# Homomorphism Closed vs. Existential Positive

Tomás Feder[*] Moshe Y. Vardi[†]

## Abstract

Preservations theorems, which establish connection between syntactic and semantic properties of formulas, are a major topic of investigation in model theory. In the context of finite-model theory, most, but not all, preservation theorems are known to fail. It is not known, however, whether the Łos-Tarski-Lyndon Theorem, which asserts that a 1st-order sentence is preserved under homomorphisms iff it is equivalent to an existential positive sentence, holds with respect to finite structures. Resolving this is an important open question in finite-model theory.

In this paper we study the relationship between closure under homomorphism and positive syntax for several non-1st-order existential logics that are of interest in computer science. We prove that the Łos-Tarski-Lyndon Theorem holds for these logics. The logics we consider are variable-confined existential infinitary logic, Datalog, and various fragments of second-order logic.

## 1 Introduction

A preservation theorem in model theory is a result that asserts that if a formula of some logic has certain preservation properties (i.e., if it is true on some structure, then it is also true on every structure obtained from that structure via certain algebraic operations), then the formula is actually equivalent to a formula in some syntactically restricted class such that every formula in that class clearly possesses the preservation property at hand. Preservations theorems, which establish connection between syntactic and semantic properties of formulas, are a major topic of investigation in model theory [4]. Several preservation theorems are known to hold for 1st-order logic. For example, the Łos-Tarski Theorem asserts that a 1st-order theory is preserved under extensions iff it is equivalent to a existential sentence. Lyndon's Theorem asserts that a 1st-order theory is preserved under surjective homomorphisms iff it is equivalent to a positive sentence [4]. (A *homomorphism* is a mapping from the universe of a structure $\mathbf{A}$ to the universe of of a structure $\mathbf{B}$ (over the same signature) that maps every tuple in a relation $R^{\mathbf{A}}$ to a tuple in the corresponding relation $R^{\mathbf{B}}$. A class $\mathcal{C}$ is closed under surjective homomorphisms if $\mathbf{A} \in \mathcal{C}$ implies that $h(\mathbf{A}) \in \mathcal{C}$, for an arbitrary mapping $h$.) The proof techniques can be easily combined to show that a 1st-order theory is preserved under (not necessarily surjective) homomorphisms iff it is equivalent to an existential positive sentence. (A class $\mathcal{C}$ is closed under homomorphisms if $\mathbf{A} \in \mathcal{C}$ implies that $\mathbf{B} \in \mathcal{C}$, whenever there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.) We slighly abuse terminology and refer to this as the Łos-Tarski-Lyndon Theorem.

In the context of finite-model theory, the model theory of finite structures, most, but not all, preservation theorems are known to fail, including, for example, the Łos-Tarski Theorem [6]. It is not known, however, whether Lyndon's Theorem holds with respect to finite structures. It is also not known whether the Łos-Tarski-Lyndon Theorem holds with respect to finite structures (for a closely related failure, see [2]). Resolving the latter problem is an important and well-known open question in finite-model theory (see Problem 1.9 on the finite-model theory website[1] and an incorrect claim in [11]). On one hand, the class of existential positive 1st-order formulas corresponds to the class of *select-project-join-union* database queries, which are the queries that most often occur in practice [1]. On the other hand, classes of unsatisfiable instances of constraint-satisfaction problems, e.g., all non-3-colorable graphs, are known to be closed under homomorphism [8]. An extensive treatment of preservation theorems in finite-model theory can be found in [9, 18].

While the main focus of model theory has been on 1st-order logic, finite-model theory studies also many non-1st-order logics, as it is well known that the expressive power of 1st-order logic over finite-structures is rather limited [6, 13]. In this paper we study the relationship between homomorphism closure and existential positive syntax for several non-1st-order logics that are of interest in computer science. We focus on existential logics, for which preservations under extensions holds trivially, and ask whether for these logics homomorphism closure yields positivity. We prove that this is the case for several logics. Most of our proofs are constructive; we start with a given formula in a logic and then show how to eliminate

[*] tomas@theory.stanford.edu

[†] Rice University, vardi@cs.rice.edu, supported in part by NSF grants CCR-9988322, CCR-0124077, IIS-9908435, IIS-9978135, and EIA-0086264.

[1] http://wwwmgi.informatik.rwth-aachen.de/FMT/

negated atoms and inequalities[2] over homomorphism-closed classes.

We start by examining the logic $\exists L^\omega$, a variable-confined infinitary logic that was studied extensively in finite-model theory, cf. [16, 15]. We show that $\exists L^\omega$ sentences with negated atoms and inequalities over homomorphism-closed classes of structures can be rewritten without inequalities or negated atoms. The proof uses an asymmetric pebble game that is known to characterize the expressive power of this logic [15]. We then prove this result also for *Datalog*, the language of function-free logic programs (known also as Horn-clause programs), which plays a major role in database theory [1]. Datalog is essentially the existential fragment of fixpoint logic, i.e., the closure of existential 1st-order logic under recursion [3], and is also an effective fragment of $\exists L^\omega$ [15]. Our proof for Datalog is constructive and uses an analysis of Datalog derivation trees.

Datalog can also be viewed as a fragment of the extension of existential 1st-order logic with universal 2nd-order quantification, so we now turn our attention to the latter logic. The extension of existential 1st-order logic with universal 2nd-order quantification is of interest because its dual, the extension of universal 1st-order logic with existential 2nd-order quantification is the class *SNP*, a syntactic fragment of NP (via the characterization of NP in terms of existential 2nd-order logic [7]), which plays an important role in the theory of approximation algorithms [17]. The restriction to monadic 2nd-order quantification is also of interest, because the extension of universal 1st-order logic with existential monadic 2nd-order quantification, the class *monadic SNP*, captures the class of *constraint-satisfaction problems*, which is of major interest in artificial intelligence and complexity theory [8]. We relate closure under homomorphism and positivity to the complements of both SNP and monadic SNP. In fact, we can extend our results also to the complement of *binary SNP*, where the 2nd-order quantification is binary (cf. [5]), but our techniques, which are based on the probabilistic method and Ramsey bounds, do not seem to be able to handle 2nd-order quantification of any fixed arity beyond two.

The proof presented here for Datalog holds over both for classes of general structures as well as classes of finite structures. The proofs for $\exists L^\omega$ and for the various SNP classes are over the class of finite structures. An application of the Compactness Theorem shows that the results for the various SNP classes hold for arbitrary structures as well. Finally we show that the main property studied in this paper, namely preservation under homomorphisms, is undecidable all of our effective logics ($\exists L^\omega$ is not effective).

[2] All the inequalities in this paper are of the form $x \neq y$.

## 2 The logic $\exists L^\omega$

.

The logic $\exists L^k(\neq, \neg)$ is a variable-confined existential infinitary logic: we start with $k$ individual variables, then form atomic formulas, negated atomic formulas, and inequalities, and then form formulas by closing under existential quantification and under infinitary conjunctions and disjunctions. We get $\exists L^k()$ if we start without negated atomic formulas and inequalities. The logic $\exists L^\omega(\neq, \neg)$ (resp., $\exists L^\omega()$) is obtained by taking the union of $\exists L^k(\neq, \neg)$ (resp., $\exists L^k()$), for all $k > 0$. These logic play an important role in finite-model theory, since they capture many other non-1st-order logics and since their expressive power has been characterized in terms of pebble games [15, 16].

The *weak existential $k$-pebble game* between the *Spoiler* and the *Duplicator* on the structures $\mathbf{A}$ and $\mathbf{B}$ is played as follows. Each player has $k$ pebbles: the Spoiler has pebbles $p_1, \ldots, p_k$ and the Duplicator has pebbles $q_1, \ldots, q_k$. A position in the game is a placement of some of the pebbles on the elements of $\mathbf{A}$ and $\mathbf{B}$; the $p_i$'s are placed on elements of $\mathbf{A}$ and the $q_i$'s are placed on elements of $\mathbf{B}$. Initially, no pebble is placed. In each round of the game, the Spoiler places (resp., removes) a pebble $p_i$ on (resp., from) an element of $\mathbf{A}$; the Duplicator responds with a similar move with the pebble $q_i$ on $\mathbf{B}$.

