Discrete Mathematics and Theoretical Computer Science **3**, 1999, 109–124

Local Normal Forms for First-Order Logic with Applications to Games and Automata

Thomas Schwentick and Klaus Barthelmann

Institut für Informatik, Johannes Gutenberg-Universität Mainz, D-55099 Mainz, Germany

received July 28, 1998, *revised August 25, 1999*, *accepted September 10, 1999*.

Building on work of Gaifman [Gai82] it is shown that every first-order formula is logically equivalent to a formula of the form $\exists x_1, \ldots, x_l \forall y \varphi$ where φ is r-local around y, i. e. quantification in φ is restricted to elements of the universe of distance at most r from y.

From this and related normal forms, variants of the Ehrenfeucht game for first-order and existential monadic second-order logic are developed that restrict the possible strategies for the spoiler, one of the two players. This makes proofs of the existence of a winning strategy for the duplicator, the other player, easier and can thus simplify inexpressibility proofs.

As another application, automata models are defined that have, on arbitrary classes of relational structures, exactly the expressive power of first-order logic and existential monadic second-order logic, respectively.

Keywords: First-order logic, existential monadic second-order logic, games, automata, locality

1 Introduction

First-order (FO) logic and its extensions play an important role in many branches of (theoretical) computer science. Examples that will be considered in this paper are automata theory and descriptive complexity. Since Büchi's and Elgot's famous characterization of the regular string languages as the sets of models of (existential) monadic second-order (MSO) sentences, (existential) MSO logic has been used as a guideline in the search for reasonable automata models for other kinds of structures like trees or graphs. In descriptive complexity, since Fagin [Fag74] showed that the complexity class **NP** coincides with the sets of models of existential second-order (Σ^1_1) sentences, many complexity classes have been characterized by extensions of FO logic [Var82, Imm86, Imm87, AV89, Gr92] and there is still hope that separations of complexity classes might be possible by separating the expressive power of the respective logics. For a recent result in this direction see the paper of Libkin and Wong [LW98].

Despite its importance as an ingredient for more expressive logics, it is well-known that the expressive power of FO logic is rather limited. It can only express properties that depend on the local appearance of a structure. This intuition has been formalized in different ways by Hanf [Han65] and Gaifman [Gai82]. Hanf showed that, for every first-order formula ψ, there is an r such that whether ψ holds in a structure A ("$A \models \psi$") only depends on the multiset of isomorphism types of all r-spheres in A. Here an r-sphere is a substructure of A which is induced by all elements of A that have distance at most r from a fixed element of A. On the other hand, Gaifman showed that, for every first-order formula ψ, there are r and d such that whether $A \models \psi$ holds depends only on how many elements with pairwise disjoint r-neighbourhoods exist that fulfil θ, for every formula θ of quantifier depth at most d.

The starting question for the present investigation was to which extent Hanf's and Gaifman's conditions could be combined. The goal was to replace the isomorphism types in Hanf's condition by something weaker and to get rid of the "disjoint r-neighbourhoods" constraint in Gaifman's condition. (For very interesting recent results concerning Hanf's and Gaifman's theorems from a different point of view see the papers of Libkin and Dong et al. [Lib97, DLW97].) It is easy to see that the straightforward attempt to replace the isomorphism type of a sphere S in Hanf's condition by its Hintikka-type for some d (i. e. by the set of formulas of quantifier depth at most d that hold in S) does not work. A counterexample is given by the set of clique graphs. For every d, the spheres of a graph consisting of one $2d$-clique fulfil exactly the same formulas of quantifier depth at most d as those of a graph which consists of two disjoint d-cliques. Nevertheless, it turns out that it is indeed possible to combine the two approaches in the following sense. For every FO formula ψ there are l and r such that $A \models \psi$ if and only if it is possible to put l pebbles onto A such that in the resulting structure all r-spheres fulfil the same FO formula φ. Put in another way, every FO formula is logically equivalent to a formula of the form $\exists x_1, \dots, x_l \forall y \varphi$ where φ is r-local around y, i. e. quantification in φ is restricted to elements of the universe with distance at most r from y. From this normal form one can easily derive normal forms for other logics like monadic second-order logic. For existential monadic second-order logic we can show a bit more. Every such formula is, on classes of connected structures, equivalent to a formula of the form $\exists X_1, \dots, X_l \exists x \forall y \varphi$ where φ is restricted as above. If every structure has an element which is uniquely definable by a local formula, we can even achieve a formula of the form $\exists X_1, \dots, X_l \forall y \varphi$.

As one application of the normal forms we get variants of the Ehrenfeucht game [Ehr61] for first-order logic and existential monadic second-order logic in which the spoiler has only restricted global access to the structures that are played. After a phase in which he can select some vertices of the first graph (before having seen the other one) and one vertex of the other graph, in the second phase all moves are restricted to the neighbourhoods of one vertex in each graph.

Another application concerns a form of automata on relational structures. It is well-known that regular sets of strings can be obtained as projections of locally testable sets, namely the sets of transition sequences of a (nondeterministic) automaton. Thomas [Tho91] used this idea to extend the notion of recognizability to other objects like grids and graphs of uniformly bounded degree. The normal forms allow to generalize the method of local testing further to sets of arbitrary relational structures. Moreover, they maintain the correspondence to definability in a natural logic.

The paper is organized as follows. In Section 2 we give basic definitions and fix some notation. In Section 3 we show the normal form theorems. In Section 4 we define the simplified games for first-order logic and monadic Σ^1_1-logic. The analogous results for automata are presented in Section 5. The ensuing automata models are also compared with the automata of [Tho91, Tho97b, Tho97a] and [Cou90]. Section 6 contains a conclusion.

2 Definitions and Notations

A *relational signature* σ is a finite set of relation symbols R, each with a fixed arity $a(R)$, and constant symbols c. We do not use function symbols. A σ*-structure* A consists of a universe U^A (the *vertices* of A), an $a(R)$-ary relation R^A on U^A, for every relation symbol R of σ and a constant c^A, for every constant symbol c of σ. All theorems in this paper are valid for infinite and finite structures. We assume that all structures contain at least two elements.

Let the *Gaifman graph* of a σ-structure A have universe U^A and edges between vertices a and b whenever a and b occur in a common tuple of a relation of A. The *distance* $\delta(a, b)$ of the vertices a and b of A is given by their (standard graph) distance in the Gaifman graph of A. For a tuple $\mathbf{b} = b_1, \ldots, b_l$ of vertices of A we define $\delta(a, \mathbf{b}) := \min\{\delta(a, b_i) \mid i \leq l\}$. For a tuple $\mathbf{b} = b_1, \ldots, b_l$ of vertices of A and $r \geq 0$ we define $S^r(\mathbf{b}) := \{a \in U^A \mid \delta(a, \mathbf{b}) \leq r\}$. We let $N^r(b)$ (resp. $N^r(\mathbf{a})$, for $\mathbf{a} = a_1, \ldots, a_l$) denote the substructure of A which is induced by the vertices of $S^r(b)$ (resp. $S^r(\mathbf{a})$) and has b (the a_i) as distinguished elements.

