Existential Second-Order Logic Over Strings

THOMAS EITER AND GEORG GOTTLOB

Technische Universität Wien, Wien, Austria

AND

YURI GUREVICH

Microsoft Research, Redmond, Washington

Abstract. Existential second-order logic (ESO) and monadic second-order logic (MSO) have attracted much interest in logic and computer science. ESO is a much more expressive logic over successor structures than MSO. However, little was known about the relationship between MSO and syntactic fragments of ESO. We shed light on this issue by completely characterizing this relationship for the prefix classes of ESO over strings, (i.e., finite successor structures). Moreover, we determine the complexity of model checking over strings, for all ESO-prefix classes. Let ESO($\mathcal{Q}$) denote the prefix class containing all sentences of the shape $\exists \mathbf{R} Q \varphi$, where $\mathbf{R}$ is a list of predicate variables, Q is a first-order quantifier prefix from the prefix set $\mathcal{Q}$, and φ is quantifier-free. We show that ESO($\exists^*\forall\exists^*$) and ESO($\exists^*\forall\forall$) are the maximal standard ESO-prefix classes contained in MSO, thus expressing only regular languages. We further prove the following dichotomy theorem: An ESO prefix-class either expresses only regular languages (and is thus in MSO), or it expresses some NP-complete languages. We also give a precise characterization of those ESO-prefix classes that are *equivalent* to MSO over strings, and of the ESO-prefix classes which are closed under complementation on strings.

Categories and Subject Descriptors: F.1.1. [**Computation by Abstract Devices**]: Models of Computation—*automata*, *relations between models*; F2.2. [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*; F.4.1. [**Mathematical Logic and Formal Languages**]: Mathematical Logic—*computational logic*; *model theory*; F.4.3. [**Mathematical Logic and Formal Languages**]: Formal Languages—*classes defined by grammars or automata*

General Terms: Theory

An extended abstract of this paper has appeared in *Proceedings of the 13th IEEE Symposium on Logic in Computer Science* (*LICS '98*) (Indianapolis, Ind., June 21–24). IEEE Computer Society Press, Los Alamitos, Calif., 1998, pp. 16–27.

Y. Gurevich did this research while on a leave of absence from the University of Michigan.

This research was supported by the Austrian Science Fund Project NZ29-1NF and by National Science Foundation (NSF) grant CCR 95-04375.

Authors' Present Addresses: T. Eiter, Institut für Informationssysteme, Technische Universität Wien, A-1040 Wein, Austria, e-mail: eiter@kr.tuwien.ac.at; G. Gottlob, Institut für Informationssysteme, Technische Universität Wien, A-1040 Wien, Austria, e-mail: gottlob@dbai.tuwien.ac.at; Y. Gurevich, Microsoft Research, One Microsoft Way, Redmond, WA 98052.

Additional Key Words and Phrases: Finite model theory, second-order logic, existential fragment, prefix classes, model checking, finite words, strings, regular languages, S1S, descriptive complexity, NP, decision problem, finite satisfiability.

1. *Introduction*

Second-order logic over finite structures has attracted the interest of logicians, mathematicians, and computer scientists for a long time. In particular, several important results have been obtained that link logic to automata theory and complexity theory.

Two fundamental results in this context are the famous Büchi Theorem [Büchi 1960], which says that monadic second-order logic (MSO) over strings precisely characterizes the regular languages, and Fagin's Theorem [Fagin 1974], which states that existential second-order logic (ESO) exactly expresses the NP properties over finite structures (in particular, over finite strings). Thus, over strings, ESO is a much more expressive logic than MSO. However, little was known about the relationship between syntactic fragments of ESO and MSO.

In this paper, we shed light on this issue by investigating prefix classes of (nonmonadic) ESO over finite strings. Before explaining in more detail both the problem studied and the results obtained, we spend a few paragraphs describing the context of our research and the most important earlier results related to it.

1.1. Logical Characterization of NP. The class NP is one of the most well-known classes of problems, and has been attracting much interest from both the practical as well as the theoretical side. To date, a large and steadily increasing number of problems in practice is known to be complete for this class, while no polynomial time algorithm for any of these problems is known; the P = NP question is one of the main challenging open problems in computer science.

