Locality of Queries Definable in Invariant First-Order Logic with Arbitrary Built-in Predicates

Matthew Anderson[1,⋆], Dieter van Melkebeek[1,⋆],
Nicole Schweikardt[2], and Luc Segoufin[3]

[1] University of Wisconsin - Madison, USA
{mwa,dieter}@cs.wisc.edu
[2] Goethe-Universität Frankfurt am Main, Germany
schweika@informatik.uni-frankfurt.de
[3] INRIA and ENS-Cachan, LSV, France
http://www-rocq.inria.fr/~segoufin

Abstract. We consider first-order formulas over relational structures which may use arbitrary numerical predicates. We require that the validity of the formula is independent of the particular interpretation of the numerical predicates and refer to such formulas as Arb-invariant first-order.

Our main result shows a Gaifman locality theorem: two tuples of a structure with n elements, having the same neighborhood up to distance $(\log n)^{\omega(1)}$, cannot be distinguished by Arb-invariant first-order formulas. When restricting attention to word structures, we can achieve the same quantitative strength for Hanf locality. In both cases we show that our bounds are tight.

Our proof exploits the close connection between Arb-invariant first-order formulas and the complexity class AC^0, and hinges on the tight lower bounds for parity on constant-depth circuits.

1 Introduction

Definability in logics plays an important and delicate role in model checking, a central tool in several areas of computer science such as databases and automated verification. The problem consists in testing whether a given structure satisfies a certain property expressed in the logic. On the one hand, wider expressibility allows for more efficient implementations of a given property. On the other hand, limits on the expressibility keep the model checking task tractable and may be desirable for other reasons. For example, a database can be stored on a disk, which induces a linear order on the elements. An implementation of a query may exploit this order for looping through all the elements of the structure and performing all kinds of numerical computations. At the same time, the result of the query should only depend on the database and not on the linear order that is specific to its current representation on disk. In the database context this requirement is known as the *data independence principle*. In logic we usually speak of *closure under isomorphisms*.

In order to capture such requirements, we consider finite relational structures and queries expressed using first-order (FO) formulas which also have access to a binary

[⋆] Partially supported by NSF grants 0728809 and 1017597.

L. Aceto, M. Henzinger, and J. Sgall (Eds.): ICALP 2011, Part II, LNCS 6756, pp. 368–379, 2011.

predicate that is always interpreted as a linear order on the domain of the relational structures. We accommodate numerical computations by also allowing arbitrary numerical predicates in the logic. We require that the result of a query not depend on the actual choice of the linear order when all numerical predicates are interpreted consistent with the linear order. We refer to this logic as *Arb-invariant* FO. The special case where the query does not use any numerical predicate except for the linear order coincides with the well-known notion of *order-invariant* FO (cf., e.g., [12]).

In terms of computational power, Arb-invariant FO expresses precisely the properties computable within the complexity class AC^0, and when restricting to formulas using only the numerical predicates to $+$ and $*$ it corresponds to the uniform version of AC^0 [11]. In particular, Arb-invariant FO is for AC^0 what Arb-invariant Least Fixed Point logic (LFP) is for $\mathrm{P}/poly$ [14], and $(+,*)$-invariant FO is for uniform AC^0 what order-invariant (LFP) is for PTime [16,10]. Note that $+$ and $*$ are definable in order-invariant LFP and therefore can be omitted from the syntax, but this is no longer the case when considering first-order logic. It should be noted, however, that Arb-invariant FO and order-invariant LFP are "logical systems" rather than "logics" in the strict formal sense, as their syntax is undecidable (cf., e.g., [12]).

In this paper we study the expressive power of Arb-invariant FO and therefore the power of the complexity class AC^0. More precisely, we investigate the locality of Arb-invariant FO queries. Locality is a central notion in the study of first-order formulas. It provides good intuition for the expressive power of such formulas, and a powerful tool for showing inexpressibility. For instance, any non-local property such as acyclicity, connectivity, or k-colorability can immediately be shown non-expressible in a logic that exposes a certain amount of locality (see, e.g., [12, Chapter 4]). Locality is also exploited in an essential way in the design of efficient algorithms for evaluating first-order definable queries on certain classes of structures [3,5].

There are two important notions of locality, known as *Gaifman locality* and *Hanf locality*. Both are based on the distance measure on the elements of a structure when viewed as the vertices of the structure's Gaifman graph (in which two elements are connected by an edge whenever they appear together in a tuple of one of the structure's relations). In a nutshell, Gaifman locality means that a query cannot distinguish between two tuples having the same neighborhood type in a given structure, while Hanf locality means that a query cannot distinguish between two structures having the same (multi-)set of neighborhood types. Here, the neighborhood type of a tuple refers to the isomorphism type of the substructure induced by the elements up to distance r from the tuple, where r is a parameter. It is known that Hanf locality implies Gaifman locality, modulo a constant factor loss in the distance parameter r (cf., e.g., [12, Theorem 4.11]).

A well-known result (see, e.g., [12]) shows that FO enjoys both Hanf and Gaifman locality with a "constant" parameter r, i.e. depending only on the query. In the sequel we refer to this property as $\omega(1)$-locality. In the presence of an extra linear order that is part of the structure, all neighborhoods of positive radius degenerate to the entire domain, so all queries are trivially 1-local. Locality becomes meaningful again in order-invariant FO, where the formulas can make use of an order, but the structure does not contain the order and the semantics are independent of the order. It is shown in [8] that order-invariant FO queries are Gaifman $\omega(1)$-local. When we allow arbitrary numerical predicates, $\omega(1)$-locality turns out to fail, even if we require Arb-invariance. However, by allowing the parameter r to depend on the number n of elements of the structure, we provide an essentially complete picture in the case of Gaifman locality.

