

On the Number of Quantifiers as a Complexity Measure

Ronald Fagin  

IBM Research, Almaden, CA, USA

Jonathan Lenchner¹  

IBM T.J. Watson Research Center, Cambridge, MA, USA

Nikhil Vyas  

MIT EECS, Cambridge, MA, USA

Ryan Williams  

MIT CSAIL and EECS, Cambridge, MA, USA

Abstract

In 1981, Neil Immerman described a two-player game, which he called the “separability game” [15], that captures the number of quantifiers needed to describe a property in first-order logic. Immerman’s paper laid the groundwork for studying the number of quantifiers needed to express properties in first-order logic, but the game seemed to be too complicated to study, and the arguments of the paper almost exclusively used quantifier rank as a lower bound on the total number of quantifiers. However, last year Fagin, Lenchner, Regan and Vyas [10] rediscovered the game, provided some tools for analyzing them, and showed how to utilize them to characterize the number of quantifiers needed to express linear orders of different sizes. In this paper, we push forward in the study of number of quantifiers as a bona fide complexity measure by establishing several new results. First we carefully distinguish minimum number of quantifiers from the more usual descriptive complexity measures, minimum quantifier rank and minimum number of variables. Then, for each positive integer k , we give an explicit example of a property of finite structures (in particular, of finite graphs) that can be expressed with a sentence of quantifier rank k , but where the same property needs $2^{\Omega(k^2)}$ quantifiers to be expressed. We next give the precise number of quantifiers needed to distinguish two rooted trees of different depths. Finally, we give a new upper bound on the number of quantifiers needed to express s - t connectivity, improving the previous known bound by a constant factor.

2012 ACM Subject Classification Theory of computation → Finite Model Theory

Keywords and phrases number of quantifiers, multi-structural games, complexity measure, s - t connectivity, trees, rooted trees

Digital Object Identifier 10.4230/LIPIcs.MFCS.2022.48

Related Version *Extended Version*: <https://arxiv.org/abs/2207.00104> [11]

1 Introduction

In 1981 Neil Immerman described a two-player combinatorial game, which he called the “separability game” [15], that captures the number of quantifiers needed to describe a property in first-order logic (henceforth FOL). In that paper Immerman remarked,

“Little is known about how to play the separability game. We leave it here as a jumping off point for further research. We urge others to study it, hoping that the separability game may become a viable tool for ascertaining some of the lower bounds which are ‘well believed’ but have so far escaped proof.”

¹ Corresponding author



Immerman’s paper laid the groundwork for studying the number of quantifiers needed to express properties in FOL, but alas, the game seemed too complicated to study and the paper used the surrogate measure of quantifier rank, which provides a lower bound on the number of quantifiers, to make its arguments. One of the reasons for the difficulty of directly analyzing the number of quantifiers is that the separability game is played on a pair $(\mathcal{A}, \mathcal{B})$ of sets of structures, rather than on a pair of structures as in a conventional Ehrenfeucht-Fraïssé game. However, last year Fagin, Lenchner, Regan and Vyas [10] rediscovered the games, provided some tools for analyzing them, and showed how to utilize them to characterize the number of quantifiers needed to express linear orders of different sizes. In this paper, we push forward in the study of number of quantifiers as a bona fide complexity measure by establishing several new results, using these rediscovered games as an important, though not exclusive, tool. Although Immerman called his game the “separability game,” we keep to the more evocative “multi-structural game,” as coined in [10].

Given a property P definable in FOL, let $Quants(P)$ denote the minimum number of quantifiers over all FO sentences that express P . **This paper exclusively considers expressibility in FOL.** $Quants(P)$ is related to two more widely studied descriptive complexity measures, the minimum quantifier rank needed to express P , and the minimum number of variables needed to express P . The quantifier rank of an FO sentence σ is typically denoted by $qr(\sigma)$. We shall denote the *minimum* quantifier rank over *all* FO sentences describing the property P by $Rank(P)$, and denote the minimum number of variables needed to describe P by $Vars(P)$. When referring to a specific sentence σ , we shall denote the analogs of $Quants()$ and $Vars()$ by $quants(\sigma)$ and $vars(\sigma)$. (That is, $quants(\sigma)$, $vars(\sigma)$ and $qr(\sigma)$ refer to the number of quantifiers, variables and quantifier rank of the particular sentence σ .) On the other hand, $Quants(P)$, $Vars(P)$ and $Rank(P)$ refer to the minimum values of these quantities among all expressions describing P . Possibly there is one sentence establishing $Quants(P)$, another establishing $Vars(P)$, and a third establishing $Rank(P)$. We investigate the *extremal* behavior of $Quants(P)$, via studying concrete properties P for which $Quants(P)$ behaves differently from the other measures.

First of all, for every property P , since every variable in a sentence describing P is bound to a quantifier, and quantifiers can only be bound to a single variable, it must be that $Vars(P) \leq Quants(P)$. The following simple proposition observes that $Vars(P)$ is also upper bounded by $Rank(P)$.

► **Proposition 1.** *For every property P : $Vars(P) \leq Rank(P)$.*

A proof is in [11, Prop. 1]. Since clearly $Rank(P) \leq Quants(P)$, we have:

$$Vars(P) \leq Rank(P) \leq Quants(P). \quad (1)$$

Furthermore, it follows from Immerman [16, Prop. 6.15] that $Quants(P)$ and $Rank(P)$ can both be arbitrarily larger than $Vars(P)$. When the property P is s - t connectivity up to path length k , Immerman shows that $Vars(P) \leq 3$, yet $Rank(P) \geq \log_2(k)$.

Summary of Results

From equation (1), we see that the number of quantifiers needed to express a property is lower-bounded by the minimum quantifier rank and number of variables. How much larger can $Quants(P)$ be, compared to the other two measures? It is known (see [8]) that there *exists* a fixed vocabulary V and an infinite sequence P_1, P_2, \dots of properties such that P_k is a property of finite structures with vocabulary V such that $Rank(P_k) \leq k$, yet $Quants(P_k)$ is

not an elementary function of k . However, the existence of such P_k are proved via counting arguments. We provide an *explicitly computable* sequence of properties $\{P_k\}$ with a high growth rate in terms of the number of quantifiers required. (By “explicitly computable”, we mean that there is an algorithm A such that, given a positive integer k , the algorithm A prints a FO sentence σ_k with quantifier rank k defining the property P_k , in time polynomial in the length of σ_k .)