This question has been tackled from different directions in the hope of utilizing tools and the rich body of knowledge from different, well-developed areas. One such attempt was a reduction of the P = NP question to problems in logic. In this context, Fagin [1974] gave a purely logical characterization of NP in terms of second-order logic, where there is no notion of machine, computation, or time. He proved that over finite structures, the properties which are decidable in NP are precisely those which are definable in existential second-order logic (ESO), that is, expressible through a sentence of the form $\exists \mathbf{R}\varphi$, where $\exists \mathbf{R}$ means existential quantification over a list $\mathbf{R} = R_1, \ldots, R_n$ of relational variables R_i and φ is a first-order formula.

Fagin's Theorem was successfully used in various areas for establishing different types of results. For example, it has been exploited in database theory for assessing the expressive power of query languages, cf. Kolaitis and Papadimitriou [1991], Schlipf [1995], Saccá [1997], or in computation theory to characterize subclasses of NP or to establish logically defined hierarchies of nondeterministic complexity classes within NP.[1] Another use is in the area of optimization theory, where based on Fagin's Theorem, logical definitions of classes of optimization problems were given, cf. Papadimitriou and Yannakakis [1991], Panconesi and Ranjan [1993], and Kolaitis and Thakur [1995].

[1] See, for example, Ajtai [1983], Grandjean [1985], Lynch [1992], and Olive [1998].

1.2. Prefix Classes. In the above investigations, syntactic subclasses of ESO were studied. In particular, *prefix classes* play an important role. Prefix classes are the most natural and the most commonly studied fragments of predicate logic. A prefix class is a class of formulas in prenex normal form whose quantifier prefixes obey a certain pattern. Denote by ESO($\mathfrak{Q}$) the prefix class consisting of all ESO-sentences $\exists \mathbf{R}\varphi$, where φ is in prenex normal form with a quantifier prefix from a first-order prefix class $\mathfrak{Q}$. Then, Fagin's Theorem actually characterizes NP as the class ESO($\forall^*\exists^*$).

The interest in prefix-classes dates back a long time ago. Actually, the classical decision problem of Hilbert is the following problem, where FO($\mathfrak{Q}$) denotes the set of prenex first-order formulas, possibly containing free occurrences of predicate variables:

Instance: formula φ in FO($\mathfrak{Q}$)

Question: Is φ (finitely) satisfiable?

Reformulated in the context of second-order logic, this is equivalent to whether for a given $\mathfrak{Q}$:

Instance: formula φ in ESO($\mathfrak{Q}$)

Question: Is φ (finitely) satisfiable?

This question has been studied in depth over the past decades, and an exhaustive classification of decidable and undecidable prefix classes is known (see Börger et al. [1997]); there are huge complexity gaps between elementarily decidable and undecidable classes. The ESO($\mathfrak{Q}$) classification played an important role in the identification of fragments of ESO that obey the 0-1 law, that is, the property that over finite structures, a sentence is almost surely true or almost surely false, cf. Kolaitis and Vardi [1990] and Pacholski and Szwast [1991]. It turned out that there is a close relationship between decidable classes and those satisfying the 0-1 law.

1.3. Ordered Structures, Strings, and Büchi's Theorem. While the above results on NP are on arbitrary finite structures, computer science mainly deals with *ordered structures*. In fact, the input to a computing device such as a finite automaton or a Turing machine is implicitly ordered by the position of the data in the input stream or on the input tape, respectively. Accordingly, many important issues in finite model theory were considered in the context of ordered structures (see below for a short description of results and references).

Strings, that is, words over a finite alphabet, are ordered structures of particular importance. The set of strings satisfying a given formula Φ is a formal language. We can thus directly compare classes of formal languages to logical formalisms.

In order to do this formally, we need to confine ourselves to a logical representation of strings. There are several possibilities and we have chosen the simplest here. We represent a string over an alphabet A as a structure over a finite universe $U = \{1, \ldots, n\}$ (representing the positions of the string), equipped with the natural successor relation *Succ* over U, constants *min* and *max* for the first and the last position, respectively, and a predicate C_a for each letter $a \in A$, such that $C_a(i)$ is true iff the ith position of the string consists of letter a.

A fundamental result relating logic to formal languages is Büchi's Theorem [Büchi 1960] (also found by Trakhtenbrot [1961]), which says that monadic second-order logic (MSO) defines over finite strings precisely the regular languages. Hence, over finite strings, MSO is much weaker than ESO, which expresses all languages in NP.

In fact, over finite strings, even the class ESO($\forall^*$) expresses all languages in NP; this follows from the more general result that over finite successor structures, that is, finite structures equipped with a successor predicate, every ESO sentence is equivalent to some ESO($\forall^*$) sentence [Leivant 1989; Eiter et al. 1996].