Let $H^d(A)$ denote the *depth-d-Hintikka-type* of A, i. e., the set of all FO sentences of quantifier-depth at most d that hold in A [EF95]. Recall that $H^d(A)$ contains only finitely many different formulas w. r. t. logical equivalence. The *elementary type* of a tuple $a_1, \ldots, a_k$ of vertices of A is the conjunction of all atomic σ-formulas with variables from $x_1, \ldots, x_k$ that hold in $\langle A, a_1, \ldots, a_k\rangle$.

Now we are going to define our notions of locality, r-locality and basic locality. Informally, a formula is r-local around its free variables $\mathbf{x}$, if its truth depends only on $S^r(\mathbf{x})$. More formally, a FO formula $\varphi(\mathbf{x}, \mathbf{u})$ with free variables from $\mathbf{x} = x_1, \ldots, x_l$ and $\mathbf{u} = u_1, \ldots, u_m$ is *r-local around* $\mathbf{x}$ if all variables that are quantified in φ are bounded to $S^r(\mathbf{x})$. I. e., if $\exists y\psi$ (resp. $\forall y\psi$) is a subformula of φ then ψ is of the form $(\delta(y, \mathbf{x}) \leq r) \wedge \chi$ (resp. $(\delta(y, \mathbf{x}) \leq r) \rightarrow \chi$), for some χ which is r-local around $\mathbf{x}$. A formula is *local around* $\mathbf{x}$ if it is r-local around $\mathbf{x}$ for some r. Here, $(\delta(y, \mathbf{x}) \leq r)$ is an abbreviation for the straightforward FO formula which expresses that the distance of y from $\mathbf{x}$ is at most r. A formula φ is *basic local* around $\mathbf{x}$ if it is a Boolean combination of formulas each of which is local around some variable x_i. I. e., properties that are expressed by basic formulas depend only on combinations of the properties of spheres. A formula is $\exists^*\forall$*–local* if it is of the form $\exists x_1, \ldots, x_l \forall y\varphi$ where φ is basic local around $x_1, \ldots, x_l, y$. It should be noted that in Gaifman's terminology [Gai82] a formula is local around a variable x only if x is its single free variable.

3 Normal Forms

By definition, every basic local formula around $\mathbf{x}$ is logically equivalent to a local formula around $\mathbf{x}$. The following lemma shows that the converse is also true. This will be an important tool in the proof of the normal form theorem.

Lemma 3.1 *Every first-order formula φ which is local around variables $\mathbf{x} = x_1, \ldots, x_l$ is logically equivalent to a formula which is basic local around $\mathbf{x}$.*

Proof. Let φ be a first-order formula with free variables $\mathbf{x} = x_1, \ldots, x_l$ and $\mathbf{u} = u_1, \ldots, u_m$ that is local around $\mathbf{x}$. The proof is by induction on the structure of φ. If φ is an atomic formula, it is basic local around $\mathbf{x}$ by definition. If φ is of the form $\neg\psi$ or $\psi_1 \vee \psi_2$ then the statement holds by the induction hypothesis.

In the only remaining case, φ is of the form $\exists y((\delta(y, \mathbf{x}) \leq r) \wedge \chi)$ for some r and some formula χ which is local around $\mathbf{x}$. By induction, χ is logically equivalent to a formula which is basic local around $\mathbf{x}$. By writing χ in disjunctive normal form, we get that χ is logically equivalent to a formula of the form $\bigvee_j \bigwedge_i \chi_{ji}$, where, for some r', every χ_{ji} is r'-local around x_i. Hence, φ is logically equivalent to a disjunction of formulas

$$\exists y((\delta(y,\mathbf{x}) \leq r) \wedge \bigwedge_i \chi_i) \ , \tag{*}$$

where every χ_i is r'-local around x_i. In the following, we assume w. l. o. g. that φ is of the form (*). In a sense, we have to distribute the quantification of y over the χ_i. The problem which arises is that some of the x_i might be close to each other, so that y might play a role for several χ_i simultaneously. To get around this problem we choose, for every "cluster" C of vertices x_i that are close to each other, a representative $v(C)$ such that quantification around any x_i of C can be replaced by quantification around $v(C)$. More formally, we proceed as follows. We associate with every structure A and tuple $\mathbf{a} = a_1, \dots, a_l$ of vertices of A a graph $G(A, \mathbf{a})$ with vertex set $V = \{1, \dots, l\}$ and edge set E, which contains the edge (i,j) whenever $\delta(a_i, a_j) \leq R := r + r' + 1$. We are going to construct, for every graph G on $\{1, \dots, l\}$, formulas θ_G and φ_G each of which is basic local around $\mathbf{x}$, such that, for every A and $\mathbf{a}$ it holds that

- $A \models \theta_G$ iff $G(A, \mathbf{a}) = G$, and
- if $G = G(A, \mathbf{a})$ then $[A \models \varphi_G \iff A \models \varphi]$.

Once we have established the existence of such formulas the statement of the lemma follows immediately because $\varphi \equiv \bigvee_G (\theta_G \wedge \varphi_G)$, where the disjunction is over all graphs G on $\{1, \dots, l\}$.

Let in the following a graph $G = (V, E)$ be fixed. The definition of θ_G is straightforward:

$$\theta_G := \bigwedge_{i \in V} (\bigwedge_{\substack{j \in V \\ (i,j) \in E}} (\delta(x_i, x_j) \leq R) \wedge \bigwedge_{\substack{j \in V \\ (i,j) \notin E}} (\delta(x_i, x_j) > R)) \ .$$

It is easy to see that, for every i, the i-th conjunct can be made R-local around x_i. In order to construct φ_G we choose from each connected component C of G a vertex $v(C)$. Then φ_G can be defined as

$$\bigvee_{C,\alpha} (\exists y[(\delta(y, x_{v(C)}) \leq R|C|) \wedge \alpha \wedge \bigvee_{i \in C} (\delta(y, x_i) \leq r) \wedge \bigwedge_{i \in C} \chi'_{i,C}] \wedge \bigwedge_{i \notin C} \chi'_{i,C,\alpha}) \ .$$

Here, the disjunction is over all connected components C of G and all (finitely many) elementary types α of $\mathbf{x}, \mathbf{u}, y$. If $i \in C$ then $\chi'_{i,C}$ is obtained from χ_i by making all quantifications R-local around $x_{v(C)}$. If $i \notin C$ then $\chi'_{i,C,\alpha}$ is obtained from χ_i by

- replacing every atomic subformula that contains the variable y and a variable that is bound in χ_i with false, and
- rewriting all other atomic formulas that refer to y according to α.

