Information and Computation 143, 154–174 (1998)
Article No. IC982703

A Restricted Second Order Logic for

Anuj Dawar*

Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, United Kingdom

We introduce a restricted version of second order logic SO^{ω} in which the second order quantifiers range over relations that are closed under the equivalence relation $\equiv^k$ of k variable equivalence, for some k. This restricted second order logic is an effective fragment of the infinitary logic $L^{\omega}_{\infty\omega}$, but it differs from other such fragments in that it is not based on a fixed point logic. We explore the relationship of SO^{ω} with fixed point logics, showing that its inclusion relations with these logics are equivalent to problems in complexity theory. We also look at the expressibility of NP-complete problems in this logic.

1. INTRODUCTION

In recent years, much research in finite model theory has focused on its connections with computational complexity theory. It turns out that there is a close relationship between the computational complexity of a problem, i.e., the amount of resources needed to solve the problem on some machine model of computation, and its descriptive complexity, i.e., the kinds of "logical resources" that are needed to describe the problem. The paradigmatic result establishing a connection between descriptive and computational complexity is the result of Fagin [12] which shows that the properties of finite structures that are definable by sentences of existential second order logic are exactly those that are in the complexity class NP. This was extended by Stockmeyer [24] to a tight correspondence between second order logic and the polynomial-time hierarchy. Further work along these lines has established logical characterizations for a wide range of complexity classes (see, for instance, [17]).

However, some of the results equating logical expressibility to computational complexity require the finite structures to have a built-in linear order. That is, the exact correspondence between expressibility in a logic and solvability within given resource bounds does not hold over the class of all finite structures, but is restricted to those structures that have a linear order as one of their relations. Thus, for

* Supported by EPSRC grants GR/H81108 and GR/L69596.

0890-5401/98 $25.00

instance, Immerman [16] and Vardi [25] independently showed that the extension FP of first order logic by means of a fixed point operator characterizes the class PTIME on the class of ordered structures. No such logical characterization of PTIME is known for arbitrary finite structures. Similarly, by results of Vardi [25] and Abiteboul and Vianu [2] it is known that the extension PFP of first order logic by a partial fixed point operator characterizes the class PSPACE on the class of ordered structures.

In general, FP is strictly weaker than PTIME. That is to say, while every property expressible in FP is decidable in PTIME, there are PTIME properties that are not expressible in FP. The same holds true of PFP and PSPACE. Nevertheless, Abiteboul and Vianu [3] were able to show that FP = PFP if, and only if, PTIME = PSPACE. Thus, even though we do not have a logical characterization of the class PTIME over all finite structures, the open complexity theoretic question about the separation of PTIME and PSPACE is equivalent to a question about the expressive power of two logics on the class of all finite structures. Extending this work, Abiteboul *et al.* [1] defined a variety of fixed point logics and showed that for a range of complexity classes between PTIME and EXPTIME, open questions about the separations of these classes are equivalent to separations of corresponding fixed point logics. They also gave characterizations of these fixed point logics in terms of computability on a *relational* machine model of computation, establishing a general result showing that inclusion relations among relational complexity classes mirror those among the usual computational complexity classes.

The interest in fixed point logics has also focused attention on the infinitary logic with finitely many variables—$L^{\omega}_{\infty\omega}$. All of the fixed point logics mentioned above can be seen as fragments of $L^{\omega}_{\infty\omega}$. Recently, considerable effort has been devoted to understanding the model theory of $L^{\omega}_{\infty\omega}$ on finite structures (see, for instance, [10, 18, 19]). One of the reasons for this is that definability in $L^{\omega}_{\infty\omega}$ has an elegant characterization in terms of two-player pebble games. Indeed, this has been the main tool used so far in establishing inexpressibility results for fixed point logic. The logic $L^{\omega}_{\infty\omega}$ has also proved a vehicle for introducing important notions from classical model theory, such as elementary equivalence and element types, into finite model theory in a meaningful way, by restricting the number of variables. A systematic study of the k-variable elementary equivalence relation $\equiv^k$ was undertaken in [10]. It is felt that the translation of important open questions in complexity theory into questions about fragments of $L^{\omega}_{\infty\omega}$, as in [1] for instance, provides a greater opportunity for the application of model-theoretic techniques to these questions.

In this paper, we continue the study of the model theory of $L^{\omega}_{\infty\omega}$ by defining a restricted version of second-order logic SO^{ω} that is contained within $L^{\omega}_{\infty\omega}$. This is obtained by restricting the interpretation of second order quantifiers to relations closed under the equivalence relation $\equiv^k$, for some k. We show that the existential fragment of this logic is the class relational NP, while SO^{ω} itself coincides with relational PH. This establishes results in the style of [3] for all levels of the polynomial time hierarchy. Moreover, these are of a somewhat different character to the results in [1] in that the characterizations are not in terms of fixed point logics. We also discuss the expressibility of NP-complete problems in our restricted second order logic, giving examples of natural problems that can be expressed in this way, as well as illustrating techniques for establishing lower bounds by showing, for instance, that 3-colourability cannot be expressed in $L^{\omega}_{\infty\omega}$.

2. BACKGROUND AND NOTATION

In this section, we fix our notation and examine the necessary background material. We assume familiarity with the basic notions of predicate logic, as well as basic definitions from complexity theory.

A *signature* σ is a finite sequence of relation symbols $(R_1, ..., R_s)$, with associated arities $a_1, ..., a_s$. A σ-structure $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, ..., R_s^{\mathfrak{A}})$ consists of a *finite* set A, referred to as the *universe* or *domain* of $\mathfrak{A}$, and interpretations of the relation symbols in σ as relations on A, i.e., $R_i^{\mathfrak{A}} \subseteq A^{a_i}$.

An m-ary *query* q is a map that takes structures over some fixed signature σ to m-ary relations on the domains of the structures and is closed under isomorphisms. That is, for any m-tuple t in a structure $\mathfrak{A}$ and any isomorphism f from $\mathfrak{A}$ to $\mathfrak{B}$, $t \in q(\mathfrak{A})$, if and only if $f(t) \in q(\mathfrak{B})$. For instance, a first order formula with m free variables defines an m-ary query. A 0-ary query, also called a *Boolean* query, is a map from the class of σ-structures to the set $\{\text{True}, \text{False}\}$ and can be identified with an isomorphism closed class of σ-structures. In general, we say that a query is expressible in a logic L if there is some formula of L that defines it. By abuse of notation, we will also use L to denote the class of queries that are definable in the logic L. When we speak of the computational complexity of a query q, we mean the complexity of deciding, given a structure $\mathfrak{A}$ and a tuple t from the domain of $\mathfrak{A}$, whether it is the case that $t \in q(\mathfrak{A})$. The measure of the size of the input is the cardinality of the domain of $\mathfrak{A}$.

We say that a logic L *captures* a complexity class C if every query that is expressible by a formula of L is in the complexity class C and, conversely, every query that is in C is expressible by a formula of L. We also say that L captures C on a class of structures S when the equivalence between L and C holds for queries whose domain is restricted to S. Note that this usage of the term "capture″ is not the same as that in [1].

We write Σ^1_1 for the collection of second order sentences in prenex normal form, in which all second order quantifiers precede the first order quantifiers, and which contain only existential second order quantifiers. Fagin [12] proved that Σ^1_1 captures NP. This result was extended by Stockmeyer [24] to show that second order logic captures the polynomial-time hierarchy. Indeed, the correspondence between second order logic and the polynomial-time hierarchy holds level by level. That is, if Σ^1_{n+1} denotes the collection of sentences of second order logic containing n alternations of second order quantifiers starting with an existential quantifier, then Σ^1_n captures Σ^p_n, the nth level of the polynomial-time hierarchy.