Theorem 1. *Arb-invariant* FO *formulas are Gaifman* $(\log n)^{\omega(1)}$*-local, and for every* $c \in \mathbb{N}$ *there exists an Arb-invariant* FO *formula that is not Gaifman* $(\log n)^c$*-local.*

The upper bound in Theorem 1 means that for any query in Arb-invariant FO and any large enough number n, if a structure has n elements and if two tuples of that structure have the same neighborhood up to distance $(\log n)^{f(n)}$ for any function $f \in \omega(1)$, then they cannot be distinguished by the query.

As in the case of order-invariant FO ([8]), the Hanf locality of Arb-invariant FO queries is still open in general. However if we restrict our attention to structures that represent strings, we can establish Hanf locality with the same bounds as in Theorem 1. Recall that order-invariant FO is known to be Hanf $\omega(1)$-local over strings [2]. In the following statement, Arb-invariant FO$(Succ)$ refers to Arb-invariant queries over string structures.

Theorem 2. *Arb-invariant* FO$(Succ)$ *sentences are Hanf* $(\log n)^{\omega(1)}$*-local, and for every* $c \in \mathbb{N}$ *there exists an Arb-invariant* FO$(Succ)$ *sentence that is not Hanf* $(\log n)^c$*-local.*

Proof Techniques. The proof of the upper bound on Gaifman locality in Theorem 1 exploits the tight connection between Arb-invariant FO formulas and the complexity class AC^0. The notion of locality in logic has a similar flavor to the notion of sensitivity in circuit complexity, and AC^0 is known to have low (polylogarithmic) sensitivity [13]. The latter result is closely related to the exponential lower bounds for parity on constant-depth circuits [9]. Rather than going through sensitivity, our argument directly uses the circuit lower bounds, namely as follows.

Given an Arb-invariant FO formula φ that distinguishes two points of the universe whose neighborhoods are of the same type up to distance r, we construct a circuit on $2m = \Theta(r)$ inputs that distinguishes inputs with exactly m ones from inputs with exactly $m + 1$ ones. In the special case of disjoint neighborhoods the circuit actually computes parity. The depth of the circuit is a constant depending on φ, and its size is polynomial in n. The known exponential circuit lower bounds then imply that r is bounded by a polylogarithmic function in n. This argument establishes the upper bound in Theorem 1 for the case of formulas with a single free variable and has some similarities with the proof of [8] establishing the $\omega(1)$-locality of order-invariant FO. However our proof is technically simpler and hinges on circuit lower bounds while the argument in [8] refers to Ehrenfeucht-Fraïssé games.

In order to handle an arbitrary number k of free variables, we follow again the same outline as [8]: we show how to reduce any case with $k > 1$ free variables to one with fewer variables. Our reduction is conceptually harder than the one in [8]. Indeed the reduction in [8] changes the size of the universe which can be done while preserving order-invariance but makes the preservation of Arb-invariance impossible.

The proof of the upper bound in Theorem 2 follows from a reduction to the upper bound in Theorem 1. This strategy differs from the one used in [2], which argues that the expressive power of order-invariant FO on strings is the same as FO (and is hence Hanf $\omega(1)$-local), because Arb-invariant FO$(Succ)$ can express non-FO properties.

The lower bounds in Theorems 1 and 2 follow because arithmetic predicates like addition and multiplication allow one to define a bijection between the elements of a first-order definable subset S of the domain of polylogarithmic size and an initial segment of the natural numbers [4]. Thus, (the binary representation of) a single element of the entire domain can be used to represent a list of elements of S. By exploiting this, Arb-invariant FO can express, e.g., reachability between two nodes in S by a path of polylogarithmic length.

We defer all proofs to the full paper, which may be found on the authors' websites.

2 Preliminaries

Arb-invariant First-Order Logic. A *relational schema* is a set of relation symbols each with an associated arity. A τ*-structure* M over a relational schema τ is a *finite* set *dom*(M), the *domain*, containing all the *elements* of M, together with an interpretation R^M of each relation symbol $R \in \tau$. If U is a set of elements of M, then $M_{|U}$ denotes the *induced substructure of* M *on* U. That is, $M_{|U}$ is the structure whose domain is U and whose relations are the relations of M restricted to those tuples containing only elements in U.

We say that two τ-structures M and M' are *isomorphic*, $M \cong M'$, if there exists a bijection $\pi : dom(M) \to dom(M')$ such that for each k-ary relation symbol $R \in \tau$, $(a_1, a_2, \ldots, a_k) \in R^M$ iff $(\pi(a_1), \pi(a_2), \ldots, \pi(a_k)) \in R^{M'}$. We write $\pi : M \cong M'$ to indicate that π is an isomorphism that maps M to M'. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are tuples (of the same length) of distinguished elements of *dom*(M) and *dom*(M'), respectively, then we write $(M, \boldsymbol{a}) \cong (M', \boldsymbol{b})$ to indicate that there is an isomorphism $\pi : M \cong M'$ which maps $\boldsymbol{a}$ to $\boldsymbol{b}$. All classes of structures considered in this paper are closed under isomorphisms.

Fix an infinite schema σ_{arb}, containing a binary symbol $<$ together with a symbol for each numerical predicate. For instance σ_{arb} contains a symbol $+$ for addition, $*$ for multiplication, and so on. Each numerical predicate is implicitly associated, for every $n \in \mathbb{N}$, with a specific interpretation as a relation of the appropriate arity over the domain $[n] := \{1, 2, \ldots, n\}$. For instance $+$ is associated with the restriction on $[n]$ of the classical relation of addition over $\mathbb{N}$. Reciprocally for each such family of relations, σ_{arb} contains an associated predicate symbol.