1.4. MAIN PROBLEMS STUDIED. Combining and extending the results of Büchi and Fagin, it is natural to ask: What about (nonmonadic) prefix classes ESO($\mathcal{Q}$) over finite strings? We know by Fagin's theorem that all these classes describe languages in NP. But there is a large spectrum of languages contained in NP ranging from regular languages (at the bottom) to NP-hard languages at the top. What can be said about the languages expressed by a given prefix class ESO($\mathcal{Q}$)? Can the expressive power of these fragments be characterized? In order to clarify these issues, we investigated, in particular, the following problems:

—Which classes ESO($\mathcal{Q}$) express only regular languages?
In other terms, for which fragments ESO($\mathcal{Q}$) is it true that for any formula $\Phi \in$ ESO($\mathcal{Q}$) the set $Mod(\Phi) = \{W | W \models \Phi\}$ of all finite strings (over a given finite alphabet) satisfying Φ constitutes a regular language? Any fragment fulfilling this condition is called *regular*. By Büchi's Theorem this question is identical to the following: Which prefix classes of ESO are (semantically) included in MSO?

Note that by Gurevich's classifiability theorem (cf. Börger et al. [1997]) and by elementary closure properties of regular languages, it follows that there is *a finite number of maximal regular prefix classes* ESO($\mathcal{Q}$), and similarly, of minimal nonregular prefix classes; the latter are, moreover, *standard*, that is, the quantifier prefix class $\mathcal{Q}$ is either the set of all prefixes or it can be described by a string over the alphabet $\{\forall, \exists, \forall^*, \exists^*\}$. It is our aim to determine the maximal regular prefix classes and the minimal nonregular prefix classes.

—What is the complexity of model checking (over strings) for the *nonregular* classes ESO($\mathcal{Q}$), that is, deciding whether $W \models \Phi$ for a given W (where Φ is fixed)?

Model checking for regular classes ESO($\mathcal{Q}$) is easy: it is feasible by a finite state automaton. We also know (e.g., by Fagin's Theorem) that some classes ESO($\mathcal{Q}$) allow us to express NP-complete languages. It is therefore important to know (i) which classes ESO($\mathcal{Q}$) can express NP-complete languages, and (ii) whether there are prefix classes ESO($\mathcal{Q}$) of intermediate complexity between regular and NP-complete classes.

—Which classes ESO($\mathcal{Q}$) *capture* the class REG of all regular languages? A class of logical sentences captures REG, if all regular languages and only those can be expressed in it. By Büchi's Theorem, this question is equivalent to the question of which classes ESO($\mathcal{Q}$) have exactly the expressive power of MSO over strings.

—For which classes ESO($\mathcal{Q}$) is finite satisfiability decidable, that is, given a formula $\Phi \in$ ESO($\mathcal{Q}$), decide whether Φ is true on some finite string?

—Which classes ESO($\mathcal{Q}$) are closed under complementation over strings?

1.5. MAIN RESULTS. The present paper answers all the above questions exhaustively. Some of our results are rather unexpected. In particular, we prove a surprising dichotomy theorem which sharply classifies all ESO($\mathcal{Q}$) classes as either regular or intractable. Our main results are summarized as follows:

(1) The class ESO($\exists^*\forall\exists^*$) is regular (Theorem 8.2). This theorem constitutes the technically most involved result of this paper. Given that this class is nonmonadic, it was not possible to exploit any of the ideas underlying Büchi's proof for proving it regular. The main difficulty consists in the fact that relations of higher arity may connect elements of a string that may be very distant from one another and it is not a priori clear how a finite state automaton should be able to guess such connections and check their global consistency. To solve this problem, we have to develop completely new methods. In particular, we prove new combinatorial results on hypergraphs and applied them to logic.

Interestingly, model checking for the fragment ESO($\exists^*\forall\exists^*$) is NP-complete over *graphs*. For example, the well-known set-splitting problem can be expressed in it. Thus, the fact that our input structures are monadic *strings* is essential (just as for MSO).

(2) The class ESO($\exists^*\forall\forall$) is regular (Theorem 9.1). The regularity proof for this fragment is easier but also requires new techniques. Note that model checking for this class, too, is NP-complete over graphs.

(3) Any class ESO($\mathcal{Q}$) not contained in the union of ESO($\exists^*\forall\exists^*$) and ESO($\exists^*\forall\forall$) is not regular (Proposition 3.1).