These replacements assure that y does not occur in $\chi'_{i,C,\alpha}$ and that every $\chi'_{i,C}$ and every $\chi'_{i,C,\alpha}$ is local around x_i. It should be pointed out that the internal subformula $[\cdots]$ can only become true if $\delta(,x_i) \leq r$ holds, for some $i \in C$, hence if $\delta(y, x_j) \leq r' + 1$ does not hold. This justifies the replacement of atomic subformulas that contain the variable y and a variable that is bound in χ_i with false. We note that already a smaller radius of y around $x_{v(C)}$ suffices if $v(C)$ is chosen closer to the "center" of C. □

Theorem 3.2 *Every FO formula is logically equivalent to a $\exists^*\forall$–local formula.*

Proof. Let Ψ be a first-order formula with free variables $\mathbf{u} = u_1, \ldots, u_m$. First we are going to show that Ψ is logically equivalent to a positive Boolean combination of $\exists^*\forall$–local formulas. In the following, if φ is r-local around its single variable, we write $\exists^r_l x\varphi(x)$ as an abbreviation for

$$\exists x_1, \ldots, x_l[\bigwedge_{i=1}^{l} \varphi(x_i) \wedge \bigwedge_{i \neq j}(\delta(x_i, x_j) > 2r)] \ .$$

Gaifman's theorem implies that Ψ is logically equivalent to a positive Boolean combination of formulas of the following three types.

1. $\exists^r_l x\varphi(x)$,
2. $\neg\exists^r_l x\varphi(x)$, and
3. local formulas around $\mathbf{u}$.

We are going to show that formulas of each of these types are logically equivalent to positive Boolean combinations of $\exists^*\forall$–local formulas. Formulas of type 1 are already $\exists^*$-local because $\delta(x_i, x_j) > 2r$ can be easily expressed by a formula which is $2r$-local around x_i. A formula $\varphi(\mathbf{u})$ of type 3 is logically equivalent to $\exists\mathbf{x}(\mathbf{x} = \mathbf{u} \wedge \varphi(\mathbf{x}))$. As, by Lemma 3.1, $\varphi(\mathbf{x})$ is logically equivalent to a formula that is basic local around $\mathbf{x}$ we obtain again a $\exists^*$-local formula. We still have to consider formulas of type 2. Let $\psi \equiv \neg\exists^r_l x\varphi(x)$. ψ is logically equivalent to

$$\neg\exists x\varphi(x) \vee \bigvee_{i=1}^{l-1}[\exists^r_i x\varphi(x) \wedge \neg\exists^r_{i+1} x\varphi(x)] \ .$$

Of course, $\neg\exists x\varphi(x)$ is equivalent to $\forall x\neg\varphi(x)$, which is of the required form. For every i, $\exists^r_i x\varphi(x) \wedge \neg\exists^r_{i+1} x\varphi(x)$ is logically equivalent to

$$\begin{aligned} \exists x_1, \ldots, x_i \forall y[&\bigwedge_{i} \varphi(x_i) \wedge \bigwedge_{i \neq j}(\delta(x_i, x_j) > 2r) \wedge \\ &\forall z_1, \ldots, z_{i+1} \neg(\bigwedge_{j}(\delta(z_j, \mathbf{x}) \leq 2r) \wedge \bigwedge_{i \neq j}(\delta(z_i, z_j) > 2r) \wedge \bigwedge_{j} \varphi(z_j)) \wedge \\ &\neg((\delta(y, \mathbf{x}) > 2r) \wedge \varphi(y))] \ , \end{aligned} \qquad (**)$$

expressing that there are $x_1, \ldots, x_i$ that fulfil φ, but

- neither there is a y which fulfils φ and is far from all the x_j,
- nor there exist different $z_1, \ldots, z_{i+1}$ all of which fulfil φ and are close to the x_j.

As the $[\cdots]$ part of this formula can be made local around $x_1, \ldots, x_i, y$, it is logically equivalent to a basic local formula around $x_1, \ldots, x_i, y$ by Lemma 3.1. We conclude that ψ is logically equivalent to a $\exists^*\forall$–local formula.

It remains to show that every positive Boolean combination of $\exists^*\forall$–local formulas is logically equivalent to a $\exists^*\forall$–local formula. Consider two $\exists^*\forall$–local formulas $\psi_1 \equiv \exists x_1, \ldots, x_k \forall y \varphi_1(\mathbf{x}, y)$ and $\psi_2 \equiv \exists x'_1, \ldots, x'_m \forall y' \varphi_2(\mathbf{x}', y')$. $\psi_1 \wedge \psi_2$ is equivalent to $\exists \mathbf{x}, \mathbf{x}' \forall y[\varphi_1(\mathbf{x}, y) \wedge \varphi_2(\mathbf{x}', y)]$ and $\psi_1 \vee \psi_2$ is equivalent to

$$\exists z, z', \mathbf{x}, \mathbf{x}' \forall y(z = z' \wedge \varphi_1(\mathbf{x}, y)) \vee (z \neq z' \wedge \varphi_2(\mathbf{x}', y)) .$$

□

In fact, by a closer inspection of Theorem 3.2, we can go one step further and show that local quantification around one single variable is enough.

Theorem 3.3 *Every first-order formula is logically equivalent to a formula of the form $\exists x_1, \ldots, x_l$ $\forall y \varphi(\mathbf{x}, y)$, where φ is local around y.*

Proof. We refer in the following to the threefold case distinction in the proof of Theorem 3.2. Formulas of type 1 can be transformed into

$$\exists x_1, \ldots, x_l \forall y \bigwedge_{i=1}^{l} [y = x_i \rightarrow (\varphi(y) \wedge \forall z((\delta(z, y) \leq 2r) \rightarrow \bigwedge_{j \neq i} (z \neq x_j)))] .$$

For formulas of type 3 we apply the same idea, obtaing a formula $\exists \mathbf{x} \forall y(\mathbf{x} = \mathbf{u} \wedge \varphi'(\mathbf{x}, y))$, where φ' is obtained by applying Lemma 3.1 and replacing every local subformula $\theta(x_i)$ by $y = x_i \rightarrow \theta(y)$.

