Counting and Locality over Finite Structures
A Survey

Leonid Libkin
Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974, USA
Email: libkin@bell-labs.com

Juha Nurmonen*
Department of Mathematics
and Computer Science
University of Leicester
University Road
Leicester LE1 7RH, UK
Email: j.nurmonen@mcs.le.ac.uk

Abstract

We survey recent results on logics with counting and their local properties. We first consider game-theoretic characterizations of first-order logic and its counting extensions provided by unary generalized quantifiers. We then study Gaifman's and Hanf's locality theorems, their connection with game characterizations, and examples of their usage in proving expressivity bounds for first-order logic and its extensions. We review the abstract notions of Gaifman's and Hanf's locality, and show how they are related. We also consider a closely related bounded degree property, and demonstrate its usefulness in proving expressivity bounds. We discuss two applications in computer science. One deals with proving lower bounds for the class TC^0. In particular, we use logical characterization of TC^0 and locality theorems for first-order with counting quantifiers to provide lower bounds. We then explain how the notions of locality are used in database theory to prove that extensions of relational calculus with aggregate functions and grouping still lack the power to express fixpoint computation.

1 Introduction

Finite model theory is an active area of research, mostly due to its connections to theoretical computer science, in particular, database theory [1] and complexity theory [13, 28]. Several important complexity classes have nice logical characterizations. For instance, the star-free languages are exactly the ones definable in first-order logic $\mathcal{FO}$, least fixpoint logic LFP captures PTIME, and partial fixpoint logic PFP captures PSPACE on ordered finite structures (see

*Supported by EPSRC grant GR/K 96564.

[13]). These logics also have their counterparts in the theory of database query languages. For example, relational calculus, that underlies majority of practical query languages, has exactly the power of first-order logic. The language Datalog with negation corresponds to LFP, and the extension of relational calculus with while loops corresponds to PFP (see [1]).

Several counting properties have also been considered in complexity and database theory. We give a few examples here. A logic can be extended with a linearly ordered second sort universe of numbers, and *counting quantifiers* 'there are at least n elements.' A very strong result in [9] shows that least fixpoint logic with such second sort counting fails to capture the complexity class PTIME. In circuit complexity theory, *counting modulo quantifiers* have been studied. Allowing gates which count inputs modulo a constant p, for every p, in the definition of AC^0, one obtains the class ACC. Again, this class has a logical characterization, and the problem whether the containment $ACC \subseteq NC^1$ is strict, is still open. In [5], it was shown that the class TC^0, that extends AC^0 with threshold gates, can be captured on ordered structures by first-order logic with the second sort counting, and simple arithmetic predicates on numbers.

In database theory, one often extends traditional first-order based languages with aggregate functions, such as summing up all values in a column in a relation, or finding the average value. One also extends those languages with grouping, that permits queries such as finding the average salary in each department. While the expressive power of relational calculus and many other query languages is well understood, much less is known about their aggregate extensions.

These applications have motivated a systematic study of the expressive power of counting properties from the viewpoint of finite model theory. In finite model theory, one normally uses games to prove expressivity bounds. For example, Ehrenfeucht-Fraïssé games are used for first-order logic, and bijective Ehrenfeucht-Fraïssé games are used for $\mathcal{FO}$ extended with unary generalized quantifiers. In Section 3 we review rules of such games for $\mathcal{FO}$ and some of its extensions with counting.

Playing a game often involves a complicated combinatorial argument. Many result on expressive power for first-order logic and its extensions with counting, in particular, those obtained as an attempt to avoid game arguments, give us the intuition that these logics can express only local properties, and lack a mechanism for fixpoint computation. In Section 4, we review several results of this kind. We consider Gaifman's theorem [18] for first-order logic, which shows that every first-order formula is equivalent to a local one, in the sense that only a small part of a structure is relevant for evaluating the query given by a formula. We also study modifications of Hanf's result [21]. In this approach one counts the number of isomorphism types of fixed radius neighborhoods of points. If the result of this counting satisfies certain criteria, then the structures considered are guaranteed to be elementary equivalent in a certain logic. This technique has been modified for first-order logic [17], first-order logic with counting modulo quantifiers [39] and first-order logic extended by all unary generalized quantifiers [38], for the case of finite structures. Proofs of applicability of Hanf's technique typically are not very difficult [17, 15, 38, 40]. We will see some examples in Section 4.

The above results have motivated a study of general notions of locality [32, 25]. We review this line of work in Section 5. We show that Gaifman's theorem gives rise to two general notions, one for sentences and one for open formulas. We formulate an abstract notion of locality that captures Hanf's condition, and study the relationship between the notions of locality. We also consider the bounded degree property, which is implied by all other notions of locality, and is particularly well suited for proving expressivity bounds, especially for properties involving fixpoint computation.

In Section 6, we discuss applications of the concept of locality in complexity theory. In particular, we study the relationship between the circuit complexity class TC^0 and other complexity classes such as L and NL. In Section 7, we consider applications in database theory. We review results on expressive power of relational languages that resemble commercial languages such as SQL. We show that queries such as transitive closure, are inexpressible in a theoretical language that has the power of core SQL.

2 Preliminaries

A relational signature σ is a finite set of relation symbols $\{R_1, ..., R_l\}$, each of which has an arity $p_i > 0$. We write σ_n for σ extended with n new constant symbols. A σ-structure is $\mathcal{A} = (A, \overline{R}_1, \ldots, \overline{R}_l)$, where A is a non-empty set and $\overline{R}_i \subseteq A^{p_i}$ interprets R_i. If the *universe* A is finite, the structure $\mathcal{A}$ is called finite. Unless mentioned otherwise, all structures considered here are assumed to be finite. When the notation is clear from the context, we write R_i in place of $\overline{R}_i$. The class of finite σ-structures is denoted by STRUCT$[\sigma]$. For instance, a graph $\mathcal{A} = (A, E)$ is a structure over a signature which consists of a single binary relation symbol E. If E is required to be symmetric, then $\mathcal{A}$ is an undirected graph.

If $X \subseteq A$, by $\mathcal{A} \restriction X$ we mean the structure with universe X, where the interpretation of each R_i is restricted to X. An *isomorphism* $f : \mathcal{A} \to \mathcal{B}$ is a bijection $A \to B$ such that $(a_1, \ldots, a_{p_i}) \in \overline{R}_i$ holds in $\mathcal{A}$ if and only if $(f(a_1), \ldots, f(a_{p_i})) \in \overline{R}_i$ holds in $\mathcal{B}$. If $a_1, \ldots, a_n \in A$ and $b_1, \ldots, b_n \in B$, we say that $\{(a_i, b_i) \mid 1 \leq i \leq n\}$ is a *partial isomorphism* $\mathcal{A} \to \mathcal{B}$ if it is an isomorphism $\mathcal{A} \restriction \{a_1, \ldots, a_n\} \to \mathcal{B} \restriction \{b_1, \ldots, b_n\}$.

Every formula $\psi(x_1, \ldots, x_m)$ with free variables $x_1, \ldots, x_m$ defines a *query* which maps a σ-structure $\mathcal{A}$ to an m-ary relation $q_\psi(\mathcal{A}) = \{\vec{a} \in A^m \mid \mathcal{A} \models \psi(\vec{a})\}$. We denote the corresponding structure by $\psi[\mathcal{A}] = (A, q_\psi(\mathcal{A}))$. An m-ary query is definable in a logic $\mathcal{L}$ if there is a formula $\varphi(\vec{x})$ of that logic such that for every $\mathcal{A} \in$ STRUCT$[\sigma]$ we have $q(\mathcal{A}) = \{\vec{a} \in A^m \mid \mathcal{A} \models \varphi(\vec{a})\}$. As an example, consider the *transitive closure* query *TRCL*. Suppose we are given a graph $\mathcal{A} = (A, E)$. Then *TRCL* consists of all pairs $(a, b) \in A^2$ such that there is an E-path from a to b, that is,

$$q_{TRCL} = \{(a, b) \in A^2 \mid \exists a_1, \ldots, a_k \in A \text{ such that } a = a_1 \text{ and } b = a_k \text{ and } E(a_i, a_{i+1}) \text{ for all } i\}.$$

This query is definable in many fixpoint logics, and in second-order logic, but we shall see that it is not definable in $\mathcal{FO}$ and its counting extensions.

The *Gaifman graph* $\mathcal{G}(\mathcal{A})$ of a σ-structure $\mathcal{A}$ is the undirected graph (A, E) where $(a, b) \in E$ if and only if there is a tuple $\vec{t} \in \overline{R}_i$ for some $R_i \in \sigma$ such that $a, b \in \vec{t}$. Note that if $\mathcal{A}$ is an undirected graph, then $\mathcal{A} = \mathcal{G}(\mathcal{A})$. The *degree* of a point is its degree in the Gaifman graph.

The *distance* $d(a,b)$ is the length of the shortest path from a to b in $\mathcal{G}(\mathcal{A})$. For $a \in A$, its *r-sphere* is $S_r^{\mathcal{A}}(a) = \{b \in A \mid d(b,a) \leq r\}$. For a tuple $\vec{t}$ we define $S_r^{\mathcal{A}}(\vec{t}) = \cup_{a \in \vec{t}} S_r^{\mathcal{A}}(a)$. The *r-neighborhood* of a tuple $\vec{t}$ is the σ_n-structure $N_r^{\mathcal{A}}(\vec{t}) = (\mathcal{A} \restriction S_r^{\mathcal{A}}(\vec{t}), \vec{t})$. For instance, if $\mathcal{A}$ is a chordless cycle of length at least $2r+2$ and $a \in A$, then $N_r^{\mathcal{A}}(a)$ is the chordless path of length $2r+1$ with one distinguished point a. We denote the *isomorphism type* of $N_r^{\mathcal{A}}(\vec{t})$ by $tp_r^{\mathcal{A}}(\vec{t})$. We emphasize here that if $h : N_r^{\mathcal{A}}(\vec{a}) \to N_r^{\mathcal{A}}(\vec{b})$ is an isomorphism, where $\vec{a} = (a_1, \ldots, a_n)$ and $\vec{b} = (b_1, \ldots, b_n)$, then $h(a_1) = b_1, \ldots, h(a_n) = b_n$, as neighborhoods $N_r^{\mathcal{A}}(\vec{a})$ and $N_r^{\mathcal{A}}(\vec{b})$ are σ_n-structures.

If the structure $\mathcal{A}$ is understood, we omit it from the notations when convenient.

3 Logics and games

In this section we introduce the logics that are considered in this paper. These logics are first-order logic $\mathcal{FO}$ or its extensions with various generalized quantifiers. All logics we consider are closed under Boolean connectives and first-order quantification, and are regular in the sense of [12, 31] (we do not go into details in this survey). We also review game-theoretic characterizations for elementary equivalence in these logics, and show how the games can be used to prove expressivity bounds.

We use the standard definitions for formulas and semantics of first-order logic $\mathcal{FO}$ (see e.g. [13]). Equality is treated as a special relation symbol which is not a member of the signature. The *quantifier rank* of a formula φ is defined to be the depth of quantifier nesting in φ, and is denoted by $qr(\varphi)$.

The rules of the *first-order Ehrenfeucht-Fraïssé game* are as follows. There are two players, *the spoiler* and *the duplicator*. Two σ-structures $\mathcal{A}$ and $\mathcal{B}$ and the number of rounds, say n, are given. In each round the spoiler first selects a point of one of the structures and the duplicator selects a point of the other structure. Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be the points selected after the last round from $\mathcal{A}$ and $\mathcal{B}$, respectively. The duplicator is declared the winner if $\{(a_i, b_i) \mid 1 \leq i \leq n\}$ is a partial isomorphism $\mathcal{A} \to \mathcal{B}$; otherwise the spoiler wins. We say that a player has *a winning strategy* if he can guarantee a win, no matter how the other player plays. This game is interesting because of the following result.

Theorem 3.1 (see **[13]**) *Let $\mathcal{A}, \mathcal{B} \in$ STRUCT$[\sigma]$. Then the duplicator has a winning strategy in the n-round Ehrenfeucht-Fraïssé game if and only if $\mathcal{A}$ and $\mathcal{B}$ agree on all first-order sentences of quantifier rank up to n.* □

We use the notation $\mathcal{A} \equiv_{\mathcal{FO}}^n \mathcal{B}$ if the duplicator has a winning strategy in the n-round game on $\mathcal{A}$ and $\mathcal{B}$. The above theorem can be used to provide the following tool for proving expressivity bounds.

Corollary 3.2 (see **[13]**) *A class $\mathcal{C} \subseteq$ STRUCT$[\sigma]$ is not definable in first-order logic if and only if for every n there are σ-structures $\mathcal{A} \in \mathcal{C}$ and $\mathcal{B} \notin \mathcal{C}$ such that the duplicator has a winning strategy in the n-round Ehrenfeucht-Fraïssé game on $\mathcal{A}$ and $\mathcal{B}$.* □

In other words, if for every n we can find one σ-structure from the class $\mathcal{C}$ and another from the complement of $\mathcal{C}$, such that the duplicator can maintain a partial isomorphism for n rounds, then $\mathcal{C}$ is not definable in $\mathcal{FO}$.

