A Crevice on the Crane Beach: Finite-Degree Predicates

Michaël Cadilhac
WSI, Universität Tübingen
Email: michael@cadilhac.name

Charles Paperman
WSI, Universität Tübingen
Email: charles.paperman@gmail.com

***Abstract*—First-order logic (FO) over words is shown to be equiexpressive with FO equipped with a restricted set of numerical predicates, namely the order, a binary predicate $\mathbf{MSB}_0$, and the *finite-degree* predicates: $\mathbf{FO}[\mathcal{A}_{\mathcal{RB}}] = \mathbf{FO}[\leq, \mathbf{MSB}_0, \mathcal{F}_{\mathcal{IN}}]$. The Crane Beach Property (CBP), introduced more than a decade ago, is true of a logic if all the expressible languages admitting a neutral letter are regular. Although it is known that $\mathbf{FO}[\mathcal{A}_{\mathcal{RB}}]$ does not have the CBP, it is shown here that the (strong form of the) CBP holds for both $\mathbf{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ and $\mathbf{FO}[\leq, \mathbf{MSB}_0]$. Thus $\mathbf{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ exhibits a form of locality and the CBP, and can still express a wide variety of languages, while being one simple predicate away from the expressive power of $\mathbf{FO}[\mathcal{A}_{\mathcal{RB}}]$. The counting ability of $\mathbf{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ is studied as an application.**

I. Introduction

Ajtai [1] and Furst, Saxe, and Sipser [2] showed some 30 years ago that *Parity*, the language of words over $\{0, 1\}$ having an even number of 1, is not computable by families of shallow circuits, namely AC^0 circuits. Since then, a wealth of precise expressiveness properties of AC^0 has been derived from this sole result [3], [4]. Naturally aiming at a better understanding of the core reasons behind this lower bound, a continuous effort has been made to provide alternative proofs of *Parity* $\notin \mathrm{AC}^0$. However, this has been a rather fruitless endeavor, with the notable exception of the early works of Razborov [5] and Smolenski [6] that develop a less combinatorial approach with an algebraic flavor. For instance, Koucký et al. [7] foray into descriptive complexity and use model-theoretic tools to obtain *Parity* $\notin \mathrm{AC}^0$, but assert that "contrary to [their] original hope, [their] Ehrenfeucht-Fraïssé game arguments are not simpler than classical lower bounds." More recent promising approaches, especially the topological ones of [8], [9], have yet to yield strong lower bounds.

A different take originated from a conjecture of Lautemann and Thérien, investigated by Barrington et al. [10]: the Crane Beach Conjecture. They noticed that the letter 0 acts as a *neutral letter* in *Parity*, i.e., 0 can be added or removed from any word without affecting its membership to the language. If a circuit family recognizes a language with a neutral letter, it seems convincing that the circuits for two given input sizes should look very similar, that is: the circuit family must be highly *uniform*. It was thus conjectured that all neutral letter languages in AC^0 were regular, and this was disproved in [10].

This however sparked an interest in the study of neutral letter languages, in particular from the descriptive complexity view. Indeed, AC^0 circuits recognize precisely the languages expressible in $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$, where $\mathcal{A}_{\mathcal{RB}}$ denotes all possible *numerical predicates* (expressing numerical properties of the positions in a word, $\mathcal{A}_{\mathcal{RB}}$ standing for $\mathcal{A}_{\mathcal{RB}}$itrary). Further, as all regular neutral letter languages of $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$ are star-free [10], i.e., in $\mathrm{FO}[\leq]$, the Crane Beach Conjecture asked:

Are all neutral letter languages of $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$ *in* $\mathrm{FO}[\leq]$*?*

Note that this echoes the above intuition on uniformity, since the numerical predicates correspond precisely to the allowed power to compute the circuit for a given input length [11]. The intuition on the logic side is even more compelling: if a letter can be introduced anywhere without impacting membership, then the only meaningful relation that can relate positions is the linear order. However, first-order logic can "count" up to $\log n$ (see, e.g., [12]), meaning that even within a word with neutral letters, $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$ can assert some property on the number of nonneutral letters. This is, in essence, why nonregular neutral letter languages can be expressed in $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$.

In the recent years, a great deal of efforts was put into studying the Crane Beach Property in different logics, i.e., whether the definable neutral letter languages are regular. Krebs and Sreejith [13], building on the work of Roy and Straubing [14], show that all first-order logics with monoidal quantifiers and $+$ as the sole numerical predicate have the Crane Beach Property. Lautemann et al. [15] show Crane Beach Properties for classes of bounded-width branching programs, with an algebraic approach relying on communication complexity. Some expressiveness results were also derived from Crane Beach Properties, for instance Lee [16] shows that $\mathrm{FO}[+]$ is strictly included in $\mathrm{FO}[\leq, \times]$ by proving that only the former has the Crane Beach Property. Notably, all these logics are quite far from full $\mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$, and in that sense, fail to identify the part of the arbitrary numerical predicates that fit the intuition that they are rendered useless by the presence of a neutral letter.

In the present paper, we identify a large class of predicates, the *finite-degree* predicates, and a predicate MSB_0 such that *any numerical predicate* can be first-order defined using them and the order; in symbols, $\mathrm{FO}[\leq, \mathrm{MSB}_0, \mathcal{F}_{\mathcal{IN}}] = \mathrm{FO}[\mathcal{A}_{\mathcal{RB}}]$. We show that, strikingly, both $\mathrm{FO}[\leq, \mathrm{MSB}_0]$ and $\mathrm{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ have the Crane Beach Property, this latter statement being our main result. This constitutes what we informally call a "crevice:" on both sides (the MSB_0 and $\mathcal{F}_{\mathcal{IN}}$ sides), one stands

978-1-5090-3018-7/17/$31.00

firmly on the Crane Beach, but trying to stroll along both edges at the same time would send the rambler into the void. This result implies that showing that some nonregular neutral letter language is not expressible in FO[$\mathcal{A}_{\mathcal{RB}}$] could be done by showing that the single predicate MSB_0 may be removed from any FO[$\leq, \mathrm{MSB}_0, \mathcal{F}_{\mathcal{IN}}$] formula expressing it.

The proof for the Crane Beach Property of FO[$\leq, \mathcal{F}_{\mathcal{IN}}$] relies on a communication complexity argument different from that of [15]. It is also unrelated to the database collapse techniques of [10] (succinctly put, no logic with the Crane Beach Property has the so-called *independence property*, i.e., can encode arbitrary large sets). We will show that in fact FO[$\leq, \mathcal{F}_{\mathcal{IN}}$] *does* have the independence property. This provides, to the best of our knowledge, the first example of a logic that exhibits *both* the independence and the Crane Beach properties.

The aforementioned counting property of FO[$\mathcal{A}_{\mathcal{RB}}$] led to the conjecture [10], [16] that a logic has the Crane Beach Property *if and only if* it cannot count beyond a constant. To the best of our knowledge, neither of the directions is known; we show however that FO[$\leq, \mathcal{F}_{\mathcal{IN}}$] can only count up to a constant, by showing that it cannot even express very restricted forms of the addition. This adds evidence to the "if" direction of the conjecture.

Structure of the paper. In Section II, we introduce the required notions, although some familiarity with language theory and logic on words is assumed (see, e.g., [4]). In Section III, we show that FO[$\leq, \mathrm{MSB}_0, \mathcal{F}_{\mathcal{IN}}$] = FO[$\mathcal{A}_{\mathcal{RB}}$]. In Section IV, we present a simple proof, relying on a much harder result from [10], that FO[$\leq, \mathrm{MSB}_0$] has the Crane Beach Property. The failing of the aforementioned collapse technique for FO[$\leq, \mathcal{F}_{\mathcal{IN}}$] is shown in Section V. We tackle the Crane Beach Property of FO[$\leq, \mathcal{F}_{\mathcal{IN}}$], our main result, in Section VI, after the necessary tools have been developed. Finally, in Section VII, we focus on the counting inabilities of FO[$\leq, \mathcal{F}_{\mathcal{IN}}$].

