in Figure 2. The leaves of the tree correspond to base cases (analyzed in Section 4.2). The pattern of the strategy is preserved along all branches.

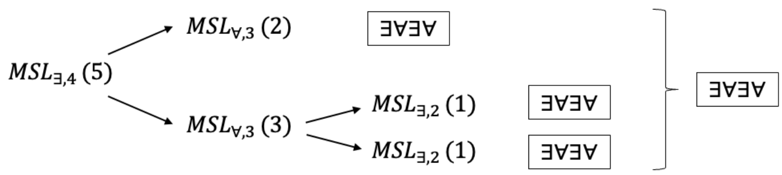

Figure 2 The $\mathrm{MSL}_{\exists,4}(5)$ game tree. Each leaf is decorated with the associated quantifier prefix. All paths can be played in parallel using Lemma 6 using the pattern $(\exists, \forall, \exists, \forall)$.

4.2 Formalizing the Strategy

The first step in formalizing the CMA strategy for **S** is to define four base cases, which we shall call *irreducible* games. We assert the following (see Appendix A in [2]).

1. $\mathrm{MSL}_{\forall,1}(1)$ is winnable with the pattern $(\forall)$.
2. $\mathrm{MSL}_{\exists,2}(1)$ is winnable with the pattern $(\exists, \forall)$.
3. $\mathrm{MSL}_{\forall,2}(2)$ is winnable with the pattern $(\forall, \forall)$.
4. $\mathrm{MSL}_{\forall,3}(2)$ is winnable with the pattern $(\forall, \exists, \forall)$.

The game $\mathrm{MSL}_{\exists,1}(1)$ is not winnable and hence not considered.

We now give a formalization of the inductive step. For a given quantifier $Q \in \{\exists, \forall\}$ and its complementary quantifier $\bar{Q}$, consider the game $\mathrm{MSL}_{Q,k}(\ell)$. Note that if **S** employs the CMA strategy the game splits into the two sub-games $\mathrm{MSL}_{\bar{Q},k-1}(\ell')$ and $\mathrm{MSL}_{\bar{Q},k-1}(\ell'')$. We designate this split as:

$$\mathrm{MSL}_{Q,k}(\ell) \to \mathrm{MSL}_{\bar{Q},k-1}(\ell') \oplus \mathrm{MSL}_{\bar{Q},k-1}(\ell'').$$

We will show in the proof of Lemma 9 that these sub-games can be played recursively, in parallel. When **S** reaches an irreducible sub-game, he plays the winning patterns asserted above. We claim the following about the rules for splitting. The proof is in [2].

▷ Claim 8 (Splitting Rules). For $k \geq 3$, we have:

$$\begin{aligned}
&\text{(i) } \mathrm{MSL}_{\exists,k}(2\ell) \to \mathrm{MSL}_{\forall,k-1}(\ell) \oplus \mathrm{MSL}_{\forall,k-1}(\ell), && \ell \geq 1\\
&\text{(ii) } \mathrm{MSL}_{\exists,k}(2\ell+1) \to \mathrm{MSL}_{\forall,k-1}(\ell) \oplus \mathrm{MSL}_{\forall,k-1}(\ell+1), && \ell \geq 1\\
&\text{(iii) } \mathrm{MSL}_{\forall,k}(2\ell) \to \mathrm{MSL}_{\exists,k-1}(\ell) \oplus \mathrm{MSL}_{\exists,k-1}(\ell-1), && \ell \geq 2\\
&\text{(iv) } \mathrm{MSL}_{\forall,k}(2\ell+1) \to \mathrm{MSL}_{\exists,k-1}(\ell) \oplus \mathrm{MSL}_{\exists,k-1}(\ell), && \ell \geq 1
\end{aligned} \tag{2}$$

Of course, the CMA strategy starts out seemingly promisingly, splitting with both initial sub-games starting on the same side; we must ensure that the strategy continues to be *well-defined*, i.e., this continues throughout the recursion stack, especially since the sub-games can have different lengths. We show this in Lemma 9, whose proof is in [2].