Let $i_1, \ldots, i_m$ be the indices of the pebbles that are placed on $\mathbf{A}$ (and $\mathbf{B}$) after the $i$-th round. Let $a_{i_1}, \ldots, a_{i_m}$ (resp., $b_{i_1}, \ldots, b_{i_m}$) be the elements of $\mathbf{A}$ (resp., $\mathbf{B}$) x pebbled by the pebbles $p_{i_1}, \ldots, p_{i_m}$ (resp., $q_{i_1}, \ldots, q_{i_m}$) after the $i$-th round. If the mapping $h$ with $h(a_{i_j}) = b_{i_j}$, for $1 \leq j \leq m$, is not a *homomorphism* between the substructures of $\mathbf{A}$ and $\mathbf{B}$ with universes $\{a_{i_1}, \ldots, a_{i_m}\}$ and $\{b_{i_1}, \ldots, b_{i_m}\}$, respectively, then the Spoiler wins the game. Otherwise, the game continues. The Duplicator wins the game if he has a *winning strategy* that allows him to continue playing "forever", i.e., if he can block the Spoiler from winning the game.

Define the *strong existential $k$-pebble game* similarly to the weak existential $k$-pebble game above, except that the Spoiler now wins if the mapping $h$ with $h(a_{i_j}) = b_{i_j}$, for $1 \leq j \leq m$, is not an *isomorphism* between the substructures of $\mathbf{A}$ and $\mathbf{B}$ with universes $\{a_{i_1}, \ldots, a_{i_m}\}$ and $\{b_{i_1}, \ldots, b_{i_m}\}$, respectively.

**Lemma 1** [16, 15]

- *A class $\mathcal{C}$ of finite structures is $\exists L^k()$-definable if and only if, for every pair of structures $\mathbf{A} \in \mathcal{C}$ and $\mathbf{B} \notin \mathcal{C}$, the Spoiler has a winning strategy for the weak existential $k$-pebble game.*
- *A class $\mathcal{C}$ of finite structures is $\exists L^k(\neq, \neg)$-definable if and only if, for every pair of structures $\mathbf{A} \in \mathcal{C}$ and*

$\mathbf{B} \notin \mathcal{C}$*, the Spoiler has a winning strategy for the strong existential* $k$*-pebble game.*

We use the characterizations given by this lemma to obtain the following result:

**Theorem 1** *Over homomorphism-closed classes of finite structures,* $L^k()$ *is as expresssive as* $L^k(\neq, \neg)$.

PROOF SKETCH: Suppose a homomorphism-closed class $\mathcal{C}$ of finite structures is not $\exists L^k()$-definable. Then, by Lemma 1, there is a pair of structures $\mathbf{A} \in \mathcal{C}$ and $\mathbf{B} \notin \mathcal{C}$ such that the Duplicator has a winning strategy for the *weak* existential $k$-pebble game. We shall then exhibit a pair of structures $\mathbf{A}' \in \mathcal{C}$ and $\mathbf{B}' \notin \mathcal{C}$ such that the Duplicator has a winning strategy for the *strong* existential $k$-pebble game. Therefore the class $\mathcal{C}$ is not $\exists L^k(\neq, \neg)$-definable either, proving the theorem.

Given the two structures $\mathbf{A}$ and $\mathbf{B}$, we let $\mathbf{A}' = \mathbf{A}$ and $\mathbf{B}' = \mathbf{A} \times \mathbf{B}$. Here $\mathbf{A} \times \mathbf{B}$ has universe $\{(a, b) : a \in \mathbf{A}, b \in \mathbf{B}\}$, and relations $R$ such that $R((a_1, b_1), \ldots, (a_l, b_l))$ iff $R(a_1, \ldots, a_l)$ holds in $\mathbf{A}$ and $R(b_1, \ldots, b_l)$ holds in $\mathbf{B}$. Note that $\mathbf{B}' \notin \mathcal{C}$, since there is a homomorphism $f$ from $\mathbf{B}'$ to $\mathbf{B}$ given by $f(a, b) = b$, and $\mathbf{B} \notin \mathcal{C}$.

The strategy for the Duplicator in the strong existential $k$-pebble game $G_2$ is based on its strategy in the weak existential $k$-pebble game $G_1$. Suppose that in $G_1$, when the Spoiler placed a pebble at $a$ in $\mathbf{A}$, then the Duplicator placed the corresponding pebble at $b$ in $\mathbf{B}$. Now in $G_2$, when the Spoiler places a pebble at $a$ in $\mathbf{A}' = \mathbf{A}$, then the Duplicator places the corresponding pebble at $(a, b)$ in $\mathbf{B}' = \mathbf{A} \times \mathbf{B}$.

The resulting mapping that sets $f(a) = (a, b)$ is clearly one-to-one. Furthermore, it is a homomorphism, since if $R(a_1, \ldots, a_l)$ holds in $\mathbf{A}$, then $R(b_1, \ldots, b_l)$ in $\mathbf{B}$ for the corresponding pebbles according to $G_1$, and so $R((a_1, b_1), \ldots, (a_l, b_l))$, that is, $R(f(a_1), \ldots, f(a_l))$ holds in $\mathbf{B}'$. Conversely, if $R(a(x_1), \ldots, f(a_l))$, that is, $R((a_1, b_1), \ldots, (a_l, b_l))$, holds in $\mathbf{B}'$, then $R(a_1, \ldots, a_l)$ holds in $\mathbf{A}$. Therefore the Duplicator maintains an isomorphism $f$, as desired. ∎

# 3 Datalog

A Datalog($\neq, \neg$) program is a finite collection of rules of the form $t_0 \leftarrow t_1, t_2, \ldots, t_l$. The expression $t_0$ is the *head* of the rule, while $t_1, t_2, \ldots, t_l$ form the *body* of the rule. The head of the rule is an atomic formula $S(x_1, \ldots, x_n)$, where $S$ is a relational symbol; $x_1, \ldots, x_n$ are called the *head variables* of this rule. The expressions in the body can be inequalities $x_j \neq x_k$, atomic formulas $R(x_1, \ldots, x_m)$, or negated atomic formulas $\neg R(x_1, \ldots, x_m)$. The relational symbols appearing in the heads of the rules form the *intensional database predicates* (IDBs), while the others are the *extensional database predicates* (EDBs). In negations of atomic formulas $\neg R(x_1, \ldots, x_m)$ appearing in the body of a rule, the symbol $R$ must be an EDB. One of the IDB predicates is designed as the *goal* predicate of the program. Here we shall assume that the goal predicate is of the form $S()$, that is, it has arity zero. We also refer to Datalog($\neg$), Datalog($\neq$), and Datalog(), depending on whether we allow negated atomic formulas and inequalities.

Datalog($\neq, \neg$) programs have a *least fixpoint* semantics, which can be viewed as a *declarative* or a *procedural* semantics [3]. For the result below, it will be more useful to describe the semantics in term of *derivation trees*. A derivation tree for a program $\pi$ over a structure $\mathbf{A}$ with universe $D$ a tree that has at the root the goal $S()$ and a rule with this goal as its head; the variables appearing in the rule are instantiated to elements of $D$. In general, an internal node of the tree is an IDB $S(x_1, \ldots, x_n)$ and a rule with this goal at its head; the variables appearing in the rule are instantiated to elements of $D$. Such an internal node has children corresponding to the atoms in the body of the rules. The chidren inherit an instantiation to elements of $D$ from their parents. The leaves of the tree are instantiations to elements of $D$ of inequalities $x_j \neq x_k$, EDB atoms $R(x_1, \ldots, x_m)$, or negated EDB atoms $\neg R(x_1, \ldots, x_m)$ such that the condition they state holds in the structure $\mathbf{A}$.