2.1. *Fixed Point Logics*

Let φ be a formula with free individual variables among $x_1, ..., x_m$, in the signature σ extended with an additional m-ary predicate symbol R. On σ-structures,

φ defines an operator mapping m-ary relations to m-ary relations. Thus, given a σ-structure $\mathfrak{A}$ and an m-ary relation P in $\mathfrak{A}$, we define $\varphi^{(\mathfrak{A}, P)}$ to be $\{s \mid (\mathfrak{A}, P) \models \varphi[s]\}$. If this operator is monotone, that is, for every P and Q such that $P \subseteq Q$, $\varphi^{(\mathfrak{A}, P)} \subseteq \varphi^{(\mathfrak{A}, Q)}$, then it has a least fixed point. While monotonicity is a semantic property, there is a syntactic condition on φ that guarantees that the corresponding operator is monotone. Namely, if φ is R-positive, that is, all occurrences of R in φ are in the scope of an even number of negations, then the operator defined by φ is monotone. We write LFP for the closure of first order logic under the operation of taking least fixed points of positive formulas. Immerman [16] and Vardi [25] independently showed that LFP captures the complexity class PTIME over the class of structures which include a linear order as one of their relations.

If φ defines a monotone operator, then its least fixed point in a structure $\mathfrak{A}$ can be obtained by iterating the operator as follows. Define φ^0 to be the empty relation $\emptyset$, and define φ^{i+1} to be $\varphi^{(\mathfrak{A}, \varphi^i)}$. Because the operator is monotone, this sequence of relations is increasing, and if $\mathfrak{A}$ has cardinality n, then for some $i \leqslant n^m$, $\varphi^i = \varphi^{i+1}$. This φ^i is then the least fixed point of φ. A similar iteration can be defined even when φ does not define a monotone operator by taking at each stage the union with the previous stage. That is, define $\varphi^{i+1} = \varphi^i \cup \varphi^{(\mathfrak{A}, \varphi^i)}$. The resulting sequence of relations is increasing for any φ, and once again reaches a fixed point for some $i \leqslant n^m$. This is the *inflationary fixed point* of φ. IFP is defined to be the closure of first order logic under the operation of taking inflationary fixed points of arbitrary formulas. Clearly, for positive formulas, the least fixed point and the inflationary fixed point coincide. Moreover, Gurevich and Shelah [14] showed that for every formula φ, the inflationary fixed point of φ is definable by a formula of LFP. It follows that the two logics IFP and LFP are equivalent on finite structures. For the rest of this paper, we will use the notation FP to denote the logic IFP.

Immerman [16] established a normal form for formulas of LFP. This, along with the result of Gurevich and Shelah mentioned above provides similar normal forms for IFP (see also [2]).

THEOREM 2.1. *For every formula* $\varphi(\bar{y})$ *of* FP *there is a formula* $\psi(\bar{x}, \bar{y})$, *which is the inflationary fixed point of a first order formula, such that* φ *is equivalent to* $\exists \bar{x} \psi$ *and* $\forall \bar{x} \psi$.

Indeed, we can even require that the first order formula of which ψ is the fixed point is itself existential (see [2]).

Consider now an arbitrary formula φ that does not necessarily define a monotone operator. The sequence of stages defined by taking $\varphi^0 = \emptyset$ and $\varphi^{i+1} = \varphi^{(\mathfrak{A}, \varphi^i)}$ is not necessarily increasing and may or may not converge to a fixed point. However, if there is an i such that $\varphi^i = \varphi^{i+1}$, then there is such an $i \leqslant 2^{n^m}$. The *partial fixed point* of φ is defined to be φ^i for i such that $\varphi^i = \varphi^{i+1}$, if such an i exists, and empty otherwise. PFP denotes the closure of first order logic under an operation defining the partial fixed points of formulas. Abiteboul and Vianu [2] showed that PFP is equivalent to the relational *while* language introduced by Chandra and Harel [6]. Vardi [25] showed that this *while* language captures the class PSPACE on the collection of structures with a linear order.

There is an apparently more general form of inductive definition, where a query is defined by simultaneous induction of a number of formulas. Let $S_0, ..., S_l$ be a sequence of relation symbols that do not occur in the signature σ, with associated arities $a_0, ..., a_l$. Further, let $\varphi_0, ..., \varphi_l$ be a sequence of formulas in the signature formed by extending σ by $S_0, ..., S_l$, where φ_j defines a query of arity a_j. We define the stages of the simultaneous induction of the sequence $(\varphi_0, ..., \varphi_l)$ by

$$\varphi_j^0 = \varnothing$$

$$\varphi_j^{i+1} = \varphi_j^{(\mathfrak{A}, \varphi_0^i, ..., \varphi_l^i)} \text{ (or } \varphi_j^i \cup \varphi_j^{(\mathfrak{A}, \varphi_0^i, ..., \varphi_l^i)} \text{ for the inflationary case).}$$

The sequence reaches a fixed point if there is an i such that $\varphi_j^i = \varphi_j^{i+1}$ for all $0 \leqslant j \leqslant l$, and the relation defined by the fixed point is then φ_0^i.

Moschovakis [22] showed that allowing simultaneous inductions does not increase the expressive power of fixed point logics (see also [20] for a discussion and [2] for the case of partial fixed points).

THEOREM 2.2. *Every query defined by a simultaneous inflationary (resp. partial) induction is definable in* FP (*resp.* PFP).

In light of Theorem 2.2, we will use simultaneous inductions in this paper wherever it makes the exposition clearer.

Abiteboul *et al.* [1] extended the above framework of fixed point logics by defining a range of fixed point logics obtained by varying two parameters, the control mechanism and the semantics of the fixed point iteration. The control mechanism can be deterministic, nondeterministic, or alternating, and the semantics can be inflationary or non-inflationary. The two fixed point logics considered above, FP and PFP, in this terminology, are both deterministic in their control, with inflationary and non-inflationary semantics, respectively. We will now consider the nondeterministic inflationary fixed point logic, introduced by Abiteboul *et al.* We denote this logic NFP, for nondeterministic fixed point logic.[1]

Given two formulas φ_0 and φ_1 in a signature σ extended by an additional m-ary predicate R, we define in any σ-structure $\mathfrak{A}$ a sequence of stages of the pair (φ_0, φ_1) indexed by binary strings

$$\varphi^{\varepsilon} = \varnothing, \text{ for the empty string } \varepsilon$$

$$\varphi^{s \cdot 0} = \varphi^s \cup \varphi_0^{(\mathfrak{A}, \varphi^s)}$$

$$\varphi^{s \cdot 1} = \varphi^s \cup \varphi_1^{(\mathfrak{A}, \varphi^s)}.$$

We now define the nondeteministic fixed point of the pair (φ_0, φ_1) in the structure $\mathfrak{A}$ to be $\bigcup_{s \in \{0,1\}^*} \varphi^s$. The logic NFP is the closure of first order logic under the operation of taking nondeterministic fixed points, with the proviso that the fixed point operator does not occur within the scope of a negation.

[1] This notation is different from [1], where NFP denotes the logic we call PFP.

We observe, without proof, that Theorems 2.1 and 2.2 extend directly to NFP as well. That is, we define a simultaneous nondeterministic induction by a sequence of pairs of formulas: $\varphi_i = (\varphi_{i,0}, \varphi_{i,1})$, $0 \leqslant i \leqslant l$ in a signature σ extended with new relation symbols $S_0, ..., S_l$. The stages of this induction on a σ-structure $\mathfrak{A}$ are defined as follows:

$$\varphi_i^{\varepsilon} = \varnothing, \qquad \text{for the empty string } \varepsilon$$
$$\varphi_i^{s \cdot 0} = \varphi^s \cup \varphi_{i,0}^{(\mathfrak{A}, \varphi_0^s \cdots \varphi_l^s)}$$
$$\varphi_i^{s \cdot 1} = \varphi^s \cup \varphi_{i,1}^{(\mathfrak{A}, \varphi_0^s \cdots \varphi_l^s)}.$$

The nondeteministic fixed point of the sequence is given by $\bigcup_{s \in \{0,1\}^*} \varphi_0^s$. The following lemma is proved in the same way as was Theorem 2.2.

LEMMA 2.3. *Every query defined by a simultaneous nondeterministic induction is definable in* NFP.

Similarly, the proof of the normal form result, Theorem 2.1, can also be extended to NFP.

LEMMA 2.4. *Every formula of NFP is equivalent to a formula of the form $\exists \bar{x} \varphi$, where φ is the nondeterministic fixed point of a pair of first order formulas.*

2.2. *Infinitary Logic*

The infinitary logic $L_{\infty\omega}$ is obtained by closing first order logic under conjunctions and disjunctions of arbitrary (not just finite) sets of formulas. This logic is of little use in the study of finite models, since every query on the class of finite structures is expressible in $L_{\infty\omega}$. However, the restriction of $L_{\infty\omega}$ where we only allow finitely many variables to appear in any given formula has proved to be of great value in studying the expressive power of fixed point logics on finite structures.