Previous works. Finite-degree predicates were introduced by the second author in [17], in the context of two-variable logics. Therein, it is shown that the two-variable fragment of FO[$\leq, \mathcal{F}_{\mathcal{IN}}$] has the Crane Beach Property, and, even stronger, that the neutral letter languages expressible with k quantifier alternations can be expressed without the finite-degree predicates with the *same* amount of quantifier alternations. The techniques used in [17] are specific to two-variable logics, relying heavily on the fact that each quantification depends on a *single* previously quantified variable. We thus stress that the communication complexity argument developed in Section VI is unrelated to [17].

The fact that two sets of predicates can both verify the Crane Beach Property while their union does not has already been witnessed in [10]. Indeed, letting $\mathcal{M}_{\mathcal{ON}}$ be the set of *monoidal* numerical predicates, the Property holds for both FO[$\leq, +$] and FO[$\leq, \mathcal{M}_{\mathcal{ON}}$] but fails for FO[$\leq, +, \mathcal{M}_{\mathcal{ON}}$], although this latter class is less expressive than FO[$\mathcal{A}_{\mathcal{RB}}$] (this can be shown using the same proof as [7, Proposition 5]).

II. Preliminaries

A. Generalities

We write $\mathbb{N} = \{0, 1, 2, \ldots\}$ for the set of nonnegative numbers. For $n \in \mathbb{N}$, we let $[n] = \{0, 1, \ldots, n-1\}$. A function $f : \mathbb{N} \to \mathbb{N}$ is *nondecreasing* if $m > n$ implies $f(m) \geq f(n)$.

An alphabet A is a finite set of letters (symbols), and we write A^* for the set of finite words. For $u = u_0 u_1 \cdots u_{n-1}$, the length n of u is denoted $|u|$. We write ε for the empty word and $A^{\leq k}$ for words of length $\leq k$.

B. Logic on words

For an alphabet A, let σ_A be the vocabulary $\{\mathbf{a} \mid a \in A\}$ of unary *letter predicates*. A (finite) word $u = u_0 u_1 \cdots u_{n-1} \in A^*$ is naturally associated with the structure over σ_A with universe $[n]$ and with $\mathbf{a}$ interpreted as the set of positions i such that $u_i = a$, for any $a \in A$. A *numerical predicate* is a k-ary relation symbol together with an interpretation in $[n]^k$ for each possible universe size n. Given a formula φ that relies on some numerical predicates and a word u, we write $u \models \varphi$ to mean that φ is true of the σ_A-structure for u augmented with the interpretations of the numerical predicates for the universe of size $|u|$. A formula φ thus *defines* or *expresses* the language $\{u \in A^* \mid u \models \varphi\}$.

C. Classes of formulas

We let $\mathcal{A}_{\mathcal{RB}}$ be the set of all numerical predicates. Given a set $\mathcal{N} \subseteq \mathcal{A}_{\mathcal{RB}}$, we write FO[$\mathcal{N}$] for the set of first-order formulas built using the symbols from $\mathcal{N} \cup \sigma_A$, for any alphabet A. Similarly, MSO[$\mathcal{N}$] denotes monadic second-order formulas built with those symbols. We further define the quantifiers Maj and $\exists_i^{\equiv}$, for $i \in \mathbb{N}$, that will only be used in discussions:

- $u \models (\mathrm{Maj}\ x)[\varphi(x)]$ iff there is strict majority of positions $i \in [|u|]$ such that $\langle u, x := i \rangle \models \varphi$;
- $u \models (\exists_i^{\equiv})[\varphi(x)]$ iff the number of positions $i \in [|u|]$ verifying $\langle u, x := i \rangle \models \varphi$ is a multiple of i.

We will write MAJ[$\mathcal{N}$] and FO+MAJ[$\mathcal{N}$] with the obvious meanings. Further, FO+MOD[$\mathcal{N}$] allows all the quantifiers $\exists_i^{\equiv}$ in FO[$\mathcal{N}$] formulas.

D. On numerical predicates

The most ubiquitous numerical predicate here will be the binary order predicate $\leq$. The predicate that zeroes the most significant bit (MSB) of a number will also be important: $(m, n) \in \mathrm{MSB}_0$ iff $n = m - 2^{\lfloor \log m \rfloor}$. Note that both predicates do *not* depend on the universe size, and we single out this concept:

Definition 1. A k-ary numerical predicate P is *unvaried* if there is a set $E \subseteq \mathbb{N}^k$ such that the interpretation of P on universes of size n is $E \cap [n]^k$. In this case, we identify P with the set E. It is *varied* otherwise.[1] We write $\mathcal{A}_{\mathcal{RB}}{}^{\mathrm{u}}$ for the set of unvaried numerical predicates.

[1] The relevance of this concept has been noted in previous works (e.g., [10]), but was left unnamed. The second author used in [17] the terms *(non)uniform*, an unfortunate coinage in this context. We prefer here the less conflicting terms *(un)varied*.

Naturally, any varied predicate can be converted to an unvaried one by turning the universe length into an argument and quantifying the maximum position; this implies in particular that $\text{FO}[\mathcal{A}_{\mathcal{RB}}] = \text{FO}[\mathcal{A}_{\mathcal{RB}}{}^{\text{u}}]$. This is however not entirely innocuous, as will be discussed in Section VII.

We will rely on the following class of unvaried predicates, generalizing a definition of [17] (see also the older notion of "finite formula" [18]):

Definition 2. An unvaried predicate $P \subseteq \mathbb{N}^k$ is of *finite degree*[2] if for all $n \in \mathbb{N}$, n appears in a finite number of tuples in P. We write $\mathcal{F}_{\mathcal{IN}}$ for the class of such predicates.

Note that this does *not* imply that there is a N that bounds the number of appearances for all n's. Some examples:

- MSB_0 is *not* a finite-degree predicate, as, e.g., $(2^n, 0) \in \text{MSB}_0$ for any n, hence 0 appears infinitely often;
- Any *unvaried* monadic numerical predicate is of finite degree, this implies in particular that any language over a unary alphabet is expressed by a $\text{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ formula;
- The graph of any nondecreasing unbounded function $f \colon \mathbb{N} \to \mathbb{N}$ defines a finite-degree predicate, since $f^{-1}(n)$ is a finite set for all n;
- The order, sum, and multiplication are not of finite degree;
- One can usually "translate" unvaried predicates to make them finite degree; for instance, the predicate true of (x, y) if $y - x < x < y$ is of finite degree, see also the proof of Proposition 4.

E. Crane Beach Property

A language $L \subseteq A^*$ is said to have a neutral letter if there is a $e \in A$ such that adding or deleting e from a word does not change its membership to L. Following [15], we say that a logic has the Crane Beach Property if all the neutral letter languages it defines are regular. We further say that it has the *strong* Crane Beach Property if all the neutral letter languages it defines can be defined using order as the sole numerical predicate.

III. $\text{FO}[\mathcal{A}_{\mathcal{RB}}]$ AND $\text{FO}[\leq, \text{MSB}_0, \mathcal{F}_{\mathcal{IN}}]$ DEFINE THE SAME LANGUAGES

In this section, we express all the numerical predicates using only finite-degree ones, MSB_0, and the order. The result is a variant of [17, Theorem 3], where it is proven for the two-variable fragment, and on neutral letter languages.

Theorem 1. $\text{FO}[\mathcal{A}_{\mathcal{RB}}]$ *and* $\text{FO}[\leq, \text{MSB}_0, \mathcal{F}_{\mathcal{IN}}]$ *define the same languages.*

Proof. We show that any $\text{FO}[\mathcal{A}_{\mathcal{RB}}{}^{\text{u}}]$ language is definable in $\text{FO}[\leq, \text{MSB}_0, \mathcal{F}_{\mathcal{IN}}]$. The main idea is to divide the set of word positions in four contiguous zones and translate the variables to the second zone, called the *work zone*, before using modified, zone-aware variations of the predicates used.