Let M be a τ-structure and $n = |dom(M)|$. An Arb-expansion of M is a structure M' over the schema consisting of the disjoint union of τ and σ_{arb} such that $dom(M) = dom(M')$, M and M' agree on all relations in τ, and $<$ is interpreted as a linear order over *dom*(M). This interpretation induces a bijection between *dom*(M) and $[n]$, identifying each element of M' with its index relative to $<$. All the numerical predicates are then interpreted over *dom*(M') via this bijection and their associated interpretation over $[n]$. For instance, $+$ is the ternary relation containing all tuples (a, b, c) of $dom(M')^3$ such that $i + j = k$, where a, b, and c are respectively the i^{th}, j^{th} and k^{th} elements of *dom*(M') relative to $<$. Note that M' is completely determined by M and the choice of the linear order $<$ on *dom*(M).

We denote by FO(τ) the first-order logic with respect to the schema τ. We use the standard syntax and semantics for FO (cf., e.g., [12]). If ϕ is a formula, we write $\phi(\boldsymbol{x})$ to denote that $\boldsymbol{x}$ is a list of the free variables of ϕ. We write $(M, \boldsymbol{a})$ when we want to emphasize the fact that $\boldsymbol{a}$ are distinguished elements of M. We also write $M \models \phi(\boldsymbol{a})$ or $(M, \boldsymbol{a}) \models \phi(\boldsymbol{x})$ to express that the tuple $\boldsymbol{a}$ of elements in *dom*(M) makes the formula $\phi(\boldsymbol{x})$ true on M.

We denote by FO(τ, Arb) the set of first-order formulas using the schema $\tau \cup \sigma_{\text{arb}}$. A formula $\phi(\boldsymbol{x})$ of FO(τ, Arb) is said to be *Arb-invariant on a finite structure M* over the schema τ, if for any tuple $\boldsymbol{a}$ of elements of *dom*(M), and any two Arb-expansions M' and M'' of M we have

$$M' \models \phi(\boldsymbol{a}) \iff M'' \models \phi(\boldsymbol{a}). \tag{1}$$

When $\phi(\boldsymbol{x})$ is Arb-invariant with respect to all finite structures M over a schema, we simply say that $\phi(\boldsymbol{x})$ is *Arb-invariant.*

When $\phi(\boldsymbol{x})$ is an Arb-invariant formula of FO(τ, Arb) on M, we write $M \models \phi(\boldsymbol{a})$ whenever there is an Arb-expansion M' of M such that $M' \models \phi(\boldsymbol{a})$. Hence Arb-invariant formulas can be viewed as formulas over τ-structures. We denote by Arb-invariant FO(τ) the set of Arb-invariant formulas of FO(τ, Arb), or simply Arb-invariant FO if τ is clear from the context. When the formula uses only the predicate $<$ of σ_{arb}, we have the classical notion of order-invariant FO (see [8] and [12]).

Locality. To each structure M we associate an undirected graph $G(M)$, known as the *Gaifman graph* of M, whose vertices are the elements of the domain of M and whose edges relate two elements of M whenever there exists a tuple in one of the relations of M in which both appear. For example, consider a relational schema τ consisting of one binary relation symbol E. Each τ-structure M is then a directed graph in the standard sense, and $G(M)$ coincides with M when ignoring the orientation. Given two elements u and v of a structure M, we denote as $\textit{dist}^M(u, v)$ the distance between u and v in M which is defined as their distance in the Gaifman graph $G(M)$. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are tuples of elements of M, then $\textit{dist}^M(\boldsymbol{a}, \boldsymbol{b})$ denotes the minimum distance between any pair of elements (one from $\boldsymbol{a}$ and one from $\boldsymbol{b}$).

For every $r \in \mathbb{N}$ and tuple $\boldsymbol{a} \in \textit{dom}(M)^k$, the *$r$-ball around $\boldsymbol{a}$ in M* is the set

$$N_r^M(\boldsymbol{a}) \ := \ \{v \in \textit{dom}(M) : \textit{dist}^M(\boldsymbol{a}, v) \leq r\}.$$

and the *r-neighborhood around $\boldsymbol{a}$ in M* is the structure

$$\mathcal{N}_r^M(\boldsymbol{a}) \ := \ \big(M_{|N_r^M(\boldsymbol{a})}, \boldsymbol{a}\big).$$

$\mathcal{N}_r^M(\boldsymbol{a})$ is the induced substructure of M on $N_r^M(\boldsymbol{a})$ with k distinguished elements $\boldsymbol{a}$.

Gaifman Locality. Let f be a function from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$. A formula $\phi(\boldsymbol{x})$ is said to be Gaifman f-local with respect to an infinite class of structures $\mathcal{M}$ if there exists $n_0 \in \mathbb{N}$ such that for any $n > n_0$, for any structure $M \in \mathcal{M}$ with $n = |\textit{dom}(M)|$, and any tuples $\boldsymbol{a}$ and $\boldsymbol{b}$ we have

$$\mathcal{N}_{f(n)}^M(\boldsymbol{a}) \cong \mathcal{N}_{f(n)}^M(\boldsymbol{b}) \implies M \models \phi(\boldsymbol{a}) \text{ iff } M \models \phi(\boldsymbol{b}). \tag{2}$$

For a set of functions F, a formula is said to be Gaifman F-local if it is Gaifman f-local for every $f \in F$.