More formally, let $L^k_{\infty\omega}$ denote the class of formulas of $L_{\infty\omega}$ in which all variables (free or bound) are among $x_1, ..., x_k$. Also, let $L^{\omega}_{\infty\omega} = \bigcup_{k \in \omega} L^k_{\infty\omega}$. The logic $L^{\omega}_{\infty\omega}$ was introduced by Barwise [4] in order to study inductive definitions on infinite structures. It was shown by Rubin [23] that for a fixed infinite structure, the least fixed point of any first order operator is expressible in $L^{\omega}_{\infty\omega}$. A similar result was obtained for the class of all finite structures by Kolaitis and Vardi [19], who showed that in this case, both FP and PFP can be seen as fragments of $L^{\omega}_{\infty\omega}$. This also applies to NFP, giving us the following picture (where inclusion is for sets of definable queries):

$$\text{FP} \subseteq \text{NFP} \subseteq \text{PFP} \subseteq L^{\omega}_{\infty\omega}.$$

The last containment in the above is a proper one, since Kolaitis and Vardi [19] show that there are nonrecursive queries that can be expressed in $L^{\omega}_{\infty\omega}$, while every query definable in PFP is computable in PSPACE. Indeed one can show that just as $L_{\infty\omega}$ is complete in its expressive power, so $L^{\omega}_{\infty\omega}$ is complete on ordered structures (for a fuller discussion of this, see [8]).

Proposition 2.5. *For every signature σ, there is a $k_\sigma \in \omega$ such that every query of arity a on ordered σ-structures is expressible in $L^k_{\infty\omega}$, where $k = \max(a, k_\sigma)$.*

We write L^k for the first order fragment of $L^k_{\infty\omega}$, i.e., the formulas of first order logic that contain only the variables $x_1, ..., x_k$.

Recall that for a structure $\mathfrak{A}$ and a tuple s of elements of $\mathfrak{A}$, the first order type of s in $\mathfrak{A}$, denoted $\text{Type}(\mathfrak{A}, s)$ is the set of formulas φ such that $\mathfrak{A} \models \varphi[s]$. The following variant of this notion was introduced in [10] and has proved to be very useful in studying expressibility in $L^\omega_{\infty\omega}$.

Definition 2.6. For any structure $\mathfrak{A}$ and a tuple s of elements of $\mathfrak{A}$, $\text{Type}^k(\mathfrak{A}, s)$ denotes the set of formulas φ of L^k such that $\mathfrak{A} \models \varphi[s]$.

We write $(\mathfrak{A}, s) \equiv^k (\mathfrak{B}, t)$ to denote $\text{Type}^k(\mathfrak{A}, s) = \text{Type}^k(\mathfrak{B}, t)$. We also write k-size($\mathfrak{A}$) to denote the number of equivalence classes of the relation $\equiv^k$ in the structure $\mathfrak{A}$.

Kolaitis and Vardi [19] showed that the equivalence relation $\equiv^k$ coincides, on finite structures, with the apparently stronger notion of equivalence in $L^k_{\infty\omega}$. That is, they showed that if $\mathfrak{A}$ and $\mathfrak{B}$ are finite structures and $(\mathfrak{A}, s) \equiv^k (\mathfrak{B}, t)$, then for every formula $\varphi \in L^k_{\infty\omega}$, $\mathfrak{A} \models \varphi[s]$ if, and only if, $\mathfrak{B} \models \varphi[t]$. Kolaitis and Vardi [19] also showed that a query q is definable in $L^k_{\infty\omega}$ if, and only if, it is closed under the relation $\equiv^k$; i.e., if $s \in q(\mathfrak{A})$ and $(\mathfrak{A}, s) \equiv^k (\mathfrak{B}, t)$, then $t \in q(\mathfrak{B})$. It was shown in [10] that if $\mathfrak{A}$ is a finite structure, there is a formula $\varphi \in \text{Type}^k(\mathfrak{A}, s)$ such that for any structure $\mathfrak{B}$, $\mathfrak{B} \models \varphi[t]$ if, and only if, $(\mathfrak{A}, s) \equiv^k (\mathfrak{B}, t)$.

The equivalence relation $\equiv^k$ has an elegant characterization in terms of Ehrenfeucht–Fraïssé style pebble games, essentially given by Barwise [4] (see also [15]). The game board consists of two structures $\mathfrak{A}$ and $\mathfrak{B}$ and a supply of k pairs of pebbles (a_i, b_i), $1 \leqslant i \leqslant k$. The pebbles $a_1, ..., a_l$ are initially placed on the elements of an l-tuple s in $\mathfrak{A}$, and the pebbles $b_1, ..., b_l$ on a tuple t in $\mathfrak{B}$. There are two players, Spoiler and Duplicator. At each move of the game, Spoiler picks up a pebble (either an unused pebble or one that is already on the board) and places it on an element of the corresponding structure. For instance, Spoiler might take pebble b_i and place it on an element of $\mathfrak{B}$. Duplicator must respond by placing the other pebble of the pair in the other structure. In the above example, Duplicator must place a_i on an element of $\mathfrak{A}$. If at the end of the move the partial map $f\colon \mathfrak{A} \to \mathfrak{B}$ given by $a_i \mapsto b_i$ is not a partial isomorphism, then Spoiler has won the game, otherwise it can continue for another move. Duplicator has a strategy to avoid losing for n moves, starting with the initial position $(\mathfrak{A}, s)$ and $(\mathfrak{B}, t)$ if, and only if, $(\mathfrak{A}, s)$ and $(\mathfrak{B}, t)$ cannot be distinguished by any formula of L^k of quantifier rank n or less. Hence, if Duplicator has a strategy to play the game indefinitely without losing, then $(\mathfrak{A}, s) \equiv^k (\mathfrak{B}, t)$.

The relation $\equiv^k$ is itself uniformly definable in FP [10, 18].

Theorem 2.7. *There is a formula η of* FP, *with $2k$ free variables, such that on any finite structure $\mathfrak{A}$, given two k-tuples s and t in $\mathfrak{A}$, $\mathfrak{A} \models \eta[s, t]$ if, and only if,* $(\mathfrak{A}, s) \equiv^k (\mathfrak{A}, t)$.

Moreover, we can also write a formula λ of FP which uniformly orders $\equiv^k$ equivalence classes (see [3, 10]). That is, on any finite structure $\mathfrak{A}$, λ defines a preorder such that the corresponding equivalence relation is $\equiv^k$.

THEOREM 2.8. *There is a formula λ of* FP, *with $2k$ free variables, such that on any finite structure $\mathfrak{A}$, λ defines a reflexive and transitive relation $\leqslant^k$ on k-tuples such that for every two k-tuples s and t, either $s \leqslant^k t$ or $t \leqslant^k s$ and both $s \leqslant^k t$ and $t \leqslant^k s$ hold if, and only if, $(\mathfrak{A}, s) \equiv^k (\mathfrak{A}, t)$.*

Thus, λ can be seen as defining a total order on the equivalence classes of $\equiv^k$. The FP definition of this order allows us to define an FP reduction which maps any structure $\mathfrak{A}$ to a quotient structure $\mathfrak{A}/\equiv^k$ which is linearly ordered. Using such a reduction, Abiteboul and Vianu [3] showed that FP = PFP if, and only if, PTIME = PSPACE (see also the exposition in [10]).

Abiteboul *et al.* [1] extend this by showing that the logic NFP captures the relational complexity class NP_r, whereby it follows that:

$$\mathrm{FP} = \mathrm{NFP} \quad \text{if, and only if,} \quad \mathrm{PTIME} = \mathrm{NP}; \quad \text{and}$$

$$\mathrm{NFP} = \mathrm{PFP} \quad \text{if, and only if,} \quad \mathrm{NP} = \mathrm{PSPACE}.$$

3. A RESTRICTED SECOND ORDER LOGIC

The fixed point logics can be viewed as effective fragments of $L^{\omega}_{\infty\omega}$, as we saw in Section 2.4. In this section, we explore a different way of obtaining an effective fragment of $L^{\omega}_{\infty\omega}$, by a restricted form of second order quantification. This provides a logical characterization of some relational complexity classes that is not based on a fixed point logic and that is closer in spirit to Fagin's characterization of NP.