For inputs of length $\ell = 2^n$, the set of positions $[\ell]$ is divided into four zones of equal size 2^{n-2}; if the input length is not a power of 2, then we apply the same split as the closest greater power of two, leaving the third and fourth zone possibly smaller than the first two.

[2]The name stems from the fact that the hypergraph defined by P, with edges of size k, is of finite degree.

As an example, suppose that the word length is $\ell = 11110$ (here and in the following, we write numbers in binary, with the MSB on the left). The four zones of $[\ell]$ will be:

1) $00000 \to 00111$; 2) $01000 \to 01111$;
3) $10000 \to 10111$; 4) $11000 \to 11101 = \ell - 1$.

This split has two salient properties: 1. Checking that a number in $[\ell]$ belongs to a given zone amounts to checking its two first bits when written using $\lceil \log \ell \rceil$ bits; 2. Any number in $[\ell]$ can be obtained from a unique number k in the work zone by replacing the two first bits 01 of k with some other bits (00, 10, and 11, for the first, third, and fourth zone, respectively); the work-zone equivalent of a number in $[\ell]$ is called the *slave* of that number, e.g., in our example above, 01101 is the slave of $00101, 01101, 10101$, and 11101.

More formally, we can define a formula $\mathsf{zone}^{(i)}(x)$, for each $1 \leq i \leq 4$, which is true iff x belongs to the i-th zone. For $i = 1$, the formula asserts that there are at least 2 powers of two greater than x; for $i = 2$, it asserts that there is exactly one power of two greater than x. In both cases, we only need the unvaried monadic predicate true on powers of two, and this is a finite-degree predicate. If x is neither in the first nor second zone, then it is in the third (resp. fourth) iff the only y such that $\text{MSB}_0(x, y)$ holds is in the first (resp. second) zone.

Using similar ideas, we can define a formula $\mathsf{slave}(x, y)$ that holds iff x is the slave of y. The formula first asserts that x is in the work zone. Next, if y is in the first zone, $\text{MSB}_0(x, y)$ should hold; if y is in the second zone, then $x = y$ should hold; if y is in the third zone then there should be a z verifying $\text{MSB}_0(y, z)$ and $\text{MSB}_0(x, z)$; and if y is in the fourth zone, then $\text{MSB}_0(y, x)$ should hold.

The strategy will then be to modify the numerical predicates so that they only take slave values, while being aware of the zones from which these values were enslaved, so as to be able to reverse the slaving.

Let $\varphi \in \text{FO}[\mathcal{A}_{\mathcal{RB}}{}^{\text{u}}]$. We sketch this strategy for binary numerical predicates. Suppose such a predicate P is used in φ. For $1 \leq i, j \leq 4$, define the numerical predicate $P^{(i,j)}$ that expects two work-zone positions $(x^{\text{s}}, y^{\text{s}})$, finds the values x and y in the i-th and j-th zone, respectively, for which x^{s} and y^{s} are the slaves, then checks whether (x, y) belongs to P. This is well defined, since the number of bits needed to write the input length is uniquely determined by x^{s} (or y^{s}), as it belongs to a work zone. For instance, continuing our example above, $(01001, 01100) \in P^{(1,3)}$ iff $(00001, 10100) \in P$. The following rewriting is then operated:

$$P(x, y) \rightsquigarrow$$
$$(\exists x^{\text{s}})(\exists y^{\text{s}}) \Big[\mathsf{slave}(x^{\text{s}}, x) \wedge \mathsf{slave}(y^{\text{s}}, y) \wedge \bigvee_{1 \leq i,j \leq 4} \big(\mathsf{zone}^{(i)}(x) \wedge \mathsf{zone}^{(j)}(y) \wedge P^{(i,j)}(x^{\text{s}}, y^{\text{s}})\big)\Big] \ .$$

Now note, crucially, that since the inputs to $P^{(i,j)}$ are work-zone positions, a pair of integers belongs to $P^{(i,j)}$ only if they share the same MSB: it is thus a finite-degree predicate. This rewriting preserves the semantic of the formula and only uses the formulas defined previously and finite-degree predicates, concluding the proof. $\square$

Remark. Theorem 1 can be shown to hold also for FO+MAJ$[\leq, \mathrm{MSB}_0, \mathcal{F_{IN}}]$, i.e., this logic is equiexpressive with FO+MAJ$[\mathcal{A_{RB}}]$. Indeed, the proof works as-is, since the number of positions verifying a formula is not changed by the rewriting. Likewise, FO+MOD$[\leq, \mathrm{MSB}_0, \mathcal{F_{IN}}]$ is equiexpressive with FO+MOD$[\mathcal{A_{RB}}]$.

IV. FO$[\leq, \mathrm{MSB}_0]$ has the Crane Beach Property

Following a short chain of rewriting, we will express MSB_0 using predicates that appear in [10] and conclude that:

Theorem 2. FO$[\leq, \mathrm{MSB}_0]$ *has the strong Crane Beach Property.*

Proof. Let $f\colon \mathbb{N} \to \mathbb{N}$ be defined by $f(n) = 2^{\left(\lfloor \log n \rfloor^2\right)}$, and let $F \subseteq \mathbb{N}^2$ be its graph. Barrington et al. [10, Corollary 4.14] show that FO$[\leq, +, F]$ has the strong Crane Beach Property; we show that MSB_0 can be expressed in that logic. First, the monadic predicate $Q = \{2^n \mid n \in \mathbb{N}\}$ is definable in FO$[\leq, F]$, since n is a power of two iff $f(n-1) \neq f(n)$. Second, given $n \in \mathbb{N}$, the greatest power of two smaller than n is $p = 2^{\lfloor \log n \rfloor}$, which is easy to find in FO$[\leq, Q]$. Finally, $\mathrm{MSB}_0(n, m)$ is true iff $m + p = n$, and is thus definable in FO$[\leq, +, F]$. $\square$

Remark. From Lange [19], MAJ$[\leq]$ and FO+MAJ$[\leq, +]$ are equiexpressive, and as MSB_0 is expressible using the unary predicate $\{2^n \mid n \in \mathbb{N}\}$ and the sum, this shows that MAJ$[\leq, \mathcal{F_{IN}}]$ is equiexpressive with FO+MAJ$[\mathcal{A_{RB}}]$. Hence MAJ$[\leq, \mathcal{F_{IN}}]$ does *not* have the strong Crane Beach Property.

V. FO$[\leq, \mathcal{F_{IN}}]$ has the Independence Property

In [10], an important tool is introduced to show Crane Beach Properties, relying on the notion of *collapse* in databases, see [20, Chapter 13] for a modern account. Specifically, let us define an ad-hoc version of the:

Definition 3 (Independence property (e.g., [21]))**.** Let $\mathcal{N}$ be a set of *unvaried* numerical predicates. Let $\vec{x}, \vec{y}$ be two vectors of first-order variables of size k and ℓ, respectively. A formula $\varphi(\vec{x}, \vec{y})$ of FO$[\mathcal{N}]$, over a single-letter alphabet, has the *independence property* if for all $n > 0$ there are vectors $\overrightarrow{a_0}, \overrightarrow{a_1}, \ldots, \overrightarrow{a_{n-1}}$, each of $\mathbb{N}^k$, for which for any $M \subseteq [n]$, there is a vector $\overrightarrow{b_M} \in \mathbb{N}^\ell$ such that:[3]

$$\langle \mathbb{N}, \vec{x} := \overrightarrow{a_i}, \vec{y} := \overrightarrow{b_M} \rangle \models \varphi \quad \text{iff} \quad i \in M \ .$$

The logic FO$[\mathcal{N}]$ has the *independence property* if it contains such a φ.