Hanf Locality. Let f be a function from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$. For any two τ-structures M, M' with domain size n we write $M \equiv_{f(n)} M'$ if there is a bijection $h : dom(M) \to dom(M')$ such that for all elements a in the domain of M we have $\mathcal{N}^{M}_{f(n)}(a) \cong \mathcal{N}^{M'}_{f(n)}(h(a))$.

A sentence ϕ is said to be Hanf f-local if there is a n_0 such that for all τ-structures M, M' with domain size $n > n_0$ we have

$$M \equiv_{f(n)} M' \quad \Longrightarrow \quad M \models \phi \text{ iff } M' \models \phi. \tag{3}$$

For a set of functions F, a sentence is said to be Hanf F-local if it is Hanf f-local for every $f \in F$.

Circuit complexity. We assume basic familiarity with Boolean circuits. A *family of circuits* is a sequence $(C_m)_{m\in\mathbb{N}}$ such that for all $m \in \mathbb{N}$, C_m is a circuit using m input variables, hence defining a function from $\{0,1\}^m$ to $\{0,1\}$. We say that a language $L \subseteq \{0,1\}^*$ is accepted by a family of circuits $(C_m)_{m\in\mathbb{N}}$ if for all $m \in \mathbb{N}$ and for all binary words w of length m, $C_m(w) = 1$ iff $w \in L$.

When dealing with structures as inputs we need to encode the structures as strings. The precise encoding is not relevant for us as long as it is generic enough. We denote by $\mathrm{Rep}(M)$ the set of all binary encodings of M. Similarly, if $\boldsymbol{a}$ is a tuple of distinguished elements of M, then $\mathrm{Rep}(M, \boldsymbol{a})$ denotes the set of all binary encodings of $(M, \boldsymbol{a})$.

AC^0 *and* FO(τ,*Arb*). A language L is in (nonuniform) AC^0 if there exists a family of circuits $(C_m)_{m\in\mathbb{N}}$ accepting L, a constant $d \in \mathbb{N}$, and a polynomial function $p(m)$ such that for all $m \in \mathbb{N}$ each circuit C_m has depth d and size at most $p(m)$. There is a strong connection between AC^0 and FO(τ, Arb) [11]. We make use of the following characterization with respect to Arb-invariant FO(τ, Arb).

Lemma 3 (Implicit in [11]). *Let $\phi(\boldsymbol{x})$ be a k-ary* FO(τ,*Arb*) *formula which is Arb-invariant on a class of τ-structures $\mathcal{M}$. There exists a family of constant-depth and polynomial-size circuits $(C_m)_{m\in\mathbb{N}}$ such that for each $M \in \mathcal{M}$, $\boldsymbol{a} \in dom(M)^k$, and $\Gamma \in \mathrm{Rep}(M, \boldsymbol{a})$,*

$$C_{|\Gamma|}(\Gamma) = 1 \quad \Longleftrightarrow \quad M \models \phi(\boldsymbol{a}).$$

Lower bounds. Our locality bounds hinge on the well-known exponential size lower bounds for constant-depth circuits that compute parity [1,6,17,9]. In fact, we use the following somewhat stronger promise version. For a binary word $w \in \{0,1\}^*$, we let $|w|_1$ denote the number of 1*s* in w.

Lemma 4 (Implicit in [9, Theorem 5.1]). *For any $d \in \mathbb{N}$, there are constants c and m_0 such that for $m > m_0$ there is no circuit of depth d and size $2^{cm^{\frac{1}{d-1}}}$ that $w \in \{0,1\}^{2m}$ accepts all inputs $w \in \{0,1\}^{2m}$ with $|w|_1 = m$ and rejects all inputs with $|w|_1 = m{+}1$.*

3 Gaifman Locality

We now prove the main result of the paper – the upper bound in Theorem 1. Recall, our theorem claims that every Arb-invariant FO formula is Gaifman $(\log n)^{\omega(1)}$-local. In fact we prove the following slightly stronger version.

Theorem 5. *For any* FO(τ, Arb) *formula* $\phi(\boldsymbol{x})$*, and infinite class* $\mathcal{M}$ *of* τ*-structures, if* $\phi(\boldsymbol{x})$ *is Arb-invariant on* $\mathcal{M}$*, then* $\phi(\boldsymbol{x})$ *is Gaifman* $(\log n)^{\omega(1)}$*-local on* $\mathcal{M}$*.*

We now briefly sketch the overall proof of Theorem 5. Suppose we have two tuples, $\boldsymbol{a}$ and $\boldsymbol{b}$, on a τ-structure M, with domain size n, such that their r-neighborhoods, $\mathcal{N}_r^M(\boldsymbol{a})$ and $\mathcal{N}_r^M(\boldsymbol{b})$, are isomorphic (for some big enough r). Suppose that there is a $\mathrm{FO}(\tau, \mathrm{Arb})$ formula $\phi(\boldsymbol{x})$ which is able to distinguish between $\boldsymbol{a}$ and $\boldsymbol{b}$ on M while being Arb-invariant on M. Using the link between Arb-invariant $\mathrm{FO}(\tau, \mathrm{Arb})$ formulas and AC^0 circuits from Lemma 3, we can view the formula $\phi(\boldsymbol{x})$ as a constant-depth circuit C.