DEFINITION 3.1.

- For an l-ary relation symbol R, and $k \geqslant l$, we define the second order quantifier $\exists^k R$ to have the following semantics: $\mathfrak{A} \models \exists^k R\varphi$ if there is an $X \subseteq A^l$ such that X is closed under the equivalence relation $\equiv^k$ in $\mathfrak{A}$, and $(\mathfrak{A}, X) \models \varphi$. As usual, $\forall^k R$ abbreviates $\neg \exists^k R \neg$.
- $\Sigma^{1,\omega}_0 = \Pi^{1,\omega}_0$ is the set of first order formulas.
- $\Sigma^{1,\omega}_{n+1}$ denotes the class of formulas of the form $\exists^{k_1} R_1 \cdots \exists^{k_m} R_m \varphi$, where φ is $\Pi^{1,\omega}_n$.
- $\Pi^{1,\omega}_{n+1}$ denotes the class of formulas of the form $\forall^{k_1} R_1 \cdots \forall^{k_m} R_m \varphi$, where φ is $\Sigma^{1,\omega}_n$.
- $\mathrm{SO}^{\omega} = \bigcup_{n \in \omega} \Sigma^{1,\omega}_n$.

The logic SO^{ω} is a restricted version of second order logic which forms an effective fragment of $L^{\omega}_{\infty\omega}$. In Theorem 3.3 below, we will see that it is in fact contained in PFP. We begin by establishing its relationship with the usual second order logic.

LEMMA 3.2. *For every n, $\Sigma^{1,\omega}_n \subseteq \Sigma^1_n$ and $\Pi^{1,\omega}_n \subseteq \Pi^1_n$.*

Proof. We present the proof for the Σ-classes. The proof for the Π-classes is analogous.

Let $\varphi \in \Sigma_n^{1,\omega}$ be a formula $\exists^{k_1} R_1 ... Q^{k_q} R_q \psi$, where Q is $\exists$ or $\forall$, depending on whether n is odd or even. Clearly, the query defined by φ can be expressed as

$$\exists R_1 \cdots Q R_q \left(\psi \wedge \bigwedge_{1 \leqslant i \leqslant q} \gamma^{k_i}(R_i) \right), \tag{1}$$

where $\gamma^{k_i}(R_i)$ asserts that R_i is $\equiv^{k_i}$-closed. By Theorem 2.7 each γ^k is definable in FP. Since $\mathrm{FP} \subseteq \Sigma_1^1 \cap \Pi_1^1$, it follows that the query (1) is definable in Σ_n^1. ∎

THEOREM 3.3. On ordered structures, for every $n \in \omega$, $\Sigma_n^{1,\omega} = \Sigma_n^1$ and $\Pi_n^{1,\omega} = \Pi_n^1$.

Proof. Since Lemma 3.2 holds on arbitrary structures, it holds on ordered structures, in particular. Thus, we only need to show the inclusions $\Sigma_n^1 \subseteq \Sigma_n^{1,\omega}$ and $\Pi_n^1 \subseteq \Pi_n^{1,\omega}$.

It follows from Proposition 2.5 that for every signature σ there is a k_σ such that, if $\mathfrak{A}$ is an ordered σ-structure and R is any l-ary relation on $\mathfrak{A}$ for $l \leqslant k_\sigma$, then R is $\equiv^{k_\sigma}$-closed. On the other hand, if the arity of R is l, for $l > k_\sigma$, R is $\equiv^l$-closed. Let $\varphi \in \Sigma_n^1$ be a σ-sentence $\exists R_1 \cdots Q R_q \psi$. For each R_i of arity a_i, let $k_i = \max(a_i, k_\sigma)$. Then it is easily seen that φ is equivalent, on ordered structures, to the sentence

$$\exists^{k_1} R_1 \cdots Q^{k_q} R_q \psi.$$

The proof for Π_n^1 sentences is similar. ∎

Theorem 3.3 establishes that the restricted second order logic SO^ω is not really restricted on ordered structures. In what follows, we establish the relationship of SO^ω to the fixed point logics.

THEOREM 3.4. $\mathrm{SO}^\omega \subseteq \mathrm{PFP}$.

Proof. It suffices to show that, given a formula ψ of PFP, there is a formula φ of PFP equivalent to $\exists^k R \psi$. Let l be the arity of R, with $l \leqslant k$.

Any relation P that is $\equiv^k$-closed on a structure $\mathfrak{A}$ can be seen as a set of $\equiv^k$ equivalence classes. Thus, the pre-order $\leqslant^k$ of Theorem 2.8, being a linear order on the collection of $\equiv^k$ equivalence classes, induces a lexicographical ordering of all $\equiv^k$-closed relations on $\mathfrak{A}$. Moreover, using the FP formula defining $\leqslant^k$, we can write an FP formula $\nu(P)$ which defines, for any $\equiv^k$-closed relation P of arity l, the lexicographically next such relation. This formula is:

$$[P(\bar{x}) \wedge \exists \bar{y} (\bar{y} \leqslant^k \bar{x}) \wedge \neg P(\bar{y})] \vee [\neg P(\bar{x}) \wedge \forall \bar{y} (\bar{y} \leqslant^k \bar{x}) \rightarrow P(\bar{y})].$$

We assume that the formula $\exists^k R \psi$ has free individual variables among $x_1, ..., x_m$ and therefore defines an m-ary query. We define this query by means of a simultaneous induction of two formulas. We therefore have two induction variables S and R of arity m and l, repectively. At successive stages of the induction R takes on, in lexicographical order, the values of $\equiv^k$-closed relations, reaching a fixed point when it contains all l-tuples. At the same time S accumulates the m-tuples that satisfy $\psi(R)$. Formally, we define the formulas φ_S and φ_R as follows:

$$\varphi_S \equiv S(\bar{x}) \vee \psi(R)$$

$$\varphi_R \equiv v(R) \vee \forall \bar{y} R(\bar{y}).$$

It can be verified that the simultaneous partial fixed point of this induction yields the query $\exists^k R\psi$. Therefore, by Theorem 2.2, we have a formula of PFP that expresses this query. ∎

We now show that the existential fragment of SO^ω is, in fact, equivalent to NFP. Note that Theorem 3.4 follows as a consequence. However, we have given a direct proof of this theorem, as it illustrates the techniques used to show the subsequent result. In the next two lemmas, we state the crucial results for proving the two directions of the equivalence of $\Sigma_1^{1,\omega}$ and NFP.

LEMMA 3.5. *For any pair of first order formulas φ_0 and φ_1, there is a formula of $\Sigma_1^{1,\omega}$ that defines the nondeterministic fixed point of* (φ_0, φ_1).

Proof. We note first that by the inflationary nature of the nondeterministic fixed point operator, for binary strings s_1 and s_2, if s_1 is a prefix of s_2, then $\varphi^{s_1} \subseteq \varphi^{s_2}$ in any structure $\mathfrak{A}$. Thus, if we consider any increasing sequence of binary strings, then the corresponding sequence of stages is increasing. It follows that, if we let k be the maximum of the number of distinct variables in φ_0 and φ_1, then for binary strings s such that length$(s) \geqslant k$-size$(\mathfrak{A})$, $\varphi^s = \varphi^{s \cdot 0} = \varphi^{s \cdot 1}$. Consider the formula

$$\exists^{2k} O \exists^{k+m} R\psi$$

with free individual variables $x_1, ..., x_m$, where ψ asserts that:

- O is a preorder of k-tuples;
- $R^0 = \varnothing$ and for every i, either $R^{i+1} = R^i \cup \varphi_0^{R^i}$ or $R^{i+1} = R^i \cup \varphi_1^{R^i}$, where R^i

is

$$\{t \mid R(s, t) \text{ for some s in the } i\text{th equivalence class determined by the preorder } O\};$$

and

- $R^m(x_1, ..., x_m)$, where m is the length of the pre-order O.

It can be verified that this formula expresses the nondeterministic fixed point of the pair (φ_0, φ_1). ∎

In the other direction, we have the following lemma.

LEMMA 3.6. *If ψ is a formula of* NFP, *then there is a formula of* NFP *that is equivalent to $\exists^k R\psi$.*

Proof. For simplicity, we assume that the arity of R is k and that the free individual variables in ψ are among $x_1, ..., x_m$.