▶ **Lemma 9.** *The* CMA *strategy is well-specified. Moreover, for $k \geq 3$, if $MSL_{Q,k}(\ell) \to MSL_{\bar{Q},k-1}(\ell_1) \oplus MSL_{\bar{Q},k-1}(\ell_2)$ with $\ell_1 \geq \ell_2$, then the pattern of* **S***'s winning strategy for $MSL_{Q,k}(\ell)$ is Q concatenated with the pattern for the winning strategy for $MSL_{\bar{Q},k-1}(\ell_1)$.*

4.3 Bounding and Characterizing the Pattern

Define $q^*_\exists(\ell)$ (resp. $q^*_\forall(\ell)$) as the minimum $r \in \mathbb{N}$ such that **S** wins the game $\mathrm{MSL}_{\exists,r}(\ell)$ (resp. $\mathrm{MSL}_{\forall,r}(\ell)$) using the CMA strategy. Of course, we must have $q_\exists(\ell) \leq q^*_\exists(\ell)$ and $q_\forall(\ell) \leq q^*_\forall(\ell)$. Let $q^*(\ell) = \min(q^*_\exists(\ell), q^*_\forall(\ell))$. The following lemma (whose proof is omitted) follows from the complete description of the strategy from Section 4.2.

▶ **Lemma 10.** *We have $q^*_\forall(1) = 1$, $q^*_\exists(1) = 2$, and $q^*_\forall(2) = 2$. Also:*

$$\begin{aligned}
&q^*_\exists(2\ell) = q^*_\forall(\ell) + 1 \quad \text{for } \ell \geq 1, && q^*_\exists(2\ell+1) = q^*_\forall(\ell+1) + 1 \quad \text{for } \ell \geq 1,\\
&q^*_\forall(2\ell) = q^*_\exists(\ell) + 1 \quad \text{for } \ell \geq 2, && q^*_\forall(2\ell+1) = q^*_\exists(\ell) + 1 \quad \text{for } \ell \geq 1.
\end{aligned}$$

From Lemma 10 it is possible to recursively compute $q^*_\forall(\ell)$ and $q^*_\exists(\ell)$, and therefore $q^*(\ell)$ for all values of $\ell \geq 1$. These values are provided for $\ell \leq 127$ in Table 1.

We now state and prove the main result of this section.

▶ **Theorem 11.** *For all $\ell \geq 1$, we have:*

$$r(\ell) \leq q(\ell) \leq r(\ell) + 1.$$

Proof. The first inequality, $r(\ell) \leq q(\ell)$, is obvious. For the second, we will show that $q^*_\exists(\ell)$ and $q^*_\forall(\ell)$ are both bounded above by $r(\ell) + 1$ (and since $q(\ell) \leq q^*(\ell) = \min(q^*_\exists(\ell), q^*_\forall(\ell))$, so too for $q(\ell)$). Lemma 10 shows that the assertion is true for $\ell \leq 2$. Now, it can be shown recursively (see, e.g., Appendix A in [2]) that $q^*_\forall(2^k) = q^*_\exists(2^k) = k+1$ for $k \geq 1$. By Theorem 7, we also know that $r(2^k) = k+1$ for $k \geq 1$. So the three functions, $r(\cdot)$, $q^*_\forall(\cdot)$, and $q^*_\exists(\cdot)$, all equal each other at successive powers of two, and increase by one between these successive powers. Since all three functions are monotonic, they differ from one another by at most one. Therefore, we have $q^*_\exists(\ell) \leq r(\ell) + 1$ and $q^*_\forall(\ell) \leq r(\ell) + 1$. ◀

Table 1 Values of $q^*_\forall(\ell), q^*_\exists(\ell), q^*(\ell)$ and $r(\ell)$ for $1 \le \ell \le 127$.

ℓ	$q^*_\forall(\ell)$	$q^*_\exists(\ell)$	$q^*(\ell)$	$r(\ell)$
1	1	2	1	1
2	2	2	2	2
3	3	3	3	2
4	3	3	3	3
5	3	4	3	3
6-7	4	4	4	3
8-9	4	4	4	4
10	5	4	4	4
11-15	5	5	5	4
16-18	5	5	5	5
19-21	5	6	5	5
22-31	6	6	6	5
32-37	6	6	6	6
38-42	7	6	6	6
43-63	7	7	7	6
64-75	7	7	7	7
76-85	7	8	7	7
86-127	8	8	8	7

We wrap up this section with two results that will be useful in Section 5. For their proofs, we refer the reader to the full version [2].