Note that we have not allowed expressions in the body of a rule to be equalities $x_j = x_k$. This limitation causes no loss of generality, because for such an equality we can replace all occurences of $x_k$ by $x_j$ throughout the rule. We also assume that the variables in an IDB $S(x_1, \ldots, x_n)$ occurring either in the head or in the body of a rule are all distinct. This also causes no loss of generality, because otherwise we can introduce a new IDB $S'$ with fewer arguments. Thus, without loss of generality we can assume the the list of head variables in a rule head is always a prefix $x_1, \ldots, x_k$ of some list $x_1, \ldots, x_n$ of variables. For example, the Datalog program that recogizes whether a directed graph has a cycle is usually written as follows:

$$P(x, y) \leftarrow E(x, y)$$

$$P(x, y) \leftarrow P(x, z), E(z, y)$$

$$C() \leftarrow P(x, x)$$

To avoid having $P(x, x)$, we introduce $P'(x)$ to stand for $P(x, x)$, remove the last rule, and add the following extra rules

$$P'(x) \leftarrow E(x, x)$$

$$P'(x) \leftarrow P(x, z), E(z, x)$$

$$P'(x) \leftarrow P'(x), E(x, x)$$

$$P(x,y) \leftarrow P'(x), E(x,y)$$

$$C() \leftarrow P'(x)$$

The remainder of this section is devoted to the proof of the following result.

**Theorem 2** *Over homomorphism-closed classes, Datalog$()$ is as expressive as Datalog$(\neq, \neg)$.*

Let $\mathcal{C}$ be a class of structures that is closed under homomorphisms and characterized by a Datalog$(\neq, \neg)$ program $\pi_0$. We define from $\pi_0$ a new Datalog$(\neq, \neg)$ program $\pi_1$ that also characterizes $\mathcal{C}$. We then define from $\pi_1$ a Datalog() program $\pi_2$ and show that $\pi_2$ also characterizes $\mathcal{C}$.

To illustrate the constructions, we consider the following simple example for $\pi_0$:

$$T(x,y,w) \leftarrow E(x,y), x \neq w, \neg E(x,w)$$

$$T(x,y,w) \leftarrow E(x,z), T(z,y,w), x \neq w$$

$$T(x,y,w) \leftarrow E(z,w), T(z,y,w), \neg E(y,w), y \neq w$$

$$S() \leftarrow T(x,y,z).$$

The program $\pi_1$ is obtained by introducing, for each IDB $T$ in $\pi_0$, several indexed IDBs corresponding to $T$. The indexing indicates which EDBs, or negations of EDBs, on the arguments $x_1, x_2, \ldots x_n$ of $T$ are needed in the derivation of $T$. We call such rules *annotated rules*. We discard annotated rules that would require both $E$ and $\neg E$ on the same arguments, since such rules cannot occur in a derivation tree.

The construction is described formally with an iterative process. Initially, the program $\pi_1$ is empty. We perform the following two steps:

(1) For each rule $r$ in $\pi_0$ with head $R(x_1, \ldots, x_k)$ and body that has no IDB atoms, let $\sigma'$ be the set of literals –atomic formulas or negated atomic formulas–that appear in the body of $r$. If $\sigma'$ does not contain a contradition, i.e., an atom and its negation, then let $\sigma = \sigma'|_{x_1,\ldots,x_k}$ be the restriction of $\sigma'$ to literals over the head variables. We create a new IDB prediacate $R_\sigma$; we call it an *annotated predicate*. Add to $\pi_1$ the annotated rule $r'$ obtained from $r$ by replacing the IDB atom $R(x_1, \ldots, x_k)$ in the head with the annotated atom $R_\sigma(x_1, \ldots, x_k)$.

(2) For each rule $r$ in $\pi_0$ with head $R(x_1, \ldots, x_k)$, and body with IDB atoms, let $\sigma'$ be the set of literals – atomic formulas or negated atomic formulas–that appear in the body of $r$. For each IDB atom $Q(u_1, \ldots, u_k)$ in the body of $r$, choose an annotated IDB atom $Q_\tau(x_1, \ldots, x_k)$ already in $\pi_1$, replace $Q(u_1, \ldots, u_k)$ by $Q_\tau(u_1, \ldots, u_k)$, apply to $\tau$ the substitution $[x_1 \mapsto u_1, \ldots, x_k \mapsto u_k]$, and add the resulting literals to $\sigma'$. If $\sigma'$ does not contain a contradition, i.e., an atom and its negation, then let $\sigma = \sigma'|_{x_1,\ldots,x_k}$ be the restriction of $\sigma'$ to the litearls over the head variables. Create a new annotated IDB atom $R'_\sigma$ for $\pi_1$. Add to $\pi_1$ the annotated rule $r'$ obtained from $r$ by replacing also the IDB atom $R(x_1, \ldots, x_k)$ in the head with the annotated atom $R_\sigma(x_1, \ldots, x_k)$. This step is iterated as long as possible.

We illustrate this construction with the above program. The first rule of $\pi_0$ yields in $\pi_1$:

$$T_{E(x,y),\neg E(x,w)}(x,y,w) \leftarrow E(x,y), x \neq w, \neg E(x,w)$$

The second rule of $\pi_0$ yields in $\pi_1$:

$$T(x,y,w) \leftarrow E(x,z), T_{E(x,y),\neg E(x,w)}(z,y,w), x \neq w$$

$$T(x,y,w) \leftarrow E(x,z), T(z,y,w), x \neq w$$

Note that the conditions appearing as subindices of $T$ in the body do not carry over to the subindices of the head $T$ in this case; this is because they all involve $x$ instantiated as $z$, which is not a head variable.

The third rule of $\pi_0$ yields in $\pi_1$:

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T(z,y,w), \neg E(y,w), y \neq w$$

Now that we have a new IDB predicate $T_{\neg E(y,w)}$, the second rule of $\pi_0$ yields in $\pi_1$:

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T_{\neg E(y,w)}(z,y,w), y \neq w$$

Note how the indices of $T$ in the body do carry over to the indices of $T$ in the head.

The third rule of $\pi_0$ now yields in $\pi_1$:

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T(z,y,w), \neg E(y,w), y \neq w,$$

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T_{\neg E(y,w)}(z,y,w),$$

$$\neg E(y,w), y \neq w$$

Note that we excluded here the rule that would have $T_{E(x,y),\neg E(x,w)}(z,y,w)$ in the body. The reason is that $\neg E(x,w)$ instantiates to $\neg E(z,w)$, but the body of the rule contains $E(z,w)$; such conflicts, as we said before, leads to excluding the rule.

Finally, the fourth rule of $\pi_0$ yields in $\pi_1$:

$$S() \leftarrow T(x,y,z)$$

$$S() \leftarrow T_{E(x,y),\neg E(x,w)}(x,y,z)$$

$$S() \leftarrow T_{\neg E(y,w)}(x,y,z)$$

**Lemma 2** *The program $\pi_1$ characterizes the same class $\mathcal{C}$ of structures characterized by $\pi_0$.*

PROOF SKETCH: In one direction, a derivation tree for a structure $\mathbf{A}$ and program $\pi_1$ can be used to obtain a derivation tree for $\mathbf{A}$ and program $\pi_0$, simply by omitting the indices from $\pi_1$ in the derivation tree.

Conversely, a derivation tree for a structure $\mathbf{A}$ and program $\pi_0$ can be used to obtain a derivation tree for $\mathbf{A}$ and program $\pi_1$, simply by traversing the tree for $\pi_0$ from the leaves up to the root, and adding to it the appropriate indices to obtain annotated rules from $\pi_1$.