As in the proof of Theorem 3.4, we are going to use the FP definition of the order $\leqslant^k$ of $\equiv^k$ equivalence classes. Intuitively, we want to define an induction that steps along this order and at each stage decides nondeterministically whether or not to include the current equivalence class in the relation R. For this, we essentially need to maintain three relations: one, S, to count the equivalence classes that have been visited, one to include those equivalence classes that have been chosen to be in R, and finally one, P, to construct the relation defined by ψ, given the candidate R. We do this by a simultaneous induction of three pairs of formulas, $(\varphi_{P,0}, \varphi_{P,1})$, $(\varphi_{R,0}, \varphi_{R,1})$ and $(\varphi_{S,0}, \varphi_{S,1})$, with relation symbols R and S of arity k and P of arity m. Note that in the definition below, $\varphi_{P,0} \equiv \varphi_{P,1}$ and $\varphi_{S,0} \equiv \varphi_{S,1}$ so the nondeterminism is confined to the pair $(\varphi_{R,0}, \varphi_{R,1})$:

$$\varphi_{P,0} \equiv \varphi_{P,1} \equiv \psi(R) \wedge \forall \bar{y} S(\bar{y})$$

$$\varphi_{S,0} \equiv \varphi_{S,1} \equiv \forall \bar{y}(\lambda(\bar{y}, \bar{x}) \rightarrow S(\bar{y}))$$

$$\varphi_{R,0} \equiv x \neq x$$

$$\varphi_{R,1} \equiv \forall \bar{y}(\lambda(\bar{y}, \bar{x}) \rightarrow S(\bar{y})).$$

In the above λ is the FP formula in Theorem 2.8 that defines the pre-order $\leqslant^k$. It is clear that any inflationary fixed point can be expressed as a nondeterministic fixed point (simply by taking $\varphi_0 = \varphi_1$). A slight complication arises because in the above formulas λ appears within the scope of a negation symbol. However, by Theorem 2.1, we know that the negation of an FP formula can be expressed without the fixed point operator appearing inside the scope of a negation.

It can be verified that the simultaneous nondeterministic fixed point of the above system defines the query $\exists^k R\psi$. Therefore, by Lemma 2.3, there is a formula of NFP that defines the same query. ∎

The following theorem is immediate from Lemmas 2.4, 3.5, and 3.6.

Theorem 3.7. $\Sigma_1^{1,\omega} = \mathrm{NFP}$.

Remark 3.8. The proof of Theorem 3.7 can be extended to show that, if we close the logic NFP simultaneously under negation and the operation of taking nondeterministic fixed points, we obtain a logic equivalent to SO^ω. Moreover, the alternations of negations and fixed points correspond exactly to the second order quantifier alternations in SO^ω. Similarly, Abiteboul *et al.* [1] also define an alternating inflationary fixed point logic, which they show to be equivalent to PFP. One can show that the fragment of this logic obtained by allowing only a bounded number of alternations is equivalent to SO^ω. Once again, the number of alternations corresponds exactly to the number of alternations of second order quantifiers in SO^ω.

The following corollaries follow immediately from Theorem 3.7.

COROLLARY 3.9. $\mathrm{FP} \subseteq \Sigma_1^{1,\omega} \cap \Pi_1^{1,\omega}$.

COROLLARY 3.10. $\mathrm{FP} = \Sigma_1^{1,\omega}$ *if, and only if,* PTIME = NP.

COROLLARY 3.11. $\Sigma_1^{1,\omega} = \mathrm{PFP}$ *if, and only if,* NP = PSPACE.

Moreover, an application of the same methods also gives us

THEOREM 3.12. $\mathrm{SO}^\omega = \mathrm{PFP}$ *if, and only if,* PH = PSPACE.

By Corollaries 3.10, and 3.11 and Theorem 3.12, the inclusion relations between SO^ω and the fixed point logics FP and PFP are equivalent to open problems in complexity theory. The next result shows that this is also true of the levels of the hierarchy within SO^ω.

THEOREM 3.13. *For every* $i, j \in \omega$, $\Sigma_i^{1,\omega} = \Sigma_j^{1,\omega}$ *if, and only if,* $\Sigma_i^p = \Sigma_j^p$; *and* $\Sigma_i^{1,\omega} = \Pi_i^{1,\omega}$ *if, and only if,* $\Sigma_i^p = \Pi_i^p$.

Finally, we also observe that when a sentence φ of $\Sigma_i^{1,\omega}$ is translated to the ordered quotient structure $\mathfrak{A}/\equiv^k$, the resulting sentence is, in fact, in the monadic fragment of Σ_1^1; i.e., it only uses quantification over sets. Writing $\mathrm{mon}.\Sigma_i^1$ for the monadic fragment of Σ_i^1, we can then extract the following result from the proof of Theorem 3.13. Note that this result is not about the logic SO^ω. It is a result about the unrestricted second order logic, although it is obtained by using facts about SO^ω in the proof.

THEOREM 3.14. *For any* $i, j \in \omega$, *if* $\mathrm{mon}.\Sigma_i^1 = \mathrm{mon}.\Sigma_j^1$ *on ordered strucutures, then* $\Sigma_i^p = \Sigma_j^p$.

4. NP-COMPLETE PROBLEMS

While the logic FP cannot express some easily computable properties, such as the property of a graph having even cardinality, which is not even expressible in $L_{\infty\omega}^\omega$, it can nevertheless express some P-complete problems such as the path systems problem and alternating transitive closure (see [19]). Similarly, Abiteboul *et al.* [1] show that there are natural PSPACE-complete problems that can be expressed in PFP. In this section, we examine the expressibility of NP-complete problems in the logic $\Sigma_1^{1,\omega}$.

In one sense, it is easy to see that there are NP-complete problems that can be expressed in $\Sigma_1^{1,\omega}$, since this logic captures NP on ordered structures (see Theorem 3.3). Thus, if we take any NP-complete problem and consider the set of its instances with linear order, we obtain a problem that is still NP-complete and is expressible in $\Sigma_1^{1,\omega}$. For instance, consider the class of structures $(V, E, \leqslant)$, where $\leqslant$ is a linear order on V, and the graph (V, E) is Hamiltonian.

However, in the absence of linear order, many natural NP-complete problems cannot be expressed in $L_{\infty\omega}^\omega$ and *a fortiori* not in $\Sigma_1^{1,\omega}$. Immerman [15] showed, essentially, that Hamiltonicity and clique are not definable in $L_{\infty\omega}^\omega$. We present, as an example, a simple proof that Hamiltonicity is not in $L_{\infty\omega}^\omega$. A proof of this result was also given by de Rougemont in [11].

EXAMPLE 4.1. Consider the complete bipartite graph $K_{m,n}$. It is easily verified that this graph is Hamiltonian if, and only if, $m=n$. An easy pebble game argument shows that $K_{k,k} \equiv^k K_{k,k+1}$. Since $K_{k,k}$ is Hamiltonian and $K_{k,k+1}$ is not, it follows that Hamiltonicity is not in $L^k_{\infty\omega}$ for any k.

Lovász and Gács [21] show that the problem of propositional satisfiability (SAT) is complete for NP even under first order reductions. Since $L^\omega_{\infty\omega}$ is closed under first order reductions, it follows that SAT is not definable in $L^\omega_{\infty\omega}$, for otherwise NP would be contained in $L^\omega_{\infty\omega}$, which we know is not the case. Similarly, Dahlhaus [7] shows that both Hamiltonicity and clique are also NP-complete under first order reductions and this provides an alternative proof that these problems are not expressible in $L^\omega_{\infty\omega}$. Essentially, it is a consequence of the completeness results that we can take the proof that some query in NP is not in $L^\omega_{\infty\omega}$, say the even cardinality query, and translate it into a proof that Hamiltonicity or clique is not in $L^\omega_{\infty\omega}$.

In contrast, 3-colourability is an NP-complete problem that is known not to be complete with respect to first order reductions. By a result of [9], we know that the class of queries that are reducible to 3-colourability obeys a 0–1 law. That is, for every Boolean query in this class, the proportion of structures of size n that are instances of the query tends to either 0 or 1 as n goes to infinity. It follows that straightforward counting arguments such as those in Example 4.1. will not suffice to show that 3-colourability is not expressible in $L^\omega_{\infty\omega}$. By taking a different approach, we are nevertheless able to show below that 3-colourability is not definable in $L^\omega_{\infty\omega}$. This answers an open question posed by Kolaitis and Vardi [19]. The proof is presented in Section 4.2. Before turning to that, however, we demonstrate natural NP-complete problems that are expressible in $\Sigma^{1,\omega}_1$.

