Counting Quantifiers, Successor Relations, and Logarithmic Space

Kousha Etessami*

DIMACS P.O. Box 1179
Piscataway, NJ 08855
Email: `etessami@dimacs.rutgers.edu`

Abstract

Given a successor relation S (i.e., a directed line graph), and given two distinguished points s and t, the problem ORD is to determine whether s precedes t in the unique ordering defined by S. We show that ORD is L-complete (via quantifier-free projections). We then show that first-order logic with counting quantifiers, a logic that captures TC^0 ([BIS90]) over structures with a built-in total-ordering, can not express ORD. Our original proof of this in the conference version of this paper ([Ete95]) employed an Ehrenfeucht-Fraïssé Game for first-order logic with counting ([IL90]). Here we show how the result follows from a more general one obtained independently by Nurmonen, [Nur96]. We then show that an appropriately modified version of the EF game is "complete" for the logic with counting in the sense that it provides a necessary and sufficient condition for expressibility in the logic.

We observe that the L-complete problem ORD is essentially sparse if we ignore reorderings of vertices, a property which we term "pseudo-sparseness". We then prove that there are no pseudo-sparse NP-complete properties (under P-time reductions) unless the Polynomial Time Hierarchy collapses (to Σ_3), revealing a structural property on which L and NP apparently differ.

1 Introduction

Complexity theory, to date, has produced few non-trivial lower bounds on general models of computation. This has led researchers to pursue lower bounds for various weaker models where the behavior of computation is more tenable than on Turing machines. There has been some success, for example, in models such as JAGs [CR80, Poo93], but for more general models things are pretty bleak.

The situation is the same in Descriptive Complexity, where the familiar classes of computational complexity have been characterized as exactly the properties expressible in corresponding logical languages over *ordered* structures (see, e.g., [Imm89, Imm87]). This ordering is crucial to giving logic the power to reference relations the way a Turing machine

*The bulk of this work was done while the author was a graduate student at the Computer Science Department of the University of Massachusetts-Amherst, during which time he was supported by NSF grant CCR-9207797.

would its input tape, or for that matter circuits would their sequence of input bits. Likewise, in descriptive complexity attempts have been made at weaker models, where, for example, only weaker forms of ordering are available, [EI94, EI95, CFI92]. In fact, a very weak form of ordering, called *one-way local ordering*, together with the logic corresponding to the class L, has been shown strictly more powerful than the JAG model and has succumbed to lower bounds [EI94, EI95]. For sufficiently strong logical languages a *total-ordering* relation, $\leq$, is adequate for capturing the corresponding complexity class. In fact, for logics corresponding to the class **L** (deterministic logarithmic space) and above, a *successor* relation suffices.

The problem of obtaining a total ordering from a given successor relation can be formulated as a decision problem by specifying two points s and t and asking if s precedes t in the ordering defined by the given successor relation. We call this problem ORD. We prove that ORD is complete for L via quantifier-free projections, a very low level form of reduction ([Imm87, IL95]).

We then look at first-order logic augmented with counting quantifiers, a logical language which over structures with a total-ordering has exactly the power of the class $\mathbf{TC^0}$ (uniform, bounded depth, polynomial size, threshold circuits) ([BIS90]). We show that this logic can not express the property ORD. This holds even in light of the fact that for a logical characterization of L itself a successor relation suffices, [Imm87].

Our original lower-bound proof (in the conference version of this paper, [Ete95]) uses a version of the Ehrenfeucht-Fraïssé Game (hereafter called EF game) for first-order logic with counting quantifiers (see [IL90, CFI92, EI95]). A recent more general result obtained independently by Nurmonen [Nur96] implies this lower bound. Here, instead of providing the full EF game proof as before, we will describe Nurmonen's result* and show how it implies the lower bound.

We also show that an appropriately modified version of the counting game is "complete" for first-order logic with counting in that it provides necessary and sufficient conditions for whether a property K is expressible in the logic (and, thus, in the corresponding complexity class TC^0 when a total-ordering is present).

We then explore some interesting issues that arise from the L-completeness of ORD in structural complexity theory. The key observation is that ORD is essentially sparse, in the sense that there is, up to graph isomorphism, only one successor graph, and there are only polynomially many placements of the two points on this graph. This leads to our definition of *pseudo-sparseness*, and we prove, analogous to Mahaney's Theorem ([Mah82]), that if NP were to also have a pseudo-sparse complete problem, as L does, then the polynomial time hierarchy would collapse (to Σ_3).

ORD's L-completeness is also of independent interest ([Lib94, LW94]) for characterizing the power of certain database query languages, and it was mentioned as an open problem in that setting. We point out the reformulation of the ORD problem that is relevant to this connection.

Section 2 presents basic background and definitions. In section 3 we define first-order reductions and quantifier-free projections. Section 4 proves the L-completeness property ORD, under quantifier-free projections. Section 5 provides the E-F game for first-order logic augmented with counting quantifiers, and sets up the lower bound proof on that language

*My thanks to Ron Fagin for bringing this work to my attention.

with successor for the property ORD. It is then explained how Nurmonen's result [Nur96] can be used to obtain the lower bound. In section 6 we discuss some of the consequences and questions raised by the completeness of ORD having to do with its pseudo-sparseness, and we prove that NP does not have pseudo-sparse complete sets unless the polynomial time hierarchy collapses. Section 8 concludes with further implications and open problems.

2 Background & Definitions

In this paper our notation follows the conventions of first-order logic and descriptive complexity. See, e.g., [Imm87, Imm89] for more detail and motivation.

A logical structure will be called *totally ordered* if it includes a relation, $\leq$, that represents a total ordering on the universe of the structure. Ordered structures will also have constants $\mathbf{1}, \mathbf{n}$ denoting the first and last elements of the universe. We will usually assume for ordered structures that the universe is just $\{1, 2, \ldots, n\}$ with the usual ordering. For weaker logics, a total ordering alone is not enough to capture computation, and we must also add a predicate $\boldsymbol{BIT(i, x)}$, meaning: "The i'th bit in the binary expansion of x's position in the ordering $\leq$, is 1".

We will often prefer not to add the full power of total ordering and BIT to our logical structures. One way to maintain some of the power that is lost without them, is to add a second universe of *numbers*, with an associated ordering and BIT predicate. Thus a structure $\mathcal{A}$ with *numbers* is a two-sorted structure,

$$\begin{aligned}\mathcal{A} \quad = \quad & \langle \{1, 2, \ldots, n\}, \{v_1, v_2, \ldots, v_n\}, \\ & \leq, \mathrm{BIT}, \mathbf{1}, \mathbf{n}, R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}} \rangle\end{aligned}$$

Here the relations $R_1^{\mathcal{A}}, \ldots, R_r^{\mathcal{A}}$ apply to the domain $\{v_1, v_2, \ldots, v_n\}$, while $\leq$, BIT, and constants $\mathbf{1}, \mathbf{n}$ refer only to the domain of numbers. In essence, the added universe of numbers gives us the ability to do some arithmetic on the side as we express a property of the input structures. *In this paper we will assume that all structures are equipped with numbers unless we explicitly state otherwise.*

Once we have numbers, a way to further increase descriptive power is to add counting quantifiers $(\exists i\, x)$. Let $(\exists i\, x)\ \varphi(x)$ mean "There exist at least i distinct x's for which $\varphi(x)$ holds." Here i *is a free variable* that ranges over the number domain, while x ranges over the "vertex" domain. Thus, for example, if we want to say that there are an even number of vertices x that satisfy $\varphi(x)$, we would write:

$$\exists i\ \exists j\ (j + j = i\ \wedge\ \exists i\ x\ \varphi(x) \wedge \forall k\ (k > i \rightarrow \neg \exists k y \varphi(y)))$$

Here $j + j = i$ and other arithmetic expressions over the number domain can be written in terms of the BIT and $\leq$ predicates (In fact, having BIT and $\leq$ is equivalent to having $+$ and $\times$, [Lin95].)

Notice that $\exists x \varphi(x) \equiv (\exists 1\, x)\varphi(x)$. Thus since $\mathbf{1}$ and $\mathbf{n}$ are provided as constants, the counting quantifier will be general enough and, when it is convenient, we won't have to consider the existential and universal quantifiers.