► **Theorem** (Theorem 4, Section 2). *There is an explicitly computable sequence of properties $\{P_k\}$ such that for all k we have $\text{Rank}(P_k) \leq k$, yet $\text{Quants}(P_k) \geq 2^{\Omega(k^2)}$.*

Next, we give an example of a setting in which one can completely nail down the number of quantifiers that are necessary and sufficient for expressing a property. Building on Fagin *et al.* [10], which gives results on the number of quantifiers needed to distinguish linear orders of different sizes, we study the number of quantifiers needed to distinguish rooted trees of different depths.

Let $t(r)$ be the maximum d such that there is a formula with r quantifiers that can distinguish rooted trees of depth d (or larger) from rooted trees of depth less than d . Reasoning about the relevant multi-structural games, we can completely characterize $t(r)$, as follows.

► **Theorem** (Theorem 14, Section 3). *For all $r \geq 1$ we have*

$$t(2r) = \frac{7 \cdot 4^r}{18} + \frac{4r}{3} - \frac{8}{9}, \quad t(2r+1) = \frac{8 \cdot 4^r}{9} + \frac{4r}{3} - \frac{8}{9}.$$

It follows from the above theorem that we can distinguish (rooted) trees of depth at most d from trees of depth greater than d using only $\Theta(\log d)$ quantifiers, and we can in fact pin down the *exact* depth that can be distinguished with r quantifiers. This illustrates the power of multi-structural games, and gives hope that more complex problems may admit an exact number-of-quantifiers characterization.

Next, we consider the question of how many quantifiers are needed to express that two nodes s and t are connected by a path of length at most n , in directed (or undirected) graphs. In our notation, we wish to determine $\text{Quants}(P)$ where P is the property of s - t connectivity via a path of length at most n . Considering the significance of s - t connectivity in both descriptive complexity and computational complexity, we believe this is a basic question that deserves a clean answer. It follows from the work of Stockmeyer and Meyer that s - t connectivity up to path length n can be expressed with $3 \log_2(n) + O(1)$ quantifiers. As mentioned earlier, s - t connectivity is well-known to require quantifier rank at least $\log_2(n) - O(1)$. We manage to reduce the number of quantifiers necessary for s - t connectivity.

► **Theorem** (Theorem 15, Section 4). *The number of quantifiers needed to express s - t connectivity is at most $3 \log_3(n) + O(1) \approx 1.893 \log_2(n) + O(1)$.*

The remainder of this manuscript proceeds as follows. In the next subsection we describe multi-structural games and compare them to Ehrenfeucht-Fraïssé games. In the subsection that follows we review related work in complexity. We then prove the theorems mentioned above. In Section 2 we prove Theorem 4. In Section 3 we prove Theorem 14. In Section 4 we prove Theorem 15. In Section 5, we give final comments and suggestions for future research.

1.1 Multi-Structural Games

The standard Ehrenfeucht-Fraïssé game (henceforth E-F game) is played by “Spoiler” and “Duplicator” on a pair (A, B) of structures over the same FO vocabulary V , for a specified number r of *rounds*. If V contains constant symbols $\lambda_1, \dots, \lambda_k$, then designated (“constant”)

elements c_i of A , and c'_i of B , must be associated with each λ_i . In each round, Spoiler chooses an element from A or from B , and Duplicator replies by choosing an element from the other structure. In this way, they determine sequences of elements $a_1, \dots, a_r, c_1, \dots, c_k$ of A and $b_1, \dots, b_r, c'_1, \dots, c'_k$ of B , which in turn define substructures A' of A and B' of B . Duplicator wins if the function given by $f(a_i) = b_i$ for $i = 1, \dots, r$, and $f(c_j) = c'_j$ for $j = 1, \dots, k$, is an isomorphism of A' and B' . Otherwise, Spoiler wins.

The equivalence theorem for E-F games [9, 12] characterizes the minimum *quantifier rank* of a sentence ϕ over V that is true for A but false for B . The quantifier rank $qr(\phi)$ is defined as zero for a quantifier-free sentence ϕ , and inductively:

$$\begin{aligned} qr(\neg\phi) &= qr(\phi), \\ qr(\phi \vee \psi) &= qr(\phi \wedge \psi) = \max\{qr(\phi), qr(\psi)\}, \\ qr(\forall x\phi) &= qr(\exists x\phi) = qr(\phi) + 1. \end{aligned}$$

► **Theorem 2** ([9, 12], Equivalence Theorem for E-F Games). *Spoiler wins the r -round E-F game on (A, B) if and only if there is a sentence ϕ of quantifier rank at most r such that $A \models \phi$ while $B \models \neg\phi$.*

In this paper we make use of a variant of E-F games, which have come to be called *multi-structural games* [10]. Multi-structural games (henceforth M-S games) make Duplicator more powerful and can be used to characterize the *number* of quantifiers, rather than the *quantifier rank*. In an M-S game there are again two players, Spoiler and Duplicator, and there is a fixed number r of rounds. Instead of being played on a pair (A, B) of structures with the same vocabulary (as in an E-F game), the M-S game is played on a pair $(\mathcal{A}, \mathcal{B})$ of *sets of structures*, all with the same vocabulary. For k with $0 \leq k \leq r$, by a *labeled structure* after k rounds, we mean a structure along with a labeling of which elements were selected from it in each of the first k rounds. Let $\mathcal{A}_0 = \mathcal{A}$ and $\mathcal{B}_0 = \mathcal{B}$. Thus, \mathcal{A}_0 represents the labeled structures from \mathcal{A} after 0 rounds, and similarly for \mathcal{B}_0 – in other words nothing is yet labelled except for constants. If $1 \leq k < r$, let \mathcal{A}_k be the labeled structures originating from \mathcal{A} after k rounds, and similarly for \mathcal{B}_k . In round $k + 1$, Spoiler either chooses an element from each member of \mathcal{A}_k , thereby creating \mathcal{A}_{k+1} , or chooses an element from each member of \mathcal{B}_k , thereby creating \mathcal{B}_{k+1} . Duplicator responds as follows. Suppose that Spoiler chose an element from each member of \mathcal{A}_k , thereby creating \mathcal{A}_{k+1} . Duplicator can then make multiple copies of each labeled structure of \mathcal{B}_k , and choose an element from each copy, thereby creating \mathcal{B}_{k+1} . Similarly, if Spoiler chose an element from each member of \mathcal{B}_k , thereby creating \mathcal{B}_{k+1} , Duplicator can then make multiple copies of each labeled structure of \mathcal{A}_k , and choose an element from each copy, thereby creating \mathcal{A}_{k+1} . Duplicator wins if there is some labeled structure A in \mathcal{A}_r and some labeled structure B in \mathcal{B}_r where the labelings give a partial isomorphism. Otherwise, Spoiler wins.