We now give some examples. Note that in those examples, we do not spell out every single detail of the game argument – this may require more space than this entire section. We shall offer much simpler proofs of the existence of winning strategies in the next section, after we have introduced the ideas of locality.

First, one can use Ehrenfeucht-Fraïssé games to show that connectivity of finite graphs cannot be expressed in first-order logic. Assume that connectivity is definable, and take as a counterexample, for each finite n, $\mathcal{A}$ to be a chordless cycle of length 2^n and $\mathcal{B}$ to be a disjoint union of two chordless cycles of length 2^n. The duplicator's strategy in the n-round Ehrenfeucht-Fraïssé game is to preserve, in each round $j \leq n$, distances up to 2^{n-j}. The only way the spoiler could win is to show that in $\mathcal{B}$ there are two points with no path between them, whereas in $\mathcal{A}$ there always is a path between any two points. But if the spoiler cannot build a path between two points in n rounds, it does not matter how far these points are, or whether there is a path between them at all. This informal reasoning can be formalized to show that the duplicator has a winning strategy in the n-round Ehrenfeucht-Fraïssé game over $\mathcal{A}$ and $\mathcal{B}$. This result holds also for ordered structures, see [13]. Note that this shows that the transitive closure query *TRCL* cannot be defined in $\mathcal{FO}$. Indeed, assume that a first-order formula $\psi(x, y)$ defines *TRCL*. Then $\forall x \forall y \psi(x, y)$ would be a first-order sentence defining connectivity.

Next, assume that we are given two distinguished points a and b of a graph $\mathcal{A}$. Then we cannot define, in first-order logic, the property that a and b have equally many neighbors. As a counterexample we can take, for every n, a to be a point which has $n + 2$ neighbors and b to be a point with $n + 1$ neighbors in some graph $\mathcal{A}$. Since there are only n rounds available, the spoiler cannot demonstrate that b has fewer neighbors.

3.1 Unary quantifiers

A commonly used way to increase the expressive power of first-order logic is to extend it with *generalized quantifiers*, cf. [31, 46]. The basic idea is that we are given a class of structures, and we can check whether a substructure defined by a given family of formulas belongs to this class. In computational complexity theory, generalized quantifiers are often considered as oracles. In this section, however, we concentrate more on logical aspects of generalized quantifiers. We now review this method in detail in the case of *unary* generalized quantifiers.

Let σ_k^{unary} be a signature of k unary symbols. Suppose $\mathcal{K}$ is an isomorphism closed class of σ_k^{unary}-structures. Then $\mathcal{FO}(Q_\mathcal{K})$ is the extension of $\mathcal{FO}$ by a new formula formation rule:

> if $\psi_i(x_i, \vec{y}_i)$ is a formula of $\mathcal{FO}(Q_\mathcal{K})$ for $i = 1, \ldots, k$, then
> $Q_\mathcal{K} x_1, \ldots, x_k(\psi_1(x_1, \vec{y}_1), \ldots, \psi_k(x_k, \vec{y}_k))$ is a formula of $\mathcal{FO}(Q_\mathcal{K})$.

The corresponding semantic rule is:

$\mathcal{A} \models Q_{\mathcal{K}} x_1 \ldots x_k(\psi_1(x_1, \vec{a}_1), \ldots, \psi_k(x_k, \vec{a}_k))$ iff $(A, \psi_1[\mathcal{A}, \vec{a}_1], \ldots, \psi_k[\mathcal{A}, \vec{a}_k]) \in \mathcal{K}$, where $\psi_i[\mathcal{A}, \vec{a}_i] = \{a \in A \mid \mathcal{A} \models \psi_i(a, \vec{a}_i)\}$.

Here the tuple $\vec{a}_i$ gives the interpretation for those free variables in $\psi_i(x_i, \vec{y}_i)$ which are not equal to x_i. The extension $\mathcal{FO}(\mathbf{Q})$ by a set $\mathbf{Q}$ of unary quantifiers is defined similarly by adding to $\mathcal{FO}$ the above rule for each $Q_{\mathcal{K}} \in \mathbf{Q}$. The quantifier rank of a formula of $\mathcal{FO}(\mathbf{Q})$ is defined as usually by the maximum depth of nesting of quantifiers (counting both first-order and generalized quantifiers). We also write $\mathcal{FO}(\mathbf{Q}_u)$ for $\mathcal{FO}$ extended by *all* (continuum many) unary quantifiers. Note that the same definition can be used with other ambient logics, not just $\mathcal{FO}$. In particular, we shall use $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)$, the infinitary logic extended with unary generalized quantifiers.

We now list some well-known examples.

Example 3.3

(1) The existential quantifier $\exists$ corresponds to the class of structures $\{(A, P) \mid \emptyset \neq P \subseteq A\}$. Similarly, the universal quantifier $\forall$ can be identified with the unary quantifier which is defined by the class $\{(A, P) \mid P = A\}$.

(2) Counting quantifier $\exists^{\geq k}$ can be defined by the class $\{(A, P) \mid card(P) \geq k\}$. Note that each $\exists^{\geq k}$ can be defined in first-order logic; however, this requires k quantifiers $\exists$, and increases the quantifier rank by k. In contrast, the counting quantifier $\exists^{\geq k}$ increases the quantifier rank by one. We denote the set of all counting quantifiers $\exists^{\geq k}$ by $\mathbf{C}$, i.e., $\mathbf{C} = \{\exists^{\geq k} \mid k \geq 1\}$.

(3) Counting modulo m quantifier D_m is defined by the class $\{(A, P) \mid card(P) \equiv 0 \pmod m\}$. An easy first-order Ehrenfeucht-Fraïssé game argument shows that D_m is not definable in $\mathcal{FO}$ whenever $m > 1$. Thus $\mathcal{FO}(D_m)$ is strictly more expressive than $\mathcal{FO}$. For instance, the sentence $D_2 x(x = x)$ of $\mathcal{FO}(D_2)$ says that the number of points in a structure is even; it is well-known that this property is not definable in first-order logic alone.

(4) Majority quantifier MAJ, which is defined by the class $\{(A, P) \mid card(P) \geq \frac{1}{2} card(A)\}$, is not definable in $\mathcal{FO}$, either. For example, in $\mathcal{FO}(\text{MAJ})$ we can say that there is a node in a graph $\mathcal{A}$ that is connected to at least half of the nodes of $\mathcal{A}$: the defining sentence is $\exists y \text{MAJ} x E(y, x)$. This quantifier is also interesting in connection with capturing complexity classes, see Section 6.

(5) Extending $\mathcal{FO}$ with *Rescher* (bigger cardinality) or *Härtig* (equicardinality) quantifiers also increases the expressive power. Rescher quantifier R is defined by the class $\{(A, P, S) \mid card(P) \leq card(S)\}$ and Härtig quantifier H by the class $\{(A, P, S) \mid card(P) = card(S)\}$. Thus, for instance, given two points a and b in an undirected graph, $Rx, y(E(a, x), E(b, y))$ says that a has at most as many neighbors as b. Similarly, $Hx, y(E(a, x), E(b, y))$ says that a and b have equally many neighbors.

For each $Q_{\mathcal{K}}$, there is a natural Ehrenfeucht-Fraïssé style game-theoretic characterization for elementary equivalence in $\mathcal{FO}(Q_{\mathcal{K}})$. We formulate the rules of this game for the counting modulo m quantifier. The rules of the game are as for first-order Ehrenfeucht-Fraïssé game except that now the spoiler may also choose a subset from one of the structures, say $X \subseteq A$. The duplicator has respond by choosing a subset of the other structure, $Y \subseteq B$, which has modulo m equal cardinality to the spoiler's choice, that is, $card(Y) \equiv card(X) \pmod m$. The spoiler then challenges the duplicator's choice by selecting a point from the duplicator's structure, $b \in B$, and the duplicator has to choose a point from the other structure, $a \in A$, such that $a \in X$ if and only if $b \in Y$. Again, if $a_1, \ldots, a_n$ are the points chosen from $\mathcal{A}$ and $b_1, \ldots, b_n$ are the points chosen from $\mathcal{B}$ during n rounds, the duplicator wins if and only if $\{(a_i, b_i) \mid 1 \leq i \leq n\}$ is a partial isomorphism $\mathcal{A} \to \mathcal{B}$. We call this game the *counting modulo m Ehrenfeucht-Fraïssé* game.

Note that the first-order Ehrenfeucht-Fraïssé game can be seen as a special case of the counting modulo m Ehrenfeucht-Fraïssé game: the spoiler can choose the empty subset from one of the structures, and, in order to win, the duplicator has to respond with the empty set. The spoiler and the duplicator then choose their points from the complement of the empty set, that is, without any restrictions, just as in the regular Ehrenfeucht-Fraïssé game.

The following theorem shows that the counting modulo m Ehrenfeucht-Fraïssé game indeed gives us the game theoretical characterization we were looking for. We use the notation $\mathcal{A} \equiv^n_{\mathcal{FO}(D_m)} \mathcal{B}$ when the duplicator has a winning strategy in the n-round counting modulo m Ehrenfeucht-Fraïssé game.

Theorem 3.4 (see **[31]**) *Let $\mathcal{A}, \mathcal{B} \in$ STRUCT$[\sigma]$. Then $\mathcal{A} \equiv^n_{\mathcal{FO}(D_m)} \mathcal{B}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ agree on all $\mathcal{FO}(D_m)$ sentences of quantifier rank up to n.* □

Corollary 3.5 *A class $\mathcal{C} \subseteq$ STRUCT$[\sigma]$ is not definable in $\mathcal{FO}(D_m)$ if and only if for every n there are σ-structures $\mathcal{A} \in \mathcal{C}$ and $\mathcal{B} \notin \mathcal{C}$ such that the duplicator has a winning strategy in the n-round counting modulo m Ehrenfeucht-Fraïssé game on $\mathcal{A}$ and $\mathcal{B}$.*

The intuition behind this theorem is that if all subsets of $\mathcal{A}$ and $\mathcal{B}$ that the spoiler and the duplicator can use in the game look similar, and there are modulo m equally many of them, then $\mathcal{FO}(D_m)$ can distinguish between $\mathcal{A}$ and $\mathcal{B}$ no more than $\mathcal{FO}$ can. In the next section we give a precise formulation for this intuition.

Theorem 3.4 can be used to show limits of expressive bounds of counting modulo quantifiers. For instance, we can show that connectivity of finite graphs is not definable in $\mathcal{FO}(D_m)$, for any m. The construction is similar to the first-order case, but we also have to require that there are modulo m equally many points in both structures. Then we can proceed very much like in the proof for first-order. Similarly, we can show that the majority quantifier MAJ (or Rescher or Härtig quantifiers) cannot be defined in $\mathcal{FO}(D_m)$. To see this, we can, for instance, take, for each n, $\mathcal{A} = (A, U)$ to be a set with a unary relation U such that $card(A) = 3nm$ and $card(U) = 2nm$. Take $\mathcal{B} = (B, U)$ which satisfies $card(B) = 3nm$ but $card(U) = nm$. It is not difficult to show that the duplicator has a winning strategy in the n-round counting modulo m Ehrenfeucht-Fraïssé game over $\mathcal{A}$ and $\mathcal{B}$. But obviously $\mathcal{A} \models \mathrm{MAJ}xU(x)$ and $\mathcal{B} \not\models \mathrm{MAJ}xU(x)$. These results can be extended to the ordered case, see [39].

If we want to give a game-theoretic method to prove expressive bounds for first-order logic with *all* unary quantifiers, different techniques must be used. The method we employ here is based on *bijective Ehrenfeucht-Fraïssé* games. The rules of the game are the following. As before, the players are the spoiler and the duplicator. There are two σ-structures $\mathcal{A}$ and $\mathcal{B}$ and the number of rounds, say n, given. In each round i the duplicator first selects a bijection $f_i : A \to B$ (if $card(A) \neq card(B)$, then the duplicator loses), and then the spoiler selects a point $a_i \in A$. The duplicator has to select the point $f_i(a_i)$ from $\mathcal{B}$. This continues for n rounds. After the last round, the duplicator is declared the winner if and only if $\{(a_i, f_i(a_i)) \mid 1 \leq i \leq n\}$ is a partial isomorphism from $\mathcal{A}$ to $\mathcal{B}$. We use the notation $\mathcal{A} \equiv^n_{bij} \mathcal{B}$ if the duplicator has a winning strategy in the n-move bijective game on $\mathcal{A}$ and $\mathcal{B}$.

It turns out that this game characterizes elementary equivalence in a logic that is stronger than $\mathcal{FO}(\mathbf{Q}_u)$. This logic is obtained from $\mathcal{FO}(\mathbf{Q}_u)$ by allowing infinite disjunctions and conjunctions, but by keeping quantifier rank bounded. More precisely, let $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^k$ be the extension of $\mathcal{FO}(\mathbf{Q}_u)$ where infinite disjunctions and conjunctions are allowed but quantifier rank of each formula is at most k. The union of all these logics $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^k$ over all natural numbers $k < \omega$ is denoted by $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ (that is, the depth of nesting of quantifiers in each formula is finite). Methods used in [22] give us the following result (a proof can be found in [25]).