We let $(\mathrm{FO} + \mathrm{COUNT})$ denote first-order logic with counting quantifiers. Note that this is a slight variation on some of the other definitions that have appeared in the literature (e.g., [CFI92]) because we have numbers present and because $(\exists i\ x)$ actually leaves i free, whereas in other definitions there is a distinct quantifier $(\exists i)$ for each integer i. However, as a logic, our definition is perhaps more appropriate because each fixed formula in the other logic is equivalent to a first-order formula by the obvious expansion of the fixed counting quantifiers.

From now on, we will use $\mathbf{ThC}^0$ instead of TC^0, to denote the class of languages with uniform, bounded depth, polynomial size, threshold circuits (we do this so as not to get our terminology confused with the transitive closure operator, TC, which we discuss later). A very particular form of counting quantifier is the *majority* quantifier: $\mathrm{M}x\ \varphi(x)$ means "There are $\geq \lceil n/2 \rceil$ x's such that $\varphi(x)$ holds", where n is the size of the universe. In [BIS90] it was shown that:

Fact 2.1 ([BIS90])
$(\mathrm{FO} + \mathrm{M} + \leq + \mathrm{BIT}) = \mathrm{ThC}^0$

Clearly, we can rewrite $\mathrm{M}x\varphi(x)$ using counting quantifiers as:

$$\exists i\ (middle(i) \wedge \exists i\ x\ \varphi(x)$$

$$middle(i) \ \equiv\ (i + i = n \vee i + i - 1 = n)$$

Note that by definition we always have the $\leq$ and BIT predicates present on our number domain (or, equivalent, we have $+$ and $\times$). Thus, if we have a total-ordering present on our "vertex" domain, then we can determine the number i corresponding to the position of a given vertex x, with the formula

$$\exists i\ y\ (y \leq x) \wedge \neg\exists(i+1)\ y\ (y \leq x)$$

and thus by the above fact we have the full power of ThC^0.

We will denote first-order logic with counting over structures with a number domain as well as an $\leq$ relation on the "vertex" domain $\{v_1, \ldots, v_n\}$, by $(\mathrm{FO} + \mathrm{COUNT} + \leq)$. We thus have:

$$(\mathrm{FO} + \mathrm{COUNT} + \leq) = (\mathrm{FO} + \mathrm{M} + \leq + \mathrm{BIT}) = \mathrm{ThC}^0$$

A different way to increase the power of first-order logic is by adding various transitive closure operators:

Let $\varphi(x_1, \ldots, x_k, x'_1, \ldots, x'_k)$ be a formula with the specified $2k$ free variables (φ might also have other free variables). We will write $(\mathrm{TC}_{x_1 \ldots x_k x'_1 \ldots x'_k}\ \varphi)$ to denote the reflexive, transitive closure of the binary relation $\varphi(\bar{x}, \bar{x}')$. In other words, $\mathrm{TC}(\varphi)(\bar{a}, \bar{b})$ holds when there is a φ-path from $\bar{a}$ to $\bar{b}$. Let $(\mathrm{FO}+\mathrm{TC})$ be the closure of first-order logic with arbitrary occurrences of TC.

Fact 2.2 ([Imm87]) $\quad (\mathrm{FO} + \mathrm{TC} + \leq)\ = \mathrm{NL}$

A weaker transitive closure operator, Deterministic Transitive Closure (DTC), can be used to capture L computations. Given a binary relation $\varphi(\bar{x}, \bar{y})$, the new relation $DTC(\varphi)(\bar{a}, \bar{b})$ means "There is a 'deterministic' path from $\bar{a}$ to $\bar{b}$ in the graph induced by φ". Where 'deterministic' here means that every edge on this path is the unique edge leaving the given vertex. In [Imm87] it was proved that first-order logic with arbitrary applications of the DTC operator, over totally ordered structures, is equivalent in power to L. Clearly, a *successor* relation, $S(x, y)$, meaning "y is the immediate successor of x in the (implicit) total ordering of the universe" would suffice instead of total ordering because $\leq (x, y)$ is just $\mathrm{DTC}(S)(x, y)$. Thus:

Fact 2.3 ([Imm87]) $\quad (\mathrm{FO} + \mathrm{DTC}\ + \mathrm{S})\ = \mathrm{L}$

Of course, the same substitution of S for $\leq$ also goes through for fact 2.2.

Let $\boldsymbol{qd(\varphi)}$, denote the quantifier depth of a formula φ in the first-order language with counting quantifiers. For a structure $\mathcal{A}$ we use A or $U^{\mathcal{A}}$ to denote the universe of $\mathcal{A}$. Let the vocabulary of $\mathcal{A}$ be $\tau = \{R_1, \ldots, R_r, c_1, \ldots, c_\varepsilon\}$. We want to define tuples of elements from A and associate them with tuples of variables in our logical language. Let us say our logic has variables $X = \{x_1, x_2, \ldots\}$. We will think of a tuple $\bar{a} = a_1, \ldots, a_k$ as an assignment to the first k variables $\{x_1, \ldots, x_k\}$, $\bar{a} : \{x_1, \ldots x_k\} \to A$, with $\bar{a}(x_i) = a_i$. It is clear, by just renaming variables, that for any formula φ with k free variables, there is an equivalent formula φ' with $free(\varphi') = \{x_1, \ldots, x_k\}$. Thus, we restrict ourselves to formulas whose free variables are exactly $\{x_1, \ldots, x_i\}$ for some i. For a k-tuple $\bar{a}$, we will say that $\bar{a}$ **interprets** φ iff $\{x_1, \ldots, x_k\} \supseteq free(\varphi)$.

For convenience later on, we want the tuple $\bar{a}$ to also include the constants of $\mathcal{A}$, so we modify $\bar{a}$ so that $a_1 = c_1^{\mathcal{A}}, \ \ldots\ , a_\varepsilon = c_\varepsilon^{\mathcal{A}}$. We may now think of a $k + \varepsilon$ tuple, $\bar{a}$, as an extended assignment $\boldsymbol{\bar{a} : (\{x_1 \ldots x_k\} \cup \{c_1, \ldots, c_\varepsilon\}) \to A}$, with $\bar{a}(c_i) = c_i^{\mathcal{A}}$. Thus, we alway have $\{c_1, \ldots, c_\varepsilon\}$ in the domain of this assignment. We now say a $k + \varepsilon$ tuple $\bar{a}$ interprets φ iff $\{x_1, \ldots, x_k\} \supseteq free(\varphi)$.

Given a structure $\mathcal{A}$ with $\bar{a}$, and $\mathcal{B}$ with $\bar{b}$, where $\bar{a}$ and $\bar{b}$ interpret φ, we'll say $(\mathcal{A}, \bar{a})$ and $(\mathcal{B}, \bar{b})$ **agree** on φ if:

$$(\mathcal{A}, \bar{a}) \models \varphi \iff (\mathcal{B}, \bar{b}) \models \varphi$$

Definition 2.4 ($\boldsymbol{qd}$-$\boldsymbol{m}$ Equivalence) *For k tuples $\bar{a}$ and $\bar{b}$, define $\boldsymbol{(\mathcal{A}, \bar{a}) \equiv_m (\mathcal{B}, \bar{b})}$ to mean that for every formula of φ interpreted by $\bar{a}$ and $\bar{b}$, with $qd(\varphi) \leq m$, $(\mathcal{A}, \bar{a})$ and $(\mathcal{B}, \bar{b})$ agree on φ.*

When $\bar{a}$ is **empty**, i.e., it evaluates nothing but the constants, we abbreviate $(\mathcal{A}, \bar{a})$ by $\mathcal{A}$. Thus $\mathcal{A} \equiv_m \mathcal{B}$, means $\mathcal{A}$ and $\mathcal{B}$ agree on all *sentences* φ with $qd(\varphi) \leq m$.

3 First-Order Reductions and Projections

This section is based on [IL95]. We want to define a reduction from sets of structures to sets of structures. Let $\mathrm{STRUC}[\tau]$ denote the set of finite structures from vocabulary τ. Given properties $K_1 \subseteq \mathrm{STRUC}[\tau_1]$ and $K_2 \subseteq \mathrm{STRUC}[\tau_2]$, our goal is to use logic to construct a mapping

$$\Gamma : \mathrm{STRUC}[\tau_1] \mapsto \mathrm{STRUC}[\tau_2]$$

such that

$$\mathcal{A} \in K_1 \iff \Gamma(\mathcal{A}) \in K_2 \tag{1}$$

Since a structure is essentially a vector of relations, a natural way to define Γ logically is by building new relations using a vector of formulas. However, in such a definition we need to take care of several issues such as how constants are defined and how different sized universes are mapped to one another. We now proceed with formal definitions.