4.1. *NFA Inequivalence*

In this section we examine examples of natural NP-complete problems that *are* expressible in the logic $\Sigma^{1,\omega}_1$. The examples are special cases of the problem of NFA inequivalence, that is, the problem of deciding, given two nondeterministic finite automata, whether or not they accept distinct languages. This problem is PSPACE-complete, and was shown by Abiteboul *et al.* [1] to be expressible in PFP. Two restrictions of this problem that are known to be NP-complete are the restriction to a finite language and the restriction to a unary alphabet (see [13]). Both of these restrictions are definable in $\Sigma^{1,\omega}_1$. We examine the second one in some detail.

The problem of determining whether two NFAs over a unary alphabet are inequivalent, which we denote UNI (for unary NFA inequivalence), can also be formulated as a problem on graphs, as follows. Given

$$N=(V, A, s_0, t_0, s_1, t_1),$$

where (V, A) is a directed graph and $s_0, s_1, t_0, t_1 \in V$ are distinguished vertices, are the two sets

$$P_0 = \{p \in \omega \mid \text{there is a path of length } p \text{ from } s_0 \text{ to } t_0\}$$

$$P_1 = \{p \in \omega \mid \text{there is a path of length } p \text{ from } s_1 \text{ to } t_1\}$$

distinct?

To see that this problem is definable in $\Sigma_1^{1,\omega}$, we first observe that if there is a $p \in \omega$ that distinguishes the two sets P_0 and P_1 in a structure N, then there is such a $p < 2^t$, where t is the number of distinct $\equiv^3$ equivalence classes in N. This is because, for every p, there is a formula $\psi^p(x, y)$ of L^3 which asserts that there is a path of length p from x to y. Thus, if we consider the set E^p of pairs (x, y) such that there is a path of length p from s_0 to x if, and only if, there is a path of length p from s_1 to y, then this set is $\equiv^3$-closed. Moreover, it can also be easily verified that E^{p+1} is definable from E^p. Since there are only 2^t distinct $\equiv^3$-closed sets, it follows that for $p \geqslant 2^t$ the sequence of sets must repeat itself. Thus, if the pair (t_0, t_1) appears in the set E^p for all $p < 2^t$, then it appears in all E^p, $p \in \omega$, which means that the sets P_0 and P_1 are identical. However, if there is a $p < 2^t$ such that $(t_0, t_1) \notin E^p$, then this p witnesses that P_0 and P_1 are distinct.

Next, we note that any number $p < 2^t$ can be represented by a pair of relations $(\leqslant^3, R)$, where $\leqslant^3$ is the ordering of $\equiv^3$-equivalence classes given by Theorem 2.8, and R is a $\equiv^3$-closed relation. This is done by interpreting the pair $(\leqslant^3, R)$ as a binary string of length t, with a 1 for each $\equiv^3$-equivalence class that is in R and a 0 for each class that is not in R. Finally, we note that, given the pair $(\leqslant^3, R)$, we can write a sentence $\varphi(\leqslant^3, R)$ of FP which asserts that the number represented by $(\leqslant^3, R)$ distinguishes the sets P_0 and P_1. We will not write down φ explicitly, noting only that we can write an inductive definition of a 5-ary relation S such that $S(x_1, x_2, \bar{y})$ holds if, and only if, whenever $\bar{y}$ is in the ith $\equiv^3$-equivalence class, there is a path of length p_i from x_1 to x_2, where p_i is the number represented by the first i bits of $(\leqslant^3, R)$.

Now, it is clear that the sentence $\exists^6 O \exists^3 R \varphi(O, R)$ expresses the problem UNI. Moreover, it follows from Corollary 3.9 that this is equivalent to a sentence of $\Sigma_1^{1,\omega}$. This enables us to establish the following theorem.

THEOREM 4.2. PTIME = NP *if, and only if,* UNI $\in$ FP.

Proof. In one direction, if UNI is definable in FP, it is solvable in polynomial time. Since the problem is NP-complete, this means that PTIME = NP.

In the other direction, if PTIME = NP, then by Theorem 3.10, $\Sigma_1^{1,\omega} = \text{FP}$. But, since UNI $\in \Sigma_1^{1,\omega}$, it follows that UNI $\in$ FP. ∎

4.2. 3-*Colourability*

In order to show that 3-colourability is not expressible in $L^{\omega}_{\infty\omega}$, we adapt a construction due to Cai *et al.* [5] which shows that there is a PTIME query that is not expressible in the extension of FP by counting.

The crucial idea in the construction of Cai *et al.* is to construct graphs X_d, which include d distinguished pairs of nodes $(a_1, b_1), ..., (a_d, b_d)$ with the following property:

(*) for every subset S of $\{1, ..., d\}$ which is of even cardinality, there is an automorphism of X_d which exchanges a_i and b_i for $i \in S$, while fixing a_i and b_i for $i \notin S$. There is no automorphism of X_d that does this for a set S of odd cardinality.

We refer to the pair of points (a_i, b_i) as the ith gate of X_d.

We can construct such an X_d by including, in addition to the d gates, 2^{d-1} nodes v_S, one for each even-sized subset S of $\{1, ..., d\}$. The graph X_d then contains the edges (a_i, v_S) for $i \in S$ and (b_i, v_S) for $i \notin S$. It can be easily verified that X_d has property (*).

EXAMPLE 4.3. The graph X_3 is depicted in Fig. 1.

Let G be a graph such that every vertex in G has degree at least 2. The graph $X(G)$ is defined as follows. Every vertex v in G is replaced by a copy of X_d, where d is the degree of v, with each edge incident on v being assigned a gate of X_d. We denote the copy of X_d that replaces v by X^v, and its ith gate by (a_i^v, b_i^v). For an edge (u, v) of G, let the gates in X^u and X^v assigned to this edge be (a_i^u, b_i^u) and (a_j^v, b_j^v). The graph $X(G)$ contains the two edges (a_i^u, a_j^v) and (b_i^u, b_j^v). We also define the graph $\tilde{X}(G)$, which is obtained from $X(G)$ by "twisting" exactly one edge. That is, for one edge (u, v) of G, in place of the edges (a_i^u, a_j^v) and (b_i^u, b_j^v), we include (a_i^u, b_j^v) and (b_i^u, a_j^v).

We now state two lemmas regarding this construction due to Cai *et al.* [5].

LEMMA 4.4. *$X(G)$ and $\tilde{X}(G)$ are not isomorphic.*

We omit a full proof of this lemma, oberving only that any isomorphism from $X(G)$ to $\tilde{X}(G)$ would, restricted to some X^v, yield an automorphism of X^v which exchanges a_i and b_i for an odd number of gates. This, however, would violate the property (*) of X^v.

Recall that a separator of a graph $G = (V, E)$ is a set of nodes $U \subseteq V$, such that the subgraph of G induced by $V \backslash U$ has no connected component containing more than $|V|/2$ nodes. This allows us to formulate the second lemma due to Cai *et al.* [5].

LEMMA 4.5. *If G has no separator of cardinality k, then $X(G) \equiv^k \tilde{X}(G)$.*

Once again, we omit a detailed proof of this lemma, and present instead an informal description of Duplicator's strategy in the pebble game. At any stage in the pebble game, there are at most k pebbles on each of $X(G)$ and $\tilde{X}(G)$. Consider the graphs formed by removing from $X(G)$ and $\tilde{X}(G)$ any X^v that contains a pebbled

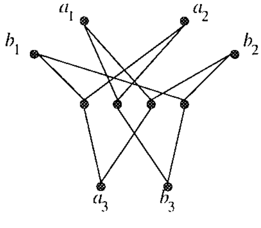

FIG. 1. X_3.

vertex. Since G has no separator of cardinality k, the resuling graphs each contain a connected component that includes more than half of all the vertices. Duplicator's strategy is essentially to "hide the twist" in this large component. Clearly, the only way Spoiler can win the game is by isolating the twist, i.e., by placing pebbles on two of the four vertices in the two gates of the twisted edge (u, v), in such a way as to force Duplicator to interchange the vertices in one of the gates, say of X^u. Ducplicator is then forced to interchange the vertices in another of the gates of X^u, effectively moving the twist to another location. Since after every move there is an unpebbled component containing more than half the vertices, these components must overlap from one move to the next. This allows Duplicator to always keep the twisted edge in the large component, and therefore Spoiler cannot isolate it.