Theorem 3.6 (see **[22, 25]**) *Let $\mathcal{A}, \mathcal{B} \in \mathrm{STRUCT}[\sigma]$. Then $\mathcal{A} \equiv^n_{bij} \mathcal{B}$ if and only if $\mathcal{A}$ and $\mathcal{B}$ agree on all $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ sentences of quantifier rank up to n.* □

Corollary 3.7 *A class $\mathcal{C} \subseteq \mathrm{STRUCT}[\sigma]$ is not definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ if and only if for every n there are $\mathcal{A} \in \mathcal{C}$ and $\mathcal{B} \notin \mathcal{C}$ such that the duplicator has a winning strategy in the n-round bijective Ehrenfeucht-Fraïssé game on $\mathcal{A}$ and $\mathcal{B}$.* □

Note that the expressive power of $\mathcal{FO}(\mathbf{Q}_u)$ is strictly weaker than that of $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$. In $\mathcal{FO}(\mathbf{Q}_u)$, it is not possible to express the second vectorization of Härtig quantifier (that is, the equicardinality quantifier for pairs) [37], while techniques used in [31] show that every vectorization of every unary quantifier can be defined in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$. It also follows from [31] that $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ is as strong in expressive power as $\mathcal{L}_{\infty\omega}(\mathbf{C})^\omega$. (In [31], this was shown for finite-variable logics, but the same proof technique works for $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ [22].)

Although $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ has strong counting power, the game characterization can be used to show that connectivity and transitive closure are not definable in it. The idea is the same as before: For each n, we can take $\mathcal{A}$ to be a chordless cycle of length 2^{n+1} and $\mathcal{B}$ a disjoint union of two chordless cycles of length 2^n. Now the duplicator's strategy is to choose in round j a bijection that preserves the distances up to 2^{n-j} between the points next to the already chosen ones. A combinatorial proof can be given that shows the existence of such a strategy [27]; however, we shall see a much easier way to establish this in the next section.

We remark that bijective games are useless in the presence of a linear order. In order to win, the duplicator has to follow the linear order when he chooses bijections (otherwise the spoiler wins in the next two rounds). Thus, there is essentially only one bijection the duplicator can choose. This gives us an example of difficulties that arise when one attempts to prove expressivity bounds in the ordered setting. In the subsequent sections, we shall face similar problems several times.

Another interesting counting logic is $\mathcal{FO} + COUNT$, cf. [15]. This is a two-sorted logic with the second sort being the sort of natural numbers. More precisely, in this approach a structure is of the form

$$\mathcal{A} = (\{v_1, \ldots, v_n\}, \{1, \ldots, n\}, \overline{R}_1, \ldots, \overline{R}_l, \leq, BIT, 1, \max).$$

Here relations R_i apply to the non-numerical domain $\{v_1, \ldots, v_n\}$, while the linear order $\leq$, the BIT predicate and the constants 1 and max (interpreted as 1 and n) refer to the numerical domain $\{1, \ldots, n\}$. Here $BIT(i, j)$ holds if and only if the ith bit in the binary representation of j is one. These two disjoint domains are connected by allowing formulas of the form $\exists ix\varphi(x)$ with the semantics that φ has at least i satisfiers, i.e., $card(\{a \in A \mid \mathcal{A} \models \varphi(a)\}) \geq i$. Here i refers to the numerical domain and x refers to the non-numerical domain; the quantifier $\exists ix$ binds x but not i. As an example, consider the sentence

$$\exists i \exists j[(j + j = i) \wedge \exists ix\varphi(x) \wedge \forall k(\exists kx\varphi(x) \rightarrow k \leq i)].$$

This sentence tests if the cardinality of $\{a \mid \varphi(a)\}$ is even. Indeed, $\exists ix\varphi(x) \wedge \forall k(\exists kx\varphi(x) \rightarrow k \leq i)$ holds iff φ has exactly i satisfiers, and i is even since it is of the form $2j$. Note that we used the fact that $+$ and $*$ are definable as ternary predicates in the presence of the BIT relation, cf. [15].

Remark 3.8 Hella [24] made the following interesting observation. While first-order logic extended by Rescher quantifier R is in general strictly weaker than $\mathcal{FO} + COUNT$, in the presence of a built-out linear order, $\mathcal{FO}(R)$ and $\mathcal{FO} + COUNT$ have the same expressive power. Clearly, $Rx, y(\varphi(x), \psi(y))$ can be written in $\mathcal{FO} + COUNT$. On the other hand, $\exists ix\varphi(x)$ can be expressed by $Rj, x(j < i, \varphi(x))$.

A game-theoretic characterization for elementary equivalence in $\mathcal{FO} + COUNT$ was introduced in [30] and used subsequently in [14]. However, we do not go into detail here, mainly due to the fact that the counting games of [30] are subsumed by the bijective games. The logic $\mathcal{FO} + COUNT$ has a number of applications in computer science, in particular, in complexity theory. This will be discussed in Section 6.

Finally, we refer the reader to [23, 46] for a more detailed overview of results on generalized quantifiers in finite-model theory.

4 Gaifman's and Hanf's conditions

The game-theoretic characterizations for elementary equivalence of logics considered in the previous section gave us a vague intuition that these logics can only express local properties. This intuition will be formalized in this section. We review theorems by Gaifman and Hanf, and their modifications.

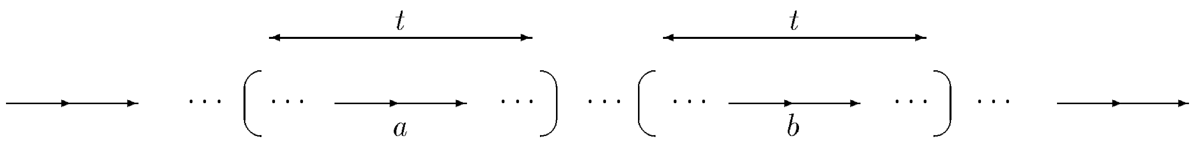

Figure 1: Formula $\varphi(x, y)$ cannot distinguish (a, b) from (b, a).

4.1 Gaifman's theorem

We start with Gaifman's theorem [18]. Let $\mathcal{A}$ be a σ-structure. Recall that the distance $d(a, b)$ is the length of the shortest path from a to b in the Gaifman graph of $\mathcal{A}$, and $S_r(\vec{a}) = \{b \mid d(b, a) \leq r, a \in \vec{a}\}$. For each fixed k, there are first-order formulas that define the relations $d(a, b) > k$, $d(a, b) = k$ and $d(a, b) < k$ (see [13]). Hence, bounded quantifications of the form $\forall x \in S_k(\vec{y})$ and $\exists x \in S_k(\vec{y})$ are expressible in first-order logic for every fixed k. A formula $\varphi^{(r)}(\vec{y})$ is called *r-local around $\vec{y}$* if every quantifier in it is of the form $\forall x \in S_k(\vec{y})$ or $\exists x \in S_k(\vec{y})$ with $k \leq r$. A sentence ψ is called *basic r-local* if it is of the form

$$\exists x_1 \ldots \exists x_m (\bigwedge_{1 \leq i \leq m} \varphi^{(r)}(x_i) \wedge \bigwedge_{1 \leq i < j \leq m} d(x_i, x_j) > 2r)$$

where $\varphi^{(r)}(x)$ is an r-local formula around x.

Theorem 4.1 ([18]) *Every first-order sentence is equivalent to a Boolean combination of basic r-local sentences, and every first-order formula $\varphi(x_1, \ldots, x_n)$ is equivalent to a Boolean combination of t-local formulas around $x_1, \ldots, x_n$ and basic r-local sentences. Furthermore, $r \leq 7^{qr(\varphi)-1}$, $t \leq (7^{qr(\varphi)-1})/2$ and $m \leq n + qr(\varphi)$.* □

Gaifman's theorem tells us that every first-order formula can see only a bounded number of local neighborhoods in a structure, i.e., only a small part of the input. This is indeed a formalization of the informal statement that first-order logic can express only local properties.

Example 4.2 We show that first-order logic cannot express the transitive closure of a directed graph. Assume, to the contrary, that there is a first-order formula $\varphi(x, y)$ in the language of a single binary relation E, such that $\varphi[\mathcal{A}] = (A, \{(a, b) \in A \mid \mathcal{A} \models \varphi(a, b)\})$ is the transitive closure of $\mathcal{A}$. Apply Gaifman's theorem to it, and find t and r. Now consider the graph shown in Figure 1. It is a successor relation, on which we select two points, a and b. Assume that $d(a, b) > 2t$, and the distances from a and b to the start and the end node of the graph are at least $t + 1$. Then the t-neighborhoods of (a, b) and (b, a) are isomorphic, and by Gaifman's theorem, φ cannot distinguish (a, b) from (b, a). Thus, φ cannot define the transitive closure, since $(a, b) \in TRCL(\mathcal{A})$, but $(b, a) \notin TRCL(\mathcal{A})$.

4.2 Hanf's theorem and its modifications

While Gaifman's theorem helps prove expressivity bounds for $\mathcal{FO}$ directly, without resorting to establishing a winning strategy for the duplicator[1], Hanf's theorem [21] and its numerous modifications [17, 25, 38, 39] provide criteria for the existence of a strategy for the duplicator that is based on counting of small neighborhoods in two structures.

Hanf's theorem was originally proved for infinite structures. It was observed by Fagin, Stockmeyer and Vardi [17] that the technique can be modified to be applicable to finite structures. The extensions of Hanf's technique [25, 38, 39] follow the ideas of [17].

Let $\mathcal{A}$ be a σ-structure and $a \in A$. Recall that the isomorphism type of $N_d^{\mathcal{A}}(a)$ is denoted by $tp_d^{\mathcal{A}}(a)$. Let τ be an isomorphism type of a σ_1-structure (σ extended with one constant). We denote the number of points $a \in A$ whose d-neighborhoods realize τ by $n_d(\mathcal{A}, \tau)$. That is,

$$n_d(\mathcal{A}, \tau) \quad = \quad card(\{a \in A \mid tp_d^{\mathcal{A}}(a) = \tau\}).$$

For example, if $\mathcal{A}$ is a chordless undirected cycle of length at least $2d+2$ then there is only one isomorphism type τ of a d-neighborhood of a point occurring in $\mathcal{A}$: the chordless path of length $2d+1$. In this case $n_d(\mathcal{A}, \tau) = card(A)$.

We call structures $\mathcal{A}$ and $\mathcal{B}$ (d,m)*-equivalent* if for every isomorphism type τ they have exactly the same number of points whose d-neighborhoods realize τ, or both structures have at least m such points, that is,

$$\min(n_d(\mathcal{A}, \tau), m) \quad = \quad \min(n_d(\mathcal{B}, \tau), m).$$

The modification of Hanf's theorem for the finite case is the following.

Theorem 4.3 ([17]) *Let n and f be positive integers. There are positive integers d and m such that whenever $\mathcal{A}$ and $\mathcal{B}$ are (d,m)-equivalent structures where every point has degree at most f, then $\mathcal{A} \equiv_{\mathcal{FO}}^n \mathcal{B}$, that is, $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of $\mathcal{FO}$ of quantifier rank up to n.* □

Note that since we consider finite models, for any $\mathcal{A}$ there is a number f that exceeds degrees of all points of $\mathcal{A}$. This leads to the following. We say that $\mathcal{A}$ and $\mathcal{B}$ are *d-equivalent*, written as $\mathcal{A} \leftrightarrows_d \mathcal{B}$, if for every type of a d-neighborhood of a point, τ, equally many points realize it in $\mathcal{A}$ and $\mathcal{B}$. That is,

$$\mathcal{A} \leftrightarrows_d \mathcal{B} \quad \text{iff} \quad n_d(\mathcal{A}, \tau) = n_d(\mathcal{B}, \tau) \;\; \text{for every} \;\; \tau.$$

Corollary 4.4 ([17]) *Let n be a positive integer. There there exists a positive integer d such that $\mathcal{A} \leftrightarrows_d \mathcal{B}$ implies $\mathcal{A} \equiv_{\mathcal{FO}}^n \mathcal{B}$.* □

[1] New winning conditions for the duplicator based on Gaifman's theorem were presented recently in [45].