Definition 3.1 First-Order Reduction. *([Imm87, IL95]) Let τ_1 and τ_2 be vocabularies, with $\tau_2 = \{R_1^{a_1}, \ldots, R_r^{a_r}, c_1, \ldots, c_t\}$. Given $K_1 \subseteq \mathrm{STRUC}[\tau_1]$ and $K_2 \subseteq \mathrm{STRUC}[\tau_2]$, a k-***ary first-order reduction (f.o.r.) from K_1 to K_2**, $\Gamma \stackrel{\text{def}}{=} \langle \varphi_0, \ldots, \varphi_{r+t} \rangle$, *consists of the given $t + r + 1$ formulas from $\mathcal{L}^{\tau_1}$ such that the following conditions hold:*

1. $\forall j \in \{0, r+1, \ldots, r+t\}: \ free(\varphi_j) \subseteq \{x_1, \ldots, x_k\}$
2. $\forall i \in \{1, \ldots, r\}: \ free(\varphi_i) \subseteq \{x_1, \ldots, x_{ka_i}\}$
3. *For all $\mathcal{A} \in \mathrm{STRUC}[\tau_1]$ the following conditions hold:*
   (a) $\mathcal{A} \models \exists x_1, \ldots, x_k \varphi_0$
   (b) *For* $j \in \{r+1, \ldots, r+t\}, \quad \mathcal{A} \models \exists!(x_1, \ldots, x_k)\varphi_j$[†]
4. Γ *augments a mapping,* $\Gamma : \mathrm{STRUC}[\tau_1] \mapsto \mathrm{STRUC}[\tau_2]$, *where, for* $\mathcal{A} \in \mathrm{STRUC}[\tau_1]$, $\Gamma(\mathcal{A})$ *is defined as follows:*
   (a) $U^{\Gamma(\mathcal{A})} \stackrel{\text{def}}{=} \{\langle \alpha_1, \ldots, \alpha_k \rangle \in (U^{\mathcal{A}})^k \ \mid \ \mathcal{A} \models \varphi_0[\alpha_1, \ldots, \alpha_k]\}$
   (b) $R_i^{\Gamma(\mathcal{A})} \stackrel{\text{def}}{=} \{\langle \alpha_1, \ldots, \alpha_k; \alpha_{k+1}, \ldots, \alpha_{2k}; \ldots; \alpha_{((a_i-1)k)+1}, \ldots, \alpha_{a_i k} \rangle \in (U^{\Gamma(\mathcal{A})})^{a_i} \mid \mathcal{A} \models \varphi_i[\alpha_1, \ldots, \alpha_{a_i k}]\}$
   (c) $c_j^{\Gamma(\mathcal{A})} \stackrel{\text{def}}{=} \langle \alpha_1, \ldots, \alpha_k \rangle$ *such that* $\mathcal{A} \models \varphi_{r+j}[\alpha_1, \ldots, \alpha_k]$.
5. *For all* $\mathcal{A} \in \mathrm{STRUC}[\tau_1]$*:*

$$\mathcal{A} \in K_1 \quad \Longleftrightarrow \quad \Gamma(\mathcal{A}) \in K_2$$

We say that K_1 is **first-order reducible** *to K_2 if there is such a reduction Γ.*

By restricting the formulas that define Γ, we can define even weaker reductions:

Definition 3.2 First-order and Quantifier-free Projections. *([Imm87, IL95]) A* **first-order** *(resp.* **quantifier-free***)* **projection***, or* **fop** *(resp.* **qfp***), is a first-order reduction $\Gamma : \mathrm{STRUC}[\tau_1] \mapsto \mathrm{STRUC}[\tau_2]$ defined by $\langle \varphi_0, \ldots, \varphi_{r+t} \rangle$, such that each φ_i satisfies the following conditions:*

1. *φ_i has the form*

$$\varphi_i \equiv \psi_0^i \vee \bigvee_{j=1}^{l_i} (\psi_j^i \wedge \delta_j^i) \tag{2}$$

[†]Here $\exists!\bar{x}\varphi$ stands for "there exists a *unique* $\bar{x}$ such that φ", and is an abbreviation for:

$$\exists \bar{x}(\varphi \wedge \forall \bar{y}(\varphi\frac{\bar{y}}{\bar{x}} \rightarrow \bar{y} = \bar{x}))$$

2. *Each δ^i_j is a literal (i.e., an atomic formula $R_d(x, y, z, ..)$ or its negation) from τ_1.*

3. *Each ψ^i_j is a* `first-order` *(resp.* `quantifier-free`*) formula containing only the built-in relations[‡] , i.e., no relations from the vocabulary τ_1. (In addition, for* `qfp`*'s we demand that the built in BIT relation, if it is there, not be used[§].).*

4. *The ψ^i_j's are mutually exclusive, i.e., for $j, m \in \{0, \dots, l_i\}$:*

$$\psi^i_j \models \neg\psi^i_m$$

We write $\boldsymbol{K_1 \leq_{f.o.r.} K_2}$, $\boldsymbol{K_1 \leq_{fop} K_2}$, and $\boldsymbol{K_1 \leq_{qfp} K_2}$ to denote the fact that there is, respectively, a first-order reduction, first-order projection, or quantifier-free projection from K_1 to K_2.

As defined (see [IL95]), even built-in predicates like a total-ordering on τ_2-structures are going to be explicitly produced by the reduction (using the built-in predicates in τ_1). This choice of definition is however somewhat arbitrary. We could just as well provide the built-in predicates on τ_2 for free after the rest of the reduction has been carried out.

The following fact can now easily be checked.

Fact 3.3 *([IL95]) The following statements hold when $\boldsymbol{\rho}$ is any of the three reductions* `f.o.r.`*,* `fop`*, or* `qfp`*.*

1. *All the logical classes $\mathcal{C}$ that we have defined are closed under ρ-reductions. In other words, if $K_1 \leq_\rho K_2$ then*

$$K_2 \in \mathcal{C} \quad \Rightarrow \quad K_1 \in \mathcal{C}$$

2. *ρ-reductions are closed under composition, i.e.:*

$$K_1 \leq_\rho K_2 \leq_\rho K_3 \quad \Rightarrow \quad K_1 \leq_\rho K_3$$

Moreover, it should be clear that `fop`'s and `qfp`'s are indeed projections in the sense of Valiant [Val82], i.e., each "bit" of the output structure depends on at most one "bit" of the input structure. This notion can be made precise and formal, but since we don't use this fact further in the paper we leave it as an informal statement.

4 A Problem Complete for L

We now describe the problem ORD, which we will show to be complete for L via quantifier-free projections.

[‡]Here, for totally-ordered structures, we assume the presence not only of a built-in total ordering, $\leq$, but also of a built in successor relation $S(x, y)$. This is redundant for fop's, but we can not define a successor relation from a total ordering via a quantifier-free formula, so it is necessary for qfp's.

[§]This is because we can not, via a quantifier free formula, build a BIT relation on the k-tuples which we map to.

Definition 4.1 *Let* $\mathrm{S}(x, y)$ *be a* successor *relation, in other words, a directed line graph on* n *vertices, and let* $\leq_{\mathrm{S}}$ *denote the unique total ordering consistent with* S. *Define the set of structures ORD by:*

$$\mathrm{ORD} = \{\langle\{v_1, \ldots, v_n\}, \mathrm{S}, s, t\rangle \mid s \leq_{\mathrm{S}} t\}$$

Theorem 4.2 *ORD is* L*-Complete via qfp's.*

Proof That ORD $\in$ L is obvious. We traverse the unique out-edges from s and accept iff we reach t before a dead end.

To show that ORD is L-hard, we use a reduction to ORD from a variant of the Iterated S_n Multiplication problem where S_n is the symmetric group of permutations on n elements. This problem is already known to be L-complete ([CM87, IL95]) in certain representations, and via stronger reductions. However, quantifier-free projections are quite sensitive to input representation, so our first task is to show how, with minor variations, the constructions of [CM87, IL95] can be used to show that a version of the iterated S_n multiplication problem is L-complete via qfp's.