[3]Note that we evaluate a formula over an *infinite* domain; this is well defined in our case since we only use *unvaried* predicates and the letter predicates are irrelevant.

Intuitively, a logic has the independence property iff it can encode arbitrary sets. Barrington et al. [10], relying on a deep result of Baldwin and Benedikt [21], show that:

Theorem 3 ([10, Corollary 4.13])**.** *If a logic does not have the independence property, then it has the strong Crane Beach Property.*

We note that this powerful tool cannot show that the logic we consider exhibits the Crane Beach Property:

Proposition 1. FO$[\leq, \mathcal{F_{IN}}]$ *has the independence property.*

Proof. Let $n > 0$, and define $a_i = 2^n + 2^i$ for $0 \leq i < n$. Now for $M \subseteq [n]$, let $b_M = 2^n + \sum_{i \in M} 2^i$. It holds that $i \in M$ iff the binary AND of a_i and b_M is a_i. Consider this latter binary predicate; its behavior on two arguments that do not share the same MSB is irrelevant, and we can thus decide that such inputs are rejected. Thanks to this, we obtain a finite-degree predicate. Consequently, the formula that consists of this single predicate has the independence property. $\square$

VI. FO$[\leq, \mathcal{F_{IN}}]$ has the Crane Beach Property

A. Communication complexity

We will show the Crane Beach Property of FO$[\leq, \mathcal{F_{IN}}]$ by a communication complexity argument. This approach is mostly unrelated to the use of communication complexity of [15], [22]; in particular, we are concerned with two-party protocols with a split of the input in two contiguous parts, as opposed to worst-case partitioning of the input among multiple players. We rely on a characterization of [23] of the class of languages expressible in *monadic second-order* with varied monadic numerical predicates. Writing this class MSO$[\leq, \mathcal{M_{ON}}]$, they state in particular the following:

Proposition 2 ([23, Theorem 2.2])**.** *Let $L \subseteq A^*$ and define, for all $p \in \mathbb{N}$, the equivalence relation $\sim_p$ over A^* as: $u \sim_p v$ iff for all $w \in A^p$, $u \cdot w \in L \Leftrightarrow v \cdot w \in L$. If there is a $N \in \mathbb{N}$ such that for all $p \in \mathbb{N}$, $\sim_p$ has at most N equivalence classes, then $L \in$* MSO$[\leq, \mathcal{M_{ON}}]$.

Lemma 1. *Let $L \subseteq A^*$. Suppose there are functions $f_{\mathrm{Alice}}\colon A^* \times \mathbb{N} \times \{0,1\}^* \to \{0,1\}$ and $f_{\mathrm{Bob}}\colon A^* \times \mathbb{N} \times \{0,1\}^*$ and a constant $K \in \mathbb{N}$ such that for any $u, v \in A^*$, the sequence, for $1 \leq i \leq K$:*

- $a_i = f_{\mathrm{Alice}}\,(u, \quad |u \cdot v|, \quad b_1 b_2 \cdots b_{i-1})$
- $b_i = f_{\mathrm{Bob}}\ \ (v, \quad |u \cdot v|, \quad a_1 a_2 \cdots a_i)$;

is such that $b_K = 1$ iff $u \cdot v \in L$. Then $L \in$ MSO$[\leq, \mathcal{M_{ON}}]$.

Proof. We adapt the (folklore) proof that L is regular iff such functions exist where f_{Alice} and f_{Bob} do not use their second parameter.

Let $p \in \mathbb{N}$. For any $u \in A^*$, let $c(u)$ be the set of pairs $(a_1 a_2 \cdots a_K, b_1 b_2 \cdots b_{K-1})$ such that for all $1 \leq i \leq K$, it holds that $a_i = f_{\mathrm{Alice}}(u, |u| + p, b_1 b_2 \cdots b_{i-1})$. Define the equivalence relation $\equiv$ by letting $u \equiv v$ iff $c(u) = c(v)$; it clearly has a finite number $N = N(K)$ of equivalence classes. Moreover, if $u \equiv v$ and $w \in A^p$, then (u, w) and (v, w) define the same sequences of a_i's and b_i's, by a simple induction. Hence $u \cdot w \in L \Leftrightarrow v \cdot w \in L$. This shows that $\equiv$ refines $\sim_p$, implying, by Proposition 2, that $L \in \text{MSO}[\leq, \mathcal{M}_{\mathcal{ON}}]$. $\square$

We shall adopt the classical communication complexity view here, and consider f_{Alice} and f_{Bob} as two players, Alice and Bob, that alternate exchanging a bounded number of bits in order to decide if the concatenation of their respective inputs is in L. To show that L is in $\text{MSO}[\leq, \mathcal{M}_{\mathcal{ON}}]$, the protocol between Alice and Bob should end in a constant number of rounds. We will then rely on the fact that:

Theorem 4 ([23, Theorem 4.6])**.** $\text{MSO}[\leq, \mathcal{M}_{\mathcal{ON}}]$ *has the Crane Beach Property.*

B. A toy example: $\text{FO}[<] \subseteq \text{MSO}[\leq, \mathcal{M}_{\mathcal{ON}}]$

We will demonstrate how the communication complexity approach will be used with a toy example. Doing so, the requirements for this protocol to work will be emphasized, and they will be enforced when showing the Crane Beach Property of $\text{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ in Section VI-C.

Let us consider the following formula over $A = \{a, b, c\}$:

$$\varphi \equiv (\exists x)(\forall y)[\psi], \quad \text{with } \psi \equiv \mathbf{a}(x) \wedge (x < y \rightarrow \mathbf{b}(y)) \ ,$$

depicted as a tree in Figure 1.a. The formula φ asserts that all the letters after the last a are b's. In this example, Alice will receive $u = aa$, and Bob $v = bb$. Naturally, φ over words of length 4 is equivalent to the formula where $\exists x$ is replaced by $\bigvee_{x=0}^{3}$, and $\forall y$ is replaced by $\bigwedge_{y=0}^{3}$; our approach will be to split this rewriting between Alice and Bob.

Consider the variable x. To check the validity of the formula over a $u \cdot v$, the variable should range over the positions of both players. In other words, the formula is true if there is a position x of Alice verifying $(\forall y)[\psi]$ *or* a position x of Bob verifying it—likewise for the universal quantifier. We thus "split" the quantifiers by enforcing the domain to be either Alice's $(\forall^{\mathsf{A}}, \exists^{\mathsf{A}})$ or Bob's $(\forall^{\mathsf{B}}, \exists^{\mathsf{B}})$, obtaining Figure 1.b.

Alice will now expand her quantifiers to range over her word; she will thus replace, e.g., $(\forall^{\mathsf{A}} y)[\psi]$ by $\bigwedge_{y=0}^{1} \psi$. Crucially, at the leaves of the formula, it is known which variables were quantified by each player, and if they are Alice's, their values. Consider for instance a leaf where Alice substituted y with a numerical value. The letter predicate $\mathbf{b}(y)$ can thus be replaced by its truth value. More importantly, the predicate $x < y$ can *also* be evaluated: Either Alice quantified x, and it has a numerical value, or she did not, and we know for sure that $x < y$ does not hold, since x will be quantified by Bob. Applied to our example, we obtain the tree of Figure 1.c.

The resulting formulas at the leaves are thus *free from the variables quantified by Alice.* Moreover, for each internal node of the tree, its children represent subformulas of bounded quantifier depth, and there are thus a finite number of possible nonequivalent subformulas. Once only one subformula per equivalence class is kept, the resulting tree is of bounded depth and each node has a bounded number of children. Hence the size of this tree is bounded by a value that *only depends on* φ. Alice can thus communicate this tree to Bob. In our example, simplifying the tree, we obtain the formula:

$$(\forall^{\mathsf{B}} y)[\mathbf{b}(y)] \vee (\exists^{\mathsf{B}} x)\Big[\mathbf{a}(x) \wedge (\forall^{\mathsf{B}} y)[\psi]\Big] \ .$$

Finally, Bob can actually quantify his variables, resulting in a formula with no quantified variable, that he can evaluate, concluding the protocol.