For formulas of type 2 we first note that in (**) the subformulas of line 1 and 3 can be transformed into the correct form easily. The subformula of the second line, which does not refer to y, can be made local around $\mathbf{x}$ and therefore, again by Lemma 3.1, basic local around $\mathbf{x}$. By applying the same idea as in the cases 1 and 3 above, we obtain eventually a formula of the correct kind. □

There are straightforward analogues of Theorem 3.2 for other logics. Of special interest is the case of monadic Σ^1_1-logic, as quantification of unary relations does not change the locality properties of a structure. We get immediately that every monadic Σ^1_1-formula is logically equivalent to a formula of the form

$$\exists X_1, \ldots, X_l \exists x_1, \ldots, x_m \forall y \varphi ,$$

where φ is local around y. For classes of structures that have a connected Gaifman graph we can show even stronger normal forms. The basic idea is that global information about the structure can be transported along the relations and collected in a designated place. This generalizes a similar procedure in [Tho97b, Tho97a], where the transport is much more deterministic.

Theorem 3.4 *Let $\mathcal{C}$ be a class of structures with a connected Gaifman graph. Then the following hold.*

(a) On $\mathcal{C}$ every monadic Σ^1_1 formula is equivalent to a formula of the form $\exists X_1, \ldots, X_l \exists x \forall y \varphi$, where φ is local around y.

(b) If there exists a formula ρ that is local around its one free variable such that, for every structure A of $\mathcal{C}$, $A \models \exists! x \rho(x)$ then on $\mathcal{C}$ every monadic Σ^1_1 formula is equivalent to a formula $\exists X_1, \ldots, X_l \forall y \varphi$, where φ is local around y.

Proof. We only give sketches of the proofs.

(a) Let $\psi \equiv \exists X_1, \dots, X_l \exists x_1, \dots, x_m \forall y \varphi$, where φ is basic r-local around y. We explain how the x_i can be eliminated in favour of one x. We have to find a way to distinguish m vertices in a structure A. The idea is to guess a vertex x and m minimal paths $p_1, \dots, p_m$ in the Gaifman graph of A such that, for every i, p_i starts in x and ends in x_i. Note that we view these graph as directed although the Gaifman graph is an undirected graph. Every p_i is encoded by two unary relations Y_i, Z_i. Y_i contains all vertices of p_i. As p_i is a shortest path we can conclude that x and x_i have degree 1 w. r. t. p_i and all other vertices of p_i have degree 2 w. r. t. p_i. Z_i is used to give p_i an orientation. $v \in Z_i$ just in case v is the j-th vertex of p_i, for some j, (where x is the 0-th vertex) and j is congruent to one of 0,1,3 modulo 6.

It is straightforward that there exists a formula $\rho(x, y)$ that is local around y such that $\langle A, \mathbf{Y}, \mathbf{Z}, x \rangle \models \forall y \rho$ if and only if $\mathbf{Y}$ and $\mathbf{Z}$ encode m paths that have their starting point in x. ($\mathbf{Y}$ and $\mathbf{Z}$ might also encode some directed cycles but this does not matter.) It remains to show how in φ all references to variables x_j can be replaced. All atomic formulas θ in which a x_j occurs together with y or with a variable that is bound in φ (around y!) can be easily replaced by a formula $\exists z((\delta(z, y) \leq r + 1) \wedge \chi_i(z) \wedge \theta')$, where $\chi_i(z)$ tests that z is the sink of p_i (this can be checked 5-locally around z, hence $(r + 6)$-locally around y) and in θ' every occurrence of x_j is replaced by z.

To replace atomic formulas that only refer to variables of $\mathbf{x}$ (and $\mathbf{u}$, the free variables) we proceed as follows. For every atomic formula α which only contains variables from $\mathbf{x}$ and $\mathbf{u}$ we introduce a unary relation T_α. The intention is that $T_\alpha = U^A$ in the case that $\langle A, \mathbf{x}, \mathbf{u} \rangle \models \alpha$ and $T_\alpha = \emptyset$ otherwise. Because A is connected, the formula

$$\forall y[y \in T_\alpha \leftrightarrow \forall z((\delta(z, y) \leq 1) \rightarrow z \in T_\alpha)]$$

checks that T_α contains either all or no vertices. It is also easy to check by a 6-local formula around y (using some χ_i as needed) that T_α contains all vertices just in case the endpoints of (some of) the m paths behave according to α.

(b) In the construction of part (a), x only occurs in θ. Instead of guessing x and paths that start in x we can guess paths that start in the vertex that is distinguished by ρ.

□

Theorem 3.4 (b) can be generalized in a straightforward way to classes of structures with a bounded number of connected components. In the respective generalization of Theorem 3.4 (a) the number of connected components bounds the number of existentially quantified FO variables in the normal form. On the other hand, it can be seen by a simple game argument (cf. Section 4 below) that the set of two-vertex-graphs can not be characterized by a formula of the type considered in Theorem 3.4 (a).

Of course, Theorem 3.2 is also true with $\forall^*\exists$-local formulas in place of $\exists^*\forall$-local formulas. Although it is easy to see that $\exists^*$-local formulas do not capture all first-order properties (e. g. they cannot express the property "every vertex is coloured black"), it follows from results of Compton [Com83] that Boolean combinations (including negations!) of $\exists^*$-formulas do.

From Theorem 3.2 we can conclude that every first-order (and monadic Σ^1_1) formula on graphs (suitably represented by adjacency lists) can be evaluated by a nondeterministic Random-Access-Machine with unit-cost measure in time $O(nd^{O(1)})$, where n denotes the number of vertices and d the maximal vertex-degree.

4 Games

Ehrenfeucht games – invented in [Ehr61] building on work of Fraïssé [Fra54] – are an important tool for proving inexpressibility results in Mathematical Logic. In fact, in Finite Model Theory, where only finite structures are considered, they are the major tool available (cf. [Fag97]). To show that a given property P of finite structures is not expressible in FO logic it is enough to prove that the duplicator, one of two players, has a winning strategy in the ordinary FO Ehrenfeucht game for P (for a definition see e. g. [EF95]). Variants of Ehrenfeucht games are available for proving inexpressibility results for many other logics, including second-order logics [Ten75], existential second-order logics [Ten75, AF90], transitive closure logics [CM91] and finite variable logics [Bar77, Imm82].