Thus, ESO($\exists^*\forall\exists^*$) and ESO($\exists^*\forall\forall$) are the *maximal regular standard prefix classes*. The unique maximal (general) regular ESO-prefix class is the union of these two classes, i.e, ESO($\exists^*\forall\exists^*$) $\cup$ ESO($\exists^*\forall\forall$) = ESO($\exists^*\forall(\forall \cup \exists^*)$) (Theorem 9.7).

It turns out that there are three minimal nonregular ESO-prefix classes, namely the standard prefix classes ESO($\forall\forall\forall$), ESO($\forall\forall\exists$), and ESO($\forall\exists\forall$). All these classes express nonregular languages by formulas whose list of second-order variables consists of a single binary predicate variable.

Therefore, results (1)–(3) give a *complete characterization* of the regular classes ESO($\mathcal{Q}$). A picture of the situation is given in Figure 1. The picture also visualizes further results, and is explained in more detail below.

(4) We obtain the following dichotomy theorem: Let ESO($\mathcal{Q}$) be any prefix class. Then, either ESO($\mathcal{Q}$) is regular, or ESO($\mathcal{Q}$) expresses some NP-complete language (Theorem 10.5). This means that model checking for ESO($\mathcal{Q}$) is either possible by a deterministic finite automaton (and thus in constant space and linear time) or it is already NP-complete. Moreover, for all NP-complete classes

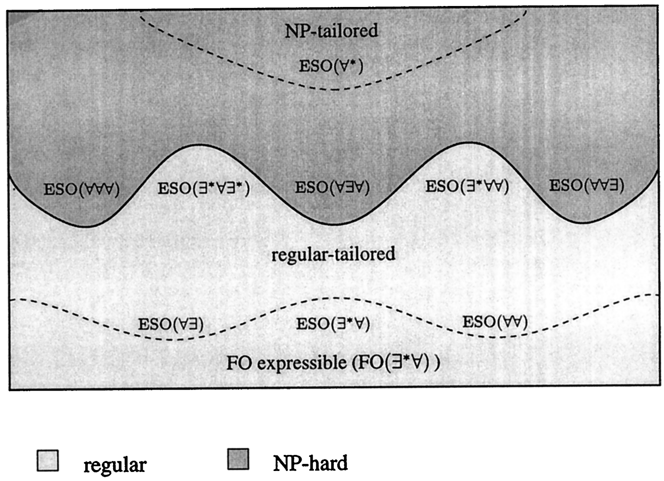


FIG. 1. Complete picture of the ESO-prefix classes on finite strings.

ESO($\mathfrak{Q}$), NP-hardness holds already for sentences whose list of second-order variables consists of a single binary predicate variable. There are no fragments of intermediate difficulty between REG and NP.

(5) The above dichotomy theorem is paralleled by the solvability of the finite satisfiability problem for ESO (and thus FO) over strings. We show that over finite strings, satisfiability of a given formula from a class ESO($\mathfrak{Q}$) is decidable if and only if ESO($\mathfrak{Q}$) is regular (Theorem 11.1).

(6) We give a precise characterization of those prefix classes of ESO that are equivalent to MSO over strings, that is, of those prefix fragments that *capture* the class REG of regular languages (Theorem 12.5). This provides us with completely new logical characterizations of REG. Moreover, we establish that any regular ESO-prefix class is over strings either equivalent to full MSO, or is contained in first-order logic, in fact, in FO($\exists^*\forall$) (Theorem 12.3). We further show that there is a unique minimal ESO prefix class which captures NP, namely ESO($\forall^*$) (Proposition 10.6). Our proof uses results in Leivant [1989] and Eiter et al. [1996] and well-known hierarchy theorems.

(7) We give a precise characterization of those regular prefix classes of ESO which, over strings, are closed under complementation. In particular, we show that any nontrivial regular class ESO($\mathfrak{Q}$) is closed under complementation iff some quantifier prefix $Q \in \mathfrak{Q}$ contains either the sequence $\forall\forall$ or the sequence $\forall\exists$; this is the case iff ESO($\mathfrak{Q}$) captures REG (Theorem 12.5). Moreover, it follows from our previously described results (1)–(4) that if NP $\neq$ co-NP, then no nonregular prefix class is closed under complementation. Assuming NP $\neq$ co-NP we have thus completely determined those prefix classes of ESO which are closed under complementation over strings (Theorem 12.7).