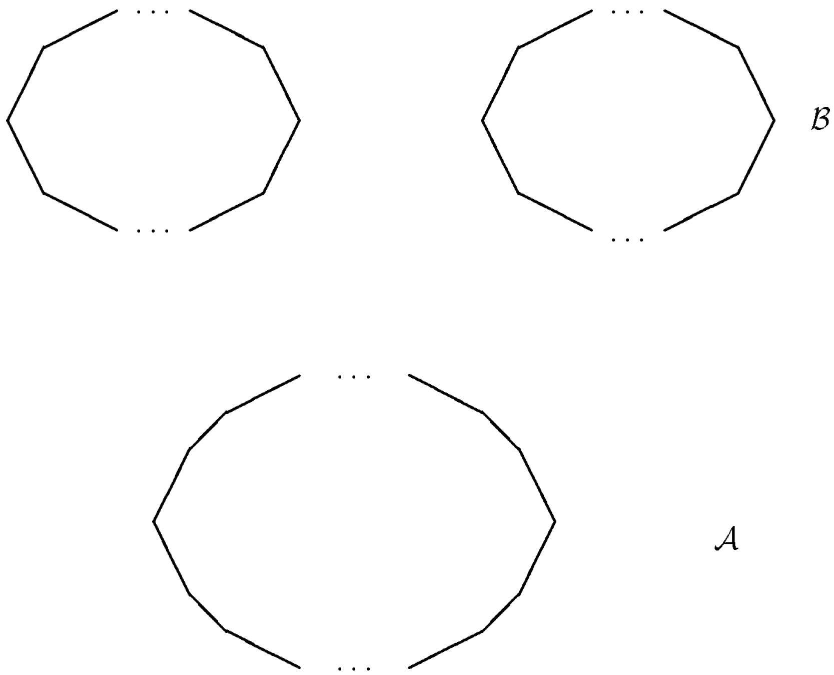

Figure 2: Hanf's technique proves that connectivity is not first-order

This result makes precise the intuition that counting power of first-order logic is rather limited. It also shows that only local neighborhoods are relevant for elementary equivalence in $\mathcal{FO}$. Most importantly, the result above yields much simpler proofs of expressivity bounds that those based on games. Below we give a canonical example of applicability of Hanf's technique.

Example 4.5 We show that connectivity of finite graphs is not definable in $\mathcal{FO}$. Assume, to the contrary, that it is definable by a $\mathcal{FO}$ sentence Φ of quantifier rank n. Apply Corollary 4.4 to find $d > 0$ such that $\mathcal{A} \leftrightarrows_d \mathcal{B}$ would imply $\mathcal{A} \models \Phi$ iff $\mathcal{B} \models \Phi$. Now let $\mathcal{A}$ be a (chordless) cycle which has length $4d + 4$, and let $\mathcal{B}$ be a disjoint union of two chordless cycles of length $2d + 2$, see Figure 2.

As we noticed before, there is only one type of a d-neighborhood that these graphs realize, namely a chain on $2d + 1$ points. Thus, $\mathcal{A} \leftrightarrows_d \mathcal{B}$, since they have the same number of vertices. At the same time, $\mathcal{A}$ is connected, but $\mathcal{B}$ is not, proving that connectivity cannot be expressed by Φ.

Other examples, proved previously with games, can be shown to be derivable from Hanf's technique. For example, with the concept of (d, m)-equivalence, it is now easy to show that the majority quantifier (or Rescher and Härtig quantifiers) are not definable in first-order logic. $\Box$

Before we outline the proof of Theorem 4.3 in [17], we state Hanf's theorem from [21]. It says that two arbitrary structures $\mathcal{A}$ and $\mathcal{B}$ are elementary equivalent if $S_r^{\mathcal{A}}(a)$ and $S_r^{\mathcal{B}}(b)$ are finite, for every finite r and every $a \in A$ and $b \in B$, and, for each $r > 0$ and each type τ of an

r-neighborhood of a point, either $n_r(\mathcal{A}, \tau) = n_r(\mathcal{B}, \tau) < \omega$, or both $\mathcal{A}$ and $\mathcal{B}$ have infinitely many realizers of type τ.

Now we explain how Theorem 4.3 is proved. First, d is taken to be $\frac{3^n-1}{2}$, and m is taken to exceed the size of any d-neighborhood of a point in a structure whose degrees are bounded by f. Define $d_j = \frac{3^{n+1-j}-1}{2}$ for $j \leq n$; in particular, $d_j = 3 \cdot d_{j+1} + 1$, $d_1 = d$ and $d_n = 1$. It is then shown that the duplicator can play in the first-order Ehrenfeucht-Fraïssé game in such a way that after each round $j \leq n$, if $\vec{a} \in A^j$ and $\vec{b} \in B^j$ are the points chosen so far, then $tp^{\mathcal{A}}_{d_j}(\vec{a}) = tp^{\mathcal{B}}_{d_j}(\vec{b})$. This suffices, since $tp^{\mathcal{A}}_1(a_1, \ldots, a_n) = tp^{\mathcal{B}}_1(b_1, \ldots, b_n)$ implies that the mapping $a_i \mapsto b_i$ is a partial isomorphism. The condition that $\mathcal{A}$ and $\mathcal{B}$ are (d, m)-equivalent allows the duplicator to make the first move by choosing a point with a given type of its d-neighborhood. Suppose that the duplicator maintained the condition successfully for the first j rounds. That is, we have $tp^{\mathcal{A}}_{d_j}(\vec{a}) = tp^{\mathcal{B}}_{d_j}(\vec{b})$. Assume that $a \in A$ is spoiler's next choice in the game. If $a \in S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a})$, then the duplicator selects the point $b \in S^{\mathcal{B}}_{2d_{j+1}+1}$ which is given by the isomorphism between $N^{\mathcal{A}}_{d_j}(\vec{a})$ and $N^{\mathcal{B}}_{d_j}(\vec{b})$. Then one easily checks that $tp^{\mathcal{A}}_{d_{j+1}}(\vec{a}a) = tp^{\mathcal{B}}_{d_{j+1}}(\vec{b}b)$. If $a \notin S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a})$ then the duplicator can choose any $b \notin S^{\mathcal{B}}_{2d_{j+1}+1}(\vec{b})$ such that $tp^{\mathcal{A}}_{d_{j+1}}(a) = tp^{\mathcal{B}}_{d_{j+1}}(b)$. If m is chosen as above, such a point b always exists. Then again we can see that $tp^{\mathcal{A}}_{d_{j+1}}(\vec{a}a) = tp^{\mathcal{B}}_{d_{j+1}}(\vec{b}b)$ holds, since no points in $S^{\mathcal{A}}_{d_{j+1}}(\vec{a})$ and $S^{\mathcal{A}}_{d_{j+1}}(a)$ belong to the same tuple of an $\mathcal{A}$-relation, and likewise for $\mathcal{B}$.

Having established Hanf's condition for $\mathcal{FO}$, we turn to counting extensions of $\mathcal{FO}$. In [39], the proof of Theorem 4.3 was modified for $\mathcal{FO}(D_k)$, $k > 1$. Two structures $\mathcal{A}$ and $\mathcal{B}$ are called (d, m, D_k)*-equivalent*, if for each isomorphism type τ they have the same number of points whose d-neighborhoods realize τ, or in both structures there are at least m such points but modulo k equally many, that is,

$$\min(n_d(\mathcal{A}, \tau), m) = \min(n_d(\mathcal{B}, \tau), m) \quad \text{and} \quad n_d(\mathcal{A}, \tau) \equiv n_d(\mathcal{B}, \tau) \pmod{k}.$$

Note that (d, m, D_1)-equivalence is just the (d, m)-equivalence. Thus, the following result extends Theorem 4.3.

Theorem 4.6 ([39]) *Let n, f and k be positive integers. There are positive integers d and m such that whenever $\mathcal{A}$ and $\mathcal{B}$ are (d, m, D_k)-equivalent structures where every point has degree at most f, then $\mathcal{A} \equiv^n_{\mathcal{FO}(D_k)} \mathcal{B}$. That is, $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of $\mathcal{FO}(D_k)$ of quantifier rank up to n.* □

The intuition behind this theorem is that the counting power of $\mathcal{FO}(D_k)$ is rather limited, and that $\mathcal{FO}(D_k)$ is not much stronger than $\mathcal{FO}$. For instance, we can use Theorem 4.6 to show that connectivity is not definable in $\mathcal{FO}(D_k)$ for any k, by using the same argument as in Example 4.5.

The proof of Theorem 4.6 in [39] follows the same idea as the proof of Theorem 4.3. One could ask whether the proof technique can be easily modified to prove elementary equivalence for $\mathcal{FO}(Q)$, where Q is an arbitrary unary quantifier. It turns out that in this technique, the additivity property of the quantifier Q is crucial, and unary quantifiers which satisfy this property are essentially counting modulo quantifiers. However, we can show a more general result that describes a winning condition in a *bijective* game, and thus covers all unary quantifiers.

Theorem 4.7 ([25, 38]) *Let n be a positive integer. There is a positive integer d such that $\mathcal{A} \leftrightarrows_d \mathcal{B}$ implies $\mathcal{A} \equiv^n_{bij} \mathcal{B}$. In particular, if $\mathcal{A} \leftrightarrows_d \mathcal{B}$, then $\mathcal{A}$ and $\mathcal{B}$ satisfy the same sentences of $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)$ of quantifier rank up to n.* □

Before we explain how this result is proved, we give the following alternative definition of d-equivalence. Two structures $\mathcal{A}$ and $\mathcal{B}$ are d-equivalent if there exists a bijection $f : A \to B$ such that for every $a \in A$,

$$tp^{\mathcal{A}}_d(a) \quad = \quad tp^{\mathcal{B}}_d(f(a)).$$

Now the proof of Theorem 4.7 is very similar to the proof of Hanf's theorem for first-order logic. We again let d_j be $\frac{3^{n+1-j}-1}{2}$, and take d to be d_1. The duplicator's strategy is to play so that after each round j in the bijective Ehrenfeucht-Fraïssé game we have $tp^{\mathcal{A}}_{d_j}(\vec{a}) = tp^{\mathcal{B}}_{d_j}(\vec{b})$, if $\vec{a} \in A^j$ and $\vec{b} \in B^j$ are chosen during those j rounds. The first round bijection is given by the reformulation of d-equivalence above. Assume j rounds have been played, and we have $tp^{\mathcal{A}}_{d_j}(\vec{a}) = tp^{\mathcal{B}}_{d_j}(\vec{b})$, where $d_j = 3 \cdot d_{j+1} + 1$. Assume that $tp^{\mathcal{A}}_{d_{j+1}}(a) = \tau$, for $a \in S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a})$. Then $tp^{\mathcal{B}}_{d_{j+1}}(h(a)) = \tau$, and $h(a) \in S^{\mathcal{B}}_{2d_{j+1}+1}(\vec{b})$, where $h : N^{\mathcal{A}}_{d_j}(\vec{a}) \to N^{\mathcal{B}}_{d_j}(\vec{b})$ is an isomorphism. Thus, the number of points realizing τ is the same in complements of $S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a})$ and $S^{\mathcal{B}}_{2d_{j+1}+1}(\vec{b})$, and hence we have a bijection $g : S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a}) \to S^{\mathcal{B}}_{2d_{j+1}+1}(\vec{b})$ with the property $tp^{\mathcal{A}}_{d_{j+1}}(x) = tp^{\mathcal{B}}_{d_{j+1}}(g(x))$. We now define a bijection f_{j+1} for the round $j+1$ to be h on $S^{\mathcal{A}}_{2d_{j+1}+1}(\vec{a})$, and g on its complement. It is routine to verify that f_{j+1} is a bijection, and that $tp^{\mathcal{A}}_{d_{j+1}}(\vec{a}a) = tp^{\mathcal{B}}_{d_{j+1}}(\vec{b}f_{j+1}(a))$ for every $a \in A$. After the last round we have vectors with isomorphic 1-neighborhoods; hence they define a partial isomorphism.

Theorem 4.7 shows the limits of expressive power provided by all unary quantifiers. It also significantly simplifies proofs of expressivity bounds, as applying bijective games is usually not a very easy task. For instance, we can use Example 4.5 to show that connectivity of finite graphs is not definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$, thus avoiding all the tedious combinatorics involved in an argument based on bijective games.

As another example, we consider classes of undirected graphs which are *closed under stretching*. More precisely, let $\mathcal{A}$ be a graph and let $\mathcal{A}_d$ be the graph obtained by replacing every edge of $\mathcal{A}$ by a path of length $2d+1$. Then each d-neighborhood of a point in $\mathcal{A}_d$ contains at most one node whose degree is greater than two. We say that a class $\mathcal{C}$ of graphs is closed under stretching if $\mathcal{A}_d \in \mathcal{C}$ for every $\mathcal{A} \in \mathcal{C}$ and for every positive integer d. Now it is easy to see that if $\mathcal{A}$ and $\mathcal{B}$ are 1-equivalent graphs, then $\mathcal{A}_d$ and $\mathcal{B}_d$ are d-equivalent. In other words, if there are the same number of points in both graphs $\mathcal{A}$ and $\mathcal{B}$ of each degree, then $\mathcal{A}_d$ and $\mathcal{B}_d$ are d-equivalent. It can then be proved that if a class $\mathcal{C}$ and its complement are closed under stretching, and there are $\mathcal{A} \in \mathcal{C}$ and $\mathcal{B} \notin \mathcal{C}$ as above, then $\mathcal{C}$ is not definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ [26]. This argument shows that graph properties such as planarity and 3-colorability are not definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$.

Hanf's technique was also used by Etessami [15] (although a preliminary conference version [14] had a proof based on counting games of [30]). It was shown in [15] that a linear order cannot be defined in $\mathcal{FO} + COUNT$ from its underlying successor relation. The proof relies on Hanf's technique for $\mathcal{FO} + COUNT$. Its applicability follows from Theorem 4.7, since every $\mathcal{FO} + COUNT$ sentence is definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$.