In our example, if $\pi_0$ was using at a leaf the rule

$$T(x,y,w) \leftarrow E(x,y), x \neq w, \neg E(x,w)$$

then $\pi_1$ uses instead the rule

$$T_{E(x,y),\neg E(x,w)}(x,y,w) \leftarrow E(x,y), x \neq w, \neg E(x,w),$$

so that while $\pi_0$ was deriving $T(x,y,w)$, the program $\pi_1$ will derive instead $T_{E(x,y),\neg E(x,w)}(x,y,w)$. Continuing up the derivation tree, if $\pi_0$ now uses the rule

$$T(x,y,w) \leftarrow E(x,z), T(z,y,w), x \neq w$$

then $\pi_1$ uses instead the rule

$$T(x,y,w) \leftarrow E(x,z), T_{E(x,y),\neg E(x,w)}(z,y,w), x \neq w,$$

and so on up the derivation tree. ■

We now show how to construct from $\pi_1$ the program $\pi_2$ without negations or inequalites. Simply remove all negations and inequalities from the bodies in the rules of $\pi_1$. In our example, we get:

$$T_{E(x,y),\neg E(x,w)}(x,y,w) \leftarrow E(x,y)$$

$$T(x,y,w) \leftarrow E(x,z), T_{E(x,y),\neg E(x,w)}(z,y,w)$$

$$T(x,y,w) \leftarrow E(x,z), T(z,y,w)$$

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(x,z), T_{\neg E(y,w)}(z,y,w)$$

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T(z,y,w)$$

$$T_{\neg E(y,w)}(x,y,w) \leftarrow E(z,w), T_{\neg E(y,w)}(z,y,w)$$

$$S() \leftarrow T(x,y,z)$$

$$S() \leftarrow T_{E(x,y),\neg E(x,w)}(x,y,z)$$

$$S() \leftarrow T_{\neg E(y,w)}(x,y,z)$$

We finally show:

**Lemma 3** *The program $\pi_2$ characterizes the same class $\mathcal{C}$ of structures characterized by $\pi_1$.*

PROOF SKETCH: In one direction. a derivation tree for a structure $\mathbf{A}$ and program $\pi_1$ can be used to obtain a derivation tree for $\mathbf{A}$ and program $\pi_2$, because the rules of $\pi_2$ have fewer conditions than the corresponding rules of $\pi_1$.

Conversely, suppose we have a derivation tree $\tau$ for a structure $\mathbf{A}$ and program $\pi_2$. Construct a different tree $\tau'$ for a structure $\mathbf{A}'$ from $\tau$ by traversing the tree $\tau$ from the root down to the leaves. Every time the rule corresponding to the instantiated rule at an internal node of the tree has a variable $x_i$ occurring in the body but not in the head of the rule, instantiate $x_i$ to a new element of $\mathbf{A}'$; this new element may only occur in the subtree rooted at this node. Once the tree is constructed put in $\mathbf{A}'$ all the tuples required by the EDB atoms in leaves of $\tau'$. The leaves of $\tau'$ have the EDBs that $\mathbf{A}'$ must have.

From the way $\tau'$ was constructed from $\tau$, we obtain a homomorphism from $\mathbf{A}'$ to $\mathbf{A}$, namely by traversing both trees simultaneously from the root down and determining for each variable in $\tau'$ the corresponding variable in $\tau$. More precisely, each element $a_i$ in $\mathbf{A}'$ is obtained from some variable $x_i$ in $\tau'$. But $x_i$ is instantiated in $\tau$ to some element $b_i$. We map $a_i$ to $b_i$.

We now consider the derivation tree $\tau'$ for $\mathbf{A}'$ and $\pi_2$, and obtain a derivation tree $\tau''$ for $\mathbf{A}'$ and $\pi_1$. The tree $\tau''$ is obtained by replacing each rule of $\pi_2$ with its corresponding rule of $\pi_1$ in $\tau'$, adding the leaves involving inequalities and negated atoms. Note that the inequalities hold, because every new variable in the body of a rule instantiated in $\tau''$ yields a new element in $\mathbf{A}'$ when we go down the tree; that is, in all instantiatiations of a rule in $\tau''$, different variables in the rule correspond to different elements in $\mathbf{A}'$. The negated atoms also hold, because every variable added down the tree is new, and an EDB that is required to hold both positively and negatively would have been detected by the consistency check for indices of the annotated program $\pi_1$ (as we said earlier, rules that yield such a situation are eleminated).

Finally, once we know that $\mathbf{A}'$ is in the class $\mathcal{C}$ of structures for $\pi_1$, we can use the fact that $\mathcal{C}$ is closed under homomorphisms to infer that $\mathbf{A}$, as a homomorphic image of $\mathbf{A}'$, is also in $\mathcal{C}$. ■

# 4 SNP

Rather than investigate the extension of existential 1st-order logic with universal 2nd-order quantifiers over classes of structures closed under homomorphism, it is more convenient to consider the dual logic, the extension of universal 1st-order logic with existential 2nd-order quantification over class of structures closed under inverse homomorphisms. (A class $\mathcal{C}$ is closed under inverse homomorphisms if $\mathbf{B} \in \mathcal{C}$ implies that $\mathbf{A} \in \mathcal{C}$, whenever there is a homomorphism from $\mathbf{A}$ to $\mathbf{B}$.) The logic SNP consists of all classes characterizable by an existential 2nd-order formula with a universal 1st-order part. We assume that the 1st-order part is written in conjunctive normal form, and view each disjunction as the negation of a conjunction of positive and negative literals. The full logic, in which we allow inequality and negated atoms, is denoted as SNP($\neq, \neg$).

For example, the 3-colorability problem can be written as

$$(\exists R, B, G)(\forall x, y)$$

$$\neg(R(x) \wedge B(x)) \wedge \neg(R(x) \wedge G(x)) \wedge \neg(B(x) \wedge G(x))$$

$$\wedge\neg(\neg R(x) \wedge \neg B(x) \wedge \neg G(x)) \wedge \neg(E(x,y) \wedge R(x) \wedge R(y))$$

$$\wedge\neg(E(x,y) \wedge B(x) \wedge B(y)) \wedge \neg(E(x,y) \wedge G(x) \wedge G(y)).$$

Note that this sentence is monadic, since the existentialy quantified relations $R, B, G$ are monadic. It is monotone, because inside each negated conjunction, the input relation $E$ appears positively. The formula does not use inequalities. Thus, 3-colorability is in 1SNP(), which is monadic, without inequalities or negated atoms. (In general, we let $k$SNP denote the restriction of SNP to existential quantification of arity at most $k$.)

A sentence in SNP($\neq, \neg$) might have a negated conjunction, say, of the form

$$\neg(\neg E(x,y) \wedge R(x) \wedge B(y) \wedge x \neq y).$$

We establish the following result:

**Theorem 3** *For classes of finite structures closed under inverse homomorphisms, 1SNP() is as expressive as 1SNP($\neq, \neg$).*

Consider a class $\mathcal{C}$ that is closed under inverse homomorphisms, and such that $\mathcal{C}$ can be defined by a sentence $Q_0$ in 1SNP($\neq, \neg$) . We define sentences $Q_1, Q_2, Q_3, Q_4, Q_5$ successively, and show that $Q_i$ describes the same set of structures as $Q_{i-1}$. The last sentence, $Q_5$, is in 1SNP(), and this completes the proof.

$Q_1$ is obtained from $Q_0$ to ensure that for all pairs of variables $x, y$ appearing in a negated conjunction, the statement $x \neq y$ is one of the conjuncts; if this is not the case, then we can split the negated conjunction into two, one involving $x \neq y$ and the other one involving $x = y$, and then removing the occurrence of $x = y$ by replacing all occurrences of $y$ by $x$ in the conjunction.