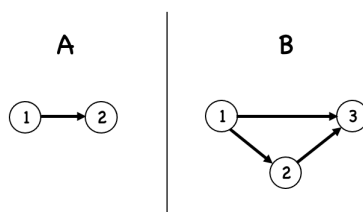
In discussing M-S games we sometimes think of the play of the game by a given player, in a given round, as taking place on one of two “sides”, the \mathcal{A} side or the \mathcal{B} side, corresponding to where the given player plays from on that round.

Note that on each of Duplicator’s moves, Duplicator can make “every possible choice”, via the multiple copies. Making every possible choice creates what we call the *oblivious strategy*. Indeed, Duplicator has a winning strategy if and only if the oblivious strategy is a winning strategy.

The following equivalence theorem, proved in [15, 10], is the analog of Theorem 2 for E-F games.

► **Theorem 3** ([15, 10], Equivalence Theorem for Multi-Structural Games). *Spoiler wins the r -round M-S game on $(\mathcal{A}, \mathcal{B})$ if and only if there is a sentence ϕ with at most r quantifiers such that $A \models \phi$ for every $A \in \mathcal{A}$ while $B \models \neg\phi$ for every $B \in \mathcal{B}$.*

In [10] the authors provide a simple example of a property P of a directed graph that requires 3 quantifiers but which can be expressed with a sentence of quantifier rank 2. P is the property of having a vertex with both an in-edge and an out-edge. P can be expressed via the sentence $\sigma = \exists x(\exists yE(x, y) \wedge \exists yE(y, x))$, where $E(,)$ denotes the directed edge relation. In [10] it is shown that while Spoiler wins a 2-round E-F game on the two graphs A and B in Figure 1, Duplicator wins the analogous 2-round M-S game starting with these two graphs.



■ **Figure 1** The graph B , on the right, contains a vertex with both an in-edge and an out-edge, while the graph A , on the left, does not.

Hence, by Theorem 3, the property P is not expressible with just 2 quantifiers.

1.2 Related Work in Complexity

Trees are a much studied data structure in complexity theory and logic. It is well known that it is impossible, in FOL, to express that a graph with no further relations is a tree [18, Proposition 3.20]. We note, however, that given a partial ordering on the nodes of a graph, it is easy to express in FOL the property that the partial ordering gives rise to a tree. The relevant sentence expresses that there is a root (i.e., a greatest element) from which all other nodes descend, and if a node x has nodes y and z as distinct ancestors then one of y and z must have the other as its own ancestor. Hence the needed sentence is the conjunction of the following two sentences:

$$\begin{aligned} &\exists x\forall y(y \neq x \rightarrow y < x), \\ &\forall x\forall y\forall z((x < y \wedge x < z \wedge y \neq z) \rightarrow (y < z \vee z < y)). \end{aligned}$$

There are also interesting models of computation and logics based on trees. See, for example, the literature on Finite Tree Automata [7] and Computational Tree Logic [6].

We now discuss s - t connectivity. In this paragraph only, n denotes the number of nodes in the graph and k the number of edges in a shortest path from s to t . The s - t connectivity problem has been studied extensively in both logic [1, 16] and complexity theory. Most complexity studies of this problem have focused on space and time complexity. Directed s - t connectivity is known to be NL-complete (see for example Theorem 16.2 in [19]), while undirected s - t connectivity is known to be in L [20]. Savitch [22] proved that s - t connectivity can be solved in $O(\log^2(n))$ space and $n^{\log_2(n)(1+O(1))}$ time. Recent work of Kush and Rossman [17] has shown that the randomized AC^0 formula complexity of s - t connectivity is at most size $n^{0.49 \log_2(k)+O(1)}$, a slight improvement. Barnes, Buss, Ruzzo and Schieber [2] gave an algorithm running in both sublinear space and polynomial time for s - t connectivity. Gopalan, Lipton, and Meka [13] presented randomized algorithms for solving s - t connectivity

with non-trivial time-space tradeoffs. The s - t connectivity problem has also been studied from the perspective of circuit and formula depth. For the weaker model of AC^0 formulas an $n^{\Omega(\log(k))}$ size lower bound is known to hold unconditionally [4, 5, 21].

There is also a natural and well-known correspondence with the number of quantifiers in FOL and circuit complexity, in particular with the circuit class AC^0 (constant-depth circuits comprised of NOT gates along with unbounded fan-in OR and AND gates). For example, Barrington, Immerman, and Straubing [3] proved that $\text{uniform}_{FO}\text{-}AC^0 = FO[<, BIT]$, thus characterizing the problems solvable in uniform AC^0 by those expressible in FOL with ordering and a BIT relation.

More generally it is known that $\text{uniform}_{FO}\text{-}AC[t(n)] = FO[<, BIT][t(n)]$ ([16], Theorem 5.22), i.e., FO formulas over ordering and BIT relations, defined via constant-sized blocks that are “iterated” for $O(t(n))$ times, are equivalent in expressibility with AC circuits of depth $O(t(n))$. (See [11, Appendix C] for a more detailed statement.) Generally speaking, the number of quantifiers of FOL sentences (with a regular form) roughly corresponds to the *depth* of a (highly uniform) AC^0 circuit deciding the truth or falsity of the given sentence. Thus the number of quantifiers can be seen as a proxy for “uniform circuit depth”.