Summing up, the combinatorial arguments in this section allow us to simplify many proofs that were originally shown by using difficult game-theoretic arguments. Furthermore, we can also prove, often quite easily, new nondefinability results. Note, however, that in the presence of a linear ordering, none of these results can be applied to derive inexpressibility results. This is because in the presence of an order, every point is a neighbor of any other point, and thus $S_1^{\mathcal{A}}(a)$ contains all elements of $\mathcal{A}$. Many results proving expressive bounds of these logics on ordered structures still use games (see [13] for first-order case and [39] for $\mathcal{FO}(D_m)$). However, bijective Ehrenfeucht-Fraïssé games cannot be applied at all in the ordered case. In fact, this is closely connected to some deep problems in circuit complexity, as will be explained in Section 6.

5 Abstract notions of locality

On the surface, Gaifman's theorem and various forms of Hanf's theorem appear to be quite unrelated. Nevertheless, we shall see soon that there is a very close relationship between these results. To make statements like this, we study abstract concepts behind locality theorems. This approach can be seen as the study of the essential ideas behind the proofs of locality theorems, rather than using the statements of the theorems for proving expressivity bounds. From these theorems, we extract abstract notions of locality, and show how they are related. We also discuss a new form of locality, the bounded degree property, and show how it is related to other forms. This property turns out to be particularly simple to use in proving expressivity bounds.

5.1 Gaifman's locality

We start by analyzing Gaifman's theorem. This theorem says that only local neighborhoods are important for elementary equivalence in first-order logic. This is captured by the following definition.

Definition 5.1 ([32, 25]) *A formula $\psi(x_1, \ldots, x_m)$ is* Gaifman-local *if there exists $r > 0$ such that for every $\mathcal{A} \in$ STRUCT$[\sigma]$ and for every two m-ary vectors $\vec{a}, \vec{b} \in A^m$,*

$$tp_r(\vec{a}) = tp_r(\vec{b}) \quad \text{implies} \quad \mathcal{A} \models \psi(\vec{a}) \text{ if and only if } \mathcal{A} \models \psi(\vec{b}).$$

The minimum r for which this holds is called the locality rank *of ψ and is denoted by* $\mathsf{lr}(\psi)$.

This definition formulates that if a logic is Gaifman-local, i.e., every formula of a logic is Gaifman-local, then indeed only small parts of the input are relevant for elementary equivalence in the logic. The part of Gaifman's theorem that deals with sentences, gives rise to the following notion.

Definition 5.2 ([32, 25])
- *A formula $\psi(x_1, \ldots, x_m)$, $m \geq 1$, is* strongly Gaifman-local *if there exists $r > 0$ such that for every $\mathcal{A}, \mathcal{B} \in$* STRUCT$[\sigma]$ *and for every two m-ary vectors $\vec{a} \in A^m$ and $\vec{b} \in B^m$, $tp_r^{\mathcal{A}}(\vec{a}) = tp_r^{\mathcal{B}}(\vec{b})$ implies $\mathcal{A} \models \psi(\vec{a})$ if and only if $\mathcal{B} \models \psi(\vec{b})$.*
- *A sentence Ψ is* strongly Gaifman-local *if it is equivalent to a Boolean combination of sentences of the form $\exists \vec{y} \psi(\vec{y})$, where $\psi(\vec{y})$ is a strongly Gaifman-local formula.*

Before going further we notice that not every first-order formula is *strongly* Gaifman local. Consider the class of directed graphs. Let $\psi(x)$ be the formula $\forall y \neg E(y,x) \wedge \exists z \forall y \neg E(z,y)$. Then $\psi(x)$ says that x has not a predecessor and there is a point which has no a successor. Suppose that ψ is strongly Gaifman-local with locality rank r. Let G_1 be a chain graph of length $r+1$ and let G_2 be a chain graph of the same length with a loop on the end node. Denote the start node of G_i by a_i. Then $tp_r^{G_1}(a_1) = tp_r^{G_2}(a_2)$ but obviously $G_1 \models \psi(a_1)$ whereas $G_2 \not\models \psi(a_2)$.

From Gaifman's theorem, we can derive the following.

Corollary 5.3 ([18, 32, 25]) *Every first-order formula is Gaifman-local and every first-order sentence is strongly Gaifman-local. Moreover, for every $\psi(\vec{x})$, $\mathsf{lr}(\psi) \leq (7^{\mathsf{qr}(\psi)} - 1)/2$.* □

5.2 Hanf's locality

When we studied Hanf's condition and proofs of Theorems 4.3, 4.6 and 4.7 we noticed that the essential part in all the proofs was the concept of d-equivalence. We now consider this concept in more detail and define the notion of Hanf's locality. Before giving the definition of the abstract notion we need more notation and definitions.

We extend the notion of d-equivalence (see Section 4) for structures with parameters. Let $\mathcal{A}, \mathcal{B} \in$ STRUCT$[\sigma]$ and $\vec{a} \in A^m$, $\vec{b} \in B^m$. Then $(\mathcal{A}, \vec{a})$ and $(\mathcal{B}, \vec{b})$ are *d-equivalent*, $(\mathcal{A}, \vec{a}) \leftrightarrows_d (\mathcal{B}, \vec{b})$, if for every isomorphism type τ,

$$n_d((\mathcal{A}, \vec{a}), \tau) \quad = \quad n_d((\mathcal{B}, \vec{b}), \tau).$$

In other words, there are as many points $a \in A$ and $b \in B$ such that $tp_d^{\mathcal{A}}(\vec{a}a) = tp_d^{\mathcal{B}}(\vec{b}b)$; or equivalently, there is a bijection $f : A \to B$ such that $tp_d^{\mathcal{A}}(\vec{a}x) = tp_d^{\mathcal{B}}(\vec{b}f(x))$ for all $x \in A$.

Definition 5.4 ([32, 25]) *A formula $\psi(x_1, \ldots, x_m)$ is* Hanf-local *if there exists a number d such that for every $\mathcal{A}, \mathcal{B} \in$* STRUCT$[\sigma]$ *and for every two m-ary vectors $\vec{a}$ and $\vec{b}$ of elements of $\mathcal{A}$ and $\mathcal{B}$ respectively,*

$$(\mathcal{A}, \vec{a}) \leftrightarrows_d (\mathcal{B}, \vec{b}) \qquad \textit{implies} \qquad \mathcal{A} \models \psi(\vec{a}) \textit{ if and only if } \mathcal{B} \models \psi(\vec{b}).$$

The minimum d for which this holds is called the Hanf locality rank *of ψ, and is denoted by* $\mathsf{hlr}(\psi)$.

Thus, a sentence φ is Hanf-local, is there exists a number d such that $\mathcal{A} \leftrightarrows_d \mathcal{B}$ implies $\mathcal{A} \models \varphi$ iff $\mathcal{B} \models \varphi$. From results of the previous section, the following theorem is immediate.

Theorem 5.5 ([32, 25]) *Every sentence φ of $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$, or $\mathcal{FO}+COUNT$, or $\mathcal{FO}$ is Hanf-local, and $\mathsf{hlr}(\varphi) \leq 3^{\mathsf{qr}(\varphi)}$.* □

This result confirms that although these logics have substantial counting power, they can only recognize properties of small neighborhoods, and cannot grasp a structure as a whole. The definitions above extract the essential notions that were crucial for proving theorems in the previous section. Now these notions allow us to *compare* different locality results.

5.3 Relationship between the notions of locality

The result below is the main technical lemma that establishes the relationship between strong Gaifman-locality, Gaifman-locality and Hanf-locality. It states that d-equivalence of structures extends to d-equivalence of structures with parameters, if large enough neighborhoods of the parameters are isomorphic.

Lemma 5.6 ([25]) *If $\mathcal{A} \leftrightarrows_d \mathcal{B}$ and $tp^{\mathcal{A}}_{3d+1}(\vec{a}) = tp^{\mathcal{B}}_{3d+1}(\vec{b})$, then $(\mathcal{A}, \vec{a}) \leftrightarrows_d (\mathcal{B}, \vec{b})$.* □

Using this, we prove the following.

Theorem 5.7 ([32, 25]) *Every Hanf-local formula is Gaifman-local.*

Proof. Suppose $\psi(x_1, \ldots, x_m)$ is a Hanf-local formula with $\mathsf{hlr}(\psi) = d$. We show that ψ is Gaifman-local. Take any two m-vectors $\vec{a}$ and $\vec{b}$ of a structure $\mathcal{A}$ such that $tp_r(\vec{a}) = tp_r(\vec{b})$, where $r = 3d+1$. Since $\mathcal{A} \leftrightarrows_d \mathcal{A}$, by Lemma 5.6 we obtain $(\mathcal{A}, \vec{a}) \leftrightarrows_d (\mathcal{A}, \vec{b})$. Thus, $\mathcal{A} \models \psi(\vec{a})$ if and only if $\mathcal{A} \models \psi(\vec{b})$. Hence ψ is Gaifman-local and $\mathsf{lr}(\psi) \leq 3d+1$. □

We now consider the relationship between Hanf's locality and strong Gaifman's locality. As a technical tool, we need to extend the notion of d-equivalence to tuples. The number of different m-tuples whose d-neighborhoods realize an isomorphism type τ^m of a σ_m-structure $\mathcal{A}$, is denoted by $n_d(\mathcal{A}, \tau^m)$. We write $\mathcal{A} \leftrightarrows_{m,d} \mathcal{B}$, if for every isomorphism type τ^m,

$$n_d(\mathcal{A}, \tau^m) \quad = \quad n_d(\mathcal{B}, \tau^m).$$

Equivalently, $\mathcal{A} \leftrightarrows_{m,d} \mathcal{B}$ if and only if there is a bijection $f : A^m \to B^m$ such that $tp^{\mathcal{A}}_d(\vec{a}) = tp^{\mathcal{B}}_d(f(\vec{a}))$ for every $\vec{a} \in A^m$. By considering m-tuples whose components are the same, we see that for all $m > 0$, $\mathcal{A} \leftrightarrows_{m,d} \mathcal{B}$ implies $\mathcal{A} \leftrightarrows_d \mathcal{B}$.

Our crucial lemma is that r-equivalence of $(m+1)$-tuples can be guaranteed by d-equivalence of tuples for large enough d that depends only on r. This can be shown by considering parametrized versions of these structures and applying Lemma 5.6.

Proposition 5.8 ([32, 25]) *Let $m > 0$ and $d \geq 0$. Then $\mathcal{A} \leftrightarrows_{m,3d+1} \mathcal{B}$ implies $\mathcal{A} \leftrightarrows_{m+1,d} \mathcal{B}$. In particular, for every $r > 0$ and $m \geq 1$ there is d such that $\mathcal{A} \leftrightarrows_d \mathcal{B}$ implies $\mathcal{A} \leftrightarrows_{m,r} \mathcal{B}$.* □

This can be used to show the following.

Theorem 5.9 ([32, 25]) *Every strongly Gaifman-local sentence is Hanf-local.*

Consider a sentence Ψ which is equivalent to $\exists x_1 \ldots \exists x_m \psi(x_1, \ldots, x_m)$, where $\psi(\vec{x})$ is strongly Gaifman-local. Let $r > 0$ witness strong Gaifman's locality of ψ. Take d given by Proposition 5.8. Then $\mathsf{hlr}(\Psi) \leq d$. Indeed, let $\mathcal{A} \leftrightarrows_d \mathcal{B}$ and $\mathcal{A} \models \Psi$. Then $\mathcal{A} \models \psi(\vec{a})$ for some $\vec{a} \in A^m$. By Proposition 5.8 we know that $\mathcal{A} \leftrightarrows_{m,r} \mathcal{B}$, and thus we find $\vec{b} \in B^m$ such that $tp_r^{\mathcal{A}}(\vec{a}) = tp_r^{\mathcal{B}}(\vec{b})$. Since ψ is strongly Gaifman-local, $\mathcal{B} \models \psi(\vec{b})$ and thus $\mathcal{B} \models \Psi$. Hence, $\mathsf{hlr}(\Psi) \leq d$. □

This implies that the two parts of Gaifman's theorem (those dealing with sentences and open formulas) are not independent. In fact, for any logic satisfying some regularity properties, strong Gaifman-locality of its sentences implies Gaifman-locality of its open formulae. See [25, 32] for details.

5.4 Bounded degree property

One of the easiest ways to prove expressivity bounds is the *bounded degree property*. It was first introduced for graph queries in studying limits of expressive power of database query languages [34]. Later it was generalized to arbitrary (finite) structures in [11]. We now review this concept, show its usefulness in proving expressivity bounds, and relate it to other notions of locality.

For a relation $\overline{R}_i$ in $\mathcal{A}$, we define $degree_j(R_i, a)$ to be the number of tuples in $\overline{R}_i$ whose jth component is a. For directed graphs, this gives us the familiar notions of in- and out-degree. The set

$$\{degree_j(R_i, a) \mid R_i \in \sigma, a \in A, j \leq p_i\}$$

of all degrees realized in $\mathcal{A}$ is denoted by $deg\_set(\mathcal{A})$. We use $deg(\mathcal{A})$ for $card(deg\_set(\mathcal{A}))$. The class of σ-structures $\mathcal{A}$ for which $deg\_set(\mathcal{A}) \subseteq \{0, \ldots, k\}$ is denoted by $\mathrm{STRUCT}_k[\sigma]$.