Our main results are summarized in Figure 1. In this figure, the set of all ESO-prefix classes is divided into four regions. The upper two regions contain all classes that express nonregular languages, and therefore, as we show, also NP-complete languages. The uppermost region contains those classes which capture NP, that is, express all languages in NP. These classes are called NP-*tailored*. The region next below, separated by a dashed line, contains those classes which can express some NP-hard languages, but not all languages in NP. Its bottom is constituted by the minimal nonregular classes, ESO($\forall\forall\forall$), ESO($\forall\exists\forall$), and ESO($\forall\forall\exists$). The lower two regions contain all regular classes. The maximal regular standard prefix classes are ESO($\exists^*\forall\exists^*$) and ESO($\exists^*\forall\forall$). The dashed line separates the classes which capture precisely REG (called *regular-tailored*), from those which do not; by our results, the expressive capability of the latter classes is restricted to first-order logic (in fact, to FO($\exists^*\forall$)). The minimal classes which capture REG are ESO($\forall\exists$) and ESO($\forall\forall$).

Furthermore, all nontrivial classes contained in the lowest region are (provably) not closed under complementation. All classes in the upper region of regular classes (regular-tailored) are closed under complementation, while those in the region above regular classes are not closed under complementation unless NP = co-NP.

1.6. Potential Applications. Monadic second-order logic over strings is currently used in the verification of hardware, software, and distributed systems. An example of a specific tool for checking specifications based on MSO is the MONA tool developed at the BRICS research lab in Denmark [Basin and Klarlund 1995; Henriksen et al. 1996; Klarlund 1998].

Observe that certain interesting desired properties of systems are most naturally formulated in *nonmonadic* second-order logic. Consider, as an unpretentious example,[2] the following property of a ring P of processors of different types, where two types may either be compatible or incompatible with each other. We call P *tolerant*, if for each processor p in P there exist two other distinct processors $backup_1(p) \in P$ and $backup_2(p) \in P$, both compatible with p, such that the following conditions are satisfied:

(1) for each $p \in P$ and for each $i \in \{1, 2\}$, $backup_i(p)$ is not a neighbor of p;
(2) for each $i, j \in \{1, 2\}$, $backup_i(backup_j(p)) \notin \{p, backup_1(p), backup_2(p)\}$.

Intuitively, we may imagine that in case p breaks down, the workload of p can be reassigned to $backup_1(p)$ or to $backup_2(p)$. Condition (1) reflects the intuition that if some processor is damaged, there is some likelihood that also its neighbors are (e.g., in case of physical affection such as radiation), thus neighbors should not be used as backup processors. Condition (2) states that the backup processor assignment is antisymmetric and anti-triangular; this ensures, in particular, that the system remains functional, even if two processors of the same type are broken (further processors of incompatible type might be broken, provided that broken processors can be simply bypassed for communication).

Let T be a fixed set of processor types. We represent a ring of n processors numbered from 1 to n where processor i is adjacent to processor $i + 1$ $(mod\ n)$ as a string of length n from T^* whose ith position is τ if the type of the ith processor is τ; logically, $C_\tau(i)$ is then true. The property of P being tolerant is

[2] Our goal here is merely to give the reader some intuition about a possible type of application.

expressed by the following second-order sentence Φ:

$$\begin{aligned}\Phi : \exists R_1 \exists R_2 \forall x \exists y_1 \exists y_2 . & compat(x, y_1) \wedge compat(x, y_2) \wedge \\ & R_1(x, y_1) \wedge R_2(x, y_2) \wedge \\ & \bigwedge_{i=1,2} \bigwedge_{j=1,2} (\neg R_i(y_j, x) \wedge \neg R_1(y_j, y_i) \wedge \neg R_2(y_j, y_i)) \wedge \\ & x \neq y_1 \wedge x \neq y_2 \wedge y_1 \neq y_2 \wedge \\ & \neg Succ(x, y_1) \wedge \neg Succ(y_1, x) \wedge \neg Succ(x, y_2) \wedge \neg Succ(y_2, x) \wedge \\ & ((x = max) \rightarrow (y_1 \neq min \wedge y_2 \neq min)) \wedge \\ & ((x = min) \rightarrow (y_1 \neq max \wedge y_2 \neq max)),\end{aligned}$$

where $compat(x, y)$ is the abbreviation for the formal statement that processor x is compatible to processor y (which can be encoded as a simple Boolean formula over C_τ atoms).