▶ **Theorem 12** (Alternation Theorem, Smaller vs. Larger). *For every $\ell \ge 1$, there is a separating sentence σ_ℓ for $(L_{\le \ell}, L_{>\ell})$ with $q^*(\ell)$ quantifiers (and so at most $\log(\ell) + 2$ quantifiers), such that the quantifier prefix of σ_ℓ strictly alternates and ends with a $\forall$.*

▶ **Theorem 13** (Alternation Theorem, One vs. All). *For every $\ell \ge 1$, there is a sentence φ_ℓ separating L_ℓ from all other linear orders having an alternating quantifier prefix (ending with a $\forall$) and consisting of $q^*(\ell) + 2$ quantifiers (and so at most $\log(\ell) + 4$ quantifiers).*

5 Strings

In this section, we pursue our main objective: string separation results, in order to characterize the complexity of Boolean functions. We would like to bound the number of quantifiers required for these separations as a function of both the length n of the strings, as well as the sizes of the sets.

In general, we would like to separate a set of n-bit strings from the set of all other n-bit strings; recall from Section 1 that we can think of this as separating the 1 instances from the 0 instances for a Boolean function on n-bit inputs. To do so, we first need to develop a basic technique for *distinguishing* one string from another.

▶ **Proposition 14** (One vs. One). ***Upper Bound:*** *For every pair w, w' of n-bit strings such that $w \ne w'$, there is a sentence $\varphi_{w,w'}$ with $\log(n) + 6$ quantifiers separating $(\{w\}, \{w'\})$. This sentence $\varphi_{w,w'}$ (in prenex form) has an alternating quantifier prefix ending with $\forall$.*
Lower Bound: *For all sufficiently large n, there exist two n-bit strings w, w', such that separating them requires $\lfloor \log(n) \rfloor$ quantifiers.*

Proof.

Upper Bound. Let $w, w' \in \{0,1\}^n$ be any two distinct n-bit strings. There is an index $i \in [n]$ such that $w_i \neq w'_i$. Let $\mathcal{A} = \{w\}$ and $\mathcal{B} = \{w'\}$. We will show that **S** wins the MS game on $(\mathcal{A}, \mathcal{B})$ in $\log(n) + 6$ rounds.

In round 1, **S** plays pebble r on the $\mathcal{A}$ side, on the element w_i in w, creating the pebbled string $\langle w \mid w_i \rangle$. Assume **D** responds obliviously on the $\mathcal{B}$ side. We can now immediately use Observation 2 to discard the resulting pebbled string $\langle w' \mid w'_i \rangle \in \mathcal{B}$, where the pebble r is on the element w'_i. Every remaining board in $\mathcal{B}$ is of the form $\langle w' \mid w'_j \rangle$, for $j \neq i$. Note that the substring $w'[1,j]$ has length j, which is different from i, the length of the substring $w[1,i]$ of $w \in \mathcal{A}$. So now, **S** can simply play the strategy from Theorem 13 to separate a linear order of length i from all other linear orders, which he wins in $\log(n) + 4$ rounds with an alternating pattern. This gives us the desired result, after at most one more dummy move to preserve alternation.

Lower Bound. Let $\ell = 2^k + 2$ for $k > 1$, and let $w = 0^{2^{k-1}}100^{2^{k-1}}$ and $w' = 0^{2^{k-1}}010^{2^{k-1}}$. If **S** plays entirely on one side of the respective 1s then he is effectively playing the MS game on $(L_{2^{k-1}}, L_{2^{k-1}-1})$. By Theorem 7, we have $r(2^{k-1}) = k = \lfloor \log(\ell) \rfloor$. Since QR lower bounds QN, the MS game played in this fashion requires at least $\lfloor \log(\ell) \rfloor$ rounds to win.