For example, if $Q_0$ has a negated conjunction

$$\neg(\neg E(x,y) \wedge R(x) \wedge B(y))$$

then $Q_1$ has instead two negated conjunctions

$$\neg(\neg E(x,y) \wedge R(x) \wedge B(y) \wedge x \neq y)$$

$$\wedge\neg(\neg E(x,x) \wedge R(x) \wedge B(x))$$

The following is immediate:

**Lemma 4** *$Q_1$ describes the same class as $Q_0$.*

$Q_2$ is obtained from $Q_1$ as follows. If an input relation $p$ occurs with all its arguments equal, say $p(x,x,x)$, then introduce a new existentially quantified monadic relation $p'$, add a conjunct $\neg(p(x,x,x) \wedge \neg p'(x))$ to the formula, and replace every occurrence of $p(x,x,x)$ by $p'(x)$. This transformation forces $p(x,x,x)$ not to appear negated inside a negated conjunction.

For example, if $Q_1$ has a negated conjunction

$$\neg(\neg E(x,x) \wedge R(x) \wedge B(x))$$

then $Q_2$ will have instead (with a new quantified relation $E'$)

$$\neg(E(x,x) \wedge \neg E'(x)) \wedge \neg(\neg E'(x) \wedge R(x) \wedge B(x))$$

**Lemma 5** *$Q_2$ describes the same class as $Q_1$.*

PROOF SKETCH: If a structure $\mathbf{A}$ satisfies $Q_1$, then we can verify that it satisfies $Q_2$ with the same assignment to existentially quantified relations, making also $p'(x)$ true precisely when $p(x,x,x)$ is true. For the converse, suppose $\mathbf{A}$ satisfies $Q_2$. If $p'(x)$ is true precisely when $p(x,x,x)$ is true, then clearly $\mathbf{A}$ satisfies $Q_1$. Otherwise, we may construct a new structure $\mathbf{A}'$ by increasing the cases where $p(x,x,x)$ is true so that it coincides with $p'(x)$ (recall $p(x,x,x) \to p'(x)$). Then $\mathbf{A}'$ satisfies $Q_1$, and with the obvious homomorphism from $\mathbf{A}$ to $\mathbf{A}'$, we have that $\mathbf{A}$ also satisfies $Q_1$, since the class described by $Q_1$ is closed under inverse homomorphisms. ■

$Q_3$ is obtained from $Q_2$ by removing the negated conjunctions that contain a conjunct $x \neq x$ (such a negated conjunction is always satisfied), and removing the negated conjunctions that contain both $p$ and $\neg p$ applied to the same arguments (such a negated conjunction is also always satisfied). The following is immediate:

**Lemma 6** *$Q_3$ describes the same class as $Q_2$.*

Finally, starting with $Q_3$, remove all negated input relations inside negated conjunctions to obtain $Q_4$, and remove all inequalities to obtain $Q_5$.

For example, if $Q_3$ has a negated conjunction (with an input relation $E$)

$$\neg(\neg E(x,y) \wedge R(x) \wedge B(y) \wedge x \neq y),$$

then $Q_4$ has instead

$$\neg(R(x) \wedge B(y) \wedge x \neq y),$$

and $Q_5$ has instead

$$\neg(R(x) \wedge B(y)).$$

Since the proof of the equivalence of $Q_4$ and $Q_5$ is simpler than the proof of the equivalence of $Q_3$ and $Q_4$, we give it first.

**Lemma 7** *If the class described by $Q_4$ is closed under inverse homomorphisms, then $Q_5$ describes the same class as $Q_4$.*

PROOF SKETCH: It is immediate that if a structure $\mathbf{A}$ satisfies $Q_5$, it also satisfies $Q_4$, because the conditions in $Q_4$ are less restrictive with the added inequalities inside the negated conjunctions.

The interesting direction is the opposite direction. Suppose $\mathbf{A}$ satisfies $Q_4$. Construct a new structure $\mathbf{A}'$ such that for each $a_i$ in $\mathbf{A}$, the structure $\mathbf{A}'$ has a large enough (see later) number $N$ of elements $a_i^j$ for $1 \leq j \leq N$. The elements in $\mathbf{A}'$ are related in the same way as the elements in $\mathbf{A}$ they correspond to. For example, if we have $E(a_1, a_2)$ in $\mathbf{A}$, then we have $E(a_1^j, a_2^{j'})$ for all $1 \leq j, j' \leq N$ in $\mathbf{A}'$.

Since there is a homomorphism from $\mathbf{A}'$ to $\mathbf{A}$, and $\mathbf{A}$ satisfies $Q_4$, it follows that $\mathbf{A}'$ also satisfies $Q_4$. Consider the assignment of the $r$ existentially quantified relations over $\mathbf{A}'$ in $Q_4$. This can be viewed as a coloring of the elements of $\mathbf{A}'$ with $c = 2^r$ colors. Then $N/c$ of the $a_i^j$ in $\mathbf{A}'$ corresponding to a given $a_i$ in $\mathbf{A}$ have the same color. Select such a color for each $a_i$ in $\mathbf{A}$, corresponding to an assignment of truth values for existentially quantified monadic relations for each $a_i$ in $\mathbf{A}$.

We claim that this is a valid coloring for $\mathbf{A}$ with respect to $Q_5$. If not, it violates some negated conjunction involving certain elements $a_i$ of $\mathbf{A}$. But this negated conjunction corresponds to a negated conjunction in $Q_4$ with some additional conditions $a_{i'} \neq a_i$; this negated conjunction in $Q_4$ is satisfied by the $N/c$ chosen elements $a_i^j$ of $\mathbf{A}'$ of the same color as the corresponding $a_i$ in $\mathbf{A}$. We may choose different such $a_i^j$ for the variables that were instantiated to the same $a_i$ in the negated conjunction, so that the conditions $a_i^j \neq a_i^{j'}$ also hold for distinct variables. This can be achieved is $N$ is large enough so that $N/c$ is at least the number of variables in $Q_4$. Thus $\mathbf{A}'$ would not satisfy the corresponding negated conjunction in $Q_4$, a contradiction. ■

It remains to show:

**Lemma 8** *$Q_4$ describes the same class as $Q_3$.*

PROOF SKETCH: As in the preceding lemma, it is immediate that if $\mathbf{A}$ satisfies $Q_4$, it will also satisfy $Q_3$, because the conditions in the negated conjunctions of $Q_3$ are less restrictive than the conditions for $Q_4$.

Suppose then that a *finite* structure $\mathbf{A}$ satisfies $Q_3$ but not $Q_4$. We use the probabilistic method to derive a contradiction. We first construct again a structure $\mathbf{A}'$ that has a large enough (see later) number $N$ of elements $a_i^j$ for each $a_i$. For an input relation applied to a single element, say $E(a_1, a_1)$ in $\mathbf{A}$ we make as before $E(a_1^j, a_1^{j'})$ for all $j, j'$ in $\mathbf{A}'$. When an input relation is applied to at least two distinct elements, say $E(a_1, a_2)$ in $\mathbf{A}$, then we make $E(a_1^j, a_2^{j'})$ hold in $\mathbf{A}'$ only with probability $1/2$, independently over all possible such situations and choices of $j, j'$.

As before, since there is a homomorphism from $\mathbf{A}'$ to $\mathbf{A}$, the structure $\mathbf{A}'$ must satisfy $Q_3$, and we get in particular $N/c$ elements $a_i^j$ for each $a_i$ that have the same color, where $c = 2^r$ and $r$ is the number of existentially quantified monadic relations. Consider the coloring of $\mathbf{A}$ corresponding to a choice of color for $a_i$ that occurs in these $(N/c)$ $a_i^j$'s of the same color in $\mathbf{A}'$.

We claim that the following outcome has positive probability: Consider the subinstances $\mathbf{A}^*$ of $\mathbf{A}$ obtained by making some subset of the input relations that hold on $\mathbf{A}$ between at least two elements become false. Each such $\mathbf{A}^*$ is an induced subinstance of $\mathbf{A}'$, under a correspondence between each $a_i$ in $\mathbf{A}^*$ and some $a_i^j$ among the chosen $N/c$ of the same color in $\mathbf{A}'$.