Consider the following representation of Iterated-S_n-Mult. We are given a sequence of n permutations $\pi_1, \ldots, \pi_n \in (S_n)^n$ as a ternary relation $R(i, j, k)$, meaning "The k'th permutation takes element i to element j", and we are also given an element j', and we are asked: does the the product of the sequence of $\Pi\pi_i$ take element 1 to an element j'' where $j'' \geq j'$? Let us call this decision problem $\mathbf{\Pi S_n}$.

Lemma 4.3 *([CM87, IL95])* ΠS_n *is L-complete via qfp's.*

Proof The s-t-connectivity problem for acyclic graphs with out-degree ≤ 1 (1GAP) is the canonical complete problem for L, [Jon75] (the configuration graph of a logspace machine can be viewed as such a graph, with acyclicity assured by associating a clock with the machine and asserting that transitions proceed according to the clock), and the problem is actually complete via qfp's, [Imm87]. There is a simple (qfp) reduction from acyclic 1GAP to undirected acyclic graph reachability (i.e., undirected forest reachability) UFA, by just cutting off the one possible out-going edge from t, taking the symmetric closure of the edges, and checking if s and t are in the same connected tree.

To show that iterated S_n multiplication is complete (under NC^1 reductions) for L, Cook and McKenzie [CM87] reduce UFA to the problem by constructing a permutation of the edges of an undirected acyclic graph such that iterating this permutation corresponds essentially to a depth-first traversal of the edges of the graph, i.e., starting at the node corresponding to the start state, the iterated product of this permutation eventually carries an edge at the start state to an edge at the final state iff the machine accepts the input (see Proposition 1 of [CM87]).

Immerman and Landau [IL95] show that this construction can be carried out using first-order projections if the input representation of the permutations is as functions $\pi(i) \rightarrow j$ so that we can view the bits of j. In their construction, by attaching a long chain to the final state they guarantee that we get from the start state to the final state iff after a specified

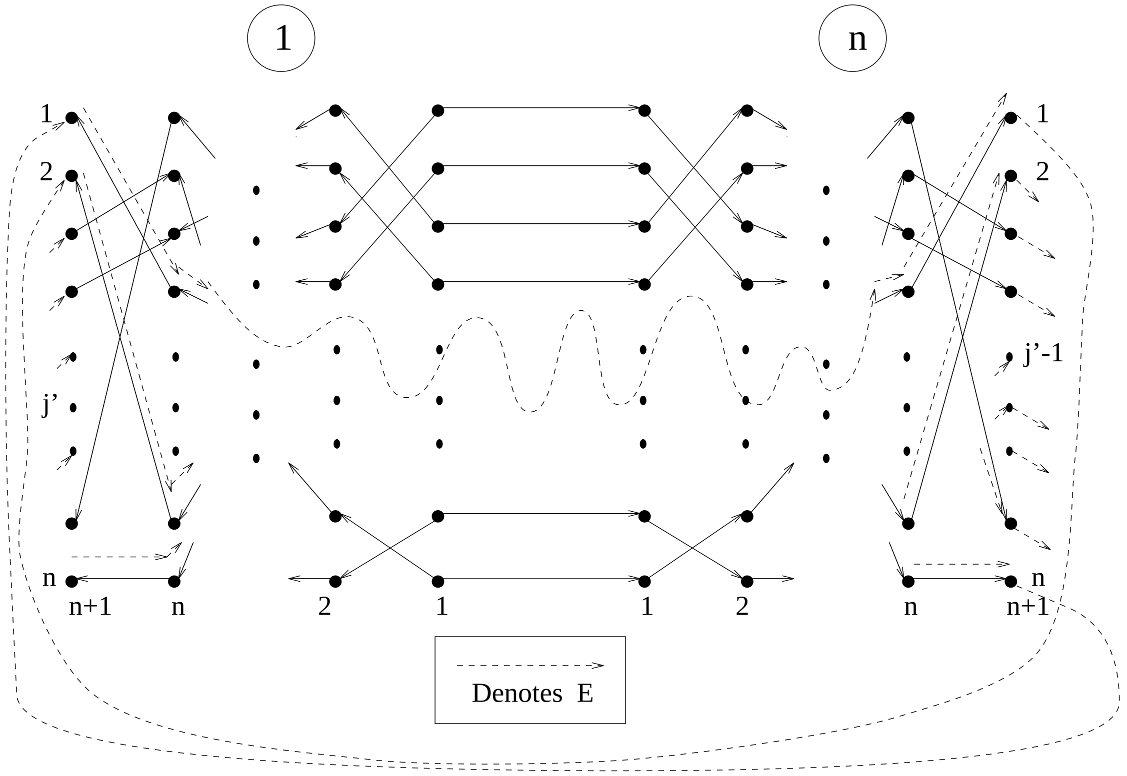

Figure 1: The Reduction.

number of iterations m of the permutation an edge at the start state has been mapped to one of the edges in the long chain (See Theorem 5.2 of [IL95]). To check that they are on one of the edges in the long chain they only need to check the highest bit of $\pi^m(start) = j''$.

The only reason their reduction is not a qfp and requires the modified representation is that they need to look at a *bit* of the resulting edge j'' in order to determine whether it is in the long chain.

We avoid this by only asking "does the start edge, *start*, lead to one of the edges greater than j' in the product $\Pi\pi_i$", where j' denotes the beginning of the long chain and can be easily defined in terms of t. It follows that our representation of ΠS_n is complete for L via qfp's. ∎

Since qfp's are closed under composition, our theorem will follow if we can show that ΠS_n is reducible to ORD via a qfp.

Lemma 4.4 $\Pi S_n \leq_{\text{qfp}}$ ORD.

Proof We want to construct a directed line graph $E(\bar{x}, \bar{y})$ and obtain tuples $\bar{a}$ and $\bar{b}$, such that $\bar{a} \leq_E \bar{b}$ if and only if the product of the permutations $\langle R(.,.,k) \mid 1 \leq k \leq n \rangle$, maps 1 to $j'' \geq j'$. We will construct E in terms of R. Please see Figure 1. Informally, E is going to connect up two copies of the graph of the input sequence of n elements of S_n together, (one labeled "**1**", the other labeled "**n**", in the figure) the first going backward, the second going forward, and the ends connected in sequence by increasing order of index, *starting at j' and* *then going to* 1 *after reaching* n. In other words we will, given a sequence $\sigma_1, \sigma_2, \ldots, \sigma_n$, construct the graph E from the graph corresponding to

$$(\Pi_{i=1}^{n} \sigma_i)^{-1}\ \Pi_{i=1}^{n} \sigma_i$$

with the ends connected as described. Formally, here is E:

$$\begin{aligned}
&E((h,v,c),(h',v',c')) \equiv \\
&\quad ((c = 1 \wedge c' = 1 \wedge h = h'+1 \wedge R(v',v,h')) \vee \\
&\quad\ (c = 1 \wedge c' = n \wedge h = 1 \wedge h' = 1 \wedge v = v') \vee \\
&\quad\ (c = n \wedge c' = n \wedge h' = h+1 \wedge R(v,v',h)) \vee \\
&\quad\ (c = n \wedge c' = 1 \wedge v \neq n \wedge v \neq j'-1 \wedge h = n+1 \wedge h' = n+1 \wedge v' = v+1) \vee \\
&\quad\ (c = n \wedge c' = 1 \wedge v = n \wedge h = n+1 \wedge h' = n+1 \wedge v' = 1))
\end{aligned}$$

Observe that E is a line graph over the following set U of triples:

$$\{(a_1,a_2,a_3) \mid a_1 \in \{1,\ldots,n+1\}\ ,\ a_2 \in \{1,\ldots,n\}\ ,\ a_3 \in \{1,n\}\}$$

Let $\bar{a} = (1,1,1)$ and $\bar{b} = (n+1,n,n)$. Then $\bar{a} \leq_E \bar{b}$ iff the iterated product of the permutations maps 1 to $j'' \geq j'$.