Informally, a query has the bounded degree property if an upper bound on the degrees in an input structure implies an upper bound on the number of degrees realized in the output structure produced by the query. Recall that the output of $\psi(x_1, \ldots, x_m)$ on $\mathcal{A}$ is the structure with one m-ary relation $(A, \{\vec{a} \in A^m \mid \mathcal{A} \models \psi(\vec{a})\})$.

Definition 5.10 ([11]) *A formula $\psi(x_1, \ldots, x_m)$ has the* bounded degree property (BDP)*, if there is a function $f_\psi : \mathbb{N} \to \mathbb{N}$ such that*

$$deg(\psi[\mathcal{A}]) \quad \leq \quad f_\psi(k) \text{ for any } \mathcal{A} \in \mathrm{STRUCT}_k[\sigma].$$

The bounded degree property is a very useful tool in proving inexpressibility results of recursive properties, i.e., for those queries that require fixpoint computation. As a simple example, we show that the transitive closure query violates the BDP. Assume that *TRCL* does have the BDP; that is, there is a function $f : \mathbb{N} \to \mathbb{N}$ such that $deg(TRCL(\mathcal{A})) \leq f(k)$ if all in- and out-degrees in $\mathcal{A}$ do not exceed k. Let $N = f(1) + 1$. Consider $\mathcal{A}$ which is a successor relation on N points (see Figure 1). Since all in- and out-degrees in $\mathcal{A}$ are at most 1, we get $deg(TRCL(\mathcal{A})) \leq f(1) < N$, but one can easily see that $deg(TRCL(\mathcal{A})) = N$. This contradiction shows that *TRCL* does not have the bounded degree property. This proof also shows that *deterministic* transitive closure violates the BDP. (Deterministic transitive closure is defined just as transitive closure, except that one only considers paths where each node other than the last one has outdegree 1, see [29].)

What makes the BDP particularly interesting, is the following result.

Theorem 5.11 ([11]) *Every Gaifman-local formula has the bounded degree property.* □

From results in the previous subsection, we conclude that first-order logic and various counting logics we considered have the bounded degree property. This confirms out intuition that these logics lack mechanisms for expressing recursive (fixpoint) computation.

Corollary 5.12 ([32, 25]) *Every Hanf-local formula has the bounded degree property. In particular, $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ formulae, $\mathcal{FO} + COUNT$ formulae (without free numerical variables), and $\mathcal{FO}$ formulae have the bounded degree property.* □

Since deterministic transitive closure does not have the BDP, we obtain the following result.

Corollary 5.13 ([32, 25]) *Deterministic transitive closure is not definable in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$ nor in $\mathcal{FO} + COUNT$.* □

This follows immediately from the BDP, and avoids all the combinatorial arguments in Section 4, and especially the ones in Section 3, that are based on game-theoretic techniques.

We now give another example that shows how the BDP can be applied to prove inexpressibility results.

Example 5.14 A *balanced binary tree* is a (directed) binary tree in which all paths from the root to the leaves are of the same length. Can this property be tested in $\mathcal{FO}$, or perhaps in more expressive logics such as $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$? We now use the BDP to give the negative answer.

Suppose that we have a sentence Φ that tests if a given graph is balanced binary tree. We next define a query $\varphi(x, y)$ as follows. It first defines a new graph, by interchanging the immediate successors of x, x' and x'', and the immediate successors of y, y' and y'', as shown in Figure 3 below, and then it tests is the resulting graph is a balanced binary tree. If either x or y fails

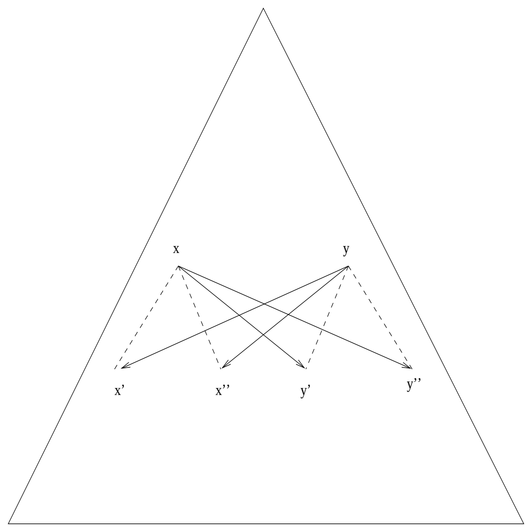

Figure 3: Changing successors of nodes in a balanced binary tree

to have exactly two immediate successors, then $\varphi(x, y)$ will evaluate to *false*. Assuming Φ is in the logic, so is $\varphi(x, y)$, for logics like $\mathcal{FO}$ and $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$.

We now show that $\varphi(x, y)$ violates the BDP. Assume it does have the BDP, and let $N = f_\varphi(2) + 1$. Let $\mathcal{A}$ be a balanced binary tree where each path from the root to a leaf has length N. Since degrees in $\mathcal{A}$ do not exceed 2, $deg(\varphi[\mathcal{A}]) < N$ by the BDP. We can see that $\mathcal{A} \models \varphi(a, b)$, for two nodes a, b, if and only if a and b are at the same level in $\mathcal{A}$. Thus, $\varphi[\mathcal{A}]$ is a disjoint union of N cliques of different sizes, and hence $deg(\varphi[\mathcal{A}]) = N$. This contradiction shows that φ does not have the BDP, and hence cannot be defined in $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$. Consequently, testing for balanced binary trees is not $\mathcal{L}_{\infty\omega}(\mathbf{Q}_u)^\omega$-definable.

One may have noticed that there is a certain asymmetry in the definition of the BDP. In the assumption, we deal with $deg\_set(\mathcal{A})$, but the conclusion puts a bound on $deg(\varphi[\mathcal{A}])$. Can the definition be made symmetric? To formalize this, define the *strong bounded degree property* of $\varphi(\vec{x})$ as follows: there exists a function $f_\varphi : \mathbb{N} \to \mathbb{N}$ such that $deg(\varphi[\mathcal{A}]) \leq f_\varphi(deg(\mathcal{A}))$.

Proposition 5.15 ([11]) *There are first-order definable graph queries that violate the strong bounded degree property.* □

In fact, [11] shows that even a weaker property is violated by some first-order queries. Define the *interval bounded degree property* of a query $\varphi(\vec{x})$ as the existence of a function $f_\varphi : \mathbb{N} \to \mathbb{N}$ such that $deg(\varphi[\mathcal{A}]) \leq f_\varphi(k)$ whenever $deg\_set(\mathcal{A}) \subseteq \{n, n+1, \ldots, n+k\}$ for some number n. Then there exist first-order definable queries on graphs that violate this property.

To summarize, we have seen four different locality conditions: strong Gaifman-locality, Hanf-locality, Gaifman-locality, and the bounded degree property. The relationship between them is shown in Figure 5.4. While Hanf-locality is closely tied to a game and relatively easy to show

Strongly Gaifman-local $\Rightarrow$ Hanf-local $\Rightarrow$ Gaifman-local $\Rightarrow$ BDP

Figure 4: The relationship between the notions of locality

for $\mathcal{FO}$ and some of its extensions, Gaifman-locality and the bounded degree property are very easy to use in expressibility proofs. Fortunately, they are implied by Hanf-locality of a logic.

6 Applications in complexity theory

Fagin's theorem, that equates existential second-order logic and complexity class NP, started a new line of research in complexity theory. In the past 20 years, many complexity classes have been characterized in logical terms, see [13, 28] for an overview. For example, polynomial time and space can be characterized by least- and partial fixpoint logics, respectively. Essential for many characterizations is the presence of a linear order on the input. The intuition behind having an order is simulating the order in which elements of the input appear on the tape of a Turing machine. While the exact order does not affect the output, its presence is required for a logic to simulate the computation of a machine. In fact, it remains an open problem to find a logic for polynomial time properties of unordered graphs, for example.

In this section, we deal with a circuit complexity class TC^0. This class is defined via Boolean circuits. Consider a family of circuits $C = \{c_1, c_2, \ldots, c_n, \ldots\}$, where the circuit c_n has n inputs and one output. Given a Boolean string x, we say that C accepts x if the output of c_n on x is 1, whenever x is of length n.

The class AC^0 is defined as the class of languages accepted by circuits C where each gate is either an AND, or an OR, or a NOT gate, with AND and OR gates having unbounded fan-in (no restriction on the number of inputs). The number of gates in c_n is polynomial in n, and the depth of circuits c_n is constant. (More generally, for AC^k, the depth of c_n is allowed to be $O(\log^k n)$.) The class TC^0 is defined as AC^0, except that majority gates MAJ are also allowed. Assume such a gate has k inputs. Then its single output is 1 iff at least $\lfloor k \rfloor + 1$ of its inputs are 1.

The class TC^0 is not an idle creation of complexity theory; in fact, it is of special importance in computer science. It characterizes the complexity of such important operations as integer multiplication, division, and sorting, and serves as a computational model for neural nets [42]. We refer the reader to survey [4] for additional information on circuit complexity.

Despite its importance, not much is known about the relationship between TC^0 and other complexity classes. We do know that $\mathrm{AC}^0 \subset \mathrm{TC}^0$, as the parity language (strings with even number of 1s) is in TC^0, but not in AC^0 [8]. We also know (see [4], for example) that

$$\mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{DLOG} \subseteq \mathrm{NLOG} \subseteq \mathrm{NC} \subseteq \mathrm{PTIME} \subseteq \mathrm{NP},$$

but we do not know if *any* of the inclusions is proper! In fact, [43] showed that there is inherent difficulty in separating TC^0 from NP, at least using conventional techniques of circuit complexity. A general notion of *natural proof* was formulated in [43]; this notion subsumes most of the existing lower bound proofs. Then [43] showed that the existence of a natural proof separating these two classes would imply that no good pseudo-random number generators are computable in TC^0. Putting it in the language of [4], it would imply that no cryptographically-secure functions can be computed in TC^0, even though cryptographers believe that such functions do exist. As the notion of natural proof is quite different from results on logical expressibility in finite model theory, one might attempt to avoid obstacles of [43] by using a logical characterization of TC^0. Below, we survey results in this direction.

Notice that in the definition of classes AC^0 and TC^0, we did not say anything about the relationship between circuits $c_n \in C$ when n varies; in fact, they can compute completely "different" things for different n. However, in most applications, those circuits compute the same property, like parity. Capturing this intuition leads to the notion of *uniformity.* The weakest notion of uniformity is PTIME-uniformity, meaning that the mapping $n \mapsto c_n$ is computable in polynomial time. Similarly, one can define logspace-uniformity (see [3] for using these notions with the class TC^0). However, the most widely used notion of uniformity is DLOGTIME-uniformity. We spare the reader the more technical definition, that can be found in [5], and instead give the characterization theorem.

Theorem 6.1 ([5]) *DLOGTIME-uniform* $\mathrm{TC}^0 \;=\; \mathcal{FO} + \mathit{COUNT} + <$. □

From now on, when we speak of TC^0, we mean its DLOGTIME-uniform version; that is, $\mathcal{FO} + \mathit{COUNT} + <$. The latter is the class of problems definable by $\mathcal{FO} + \mathit{COUNT}$ formulae in the presence of an order relation $<$. Note that those are order-independent. In general, when we deal with open formulae, not just sentences, the notion of order-independence is defined as follows. Suppose we have a $\mathcal{FO} + \mathit{COUNT}$ formula φ in the language of σ and $<$, and suppose $\mathcal{A}$ is a σ-structure. Then, for any two orderings $<_1$ and $<_2$ on $\mathcal{A}$, and for any $\vec{a}$, $\mathcal{A}_{<_1} \models \varphi(\vec{a})$ iff $\mathcal{A}_{<_2} \models \varphi(\vec{a})$, where $\mathcal{A}_<$ denotes the extension of $\mathcal{A}$ with the order $<$.

Even though we restrict our attention to order-independent properties, the mere presence of an order relation does increase the expressive power:

Proposition 6.2 ([6]) $\mathcal{FO} + \mathit{COUNT} \;\subset\; \mathcal{FO} + \mathit{COUNT} + <$. □

The example of a separating query (not the proof!) is quite simple. Let $\sigma = (E, U)$, where E is binary and U is unary. Consider the following property: If the interpretation of E is an equivalence relation, then the number of distinct sizes of equivalence classes of E equals the cardinality of U. This query is not definable in $\mathcal{FO} + \mathit{COUNT}$ [6], but, as shown in [36], can easily be defined with order, since all elements whose equivalence classes have the same size, can be canonically represented by the $<$-minimal such element. Then one just checks if the number of those elements equals the cardinality of U.

Proposition 6.2 reduces the problem of separating TC^0 from classes above it to the problem of logical expressibility; for example, to show $\mathrm{TC}^0 \neq \mathrm{NLOG}$, it would suffice to show that transitive closure is not definable in $\mathcal{FO} + COUNT + <$. Since locality gives us an easy proof that transitive closure is not in $\mathcal{FO} + COUNT$, one might try to push the ideas of locality into the ordered setting.