Φ is the natural second-order formulation of the tolerance property of a ring of processors. This formula is in the fragment ESO($\exists^*\forall\exists^*$); hence, by our results, we can immediately classify tolerance as a regular property, that is, a property that can be checked by a finite automaton.

In a similar way, one can exhibit examples of ESO($\exists^*\forall\forall$) formulas that naturally express interesting properties whose regularity is not completely obvious a priori. We thus hope that our results may find applications in the field of computer-aided verification.

1.7. FURTHER RELATED WORK. Since Büchi's logical characterization of the regular languages and Fagin's logical characterization of NP, several further logical characterizations of complexity classes or types of formal languages have been obtained.

The following are some classical results concerning general (not necessarily monadic) finite structures. Stockmeyer [1977] has shown that full second-order logic captures the polynomial hierarchy (PH). Immerman [1986] and Vardi [1984] proved that polynomial time is captured by fixpoint logic over ordered structures, and Grädel [1991; 1992] established this for ESO($\forall^*$, Horn) (see also Leivant [1989] where this result occurs implicitly). The related result that the well-known database query language *Datalog* captures P over ordered structures is already implicit in Vardi [1984] and Immerman [1986]. Abiteboul and Vianu have studied several other database query languages, and they showed that the classes of total and partial fixpoint queries coincide on arbitrary finite structures if and only if P = PSPACE (see Abiteboul et al. [1995]). Many complexity classes, including LOGSPACE and NLOGSPACE, were logically characterized by Immerman [1986]. Most of these results and many others are covered by books or surveys.[3]

[3] See, for example, Immerman [1999], Ebbinghaus and Flum [1995], Gurevich [1988], and Fagin [1993].

Our results add to previous knowledge about the relationships between nonmonadic ESO fragments and MSO over strings. They contrast with previous results on graphs. We show that existential MSO and ESO($\exists^*\forall\exists^*$) coincide over strings. This is not true for graphs. It was known that over finite graphs, disconnectivity is expressible in existential MSO [Fagin 1975], and 2-colorability and completeness of a graph are clearly in existential MSO; however, none of these properties is expressible in ESO($\exists^*\forall\exists^*$), even in presence of a successor [Eiter and Gottlob 1998]. Therefore, ESO($\exists^*\forall\exists^*$) and MSO have different expressive power over ordered graphs. Further relevant discussions of ESO and MSO fragments over graphs and general structures can be found in de Rougemont [1987], Schwentick [1994; 1995], Rosen [1999], and Durand et al. [1998].

To the best of our best knowledge, there has been no previous characterization of the regular languages by nonmonadic fragments of ESO. However, many papers cover either extensions or restrictions of MSO or REG.

Lynch [1992] for example, has studied the logic over strings obtained from existential MSO by augmenting it with addition. He proved that this logic captures NTIME(n), that is, nondeterministic linear time. Grandjean [1985] and Olive [1998] obtained interesting results related to those of Lynch. They gave logical representations of the class NLIN, that is, linear time on random access machines, in terms of second-order logic with unary functions instead of relations (in their setting, also the input string is represented by a function).

Lautemann et al. [1995] recently proved that the class CFL of context-free languages is characterized by ESO formulas of the form $\exists B\varphi$ where φ is first-order, B is a binary predicate symbol, and the range of the second-order quantifier is restricted to the class of *matchings*, that is, pairing relations without crossover. Note that this is not a purely prefix-syntactic characterization of CFL. From our results and the fact that some languages which are not context-free can be expressed in the minimal nonregular ESO-prefix classes, it follows that a syntactic characterization of CFL by means of ESO-prefix classes is impossible.

Several restricted versions of REG where studied and logically characterized by restricted versions of ESO. McNaughton and Papert [1971] showed that first-order logic with a linear ordering precisely characterizes the *star-free* regular languages. This theorem was extended by Thomas [1996] to ω-languages, that is, languages of infinite words. Later several hierarchies of the star-free languages were studied and logically characterized (see, e.g., Thomas [1996], and Pin [1986; 1994; 1996]). Straubing et al. [1995] showed that first-order logic with modular counting quantifiers characterize the regular languages whose syntactic monoids contain only solvable groups. These and many other related results can be found in the books and surveys (see, e.g., Straubing [1994], Thomas [1996], and Pin [1986; 1994; 1996]).