We are able to show that because $\phi(\boldsymbol{x})$ is Arb-invariant and distinguishes between $\boldsymbol{a}$ and $\boldsymbol{b}$ on M, we can construct from the circuit C and structure M another circuit $\tilde{C}$ that for $(2m)$-length binary strings w distinguishes between the cases when w contains m occurrences of 1 and $m+1$ occurrences, for some m depending on r. This is the *key step* in our argument. If this happens for infinitely many n, we get a family of circuits computing the promise problem described in Lemma 4. We can argue that the size of $\tilde{C}$ is polynomial in n and the depth of $\tilde{C}$ only depends on $\phi(\boldsymbol{x})$ and hence is constant. Therefore if $m \in (\log n)^{\omega(1)}$ the family of circuits $\tilde{C}$ we construct violates Lemma 4 hence $\phi(\boldsymbol{x})$ cannot distinguish between tuples which have isomorphic r-neighborhoods. Our construction is such that m is linearly related to r and therefore $\phi(\boldsymbol{x})$ is Gaifman $(\log n)^{\omega(1)}$-local.

3.1 Unary Formulas

In this subsection we consider only unary FO formulas $\phi(x)$. For didactic reasons we first assume that the r-neighborhoods of the elements a and b are disjoint. We argue that we can perform the key step in this setting, then consider the general unary case. We conclude by arguing a unary version of Theorem 5.

For clarity we describe the intuition with respect to structures that are graphs. Let M be a graph $G = (V, E)$ and take two vertices $a, b \in V$ such that $\pi : \mathcal{N}_r^G(a) \cong \mathcal{N}_r^G(b)$. Suppose, for the sake of contradiction, that there is a unary FO formula $\phi(x)$, which is Arb-invariant on G, such that $G \models \phi(a) \wedge \neg\phi(b)$. Applying Lemma 3 to ϕ gives us a circuit C which, for any vertex $c \in V$, outputs the same value for all strings in $\mathrm{Rep}(G, c)$, and distinguishes $\mathrm{Rep}(G, a)$ from $\mathrm{Rep}(G, b)$.

Disjoint neighborhoods. Let us assume that $\mathcal{N}_r^G(a) \cap \mathcal{N}_r^G(b) = \emptyset$. In this setting it turns out we can pick $r = m$. The neighborhood isomorphism, $\pi : \mathcal{N}_r^G(a) \cong \mathcal{N}_r^G(b)$, implies that the balls of radius $i < r$ around a and b are isomorphic and disjoint in G. Consider the following procedure, depicted in Figure 1. For some $i \in [m]$ cut all the edges linking nodes at distance $i - 1$ from a or b to nodes at distance i. Now, swap the positions of the $(i-1)$-neighborhoods around a and b and reconnect the edges in a way that respects the isomorphism π. The resulting graph is isomorphic to G, but the relative positions of a and b have swapped.

Using this intuition we construct a new graph G_w from G, a, and b that depends on a sequence of m Boolean variables $w := w_1 w_2 \cdots w_m$. We construct G_w so that for each variable w_i, we swap the relative positions of the radius $i - 1$ balls around a and b iff

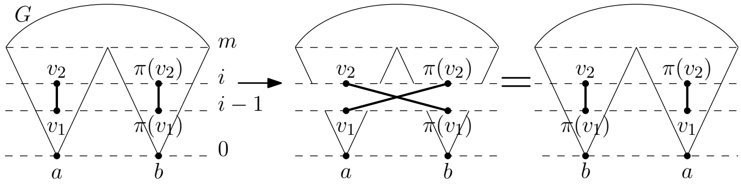

Fig. 1. Diagram for swapping the neighborhoods of a and b of radius i, conditioned on $w_i = 1$

w_i is 1. The number of such swaps is $|w|_1$. The m-neighborhood isomorphism between a and b implies that $G_w \cong G$. When $|w|_1$ is even $(G_w, a) \cong (G, a)$ and when $|w|_1$ is odd $(G_w, a) \cong (G, b)$.

Using the above construction of G_w we derive a circuit $\tilde{C}$ from C that computes parity on m bits. The circuit $\tilde{C}$ first computes a representation $\Gamma_w \in \mathrm{Rep}(G_w, a)$, and then simulates C on input Γ_w. The above distinguishing property then implies that $\tilde{C}$ computes parity on m bits. To construct Γ_w we start with a fixed string in $\mathrm{Rep}(G, a)$ and transform it into an element of $\mathrm{Rep}(G_w, a)$ by modifying the edges to switch between the shells in the manner suggested above. Observe that the presence of each edge in G_w depends on at most a single bit of w. This property implies that Γ_w consists of constants and variables in w or their negations. This means that $\tilde{C}$ is no larger or deeper than C.

We formalize this intuition for general structures and obtain the following lemma.

Lemma 6. *Let $m \in \mathbb{N}$. Let M be a structure. Let $a, b \in dom(M)$ such that $dist^M(a, b) > 2m$ and $\mathcal{N}_m^M(a) \cong \mathcal{N}_m^M(b)$. Let C be a circuit that accepts all strings in* $\mathrm{Rep}(M, a)$*, and rejects all strings in* $\mathrm{Rep}(M, b)$*. There is a circuit $\tilde{C}$ with the same size and depth as C that computes parity on m bits.*

General Unary Case. We now develop the transformation corresponding to Lemma 6 for the general unary case, where the r-neighborhoods around a and b may overlap. As before, we describe the intuition in terms of structures that are graphs.

Consider the iterative application of the isomorphism π to a. We distinguish between two cases. The first case occurs when this iteration travels far from a. That is, for some $t \in \mathbb{N}$, $\pi^t(a)$ is a point c that is far from a. We choose r large enough so that a large neighborhood around c is isomorphic to the neighborhood around a or b. By the triangle inequality, since a is far from c either: b is far from a, or c is far from a and b. In the latter case C must distinguish between either a and c or b and c. Therefore, w.l.o.g., we have a pair of vertices that are distinguished by C, and whose neighborhoods are isomorphic and disjoint. For this pair of vertices, we are in the disjoint case and Lemma 6 can be applied to produce a small circuit that computes parity.