Takeaway. This protocol relies on the fact that predicates that involve variables from both Alice and Bob can be evaluated by Alice alone. This enables Alice to remove "her" variables before sending the partially evaluated tree to Bob, who can quantify the remainder of the variables.

C. The case of $\text{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$

Theorem 5. $\text{FO}[\leq, \mathcal{F}_{\mathcal{IN}}]$ *has the strong Crane Beach Property.*

Proof. Let φ be a formula over an alphabet A in $\text{FO}[\leq, \mathcal{N}]$, for some finite subset $\mathcal{N}$ of $\mathcal{F}_{\mathcal{IN}}$, and suppose φ expresses a language L that admits a neutral letter e. We show that $L \in \text{MSO}[\leq, \mathcal{M}_{\mathcal{ON}}]$ using Lemma 1. This concludes the proof since by Theorem 4, L is a neutral letter regular language in $\text{FO}[\mathcal{A}_{\mathcal{RB}}]$, and it thus belongs to $\text{FO}[\leq]$ (see [10]; this is essentially a consequence of *Parity* $\notin \text{AC}^0$).

Let us write $u \in A^*$ for Alice's word, and v for Bob's. Both players will compute a value $N > 0$ that depends solely on φ and $|u \cdot v|$, and the protocol will then decide whether $u \cdot e^N \cdot v \in L$, which is equivalent to $u \cdot v \in L$ by hypothesis. We suppose that a large enough N has been picked for the protocol to work, and delay to the end of the proof its computation.

We will henceforth suppose that φ is given in prenex normal form and that all variables are quantified only once:

$$\varphi \equiv (Q_1 x_1)(Q_2 x_2) \cdots (Q_k x_k)[\psi] \ ,$$

with ψ quantifier-free and $Q_i \in \{\forall, \exists\}$. We again see formulas as trees with leaves containing quantifier-free formulas.

Rather than splitting the domain $[|u \cdot e^N \cdot v|]$ at a precise position, and tasking Alice to quantify over the first half and Bob over the second half, we will rely on a third group, that is "far enough" from both Alice's and Bob's words. The core of this proof is to formalize this notion. Let us first introduce the tools that will enable this formalization: one set of definitions, and two facts that will be used later on.

Definition 4. Let C be the set of pairs of integers (p_1, p_2) that appear in a same tuple of a relation in $\mathcal{N}$. Define the *link graph* $G = (\mathbb{N}, E)$ as the *undirected* graph specified by $(p_1, p_2) \in E$ iff $p_1 = p_2$ or there are integers $p_1' \leq \{p_1, p_2\} \leq p_2'$ such that $(p_1', p_2') \in C$. For $p \in \mathbb{N}$, $\mathcal{L}(p)$ (resp. $\mathcal{R}(p)$) is the greatest $q < p$ (resp. smallest $q > p$) which is not a neighbor of p in G. Equivalently, $\mathcal{L}(p)$ is the smallest neighbor of p minus 1, and $\mathcal{R}(p)$ is the greatest neighbor of p plus 1.

Note that $\mathcal{L}$ and $\mathcal{R}$ are well defined since each vertex of G has a finite number of neighbors. This directly implies the following:

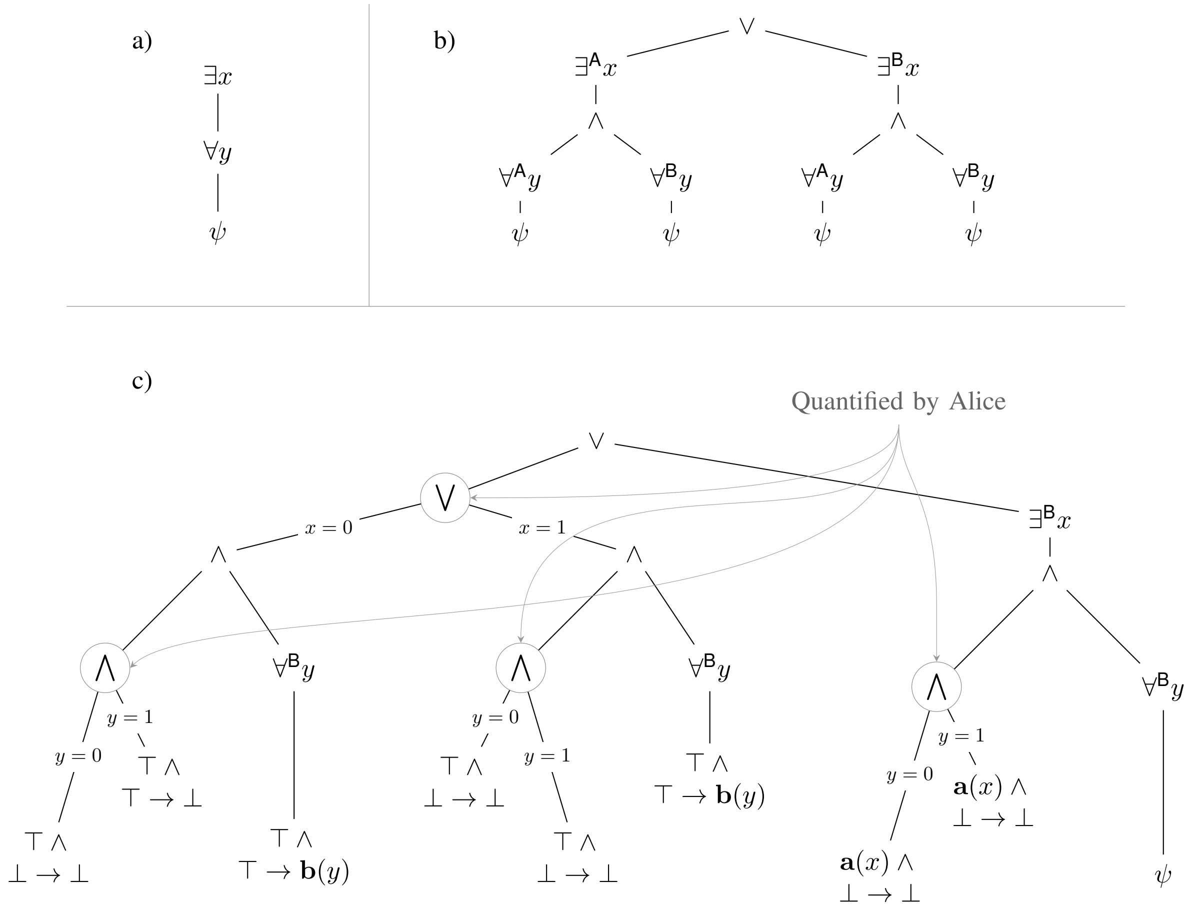

Fig. 1: The formula φ as it gets evaluated by Alice and Bob.

Fact 1. *The functions $\mathcal{L}$ and $\mathcal{R}$ are nondecreasing and unbounded. Moreover, for any $p \in \mathbb{N}$, $\mathcal{L}(p) < p < \mathcal{R}(p)$.*

Writing $\mathcal{R}^n$ for the function $\mathcal{R}$ composed n times with itself, and similarly for $\mathcal{L}$, we have:

Fact 2. *For any position p and $n > m \geq 0$:*

- $\mathcal{L}^m(\mathcal{R}^n(p)) \geq \mathcal{R}(p)$;
- $\mathcal{R}^m(\mathcal{L}^n(p)) \leq \mathcal{L}(p)$.