1.8. STRUCTURE OF THE PAPER. The rest of this paper is organized as follows. Section 2 introduces basic concepts and notation. In Section 3, we show that the classes ESO($\forall\forall\forall$), ESO($\forall\forall\exists$), and ESO($\forall\exists\forall$) all express the canonical nonregular language $L = \{a^n b^n\}$. In Section 4, we derive a new combinatorial theorem on hypergraphs, which is a crucial tool for proving that ESO($\exists^*\forall\exists^*$) is regular. The latter result, which is the technically most demanding of this paper, is gradually established in Sections 5–8. In particular, Section 6 proves the result under the restriction to successor-free ESO($\forall\exists^*$) sentences; Section 7 generalizes it to ESO($\forall\exists^*$) sentences. The general result is then proved in Section 8. The regularity of the class ESO($\exists^*\forall\forall$) is shown in Section 9. For better readability, the proof in that section is given under simplifying assumptions; a full proof is in the appendix.

The problem of model checking is considered in Section 10, where we prove a dichotomy theorem for model checking. In Section 11, we determine the classes $\mathfrak{Q}$ for which finite satisfiability is decidable. Section 12 identifies those classes ESO($\mathfrak{Q}$) that capture REG and those that are closed under complementation. The final section (Section 13) addresses further research issues and concludes the paper.

2. *Preliminaries and Notation*

We consider second-order logic with equality (unless explicitly stated otherwise) and without function symbols of positive arity. Predicates are denoted by capitals and individual variables by lower case letters; a bold face version of a letter denotes a tuple of corresponding symbols.

A *prefix* is any string over the alphabet $\{\exists, \forall\}$, and a *prefix set* is any language $\mathfrak{Q} \subseteq \{\exists, \forall\}^*$ of prefixes. A prefix set $\mathfrak{Q}$ is *trivial*, if $\mathfrak{Q} = \emptyset$ or $\mathfrak{Q} = \{\lambda\}$, that is, it consists of the empty prefix. In the rest of this paper, we focus on nontrivial prefix sets.

A *generalized prefix* is any string over the extended prefix alphabet $\{\exists, \forall, \exists^*, \forall^*\}$. A prefix set $\mathfrak{Q}$ is *standard*, if either $\mathfrak{Q} = \{\exists, \forall\}^*$ or $\mathfrak{Q}$ can be given by some generalized prefix.

For any prefix Q, the class ESO(Q) is the set of all Σ^1_1 formulas $\exists \mathbf{R}\varphi$, where φ is a prenex first-order formula with prefix Q; for any prefix set $\mathfrak{Q}$, the class ESO($\mathfrak{Q}$) is the union ESO($\mathfrak{Q}$) $= \bigcup_{Q \in \mathfrak{Q}}$ ESO(Q).

For example, ESO($\exists^*\forall\exists^*$) is the class of all formulas $\exists\mathbf{R}\exists\mathbf{y}\forall x\exists\mathbf{z}\varphi$, where φ is quantifier-free; this is the class of ESO-prefix formulas, whose first-order part is in the well-known Ackermann class with equality.

Recall that a *literal* is an atomic formula or the negation of such; equalities and inequalities are also literals. It is usual, when one deals with conjunctive normal forms, to define a clause as a set of literals interpreted as the disjunction of its members. We deal with disjunctive normal forms. Accordingly, we redefine a clause as set of literals interpreted as the *conjunction* of its members. A DNF formula can be seen as a set of clauses. For any formula $\Phi \in$ ESO($\{\forall, \exists\}^*$) whose quantifier-free part is a DNF $\varphi = \bigvee_i \delta_i$, we denote by $\Delta(\Phi)$ (simply Δ, if Φ is understood) the set of all clauses δ_i of φ.

Let $A = \{a_1, \ldots, a_m\}$ be a finite alphabet. A *string* over A is a finite first-order structure $W = \langle U, C^W_{a_1}, \ldots, C^W_{a_m}, Succ^W, min^W, max^W \rangle$, for the vocabulary $\sigma_A = \{C_{a_1}, \ldots, C_{a_m}, Succ, min, max\}$, where

—U is a nonempty finite initial segment $\{1, 2, \ldots, n\}$ of the positive integers;

—each $C^W_{a_i}$ is a unary relation over U (i.e., a subset of U) for the unary predicate C_{a_i}, for $i = 1, \ldots, m$, such that the $C^W_{a_i}$ are pairwise disjoint and $\bigcup_i C^W_{a_i} = U$.

—$Succ^W$ is the usual successor relation on U, and min^W and max^W are the first and the last element in U, respectively.