The other case occurs when the iterative application of π to a stays close to a (and b). Let S_0 be the orbit of a under π, and let S_i be the vertices at distance i from S_0, for $i \in [2m]$. Because $\pi(S_0) = S_0$ and π is a partial isomorphism on G, the shells S_i are closed under π.

We now play a game similar to the disjoint case. Consider the following procedure, depicted in Figure 2. For some $i \in [2m]$ cut all edges between the shell S_{i-1} and S_i. "Rotate" the radius $i - 1$ ball around S_0 by π relative to S_i, and reconnect the edges.

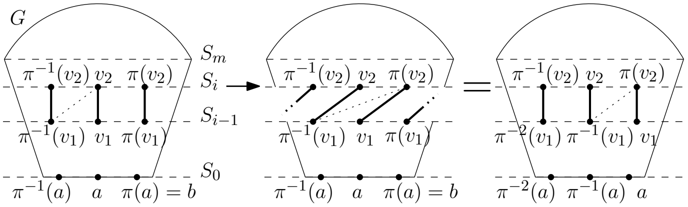
G
S_m
S_i
S_{i-1}
S_0
$\pi^{-1}(v_2)$ v_2 $\pi(v_2)$
$\pi^{-1}(v_1)$ v_1 $\pi(v_1)$
$\pi^{-1}(a)$ a $\pi(a) = b$
$\pi^{-2}(v_1)$ $\pi^{-1}(v_1)$ v_1
$\pi^{-2}(a)$ $\pi^{-1}(a)$ a

Fig. 2. Diagram for rotating the shell of radius i around S_0 when $w_i = 1$

Because the shells are closed under π, the resulting graph is isomorphic to G. Further, the positions of a and b have shifted relative to an application of π.

As before, we encode this behavior into a modified graph G_w depending on a sequence of $2m$ Boolean variables $w := w_1 w_2 \cdots w_{2m}$. When $w_i = 0$, we preserve the edges between the shells S_{i-1} and S_i. When $w_i = 1$ we rotate the edges by π. That is, an edge $(v_1, v_2) \in (S_{i-1} \times S_i) \cap E$ becomes the edge $(v_1, \pi(v_2))$ in G_w. The neighborhood isomorphism between a and b implies that $G \cong G_w$. We can argue that

$$(G_w, a) \cong (G, \pi^{|w|_1}(a)). \tag{4}$$

We define the circuit $\tilde{C}$ to simulate C on an input $\Gamma_w \in \mathrm{Rep}(G_w, a)$. The above distinguishing property implies that $\tilde{C}$ distinguishes between $|w|_1 \equiv 0 \bmod |S_0|$ and $|w|_1 \equiv 1 \bmod |S_0|$. This is not quite the promise problem defined in Lemma 4. For this reason we modify the construction to shift a by m applications of π^{-1} in Γ_w. This means that $\Gamma_w \in \mathrm{Rep}(G_w, \pi^{-m}(a))$ and $\tilde{C}$ can distinguish between $|w|_1 \equiv m \bmod |S_0|$ and $|w|_1 \equiv m+1 \bmod |S_0|$. This is ruled out by Lemma 4, completing the argument.

For general structures, the idea is formalized in the following lemma.

Lemma 7. *Let $m \in \mathbb{N}$. Let M be a structure. Let $a, b \in dom(M)$ such that $\mathcal{N}^M_{12m}(a) \cong \mathcal{N}^M_{12m}(b)$. Let C be a circuit that accepts all strings in $\mathrm{Rep}(M, a)$ and rejects all strings in $\mathrm{Rep}(M, b)$, and for each $c \in dom(M)$, C has the same output for each string in $\mathrm{Rep}(M, c)$. There is a circuit $\tilde{C}$ with the same size and depth as C that distinguishes $|w|_1 = m$ and $|w|_1 = m + 1$ for $w \in \{0,1\}^{2m}$.*

Proof of Theorem 5 in the case of unary formulas. Now that we know how to construct the circuit $\tilde{C}$ as in Lemmas 6 and 7, we are ready to finish the proof of Theorem 5 in the unary case. Assume that $\phi(x)$ is a unary formula of $\mathrm{FO}(\tau, \mathrm{Arb})$ that is Arb-invariant with respect to an infinite class of τ-structures $\mathcal{M}$.

Since $\phi(x)$ is $\mathrm{FO}(\tau, \mathrm{Arb})$, it is computable by a family of circuits in AC^0 (cf., Lemma 3). That is, there are a constant d, polynomials $s(n)$ and $r(n)$, and a circuit family $(C_{r(n)})_{n \in \mathbb{N}}$ with depth d and size $s(n)$ such that, for every $n \in \mathbb{N}$, the circuit $C_{r(n)}$ computes $\phi(x)$ on structures $M \in \mathcal{M}$ with $|dom(M)| = n$ and $r(n)$ bounds the length of the binary encoding of (M, a) for any $a \in dom(M)$. Since $\phi(x)$ is Arb-invariant on M, $C_{r(n)}$ has the same output for all strings in $\mathrm{Rep}(M, a)$ for each $a \in dom(M)$.