Assume the claim holds. Then if some negated conjunction in $Q_4$ does not hold for $\mathbf{A}$ for some elements $a_i$, we can consider the negated conjuncts in the corresponding negated conjunction in $Q_3$, and find the subinstance $\mathbf{A}^*$ of $\mathbf{A}$ for which these will hold as well. Thus the negated conjunction in $Q_3$ would not hold for $\mathbf{A}^*$, and therefore also not hold for $\mathbf{A}'$, a contradiction.

It remains to prove the claim. Consider the $k$ elements $a_i$ of $\mathbf{A}$ as integers $i = 0, 1, \ldots, k-1$, Take $k < p < N/c$ prime, and consider $p$ elements $a_i^j$ in $\mathbf{A}'$ corresponding to $a_i$ in $\mathbf{A}$, viewed as integers $j$ in the range $0, 1, \ldots, p-1$, modulo $p$. Consider subinstances $\mathbf{A}_{a,b}$ of $\mathbf{A}'$ containing for each $a_i$ the element $a_i^j$ with $j = ai + b$ modulo $p$, where $a, b$ are integers modulo $p$. Each choice of two $a_i^j$ and $a_{i'}^{j'}$ for two given $a_i$ and $a_{i'}$ occurs in some unique $\mathbf{A}_{a,b}$. Therefore different $\mathbf{A}_{a,b}$ share at most one element. Consequently, for each input relation applied to at least two distinct elements, say $E(a_1, a_2)$ in $\mathbf{A}$, the corresponding $E(a_1^j, a_2^{j'})$, which are present with probability $1/2$ in $\mathbf{A}'$, occur in at most one $\mathbf{A}_{a,b}$, and therefore the resulting possible outcomes for the different $\mathbf{A}_{a,b}$ are independent.

The total number of subinstances $\mathbf{A}_{a,b}$ being considered is $p^2$. The probability that a particular induced subinstance $\mathbf{A}^*$ does not appear in one of these $p^2$ choices of $\mathbf{A}_{a,b}$ is some constant $d < 1$, so the probability that it does not appear in any $\mathbf{A}_{a,b}$ is $d^{p^2}$, by independence. Therefore the probability that the $d'$ subinstances $\mathbf{A}^*$ do not all occur as some $\mathbf{A}_{a,b}$ is at most $d' \cdot d^{p^2}$ for some constant $d'$.

But there are only $\binom{N}{p}^k$ possible choices of subsets of $a_i^j$ of size $p$ for each $a_i$. The claim then follows by verifying that $d' \cdot d^{p^2} \binom{N}{p}^k < 1$, so that for all possible choices of subsets of $a_i^j$ of the same color we have that all possible $\mathbf{A}^*$ occur as subinstances for some choice of the $a_i$ in $\mathbf{A}^*$ from the $a_i^j$ of the same color for each $i$, with nonzero probability; for this last inequality to hold, it suffices $p = \omega(\log N)$ as $N \to \infty$, which can be satisfied since $p$ is allowed to be as large as $N/c$. ■

The proof of the monadic case extends to the binary case as well.

**Theorem 4** *For classes of finite structures closed under inverse homomorphisms, 2SNP() is as expressive as 2SNP($\neq, \neg$).*

PROOF SKETCH: As in the monadic case, we begin with a given sentence $Q_0$ in 2SNP($\neq, \neg$), and define sentences $Q_1, Q_2, Q_3, Q_4, Q_5$ successively, and show that $Q_i$ describes the same set of structures as $Q_{i-1}$. The last one $Q_5$ will be in 2SNP(), and this completes the proof.

The transformation from $Q_0$ to $Q_1$ to ensure that for all pairs of variables $x, y$ appearing in a negated conjunction, the statement $x \neq y$ is one of the conjuncts, is the same as in the monadic case. To go from $Q_1$ to $Q_2$, we handle occurrences of input conjuncts involving at most two variables, instead of just one variable as in the monadic case. That is, we replace each occurrence of say an input $p(x, x, y, y)$ with $p'(x, y)$, for an existentially quantified $p'$, adding the conjunct $\neg(p(x, x, y, y) \wedge \neg p'(x, y))$. This forces $p(x, x, y, y)$ never to appear negated inside a negated conjunction, and similarly for all input relations whose arguments involving at most two different variables. The proof that $Q_2$ describes the same class as $Q_1$ is the same as in Lemma 6 for the monadic case. The transformation from $Q_2$ to $Q_3$ is the same as in the monadic case, removing negated conjunctions with a conjunct $x \neq x$, and removing negated conjunctions containing both $p$ and $\neg p$ applied to the same arguments. As in the monadic case, we go from $Q_3$ to $Q_4$ by removing negated input atoms inside negated conjunctions.

We go from $Q_4$ to $Q_5$ as follows. The sentence $Q_5$ has an existentially quantified binary relation $eq$ that is required to be an equivalence relation. For every existentially quantified relation $r$ in $Q_4$, require in $Q_5$ when $r$ is binary (the monadic case is similar) that if ($eq(x, x')$ and $eq(y, y')$) then ($r(x, y)$ if and only if $r(x', y')$). Also replace every occurrence of an input relation, say $p(x, x, y, z)$, with ($p(x', x'', y'.z')$ and $eq(x, x')$ and $eq(x, x'')$ and $eq(y, y')$ and $eq(z, z')$), where $x', x'', y', z'$ are new variables that do not occur elsewhere in the formula. Finally replace every occurrence of $x \neq y$ with $\neg eq(x, y)$.

Again we first give the proof of the equivalence of $Q_4$ and $Q_5$. Given an instance $\mathbf{A}$, if $\mathbf{A}$ satisfies $Q_4$, then $\mathbf{A}$ also satisfies $Q_5$ with $eq$ set to be the equality relation. Conversely, if $\mathbf{A}$ satisfies $Q_5$, then the homomorphic image $\mathbf{A}'$ of $\mathbf{A}$ that identifies the elements related by $eq$ for $\mathbf{A}$ also satisfies $Q_5$, with $eq$ being the equality relation in $\mathbf{A}'$, so $\mathbf{A}'$ satisfies $Q_4$, and therefore $\mathbf{A}$ also satisfies $Q_4$, because the set of instances satisfying $Q_4$ is closed under inverse homomorphisms.

Finally, we show that $Q_4$ describes the same class as $Q_3$. The proof is a generalization of Lemma 8 in the monadic case. It is immediate that if $\mathbf{A}$ satisfies $Q_4$, it will also satisfy $Q_3$, because the conditions in the negated conjunctions of $Q_3$ are less restrictive that the conditions for $Q_4$. Suppose then that a finite structure $\mathbf{A}$ satisfies $Q_3$ but not $Q_4$. Again, we use the probabilistic methods to derive a contradiction. We construct again a structure $\mathbf{A}'$ that has a large enough number $N$ of elements $a_i^j$ for each $a_i$. For an input relation applied to a single element, say $E(a_1, a_1)$ in $\mathbf{A}$, we make as before $E(a_1^j, a_1^{j'})$ for all $j, j'$ in $\mathbf{A}'$. For an input relation applied to two distinct elements, say $E(a_1, a_2)$ in $\mathbf{A}$, we also make $E(a_1^j, a_2^{j'})$ for all $j, j'$ in $\mathbf{A}'$. For an input relation applied to at least three distinct elements, say $E(a_1, a_2, a_3)$ in $\mathbf{A}$, we make $E(a_1^j, a_2^{j'}, a_3^{j''})$ hold in $\mathbf{A}'$ only with probability $1/2$, independently over all possible such situations and choices of $j, j', j''$.