To adapt the construction of Cai *et al.* to show that 3-colourability is not definable in $L^{\omega}_{\infty\omega}$, we construct a gadget C_d for every d, along the lines of X_d above, that has some additional properties. We state the relevant properties here, and defer the explicit construction to a later point in this section.

1. C_d contains d gates, each consisting of three nodes (a_i, b_i, c_i), and these nodes are connected by edges to form triangles.

2. For every subset S of $\{1, ..., d\}$ of even size, there is an automorphism of C_d that exchanges a_i and b_i for $i \in S$, while fixing nodes in all other gates. There is no such automorphism for sets S of odd size.

3. C_d is 3-colourable, and in any valid 3-colouring of C_d all c_i are assigned the same colour. There is a valid 3-colouring of C_d in which all a_i are assigned the same colour. Finally, this 3-colouring is unique up to renaming of colours and automorphisms of C_d.

One consequence of these conditions is that if d is even, then in any 3-colouring of C_d and for any of the colours, the number of a_i that are assigned that colour must be even.

We now define, for every graph G, the graphs $C(G)$ and $\tilde{C}(G)$ along the lines of $X(G)$ and $\tilde{X}(G)$ above. The only difference is that each edge (u, v) of G is now replaced by three edges in $C(G)$, (a^u_i, a^v_j), (b^u_i, b^v_j), and (c^u_i, c^v_j). The graph $\tilde{C}(G)$ is obtained from $C(G)$ by replacing exactly one pair (a^u_i, a^v_j), (b^u_i, b^v_j) of edges by (a^u_i, b^v_j), (b^u_i, a^v_j). The following lemma is now immediate, along the lines of Lemma 4.5.

LEMMA 4.6. *If G has no separator of cardinality k, then $C(G) \equiv^k \tilde{C}(G)$.*

We proceed to construct, for every $k \in \omega$, a graph which has no separator of cardinality k, which is 3-colourable and such that its 3-colouring is unique up to a renaming of colours.

DEFINITION 4.7. A *triangular mesh* of order n, denoted T_n, is a graph with n^2 vertices: $v_{(i, j)}$, $0 \leqslant i, j < n$, and for each i and j the edges

$$[v_{(i, j)}, v_{(i+1, j)}]; \qquad [v_{(i, j)}, v_{(i, j+1)}]; \qquad [v_{(i, j)}, v_{(i+1, j+1)}],$$

where the additions are all modulo n.

We now establish the relevant properties of triangular meshes in the next three lemmas.

LEMMA 4.8. *For $n \geqslant 3$, T_n has no separator of cardinality n.*

Proof. For each i, define the row R_i to be the set of vertices $\{v_{(i,j)} \mid 0 \leqslant j < n\}$. Similarly define the column $C_j = \{v_{(i,j)} \mid 0 \leqslant i < n\}$. Note that the subgraph of T_n induced by each row and each column is a cycle of length n.

Let U be any subset of the vertices of T_n such that $|U| = n$. We first note that if U contains one vertex from every row or one vertex from every column, then the result of removing the vertices in U from T_n is a connected graph. To see this, suppose U contains one vertex from every row, then after the removal of vertices in U, every row remains connected, since it is a cycle with one vertex removed, and since two successive rows are connected by $2n$ edges, the removal of one vertex in each row will not disconnect them. Thus, such a U does not form a separator.

Next, we consider a set U of cardinality n which for some row and for some column includes at least two of its vertices. But then, there must be R_i and C_j such that $R_i \cap U = \varnothing$ and $C_j \cap U = \varnothing$. Note that $R_i \cup C_j$ contains $2n-1$ vertices. Now, for any other column C_l such that $|C_l \cap U| \leqslant 1$, all the elements of C_l are connected to $v_{(i,l)}$ by a path that does not include a vertex in U. It follows that, after U is removed, at least $n-2$ of the vertices in at least half of the remaining columns are in the same connected component as $R_i \cup C_j$. Thus, this component contains at least $(2n-1) + (n-2)^2/2$ vertices, which is more than half of the vertices of T_n. Thus, U is not a separator of T_n. ∎

LEMMA 4.9. *If n is a multiple of 3, then T_n is 3-colourable.*

Proof. We define a 3-colouring $\chi\colon T_n \to \{0, 1, 2\}$ given by:

$$\chi(v_{(i,j)}) = l \qquad \text{if, and only if,} \quad i + j \equiv l \pmod 3.$$

It is easily seen that this is a valid 3-colouring of the graph. ∎

LEMMA 4.10. *If n is a multiple of 3, then $C(T_n)$ is 3-colourable, and $\tilde{C}(T_n)$ is not 3-colourable.*

Proof. To see that $C(T_n)$ is 3-colourable, consider a valid 3-colouring of T_n, $\chi\colon T_n \to \{0, 1, 2\}$. For each vertex v of T_n, we can colour the graph C^v in such a way that all c_i^v are assigned the colour $\chi(v)$, all a_v^i are assigned the colour $(\chi(v)+1) \bmod 3$, and all b_v^i are assigned the colour $(\chi(v)+2) \bmod 3$. It can then be easily verified that this results in a valid 3-colouring of $C(T_n)$.

To show that $\tilde{C}(T_n)$ is not 3-colourable, we make the following observations. First, the edges of T_n can be partitioned into $3n$ sets, each of which forms a cycle of length n. These are given by the n rows $R_i = \{(i,j) \mid 0 \leqslant j < n\}$, the n columns $C_j = \{(i,j) \mid 0 \leqslant i < n\}$ and the n diagonals $D_k = \{(i, k + i \bmod n) \mid 0 \leqslant i < n\}$. Each vertex then appears in exactly three such cycles. Second, given any valid 3-colouring of $\tilde{C}(T_n)$, we obtain a valid 3-colouring of T_n by assigning the colour of c_i^v to the vertex v. In particular, this implies that if we consider the colours of c_i^v in an n-cycle, then they must strictly alternate among the three colours. Third, given a gadget C^v in $\tilde{C}(T_n)$, we pair off the gates of C^v into three pairs, the horizontal, vertical, and diagonal, according to which n-cycle they appear in. Given a 3-colouring of $\tilde{C}(T_n)$, if the two a_i^v in one such pair are of different colours, we say that C^v makes a switch in the corresponding n-cycle. By property (3) of the gadgets C^v, it follows that each one of them makes an even number of switches. Finally, we note that for a valid 3-colouring of $\tilde{C}(T_n)$, each n-cycle must contain an even number of switches, *except* the unique n-cycle containing the twisted edge, which must contain an odd number of switches. It can easily be verified that these last two requirements lead to a contradiction. ∎

Finally, we give an explicit construction of the gadget C_d having the properties (1)–(3) listed above. Note that, since the graph T_n is regular of degree 6, it would suffice to give a construction of C_6. However, we present a general purpose construction.

The graph C_d has $16d$ nodes altogether. It contains a spine of $3d$ nodes, $s_0, ..., s_{3d-1}$, with edges (s_i, s_{i+1}) and (s_i, s_{i+2}) for all i. Here, and in the rest of this section, addition in the subscripts is understood to be modulo $3d$. That is, the spine consists of a cycle of length $3d$, along with all its chords of length two. Thus, in any 3-colouring of C_d, the nodes s_i and s_{i+3} must be assigned the same colour, for all i, i.e., the colours along the spine strictly alternate among all three colours.

Next, adjacent to each s_i, there are two additional vertices l_i and r_i, which are also adjacent to each other. Morover, if $i \equiv 1 \pmod 3$ or $i \equiv 2 \pmod 3$, then l_i (resp. r_i) is adjacent to l_{i+1} (resp. r_{i+1}). A part of the spine is depicted in Fig. 2. For clarity, the chords of length two along the spine are omitted. It can be verified from Fig. 2 that given a 3-colouring of the spine, the colouring of the l_i and r_i is determined up to automorphism.