Now, for the sake of contradiction, suppose $\phi(x)$ is *not* Gaifman $(\log n)^{\omega(1)}$-local on $\mathcal{M}$. This implies that there is an infinite subclass of structures $\mathcal{M}' \subseteq \mathcal{M}$ and a function

$f(n)$ in $(\log n)^{\omega(1)}$, where for each $M \in \mathcal{M}'$, $\phi(x)$ distinguishes between two elements $a, b \in \mathit{dom}(M)$ having isomorphic $f(n)$-neighborhoods.

Consider some structure $M \in \mathcal{M}'$, with $n = |\mathit{dom}(M)|$. Let $m := \lfloor \frac{f(n)}{12} \rfloor$. By the above, there exists $a, b \in \mathit{dom}(M)$ such that $\mathcal{N}^M_{12m}(a) \cong \mathcal{N}^M_{12m}(b)$ and $M \models \phi(a) \wedge \neg\phi(b)$ without loss of generality. Let $C := C_{r(n)}$; this circuit then satisfies the assumptions of Lemma 7. From the lemma, we obtain a circuit $\tilde{C}$ of depth d and size $s(n)$ that distinguishes between $|w|_1 = m$ and $|w|_1 = m + 1$ for $w \in \{0, 1\}^{2m}$.

From Lemma 4 we obtain that $s(n) > 2^{cm^{1/(d-1)}}$. Noting that $m = (\log n)^{\omega(1)}$, s is polynomial, and d is constant, this yields a contradiction by choosing $M \in \mathcal{M}'$ sufficiently large, completing the proof. □

3.2 *k*-ary Formulas

To argue Theorem 5 in the case of formulas with an arbitrary number of free variables, we prove the following reduction. Given a k-ary FO(Arb) formula ϕ that is Arb-invariant on the structure M and distinguishes two k-tuples $\boldsymbol{a}$ and $\boldsymbol{b}$ that have isomorphic r-neighborhoods, we produce, for some $k' < k$, a k'-ary formula FO(Arb) ϕ' that is Arb-invariant on an extended structure M' and distinguishes between two k'-tuples $\boldsymbol{a}'$ and $\boldsymbol{b}'$ that have isomorphic r'-neighborhoods. Furthermore, r' is only slightly smaller than r. The formal statement of the reduction is as follows.

Lemma 8. *Let $k, d \in \mathbb{N}$, r a function from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$, and τ be a schema. Fix $\alpha \leq \frac{1}{7k}$. Let $\phi(\boldsymbol{x})$ be a k-ary* FO(τ, Arb) *formula of quantifier-depth d that is Arb-invariant over an infinite class of τ-structures $\mathcal{M}$ and that is not Gaifman r-local.*

Then there is $k' < k$, a schema $\tau' \supseteq \tau$, an infinite class of τ'-structures $\mathcal{M}'$ and a k'-ary FO(τ', Arb) *formula $\phi'(\boldsymbol{y})$ of quantifier-depth $(d+(k-k'))$ that is Arb-invariant over $\mathcal{M}'$ and not Gaifman αr-local.*

Repeated application of this lemma transforms a distinguishing k-ary formula into a distinguishing unary formula with slightly weaker parameters. For large enough initial radius this is sufficient to contradict the Gaifman locality of unary formulas.

4 Hanf Locality

The main result of this section is the upper bound in Theorem 2, which states that Arb-invariant FO formulas over strings are Hanf $(\log n)^{\omega(1)}$-local.

Fix a finite alphabet $\mathbb{A}$ and consider structures over the schema τ_s containing one unary predicate per element of $\mathbb{A}$ and one binary predicate E. Let $\mathcal{S}$ be the class of τ_s-structures M that interprets E as a successor relation and where each element of M belongs to exactly one of the unary predicates in τ_s. Each structure in $\mathcal{S}$ represents a string in the obvious way and we blur the distinction between a string w and its actual representation as a structure. We then consider FO$(\tau_s \cup \sigma_{\text{arb}})$ formulas that are Arb-invariant over $\mathcal{S}$ and denote the corresponding set of formulas by Arb-invariant FO(*Succ*). We say that a language $L \subseteq \mathbb{A}^*$ is definable in Arb-invariant FO(*Succ*) if there is a sentence of Arb-invariant FO(*Succ*) whose set of models in $\mathcal{S}$ is exactly L.

The proof of the upper bound in Theorem 2 has several steps. We first introduce a closure property of languages allowing, under certain conditions, substrings inside a word to be swapped without affecting language membership. We then argue that languages definable in Arb-invariant FO(Succ) have this closure property. Finally, we conclude by proving that this closure property implies that Arb-invariant FO(Succ) sentences are Hanf $(\log n)^{\omega(1)}$-local. In the following, we describe these steps in some more detail.

Arb-invariant FO(Succ) is Closed Under Swaps. Let $f : \mathbb{N} \to \mathbb{R}_{\geq 0}$. A language L is said to be *closed under $f(n)$-swaps* if there exists a $n_0 \in \mathbb{N}$ such that for all strings $w := xuyvz \in \mathbb{A}^*$, with $|w| = n > n_0$, and where i, j, i', and j' are, respectively, the positions in w immediately before the substrings u, y, v, and z, and $\mathcal{N}^w_{f(n)}(i) \cong \mathcal{N}^w_{f(n)}(i')$ and $\mathcal{N}^w_{f(n)}(j) \cong \mathcal{N}^w_{f(n)}(j')$ we have: $w := xuyvz \in L$ iff $w' := xvyuz \in L$. A language is closed under F-swaps if is it closed under $f(n)$-swaps for all $f \in F$.