2 Difference in Magnitude: Quantifier Rank vs. Number of Quantifiers

Let V be a vocabulary with at least one relation symbol with arity at least 2. It is known [8] that the number of inequivalent sentences in vocabulary V with quantifier rank k is not an elementary function of k (that is, grows faster than any tower of exponents). Since the number of sentences in vocabulary V with k quantifiers is at most only double exponential in k (e.g., a function that grows like $2^{2^{p(k)}}$ for some polynomial $p(k)$ – see [11, Appendix A] for a proof), it follows by a counting argument that for each positive integer k , there is a property P of finite structures with vocabulary V that can be expressed by a sentence of quantifier rank k , but where the number of quantifiers needed to express P is not an elementary function of k . However, to our knowledge, up to now no *explicit* examples have been given of a property P where the quantifier rank of a sentence to express P is k , but where the number of quantifiers needed to express the property P is at least exponential in k . In the proof of the following theorem, we give such an explicit example.

Let $f_V(k)$ be the number of structures with k nodes up to isomorphism in vocabulary V (such as the number of non-isomorphic graphs with k nodes). Note that in the case of graphs (a single binary relation symbol), $f_V(k)$ is asymptotic to $(2^{k^2})/k!$ [14], and Stirling’s formula implies that $f_V(k) = 2^{\Omega(k^2)}$. We have the following theorem.

► **Theorem 4.** *Assume that the vocabulary V contains at least one relation symbol with arity at least 2. There is an algorithm such that given a positive integer k , the algorithm produces a FO sentence σ of quantifier rank k where the minimum number of quantifiers needed to express σ in FOL is $k f_V(k-1)$, which grows like $2^{\Omega(k^2)}$, and where the algorithm runs in time polynomial in the length of σ .*

Proof. For simplicity, let us assume that the vocabulary V consists of a single binary relation symbol, so that we are dealing with graphs. It is straightforward to modify the proof to deal with an arbitrary vocabulary with at least one relation symbol of arity at least 2. Let us write f for f_V . Let $C_1, \dots, C_{f(k-1)}$ be the $f(k-1)$ distinct graphs up to isomorphism with $k-1$ nodes. For each j with $1 \leq j \leq f(k-1)$, derive the graph D_j that is obtained from C_j by adding one new node with a single edge to every node in C_j . Thus, D_j has k nodes. D_j uniquely determines C_j , since C_j is obtained from D_j by removing a node a that has a

single edge to every remaining node; even if there were two such nodes a , the result would be the same. Therefore, there are $f(k-1)$ distinct graphs D_j . We now give our sentence σ . Let σ_j be the sentence $\exists x_1 \cdots \exists x_k \tau_j(x_1, \dots, x_k)$, which expresses that there is a graph with a subgraph isomorphic to D_j . Then the sentence σ is the conjunction of the sentences σ_j for $1 \leq j \leq f(k-1)$. Since the sentence σ is of length $2^{\Omega(k^2)}$, it is not hard to verify that this sentence can be generated by an algorithm running in polynomial time in the length of the sentence (there is enough time to do all of the isomorphism tests by a naive algorithm).

The sentence σ has quantifier rank k . As written, this sentence has $kf(k-1)$ quantifiers. Let A be the disjoint union of $D_1, \dots, D_{f(k-1)}$. If p is a point in A , define B_p to be the result of deleting the point p from A . Let \mathcal{A} consist only of A , and let \mathcal{B} consist of the graphs B_p for each p in A . If p is in the connected component D_j of A , then B_p does not have a subgraph isomorphic to D_j . Hence, no member of \mathcal{B} satisfies σ . Since the single member A of \mathcal{A} satisfies σ , and since no member of \mathcal{B} satisfies σ , we can make use of M-S games played on \mathcal{A} and \mathcal{B} to find the number of quantifiers needed to express σ .

Assume that we have labeled copies of A and the various B_p 's after i rounds of an M-S game played on \mathcal{A} and \mathcal{B} . The labelling tells us which points have been selected in each of the first i rounds. Let us say that a labeled copy of A and a labeled copy of B_p are *in harmony* after i rounds if the following holds. For each m with $1 \leq m \leq i$, if a is the point labeled m in A , and b is the point labeled m in B_p , then $a = b$. In particular, if the labeled copies of A and B_p are in harmony, then there is a partial isomorphism between the labeled copies of A and B_p .

Let Duplicator have the following strategy. Assume first that in round i , Spoiler selects in \mathcal{A} , and selects a point a from a labeled member A of \mathcal{A} . Then Duplicator (by making extra copies of labeled graphs in \mathcal{B} as needed) does the following for each labeled B_p in \mathcal{B} . If $a \neq p$, and if the labeled copies of A and B_p before round i are in harmony, then Duplicator selects a in B_p , which maintains the harmony. If $a = p$, or if the labeled A and B_p before round i are not in harmony, then Duplicator makes an arbitrary move in B_p .

Assume now that in round i , Spoiler selects in \mathcal{B} . When Spoiler selects the point b from a labeled copy of B_p , then for each labeled A from \mathcal{A} , if the labeled copy of A is in harmony with the labeled copy of B_p before round i , then Duplicator selects b in A , and thereby maintains the harmony. We shall show shortly (Property * below) that in every round, each labeled member of \mathcal{A} is in harmony with a labeled member of \mathcal{B} , so in the case we are now considering where Spoiler selects in \mathcal{B} , Duplicator does select a point in round i in each labeled member of \mathcal{A} .

We prove the following by induction on rounds:

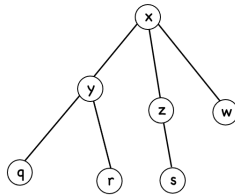
Property *: If A is a labeled graph in \mathcal{A} and if point p in A was not selected in the first i rounds, then there is a labeled copy of B_p that is in harmony with A after i rounds.