E constitutes a qfp. First, this is because the expression for E is quantifier-free. Now, in the expression for E above, for the five clauses that constitute it, call the piece of the clause that does not reference the input relation R the *prefix*. Observing the values demanded for d, d', and v in each clause, we see that for each pair of tuples $(h,v,c),(h',v',c') \in U$, at most one of the five disjuncts in the expression for E has a satisfied prefix. Thus, $E((h,v,c),(h',v',c'))$ depends on at most one position of the relation R. ∎

That concludes the completeness proof for ORD. ∎

We mention here that ORD's L-completeness is of independent interest for characterizing some database query languages, and it was mentioned as an open problem in that setting ([Lib94, LW94]). The particular problem that [Lib94] were concerned with was recognizing whether an input graph is a successor graph. But, as they observed, there is a simple first-order reduction from ORD to this problem. Given a successor graph S along with points s and t on it, we construct a new graph S':

$$S'(x,y) \equiv (x \neq s \wedge x \neq t \wedge S(x,y)) \vee (x = s \wedge S(t,y)) \vee (x = t \wedge \forall z \neg S(z,y))$$

In other words, the edge (if any) out of t is cut off and replaced by a new edge to the beginning of the line graph, and the edge out of s is cut off and replaced by a new edge to the successor of t. It is easy to see that S' is a line graph iff s precedes t in S.

5 ORD $\notin$ (FO + COUNT + S)

Observe that in order to prove ORD $\notin$ (FO + COUNT + S) we only need to exhibit, for each sentence φ in (FO + COUNT), a pair of successor structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \in$ ORD and $\mathcal{B} \notin$ ORD, such that φ can not distinguish them, i.e., φ has the same truth value for both. The required successor relation is already provided by these structures.

5.1 E-F Game for $(\mathrm{FO} + \mathrm{COUNT})$

In this section we define an E-F game for first-order logic with counting [IL90, CFI92], and we state the main lemma relating it to the logic. We use this game in the next section to prove $\mathrm{ORD} \notin (\mathrm{FO} + \mathrm{COUNT})$. We also describe a slightly modified version of the game, and in section 7 we show a tighter correspondence between this modified game and logic with counting, as well as with lower bound questions regarding ThC^0.

Definition 5.1 E-F Game for (FO + COUNT). *The m round E-F game for first-order logic with Counting Quantifiers, denoted:* $\boldsymbol{G_m((\mathcal{A}, \bar{a}), (\mathcal{B}, \bar{b}))}$*, is played between two players called I and II, and consists of m consecutive rounds. At the start and after each round PLAYER I WINS if $\bar{a} \not\cong_{\mathcal{P}_{\mathcal{A},\mathcal{B}}} \bar{b}$. Each round consists of the following:*

Player I chooses a subset A' of A (or B' of B)
Player II chooses a subset B' of B (A' of A)
such that $|A'| = |B'|$
Player I picks b' of B' (a' of A')
and sets $\bar{b} \leftarrow \bar{b},b'$ ($\bar{a} \leftarrow \bar{a},a'$)
Player II responds with a' of A' (b' of B')
and sets $\bar{a} \leftarrow \bar{a},a'$ ($\bar{b} \leftarrow \bar{b},b'$)

PLAYER II WINS if PLAYER I does not win on any of the m rounds.

Definition 5.2 Winning Strategy. *We say that PLAYER II has a* **winning strategy** *in $G_m((\mathcal{A}, \tilde{a}),(\mathcal{B}, \tilde{b}))$, and we denote this by* $\boldsymbol{(\mathcal{A}, \tilde{a}) \sim_m (\mathcal{B}, \tilde{b})}$*, if PLAYER II can win the game regardless of the moves made by PLAYER I.*

The following lemma appears in one form or another in [IL90, CFI92, EI95]. It shows that the above E-F game contains the power of first-order logic with counting quantifiers.

Lemma 5.3 ([IL90, CFI92, EI95]) $(\mathcal{A}, \bar{a}) \sim_m (\mathcal{B}, \bar{b}) \implies (\mathcal{A}, \bar{a}) \equiv_m (\mathcal{B}, \bar{b})$

In section 7 we will show that for the following modified game, the converse of the lemma also holds along with a tight correspondence to lower bounds for ThC^0. In the **Modified Game**:

1. PLAYER I is only allowed to choose subsets A' such that $|A'| = a_i$ for some number a_i that has been previously picked in the game (or one of the constants 1 or n).
2. PLAYER II will then have to choose a subset B' such that $|B'| = b_i$.

Note however that, for the lower bound, using the ordinary game suffices. It just gives us a slightly stronger result.

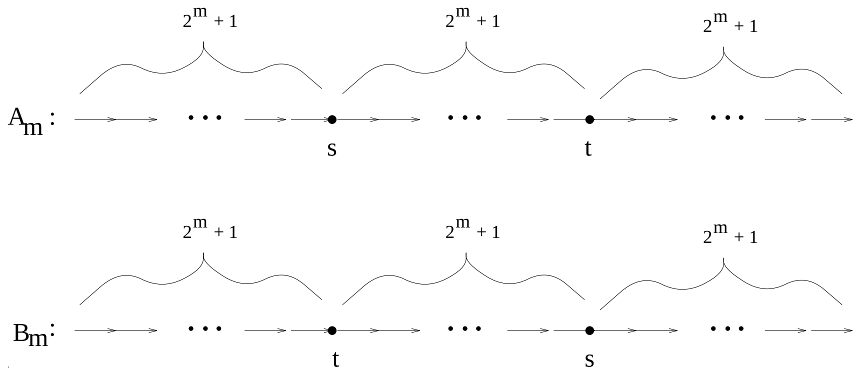

Figure 2: A_m and B_m.

5.2 Lower Bound

We provide, for each m, a pair of Successor graphs A_m and B_m, such that $A_m \in$ ORD and $B_m \notin$ ORD, but $A_m \sim_m B_m$. It then follows from Lemma 5.3, that ORD $\notin$ (FO+COUNT), and because A_m and B_m are both successor relations it follows that ORD $\notin$ (FO+COUNT+S).

The definitions of A_m and B_m are simple. Please see Figure 2. A_m is just a successor relation of length $3 \times (2^m + 1)$ with a constant s at position¶ $2^m + 1$ and t at position $2 \times (2^m + 1)$. B_m is exactly A_m with the positions of s and t switched.

Theorem 5.4 $(A_m, \bar{a}) \sim_m (B_m, \bar{b})$

In the conference version of this paper [Ete95], we proved this result by giving an explicit winning strategy for Player II in the EF game. A related but more general result obtained recently and independently by Nurmonen [Nur96] implies Theorem 5.4. In this paper rather than repeat the winning strategy argument, we only show that our lower bound follows from his result. In order to state Nurmonen's result, we need the following definitions.

Definition 5.5 *Let* $\boldsymbol{d(a,b)}$ *denote the relational distance between two points in a structure* $\mathcal{A}$*, defined inductively by:*

$$\begin{aligned} &d(a,a) = 0 \\ &d(a,b) \le l \;\; \Leftrightarrow \;\; \exists c \; d(a,c) \le l-1, \; \exists i, \; \exists \; \bar{u} \in A^k \text{ containing both } c \text{ and } b, \text{ s.t. } \mathcal{A} \models R_i(\bar{u}). \\ &d(a,b) = l \;\; \Leftrightarrow \;\; d(a,b) \le l \; \& \; d(a,b) \not\le l-1 \end{aligned}$$

Let $\boldsymbol{N^d(a)}$= $\{b \mid d(a,b) \le d\}$.

Two points a and b in structures $\mathcal{A}$ and $\mathcal{B}$ are said to have `identical` d`-neighborhoods` *if there is an isomorphism between the substructures induced by $N^d(a)$ and $N^d(b)$ that maps a to b. Two structures $\mathcal{A}$ and $\mathcal{B}$ are said to be* d`-indistinguishable` *if and only if they have exactly the same number of of points with any given d-neighborhood.*

¶We begin indexing at 0.

The following is a restatement in our setting of a theorem by [Nur96]:

Theorem 5.6 *([Nur96]) If a pair of structures $\mathcal{A}$ and $\mathcal{B}$ are 3^k-indistinguishable then $\mathcal{A} \sim_k \mathcal{B}$.*

Nurmonen [Nur96], actually proves a slightly different result about first-order logic with any set of *unary generalized quantifiers.* Each unary generalized quantifier $Qy\varphi$ can be thought of as computing a boolean query (computable or not) on the unary relation defined by the formula φ and the variable y that is bound. Our notion of a counting quantifier $(\exists i\ x)$ *where the number variable i remains free* does not exactly fit into the category of unary generalized quantifiers. However, Hella [Hel92] defined a "bijective" EF game and showed it captures at least the power of first-order logic with unary generalized quantifiers in the sense that a winning strategy for Player II in the game implies indistinguishability in the logic. In the "bijective" game, on each move Player I chooses a set of elements and Player II must respond with a bijection between the elements of the two structures, which defines a (necessarily equinumerous) response set on the other structure as the image of the set chosen by Player I. But a winning strategy for Player II in the bijective game implies a winning strategy in our counting game: we simply choose the same number as a response whenever a number is chosen on one structure, and we respond to counting moves according to the bijection in II's response in the bijective game. (We note here that the winning strategy we provided in [Ete95] was also a bijective one, and thus gives a win in the bijective game as well).