Proof. This is easily shown by induction; we prove the first item, the second being similar. For $n = 1$, this is clear. Let $n > 1$. If $m = 0$, this is immediate from Fact 1, let thus $m > 0$. We have that:

$$\mathcal{L}^m(\mathcal{R}^n(p)) = \mathcal{L}(\mathcal{L}^{m-1}(\mathcal{R}^{n-1}(p'))) \ ,$$

with $p' = \mathcal{R}(p)$. By induction hypothesis and the fact that $\mathcal{L}$ is nondecreasing, it holds that:

$$\mathcal{L}^m(\mathcal{R}^n(p)) \geq \mathcal{L}(\mathcal{R}(p')) = q \ .$$

Let $p'' = \mathcal{R}(p')$. By definition of $\mathcal{L}$, $(q+1, p'')$ is an edge in G. Now by definition of G, if $q < p'$, then (p', p'') should also be an edge in G, which contradicts the definition of p''. Hence $q \geq p'$, showing the property. $\square$ *of Fact 2.*

Let us now suppose we have two large positions $|u| \ll \ell_0 \ll r_0 \ll |v|$, the requirements on which will be made clear shortly. Let us deem a position p to be *Alicic* if $p \leq \ell_0$, *Bobic* if $p \geq r_0$, and *Neutral* otherwise; we call this the *type* of the position. We wish to ensure that two positions of two different types cannot be linked in G, so that they cannot appear in a tuple of a predicate in $\mathcal{N}$. This surely is not the case if the typing of positions does not reflect previously typed positions, e.g., $\ell_0 - 1$ is Alicic, but ℓ_0 is Neutral, and their distance may not be large enough to ensure that they do not form an edge in G. Thus the boundaries of the zones, ℓ_0 and r_0, will be moving with each new typing. Formally, let $T = \{\text{Alice}, \text{Neutral}, \text{Bob}\}$ be an alphabet. Recall that k is the number of variables in φ, and define the function $\mathsf{bounds}\colon T^{\leq k} \to [|u \cdot e^N \cdot v|]^2$ by:

$$\mathsf{bounds}(\varepsilon) = (l_0, r_0)$$

$$\mathsf{bounds}(t_1 t_2 \cdots t_i) = \begin{cases} (\mathcal{R}^n(\ell), r) & \text{if } t_i = \text{Alice} \\ (\mathcal{L}^n(\ell), \mathcal{R}^n(r)) & \text{if } t_i = \text{Neutral} \\ (\ell, \mathcal{L}^n(r)) & \text{if } t_i = \text{Bob} \end{cases}$$

with $(\ell, r) = \mathsf{bounds}(t_1 t_2 \cdots t_{i-1})$ and $n = 2^{k-i}$.

Assumption. *We henceforth assume that if* $(\ell, r) = \mathsf{bounds}(h)$ *for some word* $h \in T^{\leq k}$*, then* $|u| < \ell < r < |u| + N$*. This will have to be guaranteed by carefully picking* N*,* ℓ_0 *and* r_0*.*

The type of a position p *under type history* $t_1 t_2 \cdots t_i \in T^*$ is computed by first taking $(\ell, r) = \mathsf{bounds}(t_1 t_2 \cdots t_i)$, and reasoning as before: it is Alicic if $p \leq \ell$, Bobic if $p \geq r$, and Neutral otherwise. This is well defined since $\ell < r$ by our Assumption. The crucial property here is as follows:

Fact 3. *Let* $p_1, p_2, \ldots, p_k$ *be positions, and inductively define the type* t_i *of* p_i *as its type under type history* $t_1 t_2 \cdots t_{i-1}$*.*

1) *Two positions with different types do not form an edge in* G*;*
2) *All Alicic positions are strictly smaller than the Neutral ones, which are strictly smaller than the Bobic ones;*
3) *All Neutral positions are labeled with the neutral letter.*

Proof. (Points 1 and 2.) Suppose p_i is Alicic and p_j is Neutral, with $i < j$. Let $(\ell, r) = \mathsf{bounds}(t_1 t_2 \cdots t_{i-1})$, we thus have that p_i is maximally ℓ. Let $(\ell', r') = \mathsf{bounds}(t_1 t_2 \cdots t_{j-1})$, then p_j is minimally $\ell'+1$. By definition, once the types of $p_1, p_2, \ldots, p_i$ are fixed, the smallest ℓ' that can be obtained with the types $t_{>i}$ is by having all positions p_t, with $i < t < j$, Neutral. In that case, an easy computation shows that ℓ' would be:

$$\mathcal{L}^{2^{k-(j-1)}}(\mathcal{L}^{2^{k-(j-2)}}(\cdots(\mathcal{L}^{2^{k-(i+1)}}(\mathcal{R}^{2^{k-i}}(\ell)))\cdots)) \ .$$

That is, $\mathcal{L}$ is composed with itself m times with:

$$m = 2^{k-(i+1)} + \cdots + 2^{k-(j-1)} < \sum_{s=i+1}^{k} 2^{k-s} < 2^{k-i} = n \ .$$

Hence ℓ' is at most $\mathcal{L}^m(\mathcal{R}^n(\ell))$ with $m < n$, and by Fact 2, $\ell' \geq \mathcal{R}(\ell)$. Hence (p_i, p_j) is not an edge in G, and $p_i < p_j$.

The other cases are similar. For instance, if p_i is Neutral and p_j Bobic, with $i < j$, then, with the same notation as above, ℓ' can be at most $\mathcal{L}^m(\mathcal{R}^n(\ell))$, and by Fact 2, $\ell' \geq \mathcal{L}(\ell)$.

(Point 3.) This is a direct consequence of the Assumption. Consider $(\ell, r) = \mathsf{bounds}(\text{Neutral}^k)$; this provides the minimal ℓ and maximal r between which a position can be labeled Neutral. By the Assumption, $|u| < \ell < r < |u| + N$, hence a Neutral position has a neutral letter. ☐ *of Fact 3.*

We are now ready to present the protocol. First, we rewrite quantifiers using Alicic/Neutral/Bobic *annotated* quantifiers:

- $(\forall x)[\rho] \rightsquigarrow (\forall^{\mathsf{A}} x)[\rho] \ \wedge \ (\forall^{\mathsf{N}} x)[\rho] \ \wedge \ (\forall^{\mathsf{B}} x)[\rho]$,
- $(\exists x)[\rho] \rightsquigarrow (\exists^{\mathsf{A}} x)[\rho] \ \vee \ (\exists^{\mathsf{N}} x)[\rho] \ \vee \ (\exists^{\mathsf{B}} x)[\rho]$.

Let us further equip each node with the type history of the variables quantified before it; that is, each node holds a string $t_1 t_2 \cdots t_n \in T^{\leq k}$ where t_i is the annotation of the i-th quantifier from the root to the node, excluding the node itself.

Now if we were given the entire word $u \cdot e^N \cdot v$, a way to evaluate the formula that respects the semantics of "Alicic", "Neutral", and "Bobic" is as follows:

Algorithm 1 Formula Evaluation

foreach quantifier node $\forall^{\mathsf{A}} x$ or $\exists^{\mathsf{A}} x$ **do**
  $(\ell, r) := \mathsf{bounds}$(type history at node)
  if node is $\forall^{\mathsf{A}} x$ **then**
    Replace node with $\bigwedge_{x=0}^{\ell}$
  Similarly with $\exists$ becoming $\bigvee$
end
Evaluate the part of the leaves that can be evaluated
foreach quantifier node **do**
  $(\ell, r) := \mathsf{bounds}$(type history at node)
  if node is $\forall^{\mathsf{N}} x$ **then**
    Replace node with $\bigwedge_{x=\ell+1}^{r-1}$
  else if node is $\forall^{\mathsf{B}} x$ **then**
    Replace node with $\bigwedge_{x=r}^{|ue^N v|}$
  Similarly with $\exists$ becoming $\bigvee$
end
Finish evaluating the tree

Alice and Bob follow this algorithm. First, Alice quantifies her variables according to the bounds of the type history of each node, as in Algorithm 1. At the leaves, she thus obtains the formula ψ, and has a set of quantified Alicic variables. She can then evaluate ψ *partially*: if an atomic formula only relies on Alicic variables, she can compute its value. If an atomic formula uses a mix of Alicic and non-Alicic variables, then she can *also* evaluate it: if the formula is a numerical predicate, then by Fact 3.1, it is valued *false*; if the formula is of the form $x < y$, then it is true iff x is Alicic, by Fact 3.2. Alice now simplifies her tree: logically equivalent leaves with the same parent are merged, and inductively, each internal node keeps only a single occurrence per formula appearing as a child. Remark that the semantic of the tree is preserved. This results in a tree whose size depends *solely* on φ, and the values of N, ℓ_0, and r_0, and Alice can thus send it to Bob.