Now suppose that instead of playing entirely on the same side of the respective 1s, **S** plays on both sides of a 1 and/or on the 1 during these $\lfloor \log(\ell) \rfloor$ rounds. In this case, **D** can play obliviously to the left of the 1 when **S** plays to the left of the 1, obliviously to the right of the 1 when **S** plays to the right of the 1, and on the 1 whenever **S** plays on the 1, thereby keeping matching pairs simultaneously on both sides. The lower bound follows. ◀

We also need another helpful lemma, whose proof is in Appendix B of [2].

▶ **Lemma 15.** *Let $f\colon \mathbb{N} \to \mathbb{N}$ be a function satisfying $\lim_{n\to\infty} f(n) = \infty$, and let $t \geq 2$ be any integer. Then, for some number $N(t)$ depending on t, for all $n \geq N(t)$, we have $\lceil \log_t(f(n)) \rceil! \geq f(n)$.*

We now start with our string separation problems. The first problem we will consider will be when there is a single n-bit string in $\mathcal{A}$, and the $2^n - 1$ remaining n-bit strings in $\mathcal{B}$. Note that this corresponds to our Boolean function of interest being an indicator function.

▶ **Theorem 16** (One vs. All). *For all n, and for every $\varepsilon > 0$, it is possible to separate each n-bit string from all other n-bit strings by a sentence with $(1+\varepsilon)\log(n) + O_\varepsilon(1)$ quantifiers. This sentence (in prenex form) starts with a $\forall$, then has at most $\varepsilon \log(n) + 1$ occurrences of $\exists$, and then ends with an alternating quantifier prefix of length at most $\log(n) + O_\varepsilon(1)$.*

Proof Sketch (see [2] for full proof). Fix any $\varepsilon > 0$, and fix any integer $t \geq 2^{1/\varepsilon}$. By Lemma 15, we know there is some integer $N(t)$, such that for all $n \geq N(t)$, $\lceil \log_t(n) \rceil! \geq n$. For any such n, fix an arbitrary $w \in \{0,1\}^n$, and let $\mathcal{A} = \{w\}$, and $\mathcal{B} = \{0,1\}^n - \{w\}$.

Consider the MS game on $(\mathcal{A}, \mathcal{B})$. Every $w' \in \mathcal{B}$ differs from w in at least one bit. In round 1, **S** plays a universal move, placing a pebble on each $w' \in \mathcal{B}$ on an index that disagrees with w at that index. Assume **D** responds obliviously, so that there are n resulting pebbled strings in $\mathcal{A}$. For the next $\lceil \log_t(n) \rceil$ rounds, **S** plays only existential moves, placing the $\lceil \log_t(n) \rceil$ pebbles in distinct permutations on the n strings in $\mathcal{A}$, creating n distinct isomorphism classes[2] by Lemma 15. Once we discard structures from the two sides using Observation 2,

[2] An *isomorphism class* is a maximal set of partially isomorphic pebbled structures.

we are now left with n isomorphism classes, each of them defining a **one-vs-all** sub-game; in each of these sub-games, the round 1 pebble is placed at a different index in the single string on the left from any string on the right. Therefore, **S** can view this as a game simply about lengths, and can employ any **one-vs-all** linear order strategy. The entire game therefore reduces to n parallel instances of **one-vs-all** sub-games on linear orders.

By Lemma 6 and Theorem 13, **S** can now win these parallel games in $\log(n) + 4$ further moves. Together with the initial universal move and the preprocessing moves, the total number of rounds is:

$$\lceil \log_t(n) \rceil + \log(n) + 5 \leq \frac{\log(n)}{\log(t)} + \log(n) + 6 = \log(n)\left(1 + \frac{1}{\log(t)}\right) + 6 \leq (1+\varepsilon)\log(n) + 6.$$

Note that $N(t)$ depends only on t, which in turn depends only on ε. So when $n < N(t)$, the number of quantifiers can be absorbed directly into the $O_\varepsilon(1)$ additive term. ◀