Nurmonen's result [Nur96] is the analog of Theorem 5.6 for the bijective game, i.e., it says that if the indistinguishability condition holds then the winning strategy exists in the bijective game, and thus by the argument just given it implies Theorem 5.6.

To conclude (a slightly modified version of) Theorem 5.4 from Theorem 5.6 it suffices to observe that if we increase the length of the "gaps" in structures A_m and B_m from $2^m + 1$ to $2 \times 3^m + 1$, call these modified structures A'_m and B'_m, then A'_m and B'_m have exactly the same number of points with each given 3^m-neighborhood, and thus by the theorem $A'_m \sim_m B'_m$.

Lemma 5.3 and Theorem 5.4 yield:

Corollary 5.7 ORD $\notin$ (FO + COUNT + S)

Theorem 4.2 then yields:

Theorem 5.8
$(\mathrm{FO} + \mathrm{COUNT} + \mathrm{S}) \subsetneq (\mathrm{FO} + \mathrm{DTC} + \mathrm{S})\ (= \mathrm{L})$

6 ORD and "Pseudo-Sparseness"

We have shown that the property ORD is L-complete, and this bears more attention. As mentioned earlier, this has implications for characterizing certain database query languages and had been mentioned as an open problem in that setting, [Lib94, LW94].

For a more complexity theoretic significance, observe that ORD is remarkably "sparse". In a sense, every structure in ORD essentially looks the same. Clearly, *if we ignore the possible reorderings of the vertices*, there are only polynomially many structures of size n in ORD, based on the locations of the two constants.

Thus, we have a complete graph problem for L that is sparse up to "isomorphism", i.e., when we equate structures that differ only in the way the vertices are ordered. The following definition makes this notion precise:

Definition 6.1 *An order-independent (isomorphism invariant) property K is* **pseudo-sparse** *if there is a polynomial $p(n)$, such that for all n*

$$|\{\mathcal{A} \in K \mid |U^{\mathcal{A}}| = n\}/\cong| \leq p(n)$$

This is the same definition as ordinary sparseness except we are dealing with an order-independent property and we are only interested in the number of equivalence classes of structures modulo isomorphism, rather than the total number of structures, of a given size. We thus have

Corollary 6.2 L *has a pseudo-sparse complete problem.*

A question on the periphery that arises from this is whether the theorem of Mahaney [Mah82], which states that there are no sparse sets complete for NP unless P = NP, still goes through when we ask: Are there pseudo-sparse complete sets of structures for $SO(\exists) = \mathrm{NP}$ ([Fag74]), even after we mod out by *isomorphic* structures? The point is that this issue doesn't arise with the Turing machine characterization of NP, because no two distinct inputs are "isomorphic". But in reality, for example in the case of a complete problem like Hamiltonian Cycle, we are really more interested in the fact that there are exponentially many *non-isomorphic* graphs of a given size with hamiltonian cycles, not that there are exponentially many ways to present the same graph by reordering the vertices. Here we give a partial answer to this question*.

Theorem 6.3 *If there is a pseudo-sparse order-independent property complete for* NP *(via, e.g.,* P*-time many-one reductions) then the Polynomial Time Hierarchy collapses to Σ_3.*

Proof Suppose that a pseudo-sparse property K is NP-complete. We are going to use this assumption to show that co-NP/*poly* = NP/*poly*. This then implies that $\Sigma_3 = \Pi_3$ (see, e.g., [BDG88]) which yields the conclusion we seek.

Since K is pseudo-sparse, for each size n there are at most a polynomial number of canonical representatives $\mathcal{A}_1, \ldots, \mathcal{A}_{p(n)}$ for the structures of size $\leq n$ in K. Without loss of generality, we may assume K is a graph property (if not, it can easily be converted to one via a simple one-to-one reduction computable in P-time). It is well known that graph-non-isomorphism is in the complexity class AM, and that $AM/poly = \mathrm{NP}/poly$ (see [BM88]), i.e., graph non-isomorphism is in NP/*poly*. But we now use the graph non-isomorphism algorithm in NP/*poly* to show that $K \in$ co-$NP/poly$, and thus $NP \subseteq$ co-$NP/poly$, yielding the result.

*Thanks to Ken Regan for pointing out this approach to me.

We define M to be a non-deterministic Turing machine that is given the canons $\mathcal{A}_1, \ldots, \mathcal{A}_{p(n)}$ as advice on an input x of size n, along with the advice needed to compute graph non-isomorphism. The non-deterministic machine M will accept if and only if $x \notin K$, thus yielding $K \in$ co-$NP/poly$. Given x, M will first run the NP/poly non-isomorphism test on x and $\mathcal{A}_1$. If the answer is "yes" (i.e., they are non-isomorphic) then it will run the test on G and A_2, then A_3, and so on up to $A_{p(n)}$. If the answer is ever "no" then it will reject, otherwise after all the checks it will accept. Clearly, M will accept if and only if x is not isomorphic to any of the canons $\mathcal{A}_1, \ldots, \mathcal{A}_{p(n)}$, and thus $x \notin K$. We thus have $K \in$ co-$NP/poly$. But this implies co-NP/*poly* = NP/*poly*, and that concludes the proof. ∎

It would be interesting to know if we can obtain a stronger conclusion than that of Theorem 6.3. In particular, as in Mahaney's theorem [Mah82] for sparse sets, would the existence of pseudo-sparse complete sets for NP imply P = NP?

Theorem 6.3 thus gives us a situation where L and NP apparently differ on a fairly natural structural property. If follows from a result of Allender, Balcazar, and Immerman [ABI93] that there are no truly sparse complete problems for L (nor for P or NP or any other complexity class closed under first-order reductions) via qfp's. Furthermore, recently, Cai and Sivakumar [CS95] have shown that there is no problem that is truly sparse as well as complete via NC^1 reductions for L unless L = NP. As we have seen, ORD is pseudo-sparse and L-complete via Quantifier Free Projections. This raises the intriguing question of where in the complexity hierarchy the existence of such pseudo-sparse complete properties stops. In particular:

Question 6.4 *Are there pseudo-sparse complete properties (under, say, qfp's) for* NL *or* P*?*

7 The Power of the E-F Game

In this section we point out that the modified version of the E-F game described in section 5.1 has a tight correspondence to the logic (FO + COUNT), in that, *in principle*, if $\mathrm{ThC}^0 \neq$ L then an E-F game proof of this fact exists.

More generally we will see that for any problem K, if it can be shown to be outside of ThC^0 then this can be shown via the E-F game in a precise sense to be defined. This includes, for example, complete problems for NC^1 and thus the same statement could have been made about the ThC^0 vs. NC^1 question. This fact is known, e.g., for AC^0 and the E-F game for first-order logic (see, e.g., [Str94]), but the situation here is somewhat more subtle and is worth describing. In particular, the correspondence *does not* hold for the ordinary counting game where Player I picks a set of *any size* to which Player II replies with an equal sized subset.

To simplify our discussion, suppose that for (FO + COUNT), instead of having a separate domain of numbers, we always work on a single domain of "vertices" with a built-in $\leq$ and BIT predicate.

We now define the notion of a k-type for the logic (FO + COUNT), which is similar to the k-types for first-order and other logics (e.g., [Fra54]). It is however different in that for a given tuple from a given structure the definition of its k-type requires an ordering to be present on the domain of that structure. (Everything here would carry through on structures with numbers as well.)

Definition 7.1 **(k,m)-types.** *Given a vocabulary $\sigma = \langle R_1, \dots, R_r, c_1, \dots, c_t\rangle$ The set of (k,m)-types, $\Lambda^{k,m}$ is defined inductively as follows:*

$$\begin{array}{rcl} \Lambda^{0,m} & \doteq & \text{“The set of isomorphism types of an } m \text{ tuple from the vocabulary } \sigma\text{”} \\ \Lambda^{k+1,m} & \doteq & \{D = \{(S,i) \mid i \in \{1,\dots,m\} \ \& \ S \subseteq \Lambda^{k,m+1}\} \mid (S,i) \in D \Rightarrow \forall S' \supsetneq S \ (S',i) \notin D\} \end{array}$$

(By “the set of isomorphism types..”, we mean that each element in the set specifies uniquely and consistently all the relationships that hold and don't hold between an ordered set of m elements over the vocabulary σ plus equality and the other built-in relations.)