Property * holds after 0 rounds (with no points selected). Assume that Property * holds after i rounds; we shall show that it holds after $i+1$ rounds. There are two cases, depending on whether Spoiler moves in \mathcal{A} or in \mathcal{B} in round $i+1$. Assume first that Spoiler moves in \mathcal{A} in round $i+1$. Assume that point p was not selected in A after $i+1$ rounds. By inductive assumption, there are labeled versions of A and B_p that are in harmony after i rounds. So by Duplicator's strategy, labeled versions of A and B_p are in harmony after $i+1$ rounds. Now assume that Spoiler moves in \mathcal{B} in round $i+1$. For each labeled graph A in \mathcal{A} , if a labeled B_p is in harmony with the labeled A after i rounds, then by Duplicator's strategy, the harmony continues between the labeled A and B_p after $i+1$ rounds. So Property * continues to hold after $i+1$ rounds. This completes the proof of Property *.

After $(kf(k-1)) - 1$ rounds, pick an arbitrary labeled graph A in \mathcal{A} . Since at most $(kf(k-1)) - 1$ points have been selected after $(kf(k-1)) - 1$ rounds, and since A contains $kf(k-1)$ points (because it is the disjoint union of $f(k-1)$ graphs each with k points), there is some point p that was not selected in A in the first $(kf(k-1)) - 1$ rounds. Therefore, by Property *, a labeled version of A and of B_p are in harmony after $(kf(k-1)) - 1$ rounds, and hence there is a partial isomorphism between the labeled A and B_p . So Duplicator wins the $(kf(k-1)) - 1$ round M-S game! Therefore, by Theorem 3, the number of quantifiers needed to express σ is more than $(kf(k-1)) - 1$. Since σ has $kf(k-1)$ quantifiers it follows that the minimum number of quantifiers need to express σ is exactly $kf(k-1)$. ◀

3 Rooted Trees

Our aim in this section is to establish the minimum number of quantifiers needed to distinguish rooted trees of depth at least k from those of depth less than k using first-order formulas, given a partial ordering on the vertices induced by the structure of the rooted tree. Figure 2 gives an example of a tree where we designate x as the root node. We define the depth of



■ **Figure 2** A rooted tree with designated root node x and depth 3.

such a tree to be the maximum number of nodes in a path from the root to a leaf, where all segments in the path are directed from parent to child. Although it is more customary to denote the depth of a tree in terms of the number of *edges* along such a path, we keep to the above definition because we will often run into the special case of linear orders, which we view as trees in the natural way, and linear orders are characterized by their size (number of nodes) and we would like the size of a liner order to correspond to the depth of the associated tree. Let us denote the tree rooted at x by T_x . We make the arbitrary choice that the node x is the *largest* element in the induced partial order, so that for two nodes α, β of T_x , we have $\alpha > \beta$ if and only if there is a path (x_1, \dots, x_n) in T_x with $\alpha = x_1$ and $\beta = x_n$ such that x_i is a parent of x_{i+1} for $1 \leq i \leq n - 1$. Thus, for example, in Figure 2, $x > q$ and $z > s$, etc.

The problem of distinguishing the depth of a rooted tree via a first-order formula with a minimum number of quantifiers is similar to the analogous problem for linear orders of different sizes, since a rooted tree has depth k or greater if and only if it has a leaf node, above which there is linear order of size at least $k - 1$.

Our strategy will be to characterize a tree of depth d recursively as a graph containing a vertex v which has a subtree of depth k that includes v and everything below it, and a linear order of length $d - k$ comprising the vertices above v , where k is chosen to minimize the total number of quantifiers. We then show that this is the minimum quantifier way to characterize a tree of each given depth.

The following result is classic and key to establishing a number of fundamental inexpressibility results in FOL [18]. It is typically obtained by appeal to Theorem 2.

► **Theorem 5** ([18], Theorem 3.6). *Let $f(r) = 2^r - 1$. In an r -round E-F game played on two linear orders of different sizes, Duplicator wins if and only if the size of the smaller linear order is at least $f(r)$.*

Analogous of Theorem 5 are proven for M-S games in [10]. The following definition and theorems are from that paper.

► **Definition 6** ([10]). *Define the function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that $g(r)$ is the maximum number k such that there is a formula with r quantifiers that can distinguish linear orders of size k or larger from linear orders of size less than k .*

► **Theorem 7** ([10]). *The function g takes on the following values: $g(1) = 1, g(2) = 2, g(3) = 4, g(4) = 10$, and for $r > 4$,*

$$g(r) = \begin{cases} 2g(r-1) & \text{if } r \text{ is even,} \\ 2g(r-1) + 1 & \text{if } r \text{ is odd.} \end{cases}$$

► **Theorem 8** ([10]). *In an r -round M-S game played on two linear orders of different sizes Duplicator has a winning strategy if and only if the size of the smaller linear order is at least $g(r)$.*

For given positive integers r and k , we want to know if there exist sentences with r quantifiers that distinguish rooted trees of depth k or larger from rooted trees of depth smaller than k . For $k = r$, one such sentence is

$$\exists x_1 \cdots \exists x_r \bigwedge_{1 \leq i < r} (x_i < x_{i+1}), \quad (2)$$

which distinguishes rooted trees of depth r or larger from rooted trees of depth less than r . Here, if T_x is a rooted tree of depth exactly r then x_1 would be a deepest child. Since there are only finitely many inequivalent formulas in up to r variables that include the relations $<$ and $=$ and at most r quantifiers, there is some maximum such k , which we shall designate by $t(r)$. With $\mathbb{N} = \{1, 2, \dots\}$, we restate this definition of t formally as follows. Note that no meaningful sentence about trees can be constructed with a single quantifier, so the definition begins at $r = 2$.

► **Definition 9.** *Define the function $t : \{2, 3, \dots\} \rightarrow \mathbb{N}$ such that $t(r)$ is the maximum number z such that there is a formula with r quantifiers that can distinguish rooted trees of depth z or larger from rooted trees of depth less than z .*

By (2) above, $t(r) \geq r$ for $r \geq 2$. For an M-S game of r rounds on rooted trees of sizes $t(r)$ or larger on one side, and $t(r) - 1$ or smaller on the other side, by the Equivalence Theorem, Spoiler will have a winning strategy.