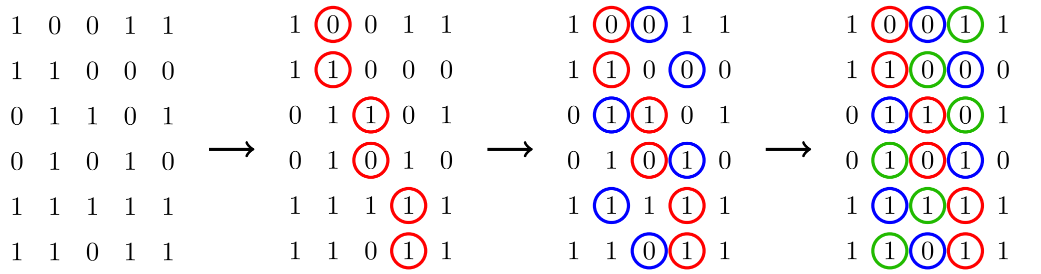

Figure 3 Illustration of the technique used by **S** to partition a set of structures into isomorphism classes. Here **S** plays three pebble moves to break the set of six strings into distinct isomorphism classes: r < b < g, r < g < b, and so on. Note that three pebbling moves suffice to give each string its own isomorphism class since $3! = 6$.

The next problem we will consider has polynomially many n-bit strings in $\mathcal{A}$, and the remaining n-bit strings in $\mathcal{B}$. This will correspond to our Boolean function of interest being a *sparse* function. Note that this immediately implies Theorem B in Section 1.

▶ **Theorem 17** (Polynomially Many vs. All). *Let $f\colon \mathbb{N} \to \mathbb{N}$ be a function that satisfies $\lim_{n\to\infty} f(n) = \infty$ and $f(n) = O(n^k)$ for some constant k. Then, for all n, and for every $\varepsilon > 0$, it is possible to separate each set of $f(n)$ n-bit strings from all other n-bit strings by a sentence with $(1+\varepsilon)\log(n) + O_{k,\varepsilon}(1)$ quantifiers.*

Proof Sketch (see [2] for full proof). Assume $n > 2$, and pick a sufficiently large constant k such that $f(n) \leq n^k$ for all n. Next, pick $\varepsilon > 0$. Let $t \geq 4$ be a large enough integer so that $t \geq 2^{2k/\varepsilon}$. By Lemma 15, we know there is some integer $N(t)$, such that for all $n \geq N(t)$, we have $\lceil \log_t(f(n)) \rceil! \geq f(n)$. **S** once again plays $\lceil \log_t(f(n)) \rceil$ existential moves, separating the $f(n)$ strings in $\mathcal{A}$ into distinct isomorphism classes by using different permutations. Now, as in the proof of Theorem 16, **S** has reduced the games to $f(n)$ parallel **one-vs-all** string separation instances. So now, using Theorem 16, he can win these instances in parallel, using $(1 + \varepsilon/2)\log(n) + 6$ quantifiers for all $n \geq \max(N(t), N'(\varepsilon))$, for some $N'(\varepsilon)$ depending only on ε. The total number of rounds used by **S** is:

$$\begin{aligned}\lceil \log_t(f(n)) \rceil + (1+\varepsilon/2)\log(n) + 6 &\leq (1+\varepsilon/2)\log(n) + k\log_t(n) + 7 \\ &= (1+\varepsilon/2)\log(n) + \frac{k\log(n)}{2k/\varepsilon} + 7 \\ &\leq (1+\varepsilon)\log(n) + 7.\end{aligned}$$

Again, $N(t)$ depends only on t, which depends only on k and ε, whereas $N'(\varepsilon)$ depends only on ε. So when $n < \max(N(t), N'(\varepsilon))$, the number of quantifiers can be absorbed into an additive term that depends only on k and ε, giving us the $O_{k,\varepsilon}(1)$ term. ◀

Our final results concern separating arbitrary sets of n-bit strings from their complements. As discussed in Section 1, this corresponds exactly to defining arbitrary Boolean functions. Note that this will immediately imply Theorem A in Section 1.