Let ϕ be an Arb-invariant FO(Succ) sentence. Suppose the strings w and w' satisfy the initial conditions for closure under $(\log n)^{\omega(1)}$-swaps, but are distinguished by ϕ. We first consider the case when the four $f(n)$-neighborhoods of i, j, i', j' are disjoint. In this case not only do the neighborhoods around the strings u and v look same, but u and v are far apart. We define a FO(Arb) formula ψ derived from ϕ and a structure M derived from w and w', such that ψ on M simulates ϕ on either w or w' depending on the input to ψ. Moreover, ψ is Arb-invariant on M and the input tuples that ψ distinguishes have large isomorphic neighborhoods, implied by the neighborhood isomorphisms among i, j, i', and j'. Applying our Gaifman locality theorem (Theorem 1) to the formula ψ induces a contradiction.

When the neighborhoods are not disjoint we reduce to the disjoint case by making two key observations. The first is that when some of the neighborhoods overlap only a small amount there is freedom to select slightly smaller neighborhoods that are pairwise disjoint, though still induce the same swap. The second insight is that when several of the neighborhoods have considerable overlap, the neighborhood isomorphisms induce periodic behavior within those neighborhoods. So much so that the substrings uyv and vyu must be identical. This contradicts the fact that w and w' are distinct. This intuition is formalized in the following lemma.

Lemma 9. *If L is a language definable in Arb-invariant* FO(Succ) *then L is closed under* $(\log n)^{\omega(1)}$*-swaps.*

Arb-invariant FO(Succ) is Hanf $(\log n)^{\omega(1)}$-local. We are now ready to prove the upper bound of Theorem 2. Consider a pair of equal length strings w, w' such that $w \equiv_{f(n)} w'$ for some bijection h. Observe that if $w = w'$, we can choose h to be the identity. The identity mapping is monotone in the sense that for each position $i \in [n]$, for all $j < i$, $h(j) < h(i)$. When $w \neq w'$, h is not the identity and not monotone. However, h is monotone when considering only the first position. We extend the set which h is monotone on by $(\log n)^{\omega(1)}$-swapping substrings of w while being careful to preserve the bijection to w'. Eventually h becomes monotone with respect to all positions and is the identity. The final insight is this, if we only perform $(\log n)^{\omega(1)}$-swaps language membership is maintained by Lemma 9. Thus, we transform between w and w' without changing language membership, so $w \in L$ iff $w' \in L$. Hence Arb-invariant FO(Succ) is Hanf $(\log n)^{\omega(1)}$-local.

5 Discussion

We have established the precise level of locality of Arb-invariant FO formulas for the Gaifman notion of locality. We leave it as an open problem whether the same bounds could be achieved for the Hanf notion of locality. We managed to prove Hanf locality for the special case of strings and we believe that a similar argument also works for trees and possibly graphs of bounded treewidth.

As pointed out in [7] "it would be interesting to see a small complexity class like uniform AC^0 [...] can be captured by a logic" (recall from the introduction that although Arb-invariant FO does capture AC^0, it does not have an effective syntax). As a (simple) first step towards a solution to this problem, in the journal version of this paper we will show that *over regular languages*, Arb-invariant $FO(Succ)$ has exactly the same expressive power as $FO(Succ, lm)$, where *lm* is the family of predicates testing the length of a string modulo some fixed number. Note that when combining this result with the one of [15], this shows all the numerical predicates do not bring any extra expressive power than the one of addition over regular languages.

References

1. Ajtai, M.: Σ_1^1-formulae on finite structures. APAL 24(1), 1–48 (1983)
2. Benedikt, M., Segoufin, L.: Towards a characterization of order-invariant queries over tame structures. JSL 74(1), 168–186 (2009)
3. Dawar, A., Grohe, M., Kreutzer, S.: Locally excluding a minor. In: Proc. of LICS (2007)
4. Durand, A., Lautemann, C., More, M.: Counting results in weak formalisms. In: Circuits, Logic, and Games. No. 06451 in Dagstuhl Seminar Proc. (2007)
5. Dvorak, Z., Král, D., Thomas, R.: Deciding first-order properties for sparse graphs. In: Proc. of FOCS (2010)
6. Furst, M., Saxe, J., Sipser, M.: Parity, circuits, and the polynomial-time hierarchy. TOCS 17(1), 13–27 (1984)
7. Grohe, M.: Fixed-point definability and polynomial time. In: Grädel, E., Kahle, R. (eds.) CSL 2009. LNCS, vol. 5771, pp. 20–23. Springer, Heidelberg (2009)
8. Grohe, M., Schwentick, T.: Locality of order-invariant first-order formulas. ACM TOCL 1(1), 112–130 (2000)
9. Håstad, J.: Computational limitations for small-depth circuits. Ph.D. thesis, MIT (1986)
10. Immerman, N.: Relational queries computable in polynomial time. Information and Control 68(1-3), 86–104 (1986)
11. Immerman, N.: Languages that capture complexity classes. SICOMP 16(4), 760–778 (1987)
12. Libkin, L.: Elements of Finite Model Theory. Springer, Heidelberg (2004)
13. Linial, N., Mansour, Y., Nisan, N.: Constant depth circuits, Fourier transform, and learnability. JACM 40(3), 620 (1993)
14. Makowsky, J.A.: Invariant definability and P/*poly*. In: Gottlob, G., Grandjean, E., Seyr, K. (eds.) CSL 1998. LNCS, vol. 1584, pp. 142–158. Springer, Heidelberg (1999)
15. Schweikardt, N., Segoufin, L.: Addition-invariant FO and regularity. In: Proc. of LICS (2010)
16. Vardi, M.: The complexity of relational query languages. In: Proc. of STOC (1982)
17. Yao, A.: Separating the polynomial-time hierarchy by oracles. In: Proc. of FOCS (1985)