Bob will now expand the remaining quantifiers (Neutral and Bobic), respecting the bounds of the type history, as in Algorithm 1. He can then evaluate all the leaves, since, by Fact 3.3, the only letter predicate true of a Neutral position is that of the neutral letter. This concludes the protocol, which clearly produces the same result as Algorithm 1.

What are N*,* ℓ_0*,* r_0*?* We check that Alice and Bob can agree on these values without communication. The requirements were made explicit in our Assumption. The values computed by the function bounds are obtained by applying $\mathcal{L}$ and $\mathcal{R}$ on ℓ_0 and r_0 *at most* $n = \sum_{i=0}^{k-1} 2^i$ times. From Fact 1, it is clear that any $(\ell, r) = \mathsf{bounds}(h)$, for $h \in T^{\leq k}$, verifies:

- $\ell_{\min} = \mathcal{L}^n(\ell_0) \leq \ell \leq \mathcal{R}^n(\ell_0) = \ell_{\max}$;
- $r_{\min} = \mathcal{L}^n(r_0) \leq r \leq \mathcal{R}^n(r_0) = r_{\max}$.

Hence we pick $\ell_0 = \mathcal{R}^{n+1}(|u|)$, ensuring, by Fact 2, that $\ell_{\min} > |u|$. Next, we pick r_0 to be $\mathcal{R}^{n+1}(\ell_{\max})$, ensuring that $r_{\min} > \ell_{\max}$ by the same Fact 2. Finally, we pick $N = \mathcal{R}^{n+1}(r_0)$, ensuring, by Fact 1, that $N > r_{\max}$, so that in particular, $r_{\max} < |u| + N$. We then indeed obtain that $|u| < \ell < r < |u| + N$, as required. Note that these computations depend *solely* on φ and the lengths of u and v. ☐ *of Theorem 5.*

Remark. It should be noted that the crux of this proof is that a relation $R(x, y)$ with x Alicic and y Neutral or Bobic can be readily evaluated by Alice. If R were monadic, then it could not mix two positions of different types, hence Alice could still remove all of her variables at the end of her evaluation. The rest of the protocol will be similar, with Bob quantifying the remaining positions. This shows that FO$[\leq, \mathcal{M}_{\mathcal{ON}}, \mathcal{F}_{\mathcal{IN}}]$ also has the Crane Beach Property.

VII. On counting

A compelling notion of computational power, for a logic, is the extent to which it is able to precisely evaluate the *number of positions* that verify a formula. This is formalized with the following standard definition:

Definition 5. For a nondecreasing function $f(n) \leq n$, a logic is said to *count up to* $f(n)$ if there is a formula $\varphi(c)$ in this logic such that for all n and $w \in \{0,1\}^n$:

$$w \models \varphi(c) \quad \Leftrightarrow \quad c \leq f(n) \wedge c = \text{number of 1's in } w \ .$$

It is known from [10] that if a logic can count up to $\log(\log(\cdots(\log n)\cdots))$, for some number of iterations of $\log$, then the logic does not have the Crane Beach Property. It has also been conjectured [10], [16] that a logic has the Crane Beach Property iff it cannot count beyond a constant. It is not known whether there exists a set of predicates $\mathcal{N}$ such that FO$[\mathcal{N}]$ can count beyond a constant but not up to $\log n$.

We define a much weaker ability:

Definition 6. For a nondecreasing function $f(n) \leq n$, a logic is said to *sum through* $f(n)$ if there is a formula $\varphi(a, b, c)$ in this logic such that for all n and $w \in \{0,1\}^n$:

$$w \models \varphi(a, b, c) \quad \Leftrightarrow \quad a = b + f(c) \ .$$

This is in general even weaker than being able to sum "up to" $f(n)$, that is, having a formula expressing that $a = b + c$ and $c \leq f(n)$. Naturally, counting and summing are related:

Proposition 3. *Let $\mathcal{N}$ be a set of unvaried numerical predicates. If* FO$[\leq, \mathcal{N}]$ *can count up to $f(n)$, it can sum through $f(n)$.*

Proof. Letting $\varphi(c)$ be the formula that counts up to $f(n)$, we modify it into $\varphi'(a, b, c)$ by changing the letter predicates to consider that there is a 1 in position p iff $b \leq p < a$. This expresses that $a = b + c$ provided that $c \leq f(n)$.

Next, the graph F of f is obtained as follows. First, modify $\varphi(c)$ into $\varphi'(c, c')$, by restricting all quantifications to c and replacing the letter predicates to have 1's in all positions below c'. Second, $(c, c') \in F$ iff c' is maximal among those that verify $\varphi'(c, c')$. This relies on the fact that $\mathcal{N}$ consists solely of unvaried predicates.

The logic can then sum through $f(n)$ by:

$$\psi(a, b, c) \equiv (\exists c')[F(c, c') \wedge a = b + c'] \ . \qquad \square$$

Remark. Proposition 3 depends crucially on the fact that the predicates are unvaried to show that the graph of the summing function is expressible. Writing $\mathcal{S}$ for the set of varied monadic predicates $S = (S_n)_{n \geq 0}$ with $|S_n| = 1$ for all n, it is easily shown that FO$[\leq, +, \times, \mathcal{S}]$ can count up to any function $\leq \log n$. However, we conjecture that there are functions whose graphs are not expressible in this logic.

Proposition 4. FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ *cannot sum through beyond a constant.*

Proof. Suppose for a contradiction that FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ can sum through a nondecreasing unbounded function f using a formula $\varphi(a, b, c)$. Let Bit be the binary predicate true of (x, y) if the y-th bit of x is 1. We define a translated version as:

$$\text{Bit}' = \{(x, y) \mid (x, y - f(x)) \in \text{Bit}\} \ .$$

We show that Bit' is of finite degree. Let $n \in \mathbb{N}$, and suppose $(n, y) \in \text{Bit}'$. This implies in particular that $0 < y - f(n) < \log n$, hence n appears a finite number of time as (n, y) in Bit'. Suppose $(x, n) \in \text{Bit}'$, then $n - f(x) > 0$, but for x large enough, $f(x) > n$, hence there can only be a finite number of pairs (x, n) in Bit'.

Now Bit can be defined in FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ using φ, since $\text{Bit}(x, y)$ holds iff $(\exists z)[\varphi(z, y, x) \wedge \text{Bit}'(x, z)]$. However, FO$[\leq, \text{Bit}]$ does not have the Crane Beach Property [10], hence this contradicts Theorem 5, concluding the proof. $\square$

Remark. Of course, the inability to sum through beyond a constant cannot characterize the Crane Beach Property since FO$[\leq, +]$ has the Crane Beach Property.