As before, since there is a homomorphism from $\mathbf{A}'$ to $\mathbf{A}$, the structure $\mathbf{A}'$ must satisfy $Q_3$. By Ramsey theory [10], there exists a constant $c'$ and a choice of $c' \log N$ elements $a_i^j$ for each $a_i$, such that all $a_i^j$ for the same $a_i$ satisfy the same existentially quantified relations $r(a_i^j)$ and $r(a_i^j, a_i^j)$, and furthermore all $a_i^j, a_{i'}^{j'}$ for two distinct $a_i, a_{i'}$ satisfy the same existentially quantified relations $r(a_i^j, a_{i'}^j)$. Consider an assignment of monadic and binary existentially quantified relations for $\mathbf{A}$ corresponding to the choice of $a_i^j$ for each $a_i$ among these ($c' \log N$) chosen $a_i^j$.

We claim that the following outcome has positive probability: Consider the subinstances $\mathbf{A}^*$ of $\mathbf{A}$ obtained by making some subset of the input relations that hold on $\mathbf{A}$ between at least three elements become false. Each such $\mathbf{A}^*$ occurs as an induced subinstance of $\mathbf{A}'$, under a correspondence between each $a_i$ in $\mathbf{A}^*$ and some $a_i^j$ among the chosen $c' \log N$ in $\mathbf{A}'$.

Assume the claim holds. Then if some negated conjunction in $Q_4$ does not hold for $\mathbf{A}$, we can consider the negated conjuncts in the corresponding negated conjunction in $Q_3$, and find the subinstance $\mathbf{A}^*$ of $\mathbf{A}$ for which these will hold as well. Thus the negated conjunction in $Q_3$ would not hold for $\mathbf{A}^*$, and therefore also not hold for $\mathbf{A}'$, a contradiction.

It remains to prove the claim. Consider the $k$ elements $a_i$ of $\mathbf{A}$ as being $i = 0, 1, \ldots, k-1$, Take $k < p < c' \log N$ prime, and consider $p$ elements $a_i^j$ in $\mathbf{A}'$ corresponding to $a_i$ in $\mathbf{A}$, viewed as as integers $j$ in the range $0, 1, \ldots, p-1$, modulo $p$. Consider subinstances $\mathbf{A}_{a,b,c}$ of $\mathbf{A}'$ containing for each $a_i$ the element $a_i^j$ with $j = ai^2 + bi + c$ modulo $p$, where $a, b, c$ are integers modulo $p$. Each choice of three $a_i^j$, $a_{i'}^{j'}$, and $a_{i''}^{j''}$ for three given $a_i$, $a_{i'}$, and $a_{i''}$ occurs in some unique $\mathbf{A}_{a,b,c}$. Therefore different $\mathbf{A}_{a,b,c}$ share at most two elements. Consequently, for each input relation applied to at least three distinct elements, say $E(a_1, a_2, a_3)$ in $\mathbf{A}$, the corresponding $E(a_1^j, a_2^{j'}, a_3^{j''})$ which are present with probability $1/2$ in $\mathbf{A}'$, occur in at most one $\mathbf{A}_{a,b,c}$, and therefore the resulting possible outcomes for the different $\mathbf{A}_{a,b,c}$ are independent.

The total number of subinstances $\mathbf{A}_{a,b,c}$ being considered considered is $p^3$. The probability that a particular induced subinstance $\mathbf{A}^*$ does not appear in one of these $p^3$ choices of $\mathbf{A}_{a,b,c}$ is some constant $d < 1$, so the probability that it does not appear in any $\mathbf{A}_{a,b,c}$ is $d^{p^3}$, by independence. Therefore the probability that the $d'$ subinstances $\mathbf{A}^*$ do not all occur as some $\mathbf{A}_{a,b,c}$ is at most $d' \cdot d^{p^3}$ for some constant $d'$.

But there are only $\binom{N}{p}^k$ possible choices of subsets of $a_i^j$ of size $p$ for each $a_i$. The claim then follows by verifying that $d' \cdot d^{p^3} \binom{N}{p}^k < 1$, so that with nonzero probability all choices of subinstances $\mathbf{A}^*$ occur for choices of subsets of $a_i^j$; here it suffices $p = \omega(\sqrt{\log N})$ as $N \to \infty$, and as we said before, Ramsey theory guarantees $p = c' \log N$. ■

This proof does not extend to the $r$-adic case for $r \geq 3$, because we need $p = \omega((\log N)^{1/r})$, but Ramsey theory gives only $p = c' \log\log N$ for $r = 3$, with further iterations of log for larger $r$. However, if we are not interested in preserving the arity of existentially quantified relations, we can, as in the monadic and in the binary case, introduce for each input relation $R$ an existentially quantified relation $R'$ of the same arity, add the condition $\neg(R \wedge \neg R')$, and use $R'$ instead of $R$ in the rest of the SNP statement. This is justified by the monotonicity of the class as before in Lemma 6, and immediately takes care of removing negations. The removal of inequalities is as in the proof of Theorem 4 for the binary case, by introducing a binary relation $eq$. Note that the new quantified relations are of the same arity as the input relations, so their arity cannot be bounded.

**Theorem 5** *For classes of finite structures closed under inverse homomorphisms, SNP() is as expressive as SNP($\neq, \neg$).*

The $k$-pebble-game characterization of $\exists L^k$-definability holds only for classes of finite structures, so Theorem 1 is stated for such classes. In contrast, the proof of Theorem 2 applies to classes finite structures as well as classes of general structures. The results for SNP fragments are proved for classes of finite structures (because of the probabilistic method). We show, however, that the Compactness Theorem applied to SNP implies that these results apply also to classes of general structures. The following lemma seems to be a "folk theorem".

**Lemma 9** *Let $\mathbf{A}$ be a structure and $\varphi$ an SNP sentence. Then $\mathbf{A}$ satisfies $\varphi$ iff all finite substructures of $\mathbf{A}$ satisfies $\varphi$.*

PROOF SKETCH: SNP is preserved under substructures, so one direction is immediate. Suppose now that $\mathbf{A}$ does not satisfy $\varphi$. Extend the vocabulary with a new constant symbol $\mathbf{c}_a$ for every element $a$ of $\mathbf{A}$. Let $diagram(\mathbf{A})$ consists of all atomic sentences and negated atomic sentences over the extended vocabulary; that is, $diagram(\mathbf{A})$ is a complete description of $\mathbf{A}$. Let $\psi$ be the 1st-order part of $\varphi$. Consider the theory $T = diagram(\mathbf{A}) \cup \{\psi\}$. We claim that $T$ is unsatisfiable. Otherwise, if it is satisfied by a structure $\mathbf{A}'$ over an expansion of the vocabulary with the quantified relations. But then, some expansion of $\mathbf{A}$ is a substructure of $\mathbf{A}'$, implying that $\varphi$ holds in $\mathbf{A}$, by substructure preservation. Since $T$ is unsatisfiable, by the Compactness Theorem, a finite subset of $T$ is unsatisfiable. Since $diagram(\mathbf{A})$ is satisfiable, we can assume that this finite subset includes $\psi$ and all atomic sentences and negated atomic sentences of a finite subset $\{\mathbf{c}_{a_1}, \ldots, \mathbf{c}_{a_n}\}$ of constants. But then $\varphi$ does not hold over the substructure of $\mathbf{A}$ induced by $\{\mathbf{c}_{a_1}, \ldots, \mathbf{c}_{a_n}\}$. ■

**Corollary 1** *For classes closed under inverse homomorphisms, SNP() (resp., 1SNP(), 2SNP()) is as expressive as SNP($\neq, \neg$) (resp., 1SNP($\neq, \neg$) , 2SNP($\neq, \neg$)).*

PROOF SKETCH: Let $\mathcal{C}$ be a class of structures closed under inverse homomorphism. Let $\mathcal{C}_f$ be the class of finite substructures of structures in $\mathcal{C}$. It is easy to see that $\mathcal{C}_f$ is also closed under inverse homomorphism. By Theorem 3,4, and 5, an SNP($\neq, \neg$) sentence (resp., 1SNP($\neq, \neg$) , 2SNP($\neq, \neg$)) $\varphi$ is equivalent over $\mathcal{C}_f$ to a setence $\varphi'$ in SNP() (resp., 1SNP(), 2SNP()). Suppose that the equivalence does not hold over $\mathcal{C}$, then $\varphi$ and $\varphi'$ do not agree on some structure $\mathbf{A} \in \mathcal{C}$. But then, by Lemma 9, $\varphi$ and $\varphi'$ must also disagree on a finite substructure $\mathbf{A}'$ of $\mathbf{A}$, which is a contradiction, since $\mathbf{A}' \in \mathcal{C}_f$. ■

# 5 Undecidability of Homomorphism Closure

In this section we prove that homomorphism closure is not effective for our effective logics ($\exists L^\omega$ is not an effective logic). Since the complement of Datalog is a fragment of 1SNP (cf. [14]), we focus on the former.