Finally, in the gap between s_{3i+2} and $s_{3(i+1)}$, we place the ith gate of C_d, by attaching the pair (a_i, b_i) to the pairs (l_{3i+2}, r_{3i+2}) and $(l_{3(i+1)}, r_{3(i+1)})$ by means of the gadget X_3 of Example 4.3. To ensure uniqueness of 3-colouring, we also connect a_i and b_i by edges to s_{3i+2}. An example is depicted in Fig. 3, where, for clarity, the edges that are part of X_3 are indicated by dashed lines. It can be verified from Fig. 3 that, if the colouring of l_{3i+2}, r_{3i+2}, $l_{3(i+1)}$ and $r_{3(i+1)}$ is fixed, this also fixes the colouring of a_i and b_i.

If we now consider an automorphism that exchanges a_i and b_i, it must also exchange exactly one of the pairs (l_{3i+2}, r_{3i+2}) or $(l_{3(i+1)}, r_{3(i+1)})$. Assuming, without loss of generality, that it is the latter, the automorphism must also exchange (l_{3i+4}, r_{3i+4}) and (l_{3i+5}, r_{3i+5}) which takes us to the next gate, where we can choose either to exchange a_{i+1} and b_{i+1} or to continue to exchange nodes l

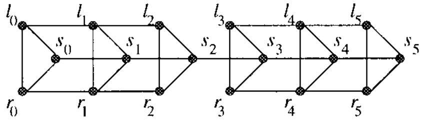

FIG. 2. A portion of the spine of C_d.

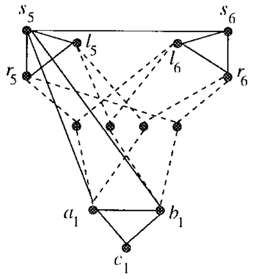

FIG. 3. A gate of C_d.

and r further along the spine. In any case, we must exchange a_j and b_j for some $j \neq i$, which gives us the required properties of the gadget C_d.

Thus, having established that there exist gadgets C_d with the properties (1)–(3), the main theorem of this section follows as a consequence of Lemmas 4.6, 4.8, and 4.10.

THEOREM 4.11. *3-colourability is not expressible in* $L^{\omega}_{\infty\omega}$.

Remark 4.12. The proof of Theorem 4.11 given above can be adapted to show that 3-colourability is not even definable in the extension of $L^{\omega}_{\infty\omega}$ with counting quantifiers. To see this, note that if we add two constants c and d to our signature and interpret them in a triangular mesh T_n by two adjacent vertices in the same row, then there is an FP formula that defines a linear order in T_n. It follows that if c and d are interpreted by vertices in adjacent gadgets in $C(T_n)$ and $\tilde{C}(T_n)$, then the sizes of the $\equiv^k$ equivalence classes in these structures are bounded. Moreover, we can now even remove the constants from our signature simply by distinguishing the two gadgets in some identifiable way, say an extra vertex that does not interfere with the relevant properties of the gadget. However, when the sizes of the $\equiv^k$ equivalence classes are bounded, then counting quantifiers do not add to the expressive power of $L^{\omega}_{\infty\omega}$ (for details see [5]).

5. CONCLUSION

A great deal of research in finite model theory has been inspired by the discovery of the close connection between logical expressibility and computational complexity. This discovery raises the possibility of applying model-theoretic methods to attack outstanding open problems in complexity theory. Unfortunately, most of the results and methods that have been developed in the study of infinite models do not apply when only finite models are considered. To a large extent, the classical subject of model theory can be seen to be the study of the relation of elementary equivalence. However, this equivalence relation turns out to be of limited interest in the logical study of finite models, since it is identical with isomorphism. Recent work has shown, nonetheless, that there is an equivalence relation (or rather a countable collection of such relations), namely $\equiv^k$, which has a close connection with logical definability and which is nontrivial on finite models. Moreover, as this paper illustrates, outstanding questions in complexity theory can be reformulated in a context where this equivalence relation corresponds with the notion of definability. This further underlines the need to study the model theory of finite variable logics.

ACKNOWLEDGMENTS

The research reported here has greatly benefited from discussions with Lauri Hella and Phokion Kolaitis. The notation introduced in Definition 3.1 was suggested by Scott Weinstein. I am also grateful to Lauri Hella and Victor Vianu for a careful reading of a draft of this paper.

Final manuscript received December 23, 1997

REFERENCES

1. Abiteboul, S., Vardi, M. Y., and Vianu, V. (1997), Fixpoint logics, relational machines, and computational complexity, *J. Assoc. Ccomput. Mach.* **44**, 30–56.
2. Abiteboul, S., and Vianu, V. (1991), Datalog extensions for database queries and updates, *J. Comput. System Sci.* **43**, 62–124.
3. Abiteboul, S., and Vianu, V. (1995), Computing with first-order logic, *J. Comput. System Sci.* **50**(2), 309–335.
4. Barwise, J. (1977), On Moschovakis closure ordinals, *J. Symbolic Logic* **42**, 292–296.
5. Cai, J.-Y., Fürer, M, and Immerman, N. (1992), An optimal lower bound on the number of variables for graph identification, *Combinatorica* **12**(4), 389–410.
6. Chandra, A., and Harel, D. (1982), Structure and complexity of relational queries, *J. Comput. System Sci.* **25**, 99–128.
7. Dahlhaus, E. (1984), Reduction to NP-complete problems by interpretation, *in* "Lecture Notes in Computer Science," Vol. 171, pp. 357–365, Springer-Verlag, Berlin/New York.
8. Dawar, A (1993), "Feasible Computation through Model Theory," Ph.D. thesis, University of Pennsylvania.
9. Dawar, A., and Grädel, E. (1995), Generalized quantifiers and 0–1 laws, *in* "Proc. 10th IEEE Symp. on Logic in Computer Science," pp. 54–64.
10. Dawar, A., Lindell, S, and Weinstein, S. (1995), Infinitary logic and inductive definability over finite structures, *Inform. and Comput.* **119**(2), 160–175.
11. de Rougemont, R. (1987), Second-order and inductive definability on finite structures, *Z. Math. Logik Grundlag. Math.* **33**, 47–63.
12. Fagin, R. (1974), Generalized first-order spectra and polynomial-time recognizable sets, *in* "Complexity of Computation" (R. M. Karp, Ed.), SIAM–AMS Proceedings, Vol. 7, pp. 43–73.
13. Garey, M. R., and Johnson, D. S. (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, New York.
14. Gurevich, Y., and Shelah, S. (1986), Fixed-point extensions of first-order logic, *Ann. Pure Appl. Logic* **32**, 265–280.
15. Immerman, N. (1982), Upper and lower bounds for first-order expressibility, *J. Comput. System Sci.* **25**, 76–98.
16. Immerman, N. (1986), Relational queries computable in polynomial time, *Inform. and Control* **68**, 86–104.
17. Immerman, N. (1989), Descriptive and computational complexity, *in* "Computational Complexity Theory, Proc. of AMS Symposia in Appl. Math." (J. Hartmanis, Ed.), Vol. 38, pp. 75–91.
18. Kolaitis, Ph. G., and Vardi, M. Y. (1992), Fixpoint logic vs. infinitary logic in finite-model theory, *in* "Proc. 7th IEEE Symp. on Logic in Computer Science," pp. 46–57.

19. Kolaitis, Ph. G., and Vardi, M. Y. (1992), Infinitary logics and 0–1 laws, *Inform. and Comput* **98**(2), 258–294.

20. Leivant, D. (1990), Inductive definitions over finite structures, *Inform. and Comput.* **89**, 95–108.

21. Lovász, L., and Gács, P. (1977), Some remarks on generalized spectra, *Z. Math. Logik Grundlag. Math.* **23**, 27–144.

22. Moschovakis, Y. N. (1974), "Elementary Induction on Abstract Structures," North-Holland, Amsterdam.

23. Rubin, A. (1975), "Free Algebras in Von Neumann–Bernays–Gödel Set Theory and Positive Elementary Inductions in Reasonable Structures," Ph.D. thesis, California Institute of Technology.

24. Stockmeyer, L. (1976), The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3**, 1–22.

25. Vardi, M. Y. (1982), The complexity of relational query languages, *in* "Proc. 14th ACM Symposium on the Theory of Computing," pp. 137–146.