Proving the existence of a winning strategy for the duplicator is often very difficult. To simplify such proofs the following approaches have been taken. There have been developed several conditions that assure that the duplicator has a winning strategy on two given structures: the Hanf-condition [FSV95], the Arora-Fagin condition [AF97] and the condition of Schwentick [Sch96] (for a survey see [Fag97]). All of these conditions exploit the fact that FO logic can only express combinations of local properties, i. e. properties of regions of bounded size. On the other hand there have been attempts to make the game easier to play for the duplicator. An important example is the invention of the Ajtai-Fagin game for existential second-order logic which allows the duplicator to choose the second structure *after* the spoiler (the duplicator's opponent) has selected relations for the first structure. The idea behind that game can be used whenever all formulas of a logic have an existential quantifier-prefix.

In this section we introduce local Ehrenfeucht games for FO logic and monadic Σ_1^1 logic. Additionally, we characterize the mentioned logics in terms of Hintikka-types (cf. Section 2). First of all, we describe the *local first-order Ehrenfeucht game* for a class $\mathcal{C}$ of σ-structures. Like the ordinary Ehrenfeucht game it is played by two players, called the spoiler and the duplicator. It has three parameters, l, r and d and consists of three stages.

Stage 1 The duplicator chooses a σ-structure $A \in \mathcal{C}$. The spoiler selects vertices $x_1, \ldots, x_l$ from A. Then the duplicator chooses a σ-structure $A' \notin \mathcal{C}$ and vertices $x'_1, \ldots, x'_l$ from A'.

Stage 2 The spoiler chooses a vertex y' from A', then the duplicator chooses a vertex y from A.

Stage 3 The spoiler and the duplicator play an ordinary d-round Ehrenfeucht game on the structures $\langle N^r(y), x_1, \ldots, x_l\rangle$ and $\langle N^r(y'), x'_1, \ldots, x'_l\rangle$.

Here $\langle N^r(y), x_1, \ldots, x_l\rangle$ denotes the structure that is induced by $S^r(y)$ and $\mathbf{x}$ and has $x_1, \ldots, x_l$ and y as distinguished elements. The spoiler wins the game if he wins the game of stage 3 in the usual sense. Otherwise, the duplicator wins.

Theorem 4.1 *Let $\mathcal{C}$ be a class of σ-structures. The following are equivalent.*

1. *$\mathcal{C}$ is first-order definable.*

2. *For some l, r and d, the spoiler has a winning strategy in the local FO Ehrenfeucht game on $\mathcal{C}$ with parameters l, r and d.*

3. *There exists a set H of Hintikka-types such that, for some l, r and d, for every σ-structure A it holds that $A \in \mathcal{C}$ if and only if there exist $x_1, \ldots, x_l \in A$ with $\{H^d(\langle N^r(y), x_1, \ldots, x_l \rangle) \mid y \in U^A\} \subseteq H$.*

Proof. $(1) \Longrightarrow (2)$: If $\mathcal{C}$ is first-order definable, by Theorem 3.2, there is a formula $\exists x_1, \ldots, x_l \forall y \varphi$ that characterizes the structures of $\mathcal{C}$. Furthermore φ is basic r-local around $\mathbf{x}, y$, for some r. Let d denote the quantifier-depth of φ. It follows in a straightforward manner that the spoiler has a winning strategy in the local first-order Ehrenfeucht game on $\mathcal{C}$ with parameters l, r and d (compare with [AF90]).

$(2) \Longrightarrow (1)$: From a winning strategy of the spoiler in the local first-order Ehrenfeucht game on $\mathcal{C}$ with parameters l, r and d one can easily derive a winning strategy in the ordinary $(l+1+r+d)$-round Ehrenfeucht game on $\mathcal{C}$. The spoiler simply plays $l+1+r$ rounds according to his winning strategy in the local game. The additional d rounds assure that the spoiler immediately has a win if, in stage 3, the duplicator does not play in the d-neighbourhoods of y and y'.

$(1) \Longrightarrow (3)$: This follows immediately from Theorem 3.3.

$(3) \Longrightarrow (1)$: This holds because Hintikka-types can be described by first-order formulas. □

The local FO Ehrenfeucht game can be easily adapted for the case of monadic Σ_1^1-logic. The resulting game has one additional parameter m. In stage 1 of the *local monadic Σ_1^1 Ehrenfeucht game* the spoiler chooses, before the duplicator chooses A', unary relations $X_1, \ldots, X_m$ and vertices $x_1, \ldots, x_l$ in A and the duplicator has to choose corresponding relations $X'_1, \ldots, X'_m$ and vertices $x'_1, \ldots, x'_l$ in A'.

Theorem 4.2 *Let $\mathcal{C}$ be a class of σ-structures. The following are equivalent.*

1. *$\mathcal{C}$ is monadic Σ_1^1 definable.*

2. *For some m, l, r and d, the spoiler has a winning strategy in the local monadic Σ_1^1 Ehrenfeucht game on $\mathcal{C}$ with parameters m, l, r and d.*

3. *There exists a set H of Hintikka-types such that, for some m, l, r and d, for every σ-structure A it holds that*

$$A \in \mathcal{C} \iff \begin{array}{l}\textit{there exist unary relations } X_1, \ldots, X_m \textit{ and elements } x_1, \ldots, x_l \\ \textit{with } \{H^d(\langle N^r(y), X_1, \ldots, X_m, x_1, \ldots, x_l \rangle) \mid y \in U^A\} \subseteq H\ .\end{array}$$

The proof is an easy generalization of the proof of Theorem 4.1.

In a similar manner one can derive respective games from Theorem 3.4. Although we have not used Theorems 4.1 and 4.2 to derive any new inexpressibility results we are optimistic that the local Ehrenfeucht games will turn out to be a useful tool to get such results. One indication in this direction is the fact that many of the inexpressibility proofs that are given in the literature can be proved by using these games (e. g. [FSV95, Sch95]).

One particular example is the proof of Ajtai and Fagin [AF90], which has already been simplified by Arora and Fagin in [AF97]. In both of these proofs the notion of the (r, d)-colour of a vertex or an edge is used. As the graphs that were used in those proofs do not contain small cycles all r-spheres are rooted trees. The definition assures that if two vertices have the same (r, d)-colour their r-neighbourhoods have the same d-Hintikka-type. As in the construction of [AF90] both structures have exactly the same multisets of (r, d)-colours Theorem 4.2 implies their result. It should be noted that the additional choice of vertices $x_1, \dots, x_l$ does not make the construction of the graphs more difficult, as they can almost be treated like l additional unary relations with the proviso that each of these colours appears in every graph exactly once.

5 Automata

The conditions (3) of Theorems 4.1 and 4.2 give rise to a generalized form of automata. In this section we are going to introduce automata models for FO logic and monadic Σ^1_1 logic, respectively. Vaguely similar machine models are known for relational databases [Lei89, GPPdB94, AV95]. Informally the FO automaton works as follows. First it nondeterministically pebbles vertices $b_1, \dots, b_g$ of its input structure A, for some g. Then, for every vertex a of A, it inspects in a constant number of steps (alternating between nondeterminism and parallelism) the neighbourhood of a. Navigation through the neighbourhood is only along edges.