Corollary 1. FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ *cannot count beyond a constant.*

VIII. Conclusion

We showed that FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ is one simple predicate away from expressing all of FO$[\mathcal{A}_{\mathcal{RB}}]$, and that it exhibits the Crane Beach Property. This logic is thus really on the brink of a crevice on the Crane Beach, and exemplifies a diverse set of behaviors that fit the intuition that neutral letters should render numerical predicates essentially useless. We emphasize some future research directions:

- As a consequence of our results, one can show that a nonregular neutral letter language L is not in AC^0 as follows. Assume $L \in \text{AC}^0$ for a contradiction, and let $\varphi \in$ FO$[\leq, \text{MSB}_0, \mathcal{F}_{\mathcal{IN}}]$ be a formula expressing it. Suppose that one can show that φ can be rewritten *without* the predicate MSB_0, then $L \in$ FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$, and thus L is regular, a contradiction. We hope to be able to apply this strategy in the future.
- As noted in [14] and [10] and studied in particular in [13], the interest in circuit complexity calls for the study of logics with more sophisticated quantifiers, notably modular quantifiers and, more generally, monoidal quantifiers. Hence the natural question here is whether FO+MOD$[\leq, \mathcal{F}_{\mathcal{IN}}]$ has the Crane Beach Property.
- As asked in [10], can we dispense from our implicit reliance on the lower bound *Parity* $\notin \text{AC}^0$? In the cases of [10], and as noted by the authors, this would be very difficult, as their results *imply* the lower bound. Here, the strong Crane Beach Property for FO$[\leq, \mathcal{F}_{\mathcal{IN}}]$ does not directly imply the lower bound. To show that

Parity $\notin$ AC^0, one could *additionally* prove that all the regular, neutral letter languages of FO$[\leq, \mathrm{MSB}_0, \mathcal{F_{IN}}]$ are in FO$[\leq, \mathcal{F_{IN}}]$—we know that this statement holds, but only thanks to *Parity* $\notin$ AC^0.

- Are we really on the brink of falling off the Crane Beach? On one side of the crevice, are there unvaried predicates that cannot be expressed in FO$[\leq, \mathcal{F_{IN}}]$ but can still be added to the logic while preserving the Crane Beach Property? We noted that all *varied* monadic predicates can be added safely, but already very simple predicates falsify the Crane Beach Property. For instance, with P the graph of the 2-adic valuation, FO$[\leq, P]$ is as expressive as FO$[\leq, +, \times]$ (see [24, Theorem 3]), which does not have the Crane Beach Property [10].
  On the other side of the crevice, FO$[\leq, \mathrm{MSB}_0]$ can be replaced by the (incomparable) logic FO$[\leq, +]$ while preserving the Crane Beach Properties and the equiexpressivity of FO$[\leq, +, \mathcal{F_{IN}}]$ and FO$[\mathcal{A_{RB}}]$. Encompassing both FO$[\leq, \mathrm{MSB}_0]$ and FO$[\leq, +]$, the logic FO$[\leq, +, F]$, with F as in the proof of Theorem 2, also verifies all of our desired properties. Can this be further extended?
- Numerical predicates correspond in a precise sense [11] to the computing power allowed to construct circuit families for a language. Is there a natural way to present FO$[\leq, \mathcal{F_{IN}}]$-uniform circuits?

Acknowledgment

The authors would like to thank Thomas Colcombet, Arnaud Durand, Andreas Krebs, and Pierre McKenzie for enlightening discussions, and Michael Blondin and Luc Dartois for their help in proofreading. This work was partly funded by the DFG Emmy Noether program (KR 4042/2).

References

[1] M. Ajtai, "Σ_1^1 formulae on finite structures," *Annals of Pure and Applied Logic*, vol. 24, pp. 1–48, 1983.

[2] M. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Theory of Computing Systems*, vol. 17, pp. 13–27, 1984.

[3] D. A. M. Barrington, K. Compton, H. Straubing, and D. Thérien, "Regular languages in NC^1," *J. Computer and System Sciences*, vol. 44, no. 3, pp. 478–499, 1992.

[4] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*. Boston: Birkhäuser, 1994.

[5] A. A. Razborov, "Lower bounds on the size of bounded depth networks over a complete basis with logical addition," *Matematicheskie Zametki*, vol. 41, pp. 598–607, 1987, in Russian. English translation in *Mathematical Notes of the Academy of Sciences of the USSR* 41:333–338, 1987.

[6] R. Smolensky, "Algebraic methods in the theory of lower bounds for Boolean circuit complexity," in *Proceedings 19th Symposium on Theory of Computing*. ACM Press, 1987, pp. 77–82.

[7] M. Koucký, S. Poloczek, C. Lautemann, and D. Therien, "Circuit lower bounds via ehrenfeucht-fraisse games," in *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, 2006, pp. 12 pp.–201.

[8] M. Gehrke, S. Grigorieff, and J.-É. Pin, "A topological approach to recognition," in *ICALP 2010, Proceedings, Part II*, S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, and P. G. Spirakis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 151–162.

[9] S. Czarnetzki and A. Krebs, "Using duality in circuit complexity," in *LATA 2016, Proceedings*, A.-H. Dediu, J. Janoušek, C. Martín-Vide, and B. Truthe, Eds. Cham: Springer International Publishing, 2016, pp. 283–294.

[10] D. A. M. Barrington, N. Immerman, C. Lautemann, N. Schweikardt, and D. Thérien, "First-order expressibility of languages with neutral letters or: The Crane Beach Conjecture," *J. Computer and System Sciences*, vol. 70, pp. 101–127, 2005.

[11] C. Behle and K.-J. Lange, "FO[<]-Uniformity," in *Proc. 21st Annual IEEE Conference on Computational Complexity (CCC'06)*, 2006, pp. 183 – 189.

[12] A. Durand, C. Lautemann, and M. More, "Counting results in weak formalisms," in *Circuits, Logic, and Games*, ser. Dagstuhl Seminar Proceedings, T. Schwentick, D. Thérien, and H. Vollmer, Eds., no. 06451. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007.

[13] A. Krebs and A. V. Sreejith, "Non-definability of Languages by Generalized First-order Formulas over (N,+)," in *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, ser. LICS '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 451–460.

[14] A. Roy and H. Straubing, "Definability of languages by generalized first-order formulas over; $(\mathbb{N}, +)$," *SIAM J. Comput.*, vol. 37, no. 2, pp. 502–521, 2007.

[15] C. Lautemann, P. Tesson, and D. Thérien, "An Algebraic Point of View on the Crane Beach Property," in *Computer Science Logic*, Z. Ésik, Ed., vol. 4207, 2006, pp. 426–440.

[16] T. Lee, "Arithmetical definability over finite structures," *Mathematical Logic Quarterly*, vol. 49, no. 4, pp. 385–392, 2003.

[17] C. Paperman, "Finite-degree predicates and two-variable first-order logic," in *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, 2015, pp. 616–630.

[18] J. F. Knight, A. Pillay, and C. Steinhorn, "Definable sets in ordered structures. ii," *Transactions of the American Mathematical Society*, vol. 295, no. 2, pp. 593–605, 1986.

[19] K.-J. Lange, "Some results on majority quantifiers over words," in *CCC '04: Proceedings of the 19th IEEE Annual Conference on Computational Complexity*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 123–129.

[20] L. Libkin, *Elements of Finite Model Theory*. Springer, 2004.

[21] J. T. Baldwin and M. A. Benedikt, "Embedded finite models, stability theory, and the impact of order," in *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science*, Jun 1998, pp. 490–500.

[22] A. Chattopadhyay, A. Krebs, M. Kouckỳ, M. Szegedy, P. Tesson, and D. Thérien, "Languages with bounded multiparty communication complexity," in *Annual Symposium on Theoretical Aspects of Computer Science*. Springer, 2007, pp. 500–511.

[23] N. Fijalkow and C. Paperman, "Monadic second-order logic with arbitrary monadic predicates," in *Mathematical Foundations of Computer Science 2014: 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, E. Csuhaj-Varjú, M. Dietzfelbinger, and Z. Ésik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 279–290.

[24] T. Schwentick, "Padding and the expressive power of existential second-order logics," in *CSL 1997*, M. Nielsen and W. Thomas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 461–477.