Since linear orders are perfectly good rooted trees, we have the following:

► **Observation 10.** *For all r we have $t(r) \leq g(r)$.*

Simple arguments establishing that $t(2) = 2$, and $t(3) = 4$, stemming from Observation 10, are given in [11, Sec. 3.1]. An analysis establishing $t(4) = 8$ is given in [11, Sec. 3.2]. Except for one paragraph, we shall not need that analysis, nor the particular result, though the reader may have an easier time understanding the rather intricate inductive argument that follows by first reading the analysis of this case.

In the proof of Theorem 7 [10], the authors provide explicit sentences that distinguish linear orders of size $g(r)$ or greater from those of size less than $g(r)$. From the proof of their Theorem 1.6, it can be seen that the distinguishing sentences Φ_r , for $r > 4$ take the form:

$$\Phi_r = \begin{cases} \exists x_1 \forall x_2 \cdots \forall x_{r-1} \exists x_r \phi_r & \text{for } r \text{ odd,} \\ \forall x_1 \exists x_2 \cdots \forall x_{r-1} \exists x_r \phi_r & \text{for } r \text{ even,} \end{cases}$$

48:10 On the Number of Quantifiers as a Complexity Measure

where ϕ_r is quantifier-free. For odd r , the formula Φ_r says that there exists a point x_1 , with a linear order of size at least $\lfloor \frac{r}{2} \rfloor$ to both sides of x_1 . For even r , the formula Φ_r says that for all x_1 , there exists a linear order of at least size $\frac{r}{2}$ to one side or the other of x_1 .

Let us denote by $t_{\forall}(r)$ the maximum number k such that rooted trees of depth k and above can be distinguished from rooted trees of depth less than k using prenex formulas with r quantifiers beginning with a universal quantifier. Equivalently, $t_{\forall}(r) = k$ is the largest depth of a rooted tree such that Spoiler has a winning strategy on r -round M-S games played on rooted trees of depth k or greater versus those of depth less than k when his first move is constrained to be on the tree of lesser depth. Analogously, when considering linear orders, let $g_{\forall}(r)$ and $g_{\exists}(r)$ denote, respectively, the maximum number k such that linear orders of size k and above can be distinguished from linear orders of size less than k using prenex formulas with r quantifiers beginning, respectively, with a universal or existential quantifier.

► **Lemma 11.** *For $r > 1$, one has $t_{\forall}(r) = t(r - 1) + 1$.*

A proof of this very simple lemma is given in [11, Proof of Lemma 18].

► **Lemma 12.** *For $r \geq 2$, the following hold:*

$$g_{\exists}(2r) = 2g_{\forall}(2r - 1) + 1, \quad (3)$$

$$g_{\forall}(2r + 1) = 2g_{\exists}(2r). \quad (4)$$

Further, there are expressions establishing the g_{\exists} relations having prenex signatures $\exists\forall\cdots\exists\forall\exists\exists$ with $r - 1$ iterations of the $\exists\forall$ pair and then a final $\exists\exists$, while there are expressions establishing the g_{\forall} relations having prenex signatures $\forall\exists\cdots\forall\exists\exists$ with r iterations of the $\forall\exists$ pair and then a final \exists .

A proof of this lemma is given in [11, Proof of Lemma 19].

► **Theorem 13.** *For $r \geq 2$, the following holds:*

$$t(r) = g_{\forall}(r - 1) + t_{\forall}(r - 1) + 1 = g_{\forall}(r - 1) + t(r - 2) + 2. \quad (5)$$

If r is odd, then $g_{\forall}(r - 1) = g(r - 1)$ so for r odd we have:

$$t(r) = g(r - 1) + t_{\forall}(r - 1) + 1 = g(r - 1) + t(r - 2) + 2. \quad (6)$$

A proof of this theorem is given in [11, Proof of Theorem 20].

We can combine the previous results to prove the following explicit expression for $t()$ [11, Proof of Theorem 21].

► **Theorem 14.** *For all $r \geq 1$ we have*

$$t(2r) = \frac{7 \cdot 4^r}{18} + \frac{4r}{3} - \frac{8}{9}, \quad t(2r + 1) = \frac{8 \cdot 4^r}{9} + \frac{4r}{3} - \frac{8}{9}.$$

A table comparing the values of f, g and t for $2 \leq r \leq 10$ is given in [11, Section 3.4].

4 s-t Connectivity

In this section we explore the number of quantifiers needed to express either directed or undirected s - t connectivity (henceforth STCON) in FOL with the binary edge relation E , as a function of the number n of edges in a shortest path between the distinguished nodes s and

t . STCON, also known as *reachability* between labelled nodes s and t , refers to the property of graphs that labelled nodes s and t are connected. STCON(n) denotes the property that s and t are connected by a path of length at most n edges.

In [11, Appendix B], we show how to describe STCON(n) using $2 \log_2(n) + O(1)$ quantifiers. The following theorem generalizes that construction, to improve the number of quantifiers to $3 \log_3(n) + O(1)$. A similar argument shows that $K \log_K(n) + O(1)$ quantifiers can be used for any positive integer K , although this quantity is minimized for $K = 3$.

► **Theorem 15.** STCON(n) can be expressed with $3 \log_3(n) + O(1)$ quantifiers.

Proof. We shall use Υ_i to denote sentences with i quantifiers, and τ_j to denote quantifier-free expressions, where the subscript j denotes the path length characterized by τ_j .

We start with the following simple expression stating that s and t are connected and $d(s, t) \leq 3$, where $d(s, t)$ denotes the length of the shortest path from s to t :

$$\Upsilon_2 = \exists x_1 \exists x_2 (\tau_3 \vee \tau_2 \vee \tau_1), \quad (7)$$

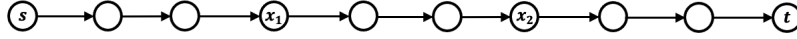
where:

$$\tau_3 = E(s, x_1) \wedge E(x_1, x_2) \wedge E(x_2, t), \quad (8)$$

$$\tau_2 = E(s, x_1) \wedge E(x_1, t), \quad (9)$$

$$\tau_1 = E(s, t). \quad (10)$$

We now iteratively add three quantifiers at each stage and slot two nodes between each of the previously established nodes, as in Figure 3.