We now define the model more formally. Let σ be a relational signature. A first-order *σ-automaton M* consists of a tuple (g, l, I, φ), where $g \geq 0$ is the size of the *global read-only store*, $l \geq 0$ is the size of the *local store*, I is a finite sequence of *instructions* and φ is a *test*. The store will hold a vector of (pointers to) elements of the structure. Instructions are of the form $\langle \text{any } i, j\rangle$ or $\langle \text{all } i, j\rangle$, where $1 \leq i, j \leq l$. The test is a quantifier-free formula with variables from $x_1, \dots, x_l, y_1, \dots, y_g$. A *configuration* $(J, \mathbf{b}, \mathbf{a})$ consists of a (possibly empty) sequence J of instructions yet to be executed and the contents $\mathbf{b}$ and $\mathbf{a}$ of the global and local store, respectively. Let $\mathbf{b} \in A^l$ and $v \in A$. A *configuration tree of M for $\mathbf{b}$ and v* is defined as follows. It is a rooted tree, directed from the root to the leaves, and has configurations of M as vertices. It has the *start configuration* $(I, \mathbf{b}, v^l)$ as its root. The leaves are *terminal configurations* $(\varepsilon, \mathbf{b}, \mathbf{a})$. An inner vertex $(\iota J, \mathbf{b}, \mathbf{a})$ has

- one child $(J, \mathbf{b}, \mathbf{a}')$, where a'_j is a_i or a neighbour of a_i and $a'_k = a_k$ for every $k \neq i$, if ι is $\langle \text{any } i, j\rangle$, and
- all children $(J, \mathbf{b}, \mathbf{a}')$, where a'_j is a_i or a neighbour of a_i and $a'_k = a_k$ for every $k \neq i$, if ι is $\langle \text{all } i, j\rangle$.

A terminal configuration $(\varepsilon, \mathbf{b}, \mathbf{a})$ is *accepting*, if $\langle A, \mathbf{a}, \mathbf{b}\rangle \models \varphi(\mathbf{x}, \mathbf{y})$, otherwise *rejecting*. A configuration tree is *accepting*, if all its leaves are accepting. We say that M *accepts* A if there is a $\mathbf{b} \in A^g$ such that for every $v \in A$ there is an accepting configuration tree of M with $\mathbf{b}$ and a.

A monadic Σ^1_1 σ-automaton similarly consists of a tuple (Q, g, l, I, φ), where g, l, I, φ are as before and Q is a finite set of *states*. It starts by nonterministically selecting a mapping $f : U^A \to Q$ (represented by unary relations F_q, one for each $q \in Q$). Afterwards it continues like the FO automaton (where the input structure is extended by the F_q).

Theorem 5.1 *A class of σ-structures is first-order definable if and only if it is accepted by a first-order σ-automaton.*

Proof. "only if": By Theorem 3.3, $\mathcal{C}$ can be defined by a $\exists^*\forall$–local formula $\exists x_1, \dots, x_g \forall y \psi$ (without free variables), where ψ is in prenex normal form with d quantifiers (r-local around y) numbered from left to right with $1, \dots, d$. Let $l = d + 1$ and let I consist of the subsequences $J_1, \dots, J_d$, where

- J_i is $\langle \text{any } l, i\rangle$ followed by $r - 1$ times $\langle \text{any } i, i\rangle$ (nondeterministic choice of an element of distance $\leq r$ from a) if the i-th quantifier is existential, and

- J_i is $\langle$all $l, i\rangle$ followed by $r - 1$ times $\langle$all $i, i\rangle$ otherwise.

φ is the quantifier-free part of ψ.

"if": Consider an automaton (g, l, I, φ). We use one variable y_i for each of the l local pointers. We assign inductively a formula ψ_J to every suffix J of I as follows:

- $\psi_\varepsilon \equiv \varphi$
- $\psi_{\iota J} \equiv \exists y((\delta(y_i, y) \leq 1) \wedge \exists y_j((y_j = y) \wedge \psi_J))$ if ι is $\langle$any i $j\rangle$. (The extra variable y ensures that this construction also works for $i = j$.)
- $\psi_{\iota J} \equiv \forall y((\delta(y_i, y) \leq 1) \wedge \forall y_j((y_j = y) \wedge \psi_J))$ if ι is $\langle$all i $j\rangle$.

The acceptance condition is then equivalent to

$$\exists x_1, \dots, x_g \forall y \exists y_1, \dots, y_l (\bigwedge_{1 \leq m \leq l} (y_m = y) \wedge \psi_I) \ .$$

$\square$

Corollary 5.2 *A class of σ-structures is monadic Σ^1_1 definable if and only if it is accepted by a monadic Σ^1_1 automaton.*

For structures of bounded degree, the monadic Σ^1_1 automata generalize those of Thomas [Tho91, Tho97b, Tho97a]. In this case a Hintikka-type of a small neighbourhood boils down to an isomorphism type, called *tile*. The automata of Thomas check that each vertex possesses a neighbourhood of one among a finite number of allowed isomorphism types. Moreover, some of them must occur *at least* a certain number of times and others *at most* a certain number of times. (This is the remainder of Hanf's condition, from which it is actually derived.) Inspecting Theorem 3.3 again with this in mind, one finds that φ describes all admissible tiles, while the center y runs over all vertices. The variables $\mathbf{x}$ are used to distinguish a finite number of vertices. It is therefore clear that it suffices to require that some tiles (namely those with a center in $\mathbf{x}$) occur *exactly* once. Moreover, for structures with a connected Gaifman graph, Theorem 3.4 implies that *one* occurrence constraint of this form suffices.

We would also like to compare monadic Σ^1_1 automata to the algebraic automata introduced by Courcelle [Cou90]. On general finite relational structures these automata are very powerful. They capture MSO logic with a built-in linear order [Cou92, Cou96]. And on grids, for example, this inclusion is strict. More suitable for a comparison are sets of *unordered, unranked trees*, which we simply call trees in the sequel. On these structures, the recognizing power of algebraic automata is exactly the same as definability in monadic Σ^1_1 logic with a built-in linear order, as will be shown below. Hence the difference between algebraic automata and monadic Σ^1_1 automata corresponds to the difference between monadic Σ^1_1 with and without a built-in linear order, two logics which have different expressive power [Cou90]. (For example, the linear order allows to count branches.) That such a linear order comes into play on an intentionally unordered structure is obviously due to the fact that trees must be encoded as terms before they are input to an automaton. The parallel inspection of local neighbourhoods in monadic Σ^1_1 automata avoids such side-effects. This is remarkable because Theorem 3.4 (b) applies and ensures that monadic Σ^1_1 automata do not need a global store in this case. Therefore, global information about a tree can be collected at the root without referring to some arbitrary order.