▶ **Theorem 18** (Arbitrary vs. Arbitrary – *Upper Bound*). *For all n, and for every $\varepsilon > 0$, any two disjoint sets of n-bit strings are separable by a sentence with $(1+\varepsilon)\frac{n}{\log(n)} + O_\varepsilon(1)$ quantifiers.*

Proof Sketch (see [2] for full proof). We first observe that for any real number $r > 2$, **S** can play $m := \lceil n/\log_r(n)\rceil$ preprocessing existential moves, putting different permutations of these m pebbles on the strings in $\mathcal{A}$ (i.e., the left side). A Stirling's approximation argument similar to Lemma 15 shows that there is some $N(r)$ such that for all $n \geq N(r)$, this number m of preprocessing moves suffices to give each string in $\mathcal{A}$ its own isomorphism class. Note that once this is done, **S** has partitioned the original instance into $|\mathcal{A}|$ disjoint instances of **one-vs-all** games.

Now, given $\varepsilon > 0$, we first choose $r > 2$ small enough that $\log(r) < 1 + \varepsilon/2$. **S** now plays the preprocessing existential moves as described above to obtain $|\mathcal{A}|$ parallel **one-vs-all** instances. Now, by Theorem 16, he can win these instances in parallel using Lemma 6, using $(1+\varepsilon/2)\log(n) + 6$ rounds for all $n \geq \max(N(r), N'(\varepsilon))$, for some $N'(\varepsilon)$ depending only on ε. The total number of rounds needed, therefore, is:

$$\begin{aligned} m + (1+\varepsilon/2)\log(n) + 6 &\leq \frac{n}{\log(n)} \cdot \log(r) + \left(1 + \frac{\varepsilon}{2}\right)\log(n) + 7 \\ &\leq \left(1+\frac{\varepsilon}{2}\right)\left(\frac{n}{\log(n)} + \log(n)\right) + 7 \\ &< (1+\varepsilon)\frac{n}{\log(n)} + 7 \end{aligned}$$

for all $n \geq \max(N(r), N'(\varepsilon), N''(\varepsilon))$, where for all $n \geq N''(\varepsilon)$, we have $(1+\varepsilon/2)\log(n) < (\varepsilon/2)\frac{n}{\log(n)}$. Since each of $N(r)$, $N'(\varepsilon)$, and $N''(\varepsilon)$ depends only on ε, the number of quantifiers for smaller n is absorbed into the $O_\varepsilon(1)$ term. ◀

Remarkably, we cannot improve the upper bound in Theorem 18 by any significant amount. The following proposition establishes this by means of a counting argument, also showing that Theorem A is tight.

▶ **Theorem 19** (Arbitrary vs. Arbitrary – *Lower Bound*). *For all sufficiently large n, there is a nonempty set of n-bit strings, $\mathcal{A} \subsetneq \{0,1\}^n$, such that every separating sentence φ for $(\mathcal{A}, \{0,1\}^n - \mathcal{A})$ must have at least $n/\log(n)$ quantifiers.*

Proof Sketch (see [2] for full proof). If we require k quantifiers to separate any instance on n-bit strings (for sufficiently large n), we can start by counting the number of pairwise inequivalent sentences that can be written with k quantifiers. Such a sentence has a quantifier prefix of length at most k ($\leq 2^{k+1}$ possibilities), followed by a quantifier-free part, which is a disjunction of types (2^{k^k} possibilities). This puts the total number of possible such formulas to be at most $2^k \cdot 2^{2^{k\log(k)}}$. We need this number to be at least $2^{2^n} - 2$, to account for all nonempty instances of the form $(\mathcal{A}, \{0,1\}^n - \mathcal{A})$, which require pairwise inequivalent sentences to separate. Solving this shows that we need $k \geq n/\log(n)$. ◀

6 Conclusions & Open Problems

We obtained nontrivial quantifier upper bounds with matching lower bounds (up to $(1+\varepsilon)$ factors) for a variety of string separation problems. All our upper bounds arise as a result of using the technique of parallel play.

Throughout this work, with very few exceptions, we used MS games to obtain *upper* bounds. It might seem unnecessary to exhibit upper bounds using game arguments, when it ordinarily suffices to exhibit separating sentences. However, the sentences implicitly arising from our game techniques are highly nontrivial to construct. In the case of QR, since taking disjunctions and conjunctions do not increase the quantifier rank, one can build up complex sentences out of simpler ones without paying any cost; we lose this convenience with QN, and therefore need more nuanced techniques, such as parallel play.