The point of this definition will hopefully become clearer when we now define the (k,m)-type for a particular m-tuple from a given structure:

Definition 7.2 *Given a structure and tuple $(A, a_1, \dots, a_m)$, we define its (k,m)-type, $\boldsymbol{\tau_A^{k,m}(\bar{a})}$, inductively. $\tau_A^{k,m}(\bar{a})$ is defined to be an element of $\Lambda^{k+1,m}$, such that:*

$$\begin{array}{rcl} \tau_A^{0,m}(\bar{a}) & = & \text{“The isomorphism type of } \bar{a}.\text{”} \\ (S,i) \in \tau_A^{k+1,m}(\bar{a}) & \iff & \exists a_i \ a' \in A \text{ such that } \tau^{k,m+1}(\bar{a},a') \in S \ \& \ S \text{ is } \texttt{minimal}, \text{ i.e.,} \\ & & \forall S' \subsetneq S \ \neg\exists a_i \ a' \text{ such that } \tau^{k,m+1}(\bar{a},a') \in S' \end{array}$$

(By “the isomorphism type of $\bar{a}$”, we mean all relationships that hold and don't hold between the ordered elements $a_1, \dots, a_m \in A^m$.)

The reader can verify that the definition of $\tau_A^{k,m}(\bar{a})$ is consistent. The goal of the definition is to capture the set-theoretic content of a structure's depth-k equivalence class in the logic with counting. The complication in the definition arises from the fact that we must only have *minimal* sets (S,i) for which $\exists a_i a' \ \tau^{k,m+1}(\bar{a},a') \in S$ in $\tau^{k+1,m}(\bar{a})$ in order for the characterization in theorem 7.4 to be true. If we did not stipulate the minimality condition Player II could have a winning strategy in the k-round game, and yet the k-types would not need to be identical. This is because when Player I chooses a set containing a certain set of $k-1$ types Player II only needs to respond with a set containing a subset of those types, rather than exactly the same set. The minimality condition forces the two k-types to be identical, as will be seen in the proof of $(1 \Rightarrow 2)$ in Theorem 7.4.

Proposition 7.3

1. *There are a finite number of elements in $\Lambda^{k,m}$.*
2. *For each $D \in \Lambda^{k,m}$, there is a quantifier depth k formula φ_D in $(\mathrm{FO} + \mathrm{COUNT})$ such that*

$$(A,\bar{a}) \models \varphi_D \iff \tau_A^{k,m}(\bar{a}) = D$$

Proof (1) is obvious from the definition. In particular, the base case says that for a finite vocabulary the isomorphism type of a k-tuple is finite.

We prove (2) by induction on k. For the base case, we can express $D \in \Lambda^{0,m}$ by a quantifier free formula that specifies precisely all the relationships.

For $D \in \Lambda^{k+1,m}$, φ_D is defined by:

$$\begin{aligned}\varphi_D \quad \equiv \quad & \Big(\bigwedge_{(S,i)\in D} (\exists x_i\ x_{k+1} \bigvee_{\lambda\in S} \varphi_\lambda \ \wedge \bigwedge_{S'\subsetneq S} \neg\exists x_i\ x_{k+1} \bigvee_{\lambda'\in S'} \varphi_{\lambda'})\Big) \ \wedge \\ & \bigwedge_{i\in[m],\ \{T\subseteq\Lambda^{k,m+1} \mid \forall T'\subseteq T\ (T',i)\notin D\}} \neg\exists x_i\ x_{k+1} \bigvee_{\lambda''\in T} \varphi_{\lambda''}\end{aligned}$$

φ_D is just a restatement as a formula, using definition 7.2 for the type $\tau_A^{k,m}(\bar{a})$, that $\tau_A^{k,m}(\bar{a}) = D$. The first line says "For each $(S,i) \in D$ there are a_i a''s such that (inductively) $\tau_A^{k,m+1}(\bar{a},a') \in S$, and furthermore S is *minimal*". The second line just has the additional stipulation that "nothing extra appears in D besides what was specified in line 1". This is specified by saying that for all (T,i) such that for all subsets T' of T $(T',i) \notin D$ it is not the case that there are a_i a''s such that $\tau_A^{k+1,m}(\bar{a},a') \in T$. ∎

When clear from the context, we will from now on use $\tau_A^k(\bar{a})$ to denote $\tau_A^{k,m}(\bar{a})$.

Theorem 7.4 *TFAE:*

1. $(A,\bar{a}) \equiv_k (B,\bar{b})$
2. $\tau_A^k(\bar{a}) = \tau_B^k(\bar{b})$
3. $(A,\bar{a}) \sim_k (B,\bar{b})$

Proof $1 \Rightarrow 2$. Suppose $\tau_A^k(\bar{a}) \neq \tau_B^k(\bar{b})$. Then, w.l.o.g., there is an element $(\{\tau_1,\ldots,\tau_l\},i) \in \tau_A^k(\bar{a})$, but $(\{\tau_1,\ldots,\tau_l\},i) \notin \tau_B^k(\bar{b})$. But then, w.l.o.g., we can assume that for all strict subsets S' of $\{\tau_1,\ldots,\tau_l\}$, $(S',i) \notin \tau_B^k(\bar{b})$. Here is why: $\{\tau_1,\ldots,\tau_l\}$ is minimal and does not belong to $\tau_B^k(\bar{b})$. If there was a smaller subset S' with $(S',i) \in \tau_B^k(\bar{b})$, then since it is also minimal it would be the element we are looking for, i.e., we would have $(S',i) \in \tau_B^k(\bar{b})$ but $(S'',i) \notin \tau_A^k(\bar{a})$ for all subsets S'' of S', because S'' would also be a strict subset of S which the minimality condition disallows.

Now, since each type τ_j has a corresponding formula φ_j such that $(A,\bar{a},a') \models \varphi_j$ if and only if $\tau_A^k(\bar{a},a') = \tau_j$, we find that $(A,\bar{a})$ and $(B,\bar{b})$ disagree on

$$\exists x_i\ x_{k+1} \bigvee_{j=1}^{l} \varphi_j$$

This is because any set of size b_i in B must have a type in it different than any from the set $\{\tau_1,\ldots,\tau_l\}$.

$2 \Rightarrow 3$. Suppose $\tau_A^k(\bar{a}) = \tau_B^k(\bar{b})$. Suppose, w.l.o.g., that Player I picks a set of size a_i in A. let S be the set of k-1 types "induced" by this set. Since $\tau_A^k(\bar{a}) = \tau_B^k(\bar{b})$, there must be a set of size b_i in B such that the types induced by it are $S' \subseteq S$. Let II pick the set S', then regardless of what point in S' player I picks, II can respond with a point in S with the same $k-1$ type.

$3 \Rightarrow 1$. This part of the proof is virtually the same as the proof in the usual game [IL90]. Let $\varphi = \exists i x_{k+1} \psi$ (this is the only important case, since for boolean combinations the fact is immediate), suppose, w.l.o.g, that $(\mathcal{A}, \bar{a}) \models \varphi$. Then let PLAYER I pick a subset A' of A of size a_i such that, for each $a' \in A'$, $(\mathcal{A}, \tilde{a}, a') \models \psi$. PLAYER II answers according to its winning strategy with a subset B' of B such that $b_i = |B'|$. Now, for any arbitrary $b' \in B'$ that PLAYER I chooses, there is an $a' \in A'$ such that $(\mathcal{A}, \bar{a}, a') \sim_{m,p} (\mathcal{B}, \bar{b}, b')$. Thus, by induction, $(\mathcal{A}, \bar{a}, a') \models \psi \Leftrightarrow (\mathcal{B}, \bar{b}, b') \models \psi$. Thus, since for each $a' \in A'$, $(\mathcal{A}, \bar{a}, a') \models \psi$, we have, for each $b' \in B'$, $(\mathcal{B}, \bar{b}, b') \models \psi$. Hence $(\mathcal{B}, \tilde{b}) \models \varphi$. Thus $(\mathcal{A}, \tilde{a}) \models \varphi \iff (\mathcal{B}, \tilde{b}) \models \varphi$. ∎

From Proposition 7.3 and Theorem 7.4, we can now conclude the following general connection between lower bounds against ThC^0 and E-F game winning strategies.