**Theorem 6** *It is undecidable whether the class of (finite) structures characterized by a program in monadic Datalog($\neq$) is closed under homomorphisms.*

PROOF SKETCH: Hillebrand et al. [12] consider a Turing machine $M$, and define a program $\Pi$ in monadic Datalog($\neq$) that accepts a structure $\mathbf{A}$ if and only if $\mathbf{A}$ does not describe a correct computation of the machine $M$. We can modify the program $\Pi$ so that it accepts a structure $\mathbf{A}$ if and only if $\mathbf{A}$ describes a computation of $M$ reaching an accepting state and the computation of the machine $M$ described by $\mathbf{A}$ contains an error.

We now describe the rules of $\Pi$ from [12]. Let $M$ be given by its state space $Q = \{q_1, \ldots, q_n\}$, its tape alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$, and its transition relation $\Delta \subseteq Q \times \Sigma \times \{Left, Stay, Right\} \times Q \times \Sigma$. In a rule, we use the expression $C = (T, L, S, R, Q)$ as an abbreviation for the sequence of clauses $Cons(T, U, C)$, $Cons(S, W, V)$, $Cons(R, Q, W)$, where $Cons(T, U, C)$ indicates that $C$ is a list with *Car* part $T$, and *Cdr* part $U$. Thus $C = (T, L, S, R, Q)$ says that $C$ represents a five element list with components $T, L, S, R, Q$.

The rule corresponding for instance to a transition $(q_i, \sigma_k, Right, q_j, \sigma_l)$ is

$$CONF(C') \leftarrow CONF(C),$$

$$C = (T, L, S, R, Q), C' = (T', L', S', R', Q'),$$

$$Cons(S'', L, L'), Cons(S', R', R),$$

$$Int_i(Q), Int_j(Q'), Int_k(S), Int_l(S''),$$

$$Cons(Y, T, T').$$

Similar rules correspond to transitions with *Left* and *Stay*. These rules are used to verify that $\mathbf{A}$ describes a computation of $M$ reaching an accepting state. Since these rules do not involve an inequality, the class of sructures $\mathbf{A}$ describing a computation of $M$ reaching an accepting state is closed under homomorphisms.

An error is encountered when an element of the database codes more than one integer or a list does not have a unique *Car* or *Cdr* part:

$$Err \leftarrow Int_i(X), Int_j(X).$$

$$Err \leftarrow Cons(X, Y, Z), Cons(X', Y', Z), X \neq X'.$$

$$Err \leftarrow Cons(X, Y, Z), Cons(X', Y', Z), Y \neq Y'.$$

Thus if $\mathbf{A}$ describes a computation of $M$ reaching an accepting state and not containing an error, we can define a structure $\mathbf{A}'$ by making two copies of an element appearing in the *Car* position, with both elements related identically, so that there is a homomorphism from $\mathbf{A}'$ to $\mathbf{A}$, and $\mathbf{A}'$ describes a computation of $M$ reaching an accepting state and containing an error.

Suppose $M$ does not terminate. If $\mathbf{A}$ describes a computation of $M$ reaching an accepting state, then the description $\mathbf{A}$ of the computation of $M$ contains an error, and therefore $\Pi$ accepts $\mathbf{A}$. If $\mathbf{A}$ does not describe a computation of $M$ reaching an accepting state, then $\Pi$ does not accept $\mathbf{A}$. Therefore the language of $\Pi$ (i.e., the class of structures characterized by $\Pi$) is the same as the language of structures $\mathbf{A}$ describing a computation of $M$ reaching an accepting state. As said before, a homomorphic image of such a structure $\mathbf{A}$ also reaches an accepting state. Therefore the language of $\Pi$ is closed under homomorphisms.

Suppose now $M$ terminates. Then there is a structure $\mathbf{A}$ describing a correct computation of $M$ reaching an accepting state. The program $\Pi$ does not accept this structure $\mathbf{A}$ since the computation does not contain an error. As said before, there also exists a structure $\mathbf{A}'$ describing a computation of $M$ reaching and accepting state and containing an error such that $\mathbf{A}'$ homomorphically maps to $\mathbf{A}$. The program $\Pi$ accepts the structure $\mathbf{A}'$. Therefore the language of $\Pi$ is not closed under homomorphisms.

Consequently, whether the language of $\Pi$ is closed under homomorphisms depends on whether $M$ terminates, and the undecidability of whether the language of $\Pi$ is closed under homomorphisms follows from the undecidability of the halting problem for $M$. ■

# Acknowledgements

The authors are grateful to Miki Ajtai for valuable conversations and to several referees for useful comments.

# References

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

[2] M. Ajtai and and Y. Gurevich, "Monotone versus positive," *J. ACM* 34(1987), 1004–1015.

[3] A. Chandra and D. Harel,
Horn-clause queries and generalizations", *J. Logic Programming*, 1(1985), 1–15.

[4] C.C. Chang and H.J. Keisler, *Model Theory*, North-Holland, 3rd edition, 1990.

[5] A .Durand, C. Lautemann, and T Schwentick, "Subclasses of binary NP", *J. of Logic and Computation* 8(1998), 189–207.

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

[7] R. Fagin, "Generalized first–order spectra and polynomial–time recognizable sets', in *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7* (R. M. Karp, ed.), 1974, 43–73.

[8] T. Feder and M. Y. Vardi, "The computational structure of monotone monadic SNP and constraint satisfaction: a study through Datalog and group theory," *SIAM J. Computing* 28 (1998), 57–104.

[9] E. Grädel and E. Rosen, "Preservation theorems for two-variable logic", *Mathematical Logic Quarterly*, 45(1999), 315–325.

[10] R.L. Graham, B.L. Rothschild, and J. H. Spencer, *Ramsey Theory*, Wiley, 1980.

[11] Y. Gurevich, "On Finite Model Theory", in *Feasible Mathematics* (S.R. Buss and P.J. Scott, eds.), Birkhaeuser, Boston, 1990, 211–219.

[12] G. G. Hillebrand, P.C. Kanellakis, H. G. Mairson, and M. Y. Vardi, "Undecidable boundedness problems for database logic programs," *J. Logic Programming* 25(1995), 163–190.

[13] N. Immerman, *Descriptive Complexity*, Springer-Verlag, 1998

[14] P.G. Kolaitis, "Implicit Definability on Finite Structures and Unambiguous Computations", *Proc. 5th IEEE Symp. on Logic in Computer Science*, 1990, 168–180.

[15] P. G. Kolaitis and M. Y. Vardi, "On the expressive power of Datalog: tools and a case study," *J. Computer and System Sciences* 51(1995), 110–134.

[16] P.G. Kolaitis and M.Y. Vardi, "Infinitary logics and 0–1 laws," *Information and computation* 98(1992), 258–194.

[17] C.H. Papadimitriou and M. Yannakakis, "Optimization, approximation, and complexity," *Proc. 20th ACM Symp. on Theory of Computing*, 1988, 229–234.

[18] E. Rosen, Some aspects of model theory and finite structures, *Bulletin of Symbolic Logic*, 8(2002), 380–403.