We do not know whether the above expressivity bound on $\mathcal{FO} + COUNT + <$ is true, although we conjecture that it is. Below, we survey some of the partial results confirming the intuition. We state the results for the NLOG-complete problem of computing the transitive closure, but they also hold for *deterministic* transitive closure, which is complete for DLOG.

Assume that instead of an order relation, we have a *successor* relation $SUCC$. Then, as an immediate consequence of the bounded degree property of $\mathcal{FO} + COUNT$, we obtain

Corollary 6.3 ([14]) *Transitive closure is not definable in* $\mathcal{FO} + COUNT + SUCC$. □

Note that $\mathcal{FO}$ plus transitive closure $TRCL$ plus successor relation capture NLOG (cf. [13, 28]); hence, $\mathcal{FO} + COUNT + SUCC \subset \text{NLOG} = \mathcal{FO} + TRCL + SUCC$. This result was first shown in [14], via a rather complex argument based on games of [30]. Later, using the results of [38], the journal version of [14] (see [15]) gave a much simpler proof based on Hanf's condition. Finally, using the bounded degree property, we gave a completely elementary proof.

The use of bounded degree property allows us to substitute any auxiliary relation for $SUCC$, as long as its degrees are bounded by a constant. For example, we could use balanced binary trees (note that using such a structure would most certainly make a game-based proof unmanageable). The next question is: How can we lift the results for $\mathcal{FO} + COUNT$ from the constant world to that where degrees are allowed to depend on the size of a structure?

First such result was given in [32], and it used the notion of *moderate* degree of [17]. Let $\mathcal{C}$ be a class of structures. Let $maxdeg_{\mathcal{C}}(n)$ denote the maximal degree of a structure in $\mathcal{C}$, whose cardinality is n. Then we say that $\mathcal{C}$ is a class of relations of moderate degree if $maxdeg_{\mathcal{C}}(n) \leq \log^{o(1)} n$. That is, for some function $\delta : \mathbb{N} \to \mathbb{N}$ with $\lim_{n\to\infty} \delta(n) = 0$, we have $maxdeg_{\mathcal{C}}(n) \leq \log^{\delta(n)} n$. Combination of results from [11] and [32] led to the following.

Proposition 6.4 ([32]) *Transitive closure is not definable in* $\mathcal{FO} + COUNT$ *in the presence of relations of moderate degree.* □

A linear order on an n-element set realizes n different degrees, from 0 to $n-1$. Thus, we need to lift the results from relations of small (constant or moderate) degree to relations of large (comparable with the size of the input) degree. The concept of moderate degree was introduced in [17] to show that connectivity is not definable in monadic Σ_1^1 in the presence of those relations. Later, [44] extended to linear orders. Thus, one may ask if a similar avenue of attack on the separation problem can be pursued in the case of $\mathcal{FO} + COUNT$.

A partial result in this direction was proved recently. Let $\mathcal{O}_k$ stand for the class of relations which are pre-orders $\langle A, \prec \rangle$ (i.e., $\prec$ is reflexive and transitive), and each equivalence class of the relation $x \sim y \equiv (x \prec y) \wedge (y \prec x)$ has at most k elements. In particular, $\mathcal{O}_1$ is the class of linear orders.

Theorem 6.5 ([36]) *Transitive closure is not definable in $\mathcal{FO}+COUNT+\mathcal{O}_k$, for any $k>1$.* □

In fact, the result of [36] is even stronger as it allows relations from $\mathcal{O}_2$ to be of special form: those that have at most $g(n)$ elements in equivalence classes of size 2, and the rest, $n-g(n)$ elements, form a linear order. The function $g(n)$ can be chosen to be arbitrarily small, e.g., $\log\log\ldots\log n$, but it cannot be bounded by a constant. The version of Theorem 6.5, proved in [36] for deterministic transitive closure, implies DLOG $\not\subseteq \mathcal{FO}+COUNT+\mathcal{O}_k$ for any $k>1$. But this still falls short of resolving the most important case of $k=1$.

One of the main goals of pursuing this line of research was to avoid the obstacles on the path towards separating TC^0 from other classes posed by the natural proofs of [43]. At the first glance, our expressivity bounds look nothing like natural proofs, although we must admit that there was no systematic study conducted on the relationship between classical lower bounds proofs in circuit complexity and logical expressivity bounds for $\mathcal{FO}+COUNT$ and the likes.

However, while avoiding one kind of problems, we encountered (perhaps even created!) different ones. For example, the proof of Theorem 6.5 is based on the following technique. One first shows that if for a class $\mathcal{C}$ of auxiliary relations, there exists a class of graphs $\mathcal{G}$ such that the pair $(\mathcal{C},\mathcal{G})$ satisfies two properties, P1 and P2, then transitive closure of graphs in $\mathcal{G}$ is not definable in $\mathcal{FO}+COUNT+\mathcal{C}$. Then, [36] showed how to construct $\mathcal{G}$ for $\mathcal{C}$ being $\mathcal{O}_k$, with $k>1$, thus proving inexpressibility of transitive closure in $\mathcal{FO}+COUNT+\mathcal{O}_k$. Unfortunately, [36] also showed that if $\mathcal{C}=\mathcal{O}_1$, the class of linear orders, then there is *no* class of graphs $\mathcal{G}$ such that $(\mathcal{C},\mathcal{G})$ satisfies P1 and P2. That is, the approach is inherently limited for showing results in the ordered setting.

Another large obstacle to using the ideals of locality for proving the separation is a very recent (unpublished) result:

Theorem 6.6 ([24]) *There exist order-invariant formulae in $\mathcal{FO}+COUNT+<$ that do not have the bounded degree property. Consequently, order-invariant $\mathcal{FO}+COUNT+<$ is not Gaifman-local.* □

Nevertheless, we believe it was a useful exercise to study the expressivity of $\mathcal{FO}+COUNT$ with relations of large degree, viewing it as a reasonable approximation of uniform TC^0. The bounds of [14], obtained less than 3 years ago with a great deal of effort, were shown here by a simple application of the bounded degree property. The best lower bound given by locality arguments (Theorem 6.5) shows that one can get very close to a linear order, and thus very close to uniform TC^0, without achieving the power of logspace computation. In a way, results of this section can be viewed as separating "very uniform" versions of TC^0 from DLOG and classes above it.

7 Applications in database theory

The theory of database query languages is firmly grounded in finite-model theory, and also provides a major motivation for finite-model theory research. Traditional databases query languages, such as relational algebra and calculus, have precisely the power of first-order logic. An important subclass, called conjunctive queries, is simply the $\{\exists, \wedge\}$-fragment of first-order logic. Various extensions of relational calculus, such as Datalog, Datalog with negation, and the language of *while* loops, correspond to various fixpoint extensions of first-order logic, see [1].

In the relational model, data is stored in relations. For example, a database may have two relations, for storing information about employees (called `Emp`) and departments (`called Dept`). Assume that `Emp` stores triples containing employee name, department, and salary, and `Dept` stores triples containing department name, manager, and manager's salary. Below is an example of a database:

EName	EDept	ESalary
John	A1	50
Ann	A1	60
Jim	B2	75

DName	Manager	MSalary
A1	Bob	80
B2	Steve	85
C3	Mary	80

An example of a query is *"For each employee, find his or her manager."* This can be written in first-order logic as:

$$q(emp, manager) \quad \equiv \quad \exists dept, esal, msal\ \texttt{Emp}(emp, dept, esal) \wedge \texttt{Dept}(dept, manager, msal)$$

In real life, of course, programmers do not write first-order formulae; instead they write queries in the language called SQL, which is the *lingua franca* of the commercial database world. The above query, in SQL, will look like:

```
SELECT  E.EName, D.Manager
FROM    Emp E, Dept D
WHERE   E.Edept = D.Dname
```

The best way to read such statements is as set-theoretic comprehensions: the above becomes

$$\{(EName,\ D.Manager) \mid (EName, EDept, ESalary) \in \texttt{Emp}, (DName, Manager, MSalary) \in \texttt{Dept}, \\ EDept = DNname\}$$

The basis of SQL is the select-from-where statement, with the addition union and difference, and features such as view creation, which make the language compositional. This basis has precisely the power of the first-order logic. However, all practical implementations of SQL come equipped with two additional features: *arithmetic operations*, and *aggregate functions*. For example, consider the following query: *"Find all departments that have more than 5 employees, together with the name of the manager and the average salary of the employees."*

In SQL, this will be written as

```
SELECT   D.DName, D.Manager, AVG(E.ESalary)
FROM     Emp E, Dept D
WHERE    E.Edept = D.Dname
GROUPBY  D.DName
HAVING   COUNT(E.EName) > 5
```

There are two key new features in this query. *Grouping* is given by the clauses `GROUPBY` and `HAVING`: for each department manager, we group together all the employees in his/her department, provided there are more than five of them. The other feature is *aggregate function*: these are `AVG`, for computing the average salary, and `COUNT`, for counting the number of employees.

Let us now see why adding these features is indeed an extension of first-order logic. We consider a query that we know is inexpressible in first-order logic alone: given a graph G, find the set of nodes x with $in\text{-}deg(x) = out\text{-}deg(x)$. We let graphs be represented as a binary relation `Edges` with two attributes, From and To. As the first step, we create two new relations, one storing nodes together with their in-degrees, and the other storing nodes with their out-degrees. Such intermediate relations are called *views* in SQL, and are created as

```
CREATE VIEW INDEGV(Node, Indeg) AS
  SELECT  E.To, COUNT(E.From)
  FROM    Edges E
  GROUPBY E.To
```

```
CREATE VIEW OUTDEGV(Node, Outdeg) AS
  SELECT  E.From, COUNT(E.To)
  FROM    Edges E
  GROUPBY E.From
```

Using these views, one computes the answer to the query as

```
SELECT INDEGV.Node
FROM   INDEGV, OUTDEGV
WHERE  (INDEGV.Node = OUTDEGV.Node)  AND  (INDEGV.Indeg = OUTDEGV.Outdeg)
```

By now, the reader must be convince that Härtig and Rescher quantifiers can be expressed in SQL. Thus, it is more powerful than first-order logic. The question now is:

How expressive is SQL?

More precisely, there seems to be a "folk result" saying that SQL cannot compute recursive queries, such as transitive closure. That is, it lacks a mechanism for recursive computation. The question now is: What kind of formal statement can one prove to confirm this intuition?

The approach of some textbooks is to restrict SQL to its subset which is essentially first-order logic, and use expressivity bounds for the latter. However, as we have just seen, this is not satisfactory. The difficulty with answering the question above is that there are dozens of different version of SQL (see [41] for an overview of standards and dialects), and they often support different sets of operators. For example, some versions even add the transitive closure operator. Thus, we restrict our attention to features that are common to *all* versions of SQL; that is, grouping and aggregation (cf. [1, 41]). The first result on the expressive power of such a language was based on yet unproven assumption from complexity theory, and two observations: SQL queries can be evaluated in deterministic logspace, and transitive closure is complete for nondeterministic logspace. Thus,

Proposition 7.1 ([10]) *Assume that DLOGSPACE $\neq$ NLOGSPACE. Then transitive closure cannot be expressed in SQL.* □

Can we get rid of the unproven assumption? The problem we face is that SQL per se is quite inconvenient to work with – its syntax is quite awkward, and in fact it has been an object of persistent criticism. SQL combines sets and multisets in order to evaluate aggregates: for example, computing the average salary, one *cannot* first project out the salary attribute and then compute its average, as the elimination of duplicates will produce an incorrect result. Following [19, 33], a language that deals correctly with multiset and set semantics was proposed in [34]. As this language can model the main features of SQL, and extends first-order logic, it was suggested to use it as a rational reconstruction of SQL.

Let us now give an informal introduction into this language, which we call here $\mathsf{AGGR}_{\mathbb{Q}}$. It deals with objects which can freely combine rational numbers, elements of the domain of atomic values, D, tuples, and sets. In particular, it permits sets of sets. The language is *statically typed* (cf. [20, 34]), but we shall not go into detail here. The full description can be found in [20, 34]; here we highlight the salient features. The language allows one to apply a function to each element of a set, that is, obtain a set $\{f(x) \mid x \in X\}$ from X. If all elements of a set $\mathcal{X}$ are sets themselves, their union can be taken: $\bigcup_{X \in \mathcal{X}} X$. Basic arithmetic operations $(+, -, *, \div, <)$ are available on $\mathbb{Q}$. On the domain of basic values D, only equality test is available (notice the absence of order!). Finally, if f is a function into $\mathbb{Q}$, its values on a set X can be added up; that is, one can compute $\sum_{x \in X} f(x)$. Note that if f is identically 1, then the above is the cardinality of X.

It was proved in [34] that these features model the main features of SQL. Furthermore, extending a result from [47], a conservativity property was shown in [34]. It says that nesting of sets is in a sense superfluous: every query from relational databases to relational databases in $\mathsf{AGGR}_{\mathbb{Q}}$ can be written in a way that does not use sets of sets. Nesting is essential for modeling grouping, and thus the (nontrivial) conservativity result gives us rather pleasant language (without higher-order features) to model *all* the features of SQL. In [34], the following was proved:

Proposition 7.2 ([34]) *Transitive closure cannot be expressed in* $\mathsf{AGGR}_{\mathbb{Q}}$. □

While it does provide useful bounds, there are two problems with the proof of [34]. First, it is very syntactic. It proceeds by establishing a normal form result for queries on a special class of inputs; the property depends both on the class of inputs, and the properties of the chosen syntax of the language. Thus, making a minor change in the syntax that does not affect expressiveness would mean that the proof must be redone from scratch. Even more unpleasantly, the proof in [34] establishes inexpressibility of transitive closure, but fails to establish a general property that will give us expressivity bounds.