We now introduce trees and algebraic automata more formally. (Compare [Cou90].) A tree contains a distinguished vertex, its root. Edges are directed from the root to the leaves. They carry labels from a finite set L_E. Vertices are labelled with sets of elements from a finite set L_V. There is no a priori upper bound on the degree of a vertex. In particular, the degree of a vertex is not determined by its labelling. The class of trees (up to isomorphism) over L_V, L_E can be obtained inductively as follows. A few simple trees are assumed to be given. They consist solely of their root, which can be unlabelled or carry a label $l \in L_V$. We denote these one-vertex trees by constants 1 and l, respectively. Then certain operations allow to construct new trees out of one or two others. For every label $l \in L_E$ there is a unary operation $\hat{l}$ adding to its argument a new (unlabelled) root and an l-labelled edge to the old root. Finally, a binary operation $\cdot$ glues its arguments at their roots, taking the union of their respective sets of root labels. It is easy to see that these operations can generate all trees and only those. In general, there is more than one way to obtain a particular tree.

An algebraic automaton recognizing a set of trees is an ordinary deterministic tree automaton working on the inductive representations of the trees (their derivation trees with respect to a grammar generating all trees). It is required that it accepts or rejects a tree independently of its representation, that is, all runs on different representations of the same tree produce the same answer. (Because logical formulas are also compatible to these inductive representations, algebraic automata are useful to obtain decision procedures.) Formally, an algebraic automaton consists of a finite set $A = \{1, \ldots, m\}$ of states, a transition function $\delta\colon (\{1\} \cup L_V \cup \{\hat{l} \mid l \in L_E\} \times A \cup A \times \{\cdot\} \times A) \to A$ and a subset $F \subseteq A$ of accepting states. δ turns A into an algebra with respect to the tree-building operations: We write t for $\delta(t)$, $t \in \{1\} \cup L_V$, $\hat{l}(a)$ for $\delta(\hat{l}, a)$, and $a_1 \cdot a_2$ for $\delta(a_1, \cdot, a_2)$. The information contained in all the possible runs of the automaton can conveniently be expressed as a homomorphism from the set of all terms into A. We may assume that the homomorphism is surjective, that is, A does not contain superfluous states. Therefore, because results do not depend on representations, A is a commutative monoid with respect to $\cdot$ and has 1 as its neutral element. Moreover, $l \cdot l = l$ for every $l \in L_V$. For all $a_1, \ldots, a_n \in A$, $n \geq 0$, the value of $\prod_{i=1}^{n} a_i$ only depends on the "Parikh image" $\|a_1 \cdots a_n\| \in \mathbb{N}^m$, where the ith component is $|\{k \mid a_k = i\}|$. This means that there is a mapping $f\colon \mathbb{N}^m \to A$ such that $\prod_{i=1}^{n} a_i = f(\|a_1 \cdots a_n\|)$.

Theorem 5.3 *A set of trees is recognizable (by algebraic automata) if and only if it is definable in monadic Σ_1^1 logic with the help of a built-in linear order.*

That is, we consider monadic Σ_1^1 formulas which may use a linear order on the vertices of a tree but must remain invariant if the linear order is replaced with a different one.

We adapt the proof in [Cou90], which started from slightly different operations and aimed at a slightly different statement. There is no substantial new idea, only the defining formula has a more restricted form.

Proof. The only-if part was shown in [Cou92, Cou96]. Since our operations are derivable from the general (so-called quantifier-free definable) operations on graphs [Cou92], a tree which is recognizable as a graph is a fortiori recognizable with respect to tree operations. For the if-part consider a tree T with set of vertices V, set of edges $E \subseteq V \times V$ and root r. Let $V_l \subseteq V$ be the set of vertices labelled with $l \in L_V$, and let $E_l \subseteq E$ be the set of vertex pairs which are connected by an edge labelled with $l \in L_E$. We write T_v for the subtree of T rooted at $v \in V$. Of course, $T_r = T$. We determine a mapping $\partial\colon V \to A$

inductively, moving from the leaves to the root. Since we can choose the representation of T such that

$$T_v = \prod_{\substack{l \in L_V \\ v \in V_l}} l \cdot \prod_{\substack{l \in L_E \\ v' \in V \\ (v,v') \in E_l}} \hat{l}(T_{v'}) \ ,$$

we have

$$\begin{aligned} \partial(v) &= \prod_{\substack{l \in L_V \\ v \in V_l}} l \cdot \prod_{\substack{l \in L_E \\ v' \in V \\ (v,v') \in E_l}} \hat{l}(\partial(v')) \\ &= \prod_{\substack{l \in L_V \\ v \in V_l}} l \cdot f(\|\langle \hat{l}(\partial(v')) \mid l \in L_E, v' \in V, (v,v') \in E_l \rangle\|) \ . \end{aligned} \tag{$*$}$$

Acceptance of T is equivalent to the existence of ∂ satisfying $(*)$ and the condition $\partial(r) \in F$. This can be expressed in monadic Σ_1^1 logic with the help of a built-in linear order. As usual, the existence of pairwise disjoint sets $X_i = \partial^{-1}(i)$, $i \in A$, is required. Besides that, each set $\bigcup_{v \in X_i} Y_{i,v}$, $i \in A$, where

$$Y_{i,v} = \{v' \in X_j \mid j \in A, (\exists l \in L_E)\,((v,v') \in E_l \wedge \hat{l}(j) = i)\}$$

is partitioned into q_i sets such that they can be used (together with the built-in linear order) to determine the cardinality of $Y_{i,v}$ modulo q_i for every $v \in X_i$. For suitable $q_i \geq 1$, $i \in A$, this is enough to compute f, hence express $(*)$.