■ **Figure 3** An illustration of slotting two nodes between each of the pre-established nodes s, x_1, x_2 and t in order to express a distance 9, $s - t$ path, using 5 quantifiers as in expressions (11) and (12).

We express that there is a path of length at most 9 from s to t , using 5 quantifiers as follows:

$$\Upsilon_5 = \exists x_1 \exists x_2 \forall x_3 \exists x_4 \exists x_5 (\tau_9 \vee \tau_8 \vee \dots \vee \tau_1). \quad (11)$$

In this case, we just show τ_9 . The simplifications required to get from τ_8 down to τ_4 are analogous to those for getting from τ_3 down to τ_1 , but where we apply (8) – (10) separately to each of (12) – (14).

$$\tau_9 = ((x_3 = s) \rightarrow E(s, x_4) \wedge E(x_4, x_5) \wedge E(x_5, x_1)) \wedge \quad (12)$$

$$((x_3 = t) \rightarrow E(x_1, x_4) \wedge E(x_4, x_5) \wedge E(x_5, x_2)) \wedge \quad (13)$$

$$((x_3 \neq s \wedge x_3 \neq t) \rightarrow E(x_2, x_4) \wedge E(x_4, x_5) \wedge E(x_5, t)). \quad (14)$$

Using 8 quantifiers, we can slot two new nodes between each node established in the prior step, as depicted in Figure 4. The associated logical expression is

$$\Upsilon_8 = \exists x_1 \exists x_2 \forall x_3 \exists x_4 \exists x_5 \forall x_6 \exists x_7 \exists x_8 (\tau_{27} \vee \tau_{26} \vee \dots \vee \tau_1), \text{ and} \quad (15)$$

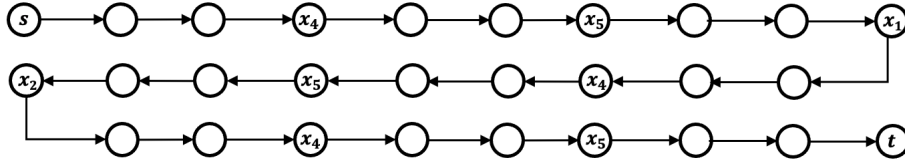
$$\tau_{27} = ((x_3 = s \wedge x_6 = s) \rightarrow E(s, x_7) \wedge E(x_7, x_8) \wedge E(x_8, x_4)) \wedge \quad (16)$$

$$((x_3 = s \wedge x_6 = t) \rightarrow E(x_4, x_7) \wedge E(x_7, x_8) \wedge E(x_8, x_5)) \wedge \quad (17)$$

$$((x_3 = s \wedge (x_6 \neq s \wedge x_6 \neq t)) \rightarrow E(x_5, x_7) \wedge E(x_7, x_8) \wedge E(x_8, x_1)) \wedge \quad (18)$$

$$\dots \quad (19)$$

$$(((x_3 \neq s \wedge x_3 \neq t) \wedge (x_6 \neq s \wedge x_6 \neq t)) \rightarrow E(x_5, x_7) \wedge E(x_7, x_8) \wedge E(x_8, t)). \quad (20)$$



■ **Figure 4** Slotting two nodes between each of the pre-established nodes s, x_1, x_2, x_4, x_5 , and t in order to express a distance 27, $s - t$ path, using 8 quantifiers as in expressions (15) and (16)-(20).

The expression τ_{27} will have the two “pivot points” around the universally quantified variables x_3 and x_6 and so have $3^2 = 9$ antecedent conditions corresponding to the possible ways the universally quantified variables x_3 and x_6 can each take the values s, t or neither s nor t . The right hand side of each equality condition describes how to fill in the edges in Figure 4 with two new vertices (utilizing the two newest existentially quantified variables, x_7 and x_8) and three new edges.

In this way we obtain sentences with $3n - 1$ quantifiers that can express STCON instances of path length up to 3^n . Thus, when n is a power of 3, we can express STCON instances of length n with $3 \log_3(n) - 1 \approx 1.893 \log_2(n) - 1$ quantifiers, and when n is *not* a power of 3, with $\lceil 3 \log_3(n) + 2 \rceil$ quantifiers. The theorem therefore follows. ◀

A Remark on Lower Bounds on Quantifier Rank and hence on Number of Quantifiers

Lower bounds on the number of quantifiers for $s-t$ connectivity follow readily from the literature. The well-known proof that connectivity is not expressible in FOL ([16, Prop. 6.15] or [18, Corollary 3.19]) can be used to establish that $s-t$ connectivity with path length n is not expressible as a formula of quantifier rank $\log_2(n) - c$ for some constant c .

► **Theorem 16** (Immerman, Proposition 6.15 [16]). *There exists a constant c such that $s-t$ connectivity to path length n is not expressible as a formula of quantifier rank $\log_2(n) - c$.*

Since the quantifier rank is a lower bound on the number of quantifiers, the previous theorem immediately implies a lower bound on the number of quantifiers as well. While we have shown that $\text{STCON}(n)$ can be expressed with $3 \log_3(n) + O(1)$ quantifiers, we note that the minimum quantifier *rank* of $\text{STCON}(n)$ is well-known to be lower.

► **Theorem 17** ([18]). *$s-t$ connectivity to path length n can be expressed with a formula of quantifier rank $\log_2(n) + O(1)$.*

5 Final Comments and Future Directions

Although progress on M-S games did not come until 40 years after their initial discovery in [15], the results of this paper show that these games are quite amenable to analysis, and the more detailed information they give about the requisite quantifier structure has the potential to yield many new and interesting insights.

Theorem 4 tells us that the number of quantifiers can be more than exponentially larger than the quantifier rank. This shows that the number of quantifiers is a more refined measure than the quantifier rank, and gives an interesting and natural measure of the complexity of a FO formula. It would be interesting to find explicit examples where the quantifier rank is k , but where the required number of quantifiers grows even faster than in our example in the proof of Theorem 4. Ideally, we would even like to find explicit examples where the required number of quantifiers is non-elementary in k .