In [20, 34] it was conjectured that *relational queries* in $\mathsf{AGGR}_{\mathbb{Q}}$ have the BDP. By relational queries we mean those whose inputs and outputs only contain elements of D, but no numbers, although rational numbers can be used in the process of evaluating a query. An SQL example of finding nodes with equal in- and out-degrees shows that there are relational queries definable in SQL but not in first-order. Clearly, proving the above conjecture would resolve the problem for queries such as transitive closure or deterministic transitive closure. By extending the normal form result $\mathsf{AGGR}_{\mathbb{Q}}$, the following was proved.

Proposition 7.3 ([11]) *Every relational query in* $\mathsf{AGGR}_{\mathbb{Q}}$ *has the bounded degree property.* □

It is still unpleasant that the proof depends on a particular syntax for the language. Also, it would be interesting to know if relational queries are Gaifman-local. These two questions were considered in [32]. The approach taken by [32] was the following. Restrict the language to *natural numbers* only. That is, one can compute `COUNT` but not `AVG`. Let us call this language $\mathsf{AGGR}_{\mathbb{N}}$. With rationals out of the way, try to embed it into $\mathcal{FO} + COUNT$ to prove locality.

Embedding into $\mathcal{FO} + COUNT$ turns out to be problematic, as in $\mathcal{FO} + COUNT$ the sizes of the first-sort and second-sort universes are the same. In contrast, in SQL, one can create numbers much bigger than the size of the database. A simple example is this: consider an n-element set X, and compute $\sum_{x \in X} f(x)$ where $f(x) = 2$ for every x. The result is $2n$. The solution to the problem is to modify a query, essentially by putting a huge linear order "on the side" and having all arithmetic done on that linear order. This technique led to the following result.

Proposition 7.4 ([32]) *For any relational query Q in* $\mathsf{AGGR}_{\mathbb{N}}$*, another query Q' can be found such that (a) Q is Gaifman-local iff Q' is, and (b) Q' can be defined in $\mathcal{FO} + COUNT$. Consequently, every relational query in* $\mathsf{AGGR}_{\mathbb{N}}$ *is Gaifman-local.* □

Using this technique, [35] returned to the main question: what is the expressive power of SQL? It defined a new language, called $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$, which first restricts $\mathsf{AGGR}_{\mathbb{N}}$ to objects that do not contain sets of sets, and then adds new arithmetic operation, and *product* over a set. That is, given a set X and a function f into $\mathbb{N}$ definable in $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$, one can compute $\prod_{x \in X} f(x)$ in $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$. Then the following sequence of results was proved.

Theorem 7.5 ([35]) *1) Every relational* $\mathsf{AGGR}_{\mathbb{Q}}$ *query can be expressed in* $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$.
2) For any relational query Q in $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$*, another query Q' in* $\mathsf{AGGR}_{\mathbb{N}}^{\mathsf{flat}}$ *can be found such that (a) Q is Gaifman-local iff Q' is, and (b) Q' can be defined in $\mathcal{FO} + COUNT$.* □

The bulk of the proof of this theorem is in showing (b) of part 2). Combining these results with locality of $\mathcal{FO} + COUNT$, we obtain:

Corollary 7.6 *Every relational query in* $\mathsf{AGGR}_{\mathbb{Q}}$ *is Gaifman-local.* □

Consequently, plain SQL queries are Gaifman-local and have the bounded degree property. Therefore, queries that need a recursion mechanism (transitive closure, deterministic transitive closure, connectivity test, etc.) *cannot* be computed in SQL.

Thus, locality helped us answer important questions about expressive power of real world database query languages.

We conclude this section by a remark about the set of basic operations on the domain of atomic values, D. In all the above results, we assume that it is only possible to test if two elements of D are equal. In many applications, there is a meaningful linear order on such domains (for examples, salaries can be compared). What happens if D is ordered? As it often happens, things become a lot more complicated in the ordered setting. In a way, we saw it in the last section. For SQL, one can show that it is possible to express any query from $\mathcal{FO} + COUNT + <$ if D is ordered. Thus, every uniform TC^0 property becomes definable in SQL, and hence the problems of expressivity in of recursive queries (such as transitive closure) hinges on the separation of complexity classes, thus confirming the original intuition of [10]. Note that there are other well-known examples of close connection between separation of complexity classes and expressivity bounds of query languages, see [2].

One can go further and add more operations to D. For example, one can assume that D is the field of real numbers with the usual operations $+, *, -$, or perhaps more complex such as e^x. The results of [7] show that for the class of *generic* [1] queries (those that commute with permutations of D), these extra operations do not add expressive power, beyond a linear order. Most examples of queries we consider – transitive closure, connectivity test, etc. – are generic. Thus, adding extra operations beyond $<$ does not lead to an increase in power of languages such as first-order logic or SQL. This assertion, in the case of an interpreted domain D, depends on what one means by quantification $\exists x$. In most application, quantification in queries assumes the finite database; however, it is conceivable (and in some applications, important) to quantify over D. We refer the reader to [7] for the discussion on this topic.

8 Conclusion

In this paper, we reviewed some results that were developed for proving lower bounds for logical expressibility. We considered first-order logic and its extensions with several kinds of counting mechanisms. We presented the usual game-theoretic characterizations of those logics, as well as Gaifman's and Hanf's theorems, and general notions behind these results. We also studied the relationship between these notions and the bounded degree property. We reviewed applications of these notions in descriptive complexity theory and database theory.

Acknowledgement We thank Jouko Väänänen for the invitation to write this survey. We are grateful to Lauri Hella and Limsoon Wong for their comments.

References

[1] S. Abiteboul, R. Hull and V. Vianu. *Foundations of Databases*, Addison Wesley, 1995.

[2] S. Abiteboul and V. Vianu. Computing with first-order logic. *Journal of Computer and System Sciences* 50 (1995), 309–335.

[3] M. Agrawal, E. Allender and S. Datta. On TC^0, AC^0, and arithmetic circuits. In *Proc. 12th IEEE Conf. on Computational Complexity*, 1997.

[4] E. Allender. Circuit complexity before the dawn of the new millennium. In *Proc. 16th Conf. on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'96)*, Springer LNCS vol. 1180, 1996, 1–18.

[5] D.A. Barrington, N. Immerman and H. Straubing. On uniformity within NC^1. *J. Comput. and Syst. Sci.*, 41:274–306,1990.

[6] M. Benedikt, H.J. Keisler. On expressive power of unary counters. *Proc. Int. Conf. on Database Theory (ICDT'97)*, Springer LNCS 1186, January 1997.

[7] M. Benedikt and L. Libkin. Languages for relational databases over interpreted structures. In *Proc. 16th ACM Symp. on Principles of Database Systems (PODS'97)*, Tucson, AZ, May 1997, pages 87–98.

[8] R.B. Boppana and M. Sipser. The Complexity of Finite Functions. In *"Handbook of Theoretical Computer Science,"* Volume A, Chapter 14, pages 759–804, North Holland, 1990.

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

[10] M. Consens and A. Mendelzon. Low complexity aggregation in GraphLog and Datalog, *Theoretical Computer Science* 116 (1993), 95–116. Extended abstract in *ICDT'90.*

[11] G. Dong, L. Libkin and L. Wong. Local properties of query languages. *Proc. Int. Conf. on Database Theory (ICDT'97)*, Springer LNCS 1186, 1997, pages 140–154.

[12] H.-D. Ebbinghaus. Extended logics: the general framework. In J. Barwise and S. Feferman, editors, *Model-Theoretic Logics*, Springer-Verlag, 1985, pages 25–76.

[13] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer Verlag, 1995.

[14] K. Etessami. Counting quantifiers, successor relations, and logarithmic space, In *Proc. Structure in Complexity Theory*, 1995.

[15] K. Etessami. Counting quantifiers, successor relations, and logarithmic space, *Journal of Computer and System Sciences*, 54 (1997), 400–411.

[16] R. Fagin. Easier ways to win logical games. In *Proc. DIMACS Workshop on Descriptive Complexity and Finite Models*, AMS 1997.

[17] R. Fagin, L. Stockmeyer and M. Vardi, On monadic NP vs monadic co-NP, *Information and Computation*, 120 (1994), 78–92.

[18] H. Gaifman. On local and non-local properties, *Proceedings of the Herbrand Symposium, Logic Colloquium '81*, North Holland, 1982.

[19] S Grumbach and T. Milo. Towards tractable algebras for bags. In *Proc. 12th ACM Symp. on Principles of Database Systems*, Washington, DC, May 1993, pages 49–58.

[20] S. Grumbach, L. Libkin, T. Milo and L. Wong. Query languages for bags: expressive power and complexity. *SIGACT News*, 27 (1996), 30–37.

[21] W. Hanf. Model-theoretic methods in the study of elementary logic. In J.W. Addison et al, eds, *The Theory of Models*, North Holland, 1965, pages 132–145.

[22] L. Hella. Logical hierarchies in PTIME. *Information and Computation*, 129 (1996), 1–19.

[23] L. Hella. Generalized quantifiers in finite model theory. This volume.

[24] L. Hella. Personal communication.

[25] L. Hella, L. Libkin and J. Nurmonen. Notions of locality and their logical characterizations over finite models. Unpublished manuscript, 1997.

[26] L. Hella and J. Nurmonen. Vectorization hierarchies of some graph quantifiers. *Archive for Mathematical Logic*, to appear.

[27] L. Hella and G. Sandu. Partially ordered connectives and finite graphs. In M. Krynicki, M. Mostowski and L. Szczerba, eds, *Quantifiers: Logics, Models and Computation II*, Kluwer Academic Publishers, 1995, pages 79–88.

[28] N. Immerman. Descriptive complexity: A logician's approach to computation. *Notices of the AMS* 42 (1995), 1127–1133.

[29] N. Immerman. Languages that capture complexity classes. *SIAM J. Comput.* 16 (1987), 760–778.

[30] N. Immerman and E. Lander. Describing graphs: A first order approach to graph canonization. In *"Complexity Theory Retrospective"*, Springer Verlag, Berlin, 1990.

[31] Ph. Kolaitis and J. Väänänen. Generalized quantifiers and pebble games on finite structures. *Annals of Pure and Applied Logic*, 74 (1995), 23–75.

[32] L. Libkin. On the forms of locality over finite models. In *Proc. 12th IEEE Symp. on Logic in Computer Science (LICS'97)*, Warsaw, Poland, June–July 1996, pages 204–215.

[33] L. Libkin and L. Wong. Some properties of query languages for bags. In *Proc. Database Programming Languages 1993*, Springer Verlag, 1994, pages 97–114.

[34] L. Libkin and L. Wong. Query languages for bags and aggregate functions. *Journal of Computer and System Sciences* 55 (1997), 241–272. Extended abstract in *PODS'94*, pages 155–166.

[35] L. Libkin and L. Wong. On the power of aggregation in relational query languages. In *Proc. Database Programming Languages 1997*, Springer LNCS, to appear.

[36] L. Libkin and L. Wong. Unary quantifiers, transitive closure, and relations of large degree. In *Proc. 15th Symp. on Theoretical Aspects of Computer Science (STACS'98)*, Springer Verlag, 1998, to appear.

[37] K. Luosto. Hierarchies of monadic generalized quantifiers. *Journal of Symbolic Logic*, to appear.

[38] J. Nurmonen. On winning strategies with unary quantifiers. *J. Logic and Computation*, 6 (1996), 779–798.

[39] J. Nurmonen. Counting modulo quantifiers on finite structures. *Information and Computation*, to appear. Extended abstract in *Proc. 11th IEEE Symp. on Logic in Computer Science (LICS'96)*, New Brunswick, NJ, July 1996, pages 484–493.

[40] J. Nurmonen. Unary quantifiers and finite structures. PhD Thesis, University of Helsinki, 1996.

[41] P. O'Neil. *Database.* Morgan Kaufmann, 1994.

[42] I. Parberry and G. Schnitger. Parallel computation and threshold functions. *Journal of Computer and System Sciences* 36 (1988), 278–302.

[43] A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences* 55 (1997), 24–35.

[44] T. Schwentick. On winning Ehrenfeucht games and monadic NP. *Annals of Pure and Applied Logic*, 79 (1996), 61–92.

[45] T. Schwentick and K. Barthelmann. Local normal forms for first-order logic with applications to games and automata. In *Proc. 15th Symp. on Theoretical Aspects of Computer Science (STACS'98)*, Springer Verlag, 1998, to appear.

[46] J. Väänänen. Generalized quantifiers. *Bulletin of the European Association for Theoretical Computer Science*, 62 (1997), 115–136.

[47] L. Wong. Normal forms and conservative properties for query languages over collection types. *Journal of Computer and System Sciences* 52 (1996), 495–505. Extended abstract in *PODS 93.*