Theorem 7.5 *Given a problem K, TFAE:*

1. $K \notin \mathrm{ThC}^0$
2. *There is a sequence (A_i, B_i) of structures, with built in $\leq$ and* BIT *predicates, such that, for all i, $A_i \in K$ and $B_i \notin K$, and $A_i \sim_i B_i$.*

Proof $2 \Rightarrow 1$ is obvious. For $1 \Rightarrow 2$, suppose to the contrary that for some i, for all $A \in K$ and $B \notin K$ we have $A \not\sim_i B$. But then K is a union of equivalence classes of $\sim_i$. From Proposition 7.3 and Theorem 7.4 we know that each such equivalence class can be expressed as a quantifier depth i formula. Moreover, since there are only a finite number of such equivalence classes, their union can also be expressed, via a disjunction, as a quantifier depth i formula. Thus $K \in (\mathrm{FO} + \mathrm{COUNT})$ over structures with ordering and BIT, which means exactly that $K \in \mathrm{ThC}^0$. ∎

Thus, for example, $\mathrm{ThC}^0 \neq \mathrm{L}$ if and only if part 2 of the theorem holds for any L-complete problem K.

Note that we can adjust the amount of non-uniformity in Theorem 7.5 to whatever we like, according to the built-in predicates we provide. As it is given, the theorem equates separation from first-order uniform ThC^0 to the existence of a winning strategy for Player II. With arbitrary built in predicates, the separation would be from non-uniform ThC^0. However, these results only tell us that winning strategies in these EF games are *in principle* necessary and sufficient to prove lower bounds. It is indeed a very challenging open problem to come up with explicit strategies even in the first-order game with, say, a built-in BIT predicate, where we know of the *existence* of a winning strategy because of the lower bounds [Ajt83, FSS84, Has86] that prove PARITY is not in the class AC^0.

8 Conclusion

We have proven that the property ORD is L-complete and can not be expressed in $(\mathrm{FO} + \mathrm{COUNT} + \mathrm{S})$. This holds even as $(\mathrm{FO} + \mathrm{DTC} + \mathrm{S}) = \mathrm{L}$ and $(\mathrm{FO} + \mathrm{COUNT} + \leq) = \boldsymbol{ThC^0}$. The goal here has been to approach the ThC^0 vs. L question via weak forms of ordering.

If we weaken the form of ordering on numbers to just a total ordering *without* BIT, and strengthen the ordering on vertices from a successor relation to a total ordering, can we still prove a lower bound? Note that we can still express things like "the size of the universe is even" because we can express addition:

$$+(x, y, z) \equiv \exists ! y \; w \; (x < w < z)$$

and then EVEN is expressed by $\exists j(j + j = n)$. Lindell [Lin95] has recently given such a lower bound: in such a setting we can not even tell if the size of our structure is prime. Thus, removing the BIT predicate from the number domain would significantly weaken the logical language.

We have pointed out that ORD is a pseudo-sparse (sparse up to reordering of vertices) complete problem for L, whereas we have shown that such complete problems for NP would imply a collapse of the polynomial time hierarchy, thus pointing out a fairly natural structural property on which L and NP apparently differ. This leads to obvious questions about whether NL and P possess such complete problems.

We have observed that for a modified version of the E-F game for first-order logic with counting quantifiers, there is a very tight correspondence between separation questions from the class ThC^0 and the existence of winning strategies for Player II. This and the analogous tight correspondences to E-F games for other complexity classes highlight the potential of descriptive complexity, while at the same time they present it with a fundamental challenge: Can we prove separations of complexity classes using E-F games (explicitly), even in settings such as AC^0 where we already know separations?

Acknowledgments

Thanks to Neil Immerman for pointing out what led to the proof that ORD is L-hard, and for all his help and guidance. Thanks to Ron Fagin for bringing Nurmonen's work to my attention. Thanks to Ken Regan for pointing out the approach of Theorem 6.3. Thanks to Howard Straubing for some helpful comments. Thanks to Leonid Libkin for pointing out the application of ORD's L-completeness to his work on query languages with L. Wong. Thanks finally to the anonymous referees for a number of useful comments that have improved this paper.

References

[ABI93] E. Allender, J. Balcazar, and N. Immerman. A first-order isomorphism theorem. In *Annual Symposium on Theoretical Aspects of Computer Science*, 1993.

[Ajt83] M. Ajtai. Σ^1_1-formulae on finite structures. *Ann. Pure and Appl. Logic*, 24:1–48, 1983.

[BDG88] J. L. Balcazar, J. Diaz, and J. Gabarro. *Structural Complexity I and Structural Complexity II.* EATCS Monographs on Theoretical Computer Science. Springer Verlag, 1988.

[BIS90] D. Mix Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *JCSS*, 41:274–306, 1990.

[BM88] L. Babai and S. Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36, 1988.

[CFI92] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, **12** (4):389–410, 1992.

[CM87] S. Cook and P. McKenzie. Problems complete for deterministic logarithmic space. *Journal of Algorithms*, 8:385–394, 1987.

[CR80] S. A. Cook and C. W. Rackoff. Space lower bounds for maze threadability of restricted machines. *SIAM J. Comput.*, 9 (3):636–652, 1980.

[CS95] J. Y. Cai and D. Sivakumar. The resolution of a Hartmanis conjecture. pages 362–371, 1995. FOCS'95.

[EI94] K. Etessami and N. Immerman. Reachability and the power of local ordering. In *11th Symp. on Theoretical Aspects of Comput. Sci.*, pages 123–135. To appear in *Theoretical Computer Science*, 1994.

[EI95] K. Etessami and N. Immerman. Tree canonization and transitive closure. In *10th Symp. on Logic in Comp. Sci.*, 1995.

[Ete95] K. Etessami. Counting quantifiers, successor relations, and logarithmic space. In *10th Structure in Complexity Theory Conf.*, pages 2–11, 1995.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. Karp, editor, *The Complexity of Computation*, volume 7. SIAM-AMS Proceedings, 1974.

[Fra54] R. Fraïssé. Sur quelques classifications des systèms de relations. *Publ. Sci. Univ. Alger. I.*, 1:35–182, 1954.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial time hierarchy. *J Math Systems Theory*, 18:13–27, 1984.

[Has86] J. Hastad. Almost optimal lower bounds for small depth circuits. In *Proc. 18th ACM Symposium on Theory of Computing (STOC)*, pages 6–20, 1986.

[Hel92] L. Hella. Logical hierarchies in PTIME. In *7th Symposium on Logic in Computer Science*, pages 360–368, 1992.

[IL90] N. Immerman and E. Lander. Describing graphs: A first-order approach to graph canonization. In Alan Selman, editor, *Complexity Theory Retrospective*, pages 59–81. Springer-Verlag, 1990.

[IL95] Neil Immerman and Susan Landau. The complexity of iterated multiplication. *Information and Computation*, 116(1):103–116, 1995.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM J. Comput.*, 16:4:760–778, 1987.

[Imm89] N. Immerman. Descriptive and computational complexity. In J. Hartmanis, editor, *Computational Complexity Theory*, volume 38, pages 75–91. Proc. Symp. in Applied Math., American Mathematical Society, 1989.

[Jon75] N. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11, 1975.

[Lib94] L. Libkin. personal communication, 1994.

[Lin95] S. Lindell. unpublished manuscript. available from slindell@haverford.edu, 1995.

[LW94] L. Libkin and L. Wong. New techniques for studying set languages, bag languages, and aggregate functions. In *PODS*, pages 155–166, 1994.

[Mah82] S. Mahaney. Sparse complete sets for np: Solution of a conjecture of Berman and Hartmanis. *JCSS*, 25:130–143, 1982.

[Nur96] J. Nurmonen. On winning strategies with unary quantifiers, 1996. To appear in *Journal of Logic and Computation.*

[Poo93] C. K. Poon. Space bounds for graph connectivity problems on node-named JAGs and node-ordered JAGs. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 218–227, 1993.

[Str94] H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhauser, 1994.

[Val82] L. G. Valiant. Reducibility by algebraic projections. *L'Enseignment Mathèmatic*, 28(3-4):253–268, 1982.