Natural directions to extend this work include the following:

1. It would be illuminating to understand the QN required to express particular string and graph properties. While our lower bound for the **one-vs-one** problem (Proposition 14) gave a pair of strings requiring $\log(n)$ quantifiers to separate, the counting argument in Proposition 19 does not exhibit a *specific* instance on n-bit strings that provably requires $n/\log(n)$ quantifiers to separate. Note that by (1), if we can find any property that requires $\omega(\log(n))$ quantifiers to capture, then that property lies outside of NL.
2. Is it possible to use our upper bound in Theorem 18 to obtain Lupanov's upper bound of $(1+\varepsilon)2^n/\log(n)$ on the minimum formula size needed separate two sets in propositional logic (or vice versa)?
3. It is known for ordered structures that with $O(\log n)$ quantifiers, one can express the BIT predicate, or equivalently, all standard arithmetic operations on elements of the universe [12]. In particular, with BIT, some properties that would otherwise require $\log(n)$ quantifiers can be expressed using $O(\log(n)/\log\log(n))$ quantifiers. Understanding the use of BIT and other numeric relations would be valuable.

References

1 Marco Carmosino, Ronald Fagin, Neil Immerman, Phokion Kolaitis, Jonathan Lenchner, and Rik Sengupta. Multi-structural games and beyond, 2023. doi:10.48550/arXiv.2301.13329.

2 Marco Carmosino, Ronald Fagin, Neil Immerman, Phokion Kolaitis, Jonathan Lenchner, and Rik Sengupta. On the number of quantifiers needed to define boolean functions, 2024. arXiv:2407.00688.

3 Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961. doi:10.4064/fm-49-2-129-141.

4 Ronald Fagin, Jonathan Lenchner, Kenneth W. Regan, and Nikhil Vyas. Multi-structural games and number of quantifiers. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13, Rome, Italy, 2021. IEEE. doi:10.1109/LICS52264.2021.9470756.

5 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, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs*, pages 48:1–48:14, Vienna, Austria, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2022.48.

6 Steven Fortune. A note on sparse complete sets. *SIAM Journal on Computing*, 8(3):431–433, 1979. doi:10.1137/0208034.

7 Roland Fraïssé. Sur quelques classifications des systèmes de relations. *Université d'Alger, Publications Scientifiques, Série A*, 1:35–182, 1954. doi:10.2307/2963939.

8 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.

9 L. Hella and K. Luosto. Game characterizations for the number of quantifiers. *Mathematical Structures in Computer Science*, pages 1–20, 2024.

10 Lauri Hella and Jouko Väänänen. The size of a formula as a measure of complexity. In Asa Hirvonen, Juha Kontinen, Roman Kossak, and Andrés Villaveces, editors, *Logic Without Borders: Essays on Set Theory, Model Theory, Philosophical Logic and Philosophy of Mathematics*, pages 193–214. De Gruyter, Berlin, München, Boston, 2015. doi:doi:10.1515/9781614516873.193.

11 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.

12 Neil Immerman. *Descriptive Complexity.* Springer, New York USA, 1999.

13 Oleg Lupanov. On a method of circuit synthesis. *Izvestia VUZ Radiofizika*, 1(1):120–140, 1958.

14 Oleg Lupanov. On the realization of functions of logical algebra by formulae of finite classes (formulae of limited depth). *Problems of Cybernetics*, 6(6):1–14, 1965. Upper bounds on sizes of formulas for all functions (English translation of Problemy Kibernetiki 6 (1961) 5-14.).

15 Stephen R. Mahaney. Sparse complete sets for NP: Solution of a conjecture of berman and hartmanis. *Journal of Computer and System Sciences*, 25(2):130–143, 1982. doi:10.1016/0022-0000(82)90002-2.

16 John Riordan and C E Shannon. The number of two-terminal series-parallel networks. *Journal of Mathematics and Physics*, 21(1-4):83–93, 1942. doi:10.1002/sapm194221183.

17 Joseph G. Rosenstein. *Linear Orderings.* Academic Press, New York USA, 1982.

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