For every $a \in A$, the sequence $a^0, a^1, a^2, \ldots$ becomes ultimately periodic. Let $p_a \geq 0$, $q_a \geq 1$ be the smallest values such that $a^{p_a} = a^{p_a + q_a}$. It is easy to see that $f^{-1}(a)$ is a finite union of sets of the form

$$\{(n_1, \ldots, n_m) \in \mathbb{N}^m \mid n_i = \bar{p}_i + k_i \bar{q}_i, k_i \geq 0\}$$

with $\bar{p}_i, \bar{q}_i \geq 0$. More precisely, for each $i \in A$ independently, either $\bar{p}_i < p_i$ and $\bar{q}_i = 0$, or $p_i \leq \bar{p}_i < p_i + q_i$ and $\bar{q}_i = q_i$. Therefore, $f^{-1}(a)$ can be described by a first-order formula for every $a \in A$. □

6 Conclusion

We introduced a normal form for first-order logic that formalizes the intuition that this logic is only able to express properties of the form "there are some important parts of the structure that fulfil given conditions and everywhere else nothing forbidden happens". Although our normal forms are mainly useful when structures of unbounded diameter are considered (otherwise the whole structure is contained in the neighbourhood of all of its vertices and local formulas are just general first-order formulas) their translation into the language of automata gives uniform means of evaluating first-order properties "along the edges" of structures. On the other hand it is applicable to structures of unbounded degree, even in situations where the Hanf argument does not work.

Our main open question is whether the normal forms have other meaningful applications and to find new inexpressibility results with the help of the simplified games. Another question is whether automata models can be designed that inspect the neighbourhoods of vertices in a more deterministic fashion.

Acknowledgements

This investigation was inspired by a talk that was given by Wolfgang Thomas in Mainz in December 1996. We would like to thank him, Clemens Lautemann, Juha Nurmonen, Ron Fagin for many fruitful discussions and suggestions. Thanks also to the anonymous referees.

References

[AF90] M. Ajtai and R. Fagin. Reachability is harder for directed than for undirected finite graphs. *Journal of Symbolic Logic*, 55(1):113–150, 1990.

[AF97] Sanjeev Arora and Ronald Fagin. On winning strategies in Ehrenfeucht–Fraïssé games. *Theoretical Computer Science*, 174(1–2):97–121, 1997.

[AV89] S. Abiteboul and V. Vianu. Fixpoint extensions of first-order logic and Datalog-like languages. In *Proc. 4th IEEE Symp. on Logic in Computer Science*, pages 71–79, 1989.

[AV95] S. Abiteboul and V. Vianu. Computing with first-order logic. *Journal of Computer and System Sciences*, 50:309–335, 1995.

[Bar77] J. Barwise. On Moschovakis closure ordinals. *Journal of Symbolic Logic*, 42:292–296, 1977.

[CM91] A. Calò and J. Makowsky. The Ehrenfeucht-Fraïssé games for transitive closure logic. Unpublished manuscript, 1991.

[Com83] K. Compton. Some useful preservation theorems. *Journal of Symbolic Logic*, 48:427–440, 1983.

[Cou90] B. Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[Cou92] B. Courcelle. The monadic second-order logic of graphs VII: Graphs as relational structures. *Theoretical Computer Science*, 101:3–33, 1992.

[Cou96] B. Courcelle. The monadic second-order logic of graphs X: Linear orderings. *Theoretical Computer Science*, 160:87–143, 1996.

[DLW97] G. Dong, L. Libkin, and L. Wong. Local properties of query languages. In *Proc. Int. Conf. on Database Theory*, LNCS, pages 140–154. Springer-Verlag, 1997.

[EF95] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[Ehr61] A. Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fund. Math.*, 49:129–141, 1961.

[Fag74] R. Fagin. Generalized first–order spectra and polynomial–time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7*, pages 43–73, 1974.

[Fag97] R. Fagin. Easier ways to win logical games. In *Proceedings of the DIMACS Workshop on Finite Models and Descriptive Complexity*. American Mathematical Society, 1997.

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

[FSV95] R. Fagin, L. Stockmeyer, and M. Vardi. On monadic NP vs. monadic co-NP. *Information and Computation*, 120:78–92, 1995. Preliminary version appeared in 1993 IEEE Conference on Structure in Complexity Theory, pp. 19-30.

[Gai82] H. Gaifman. On local and nonlocal properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North Holland, 1982.

[GPPdB94] M. Gemis, J. Paredaens, P. Peelman, and J. Van den Bussche. A computational model for generic graph functions. In H.-J. Schneider and H. Ehrig, editors, *Graph Transformations in Computer Science*, number 776 in Lecture Notes in Computer Science, pages 170–187. Springer, 1994.

[Grä92] E. Grädel. Capturing complexity classes by fragments of second order logic. *Theoretical Computer Science*, 101:35–57, 1992. A preliminary version appeared in Proceedings of 6th IEEE Conference on Structure in Complexity Theory, Chicago 1991, 341–352.

[Han65] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*, pages 132–145. North Holland, 1965.

[Imm82] N. Immerman. Upper and lower bounds for first-order expressibility. *Journal of Computer and System Sciences*, 25:76–98, 1982.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Imm87] N. Immerman. Expressibility as a complexity measure: results and directions. In *Second Structure in Complexity Conference*, pages 194–202, 1987.

[Lei89] D. Leivant. Descriptive characterizations of computational complexity. *Journal of Computer and System Sciences*, 39:51–83, 1989.

[Lib97] L. Libkin. On the forms of locality over finite models. In *Proc. 12th IEEE Symp. on Logic in Computer Science*, 1997.

[LW98] L. Libkin and L. Wong. Unary quantifiers, transitive closure, and relations of large degree. In M. Morvan, Ch. Meinel, and D. Krob, editors, *STACS 98*, number 1373 in Lecture Notes in Computer Science, pages 183–193. Springer, 1998.

[Sch95] T. Schwentick. Graph connectivity, monadic NP and built-in relations of moderate degree. In *Proc. 22nd International Colloq. on Automata, Languages, and Programming (ICALP 95)*, pages 405–416, 1995.

[Sch96] T. Schwentick. On winning Ehrenfeucht games and monadic NP. *Annals of Pure and Applied Logic*, 79:61–92, 1996.

[Ten75] R. Tenney. Second-order Ehrenfeucht games and the decidability of the second-order theory of an equivalence relation. *Journal of the Australian Mathematical Society*, 20:323–331, 1975.

[Tho91] W. Thomas. On logics, tilings and automata. In J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, editors, *Proc. ICALP'91*, number 510 in Lecture Notes in Computer Science, pages 441–453. Springer, 1991.

[Tho97a] W. Thomas. Automata theory on trees and partial orders. In M. Bidoit and M. Dauchet, editors, *Proc. TAPSOFT'97*, number 1214 in Lecture Notes in Computer Science, pages 20–38. Springer, 1997.

[Tho97b] W. Thomas. Elements of an automata theory over partial orders. In D. A. Peled, V. R. Pratt, and G. J. Holzmann, editors, *Partial Order Methods in Verification*, number 29 in DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1997.

[Var82] M. Y. Vardi. The complexity of relational query languages. In *Proc. 14th ACM Symp. on Theory of Computing*, pages 137–146, 1982.