We have extended the results on the number of quantifiers needed to distinguish linear orders of different sizes [10] to distinguish rooted trees of different depths. Can this line of attack be carried further to incorporate other structures, say to other structures with induced partial orderings such as finite lattices?

The most immediate question arising from our work is whether one can improve the known upper or lower bounds on the number of quantifiers needed to express s - t connectivity. In particular, what is the smallest constant $c \geq 1$ such that s - t connectivity (up to path length n) is expressible using $c \log_2(n)$ quantifiers? Our Theorem 15 shows that $\text{Quants}(\text{STCON}(n))$ is at most $3 \log_3(n) + O(1) \approx 1.893 \log_2(n) + O(1)$. The well-known lower bound of $\text{Rank}(\text{STCON}(n)) \geq \log_2(n) - O(1)$ (cited as Theorem 16) yields the only lower bound we know on $\text{Quants}(\text{STCON}(n))$, but we also know the upper bound $\text{Rank}(\text{STCON}(n)) \leq \log_2(n) + O(1)$ (cited as Theorem 17). As these upper and lower bounds for the quantifier rank of $\text{STCON}(n)$ essentially match, in order to improve the lower bound on $\text{Quants}(\text{STCON}(n))$ further (by a multiplicative constant), we *cannot* rely on a rank lower bound: we will have to resort to other methods, such as M-S games.

Another question is whether we can find other problems with even larger quantifier number lower bounds than logarithmic ones. Let us stress that substantially larger lower bounds on the number of quantifiers would have major implications for circuit complexity lower bounds. For example, by the standard way of expressing uniform circuit complexity classes in FOL [16], a property (over the $<$ relation) that requires $\log^{\alpha(n)}(n)$ quantifiers, where $\alpha(n)$ is an unbounded function of n , would imply a lower bound for uniform $_{\text{FO}}$ -NC. See [11, Appendix C] for an exact statement.

Another interesting direction to push this research is to extend the notion of multi-structural games to 2nd-order logic, FOL with counting, or to fixed point logic.

References

- 1 Miklós Ajtai and Ronald Fagin. Reachability is harder for directed than for undirected finite graphs. *J. Symbolic Logic*, 55(1):113–150, March 1990.
- 2 Greg Barnes, Jonathan F. Buss, Walter L. Ruzzo, and Baruch Schieber. A sublinear space, polynomial time algorithm for directed s - t connectivity. *SIAM J. Comput.*, 27(5):1273–1282, 1998. doi:10.1137/S0097539793283151.
- 3 David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1 . In *Proceedings: Third Annual Structure in Complexity Theory Conference, Georgetown University, Washington, D. C., USA, June 14-17, 1988*, pages 47–59. IEEE Computer Society, 1988. doi:10.1109/SCT.1988.5262.
- 4 Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth lower bounds for small distance connectivity. *Comput. Complex.*, 7(4):325–345, 1998. doi:10.1007/s000370050014.
- 5 Xi Chen, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 612–625. ACM, 2016. doi:10.1145/2897518.2897534.
- 6 Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on the Logic of Programs, Lecture Notes in Computer Science*, volume 131, pages 52–71, 1981.
- 7 Hubert Comon, Max Dauchet, Rémi Gilleroz, Florent Jacquemard, Denis Liguiez, Sophie Tison, and Marc Tommasi. Tree automata techniques and applications, 1999. URL: <http://www.eecs.harvard.edu/~shieber/Projects/Transducers/Papers/comon-tata.pdf>.
- 8 A. Dawar, M. Grohe, S. Kreutzer, and N. Schweikardt. Model theory makes formulas large. In L. Arge, C. Cachin, T. Jurdziński, and A. Tarlecki, editors, *ICALP07*, volume 4596, pages 913–924. Springer, 2007.

- 9 Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.
- 10 Ronald Fagin, Jonathan Lenchner, Kenneth W. Regan, and Nikhil Vyas. Multi-structural games and number of quantifiers. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2021.
- 11 Ronald Fagin, Jonathan Lenchner, Nikhil Vyas, and Ryan Williams. On the number of quantifiers as a complexity measure, 2022. [arXiv:2207.00104](https://arxiv.org/abs/2207.00104).
- 12 Roland Fraïssé. Sur quelques classifications des systèmes de relations. *Université d’Alger, Publications Scientifiques, Série A*, 1:35–182, 1954.
- 13 Parikshit Gopalan, Richard J. Lipton, and Aranyak Mehta. Randomized time-space tradeoffs for directed graph connectivity. In *FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, 23rd Conference, Mumbai, India, December 15-17, 2003, Proceedings*, volume 2914 of *Lecture Notes in Computer Science*, pages 208–216. Springer, 2003. doi:10.1007/978-3-540-24597-1_18.
- 14 Frank Harary. Note on Carnap’s relational asymptotic relative frequencies. *J. Symb. Log.*, 23(3):257–260, 1958.
- 15 Neil Immerman. Number of quantifiers is better than number of tape cells. *J. Comput. Syst. Sci.*, 22(3):384–406, 1981.
- 16 Neil Immerman. *Descriptive complexity*. Graduate Texts in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0539-5.
- 17 Deepanshu Kush and Benjamin Rossman. Tree-depth and the formula complexity of subgraph isomorphism. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 31–42. IEEE, 2020. doi:10.1109/FOCS46700.2020.00012.
- 18 Leonid Libkin. *Elements of Finite Model Theory*. Texts in Theoretical Computer Science. Springer, 2012. URL: <https://www.springer.com/gp/book/9783540212027>.
- 19 Christos Papdimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Reading, MA USA, 1995.
- 20 Omer Reingold. Undirected st-connectivity in log-space. *Proceedings of the 37th annual ACM symposium on the theory of computing*, pages 376–385, 2005.
- 21 Benjamin Rossman. Formulas vs. circuits for small distance connectivity. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*, pages 203–212. ACM, 2014. doi:10.1145/2591796.2591828.
- 22 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.