We refer to the predicates C_{a_i} also as *colors*.

Observe that this representation of a string is a successor structure as discussed, for example, in Eiter et al. [1996]. An alternative representation uses the standard linear order $<$ on U instead of the successor *Succ*. In full ESO, $<$ is tantamount to *Succ* since either predicate can be defined in terms of the other. We come back to this issue in Section 13.

The strings W for A correspond to the nonempty finite words over A in the obvious way; in abuse of notation, we often use W in place of the corresponding word from A^* and vice versa.

A *SO sentence* Φ over the vocabulary σ_A is a second-order formula whose only free variables are the predicate variables of the signature σ_A, and in which no constant symbols except *min* and *max* occur. Such a sentence defines a language over A, denoted $\mathscr{L}(\Phi)$, given by $\mathscr{L}(\Phi) = \{ W | W \models \Phi\}$. We say that a language $L \subseteq A^*$ is *expressed* by Φ, if $\mathscr{L}(\Phi) = L \cap A^+$ (thus, for technical reasons, without loss of generality we disregard the empty string); L is *expressed by a set S* of sentences, if L is expressed by some $\Phi \in S$. We say that S *captures* a class C of languages, if S expresses all and only the languages in C.

Let A be a finite alphabet. A sentence Φ over σ_A is called *regular*, if $\mathscr{L}(\Phi)$ is a regular language. A set of sentences S (in particular, any ESO-prefix class) is *regular*, if for every finite alphabet A, all sentences $\Phi \in S$ over σ_A are regular.

Büchi [1960] has shown the following fundamental theorem. Let MSO denote the fragment of second order logic in which all predicate variables have arity at most one,[4] and let REG denote the class of regular languages.

PROPOSITION 2.1 (BÜCHI'S THEOREM). *MSO captures REG.*

Note that Büchi's Theorem was independently found by Trakhtenbrot [1961].

That MSO can express all regular languages is easy to see, since it is straightforward to describe the behavior of a finite state automaton by an existential MSO sentence. In fact, this is easily possible in monadic ESO($\forall\exists$) as well as in monadic ESO($\forall\forall$). Thus, we have the following lower expressiveness bound on ESO-prefix classes over strings.

PROPOSITION 2.2. *Let $\mathfrak{Q}$ be any prefix set. If $\mathfrak{Q} \cap \{\exists, \forall\}^*\forall\{\exists, \forall\}^+ \neq \emptyset$, then ESO($\mathfrak{Q}$) expresses all languages in REG.*

3. *Nonregular ESO-Prefix Classes*

In this section, we present some ESO-prefix classes that are not regular. In particular, we show that ESO($\forall\forall\forall$), ESO($\forall\forall\exists$), and ESO($\forall\exists\forall$) include a well-known nonregular language. This means that whenever we have in a prefix Q two universal FO quantifiers separated or followed by some other FO quantifier, then any class containing Q is nonregular. As it will appear later, these three prefix-classes are the minimal nonregular standard prefix-classes of ESO.

PROPOSITION 3.1. *The language $L = \{a^n b^n | n \geq 0\}$ can be expressed by sentences*

(i) $\exists R \forall x \forall y \exists z.\varphi_1$,

(ii) $\exists R \forall x \forall y \forall z.\varphi_2$, *and*

[4] Observe that we assume MSO allows one to use nullary predicate variables (i.e., propositional variables) along with unary predicate variables. Obviously, Büchi's Theorem survives.

(iii) $\exists R \forall x \exists y \forall z \cdot \varphi_3$,

where R *is binary and the* φ_i *are quantifier-free.*

PROOF. The language L is defined by the sentence

$$\Phi = \exists R \forall x \forall y \exists z . R(min, max) \wedge C_a(min) \wedge$$
$$[R(x, y) \rightarrow (\neg R(y, x) \wedge (C_a(x) \leftrightarrow C_b(y)))] \wedge$$
$$[(R(x, y) \wedge C_a(x)) \rightarrow (Succ(x, z) \wedge (z \neq y \rightarrow R(y, z)))] \wedge$$
$$[(R(x, y) \wedge C_b(x)) \rightarrow (Succ(z, x) \wedge R(y, z))].$$

The formula says that R is a directed graph such that an arc goes from *min* to *max*, that the first letter of the word is an a, that R is asymmetric and that arcs are between letters of different colors; the penultimate conjunct says that if an arc leads from an element e_a colored with C_a to an element e_b colored with C_b, then an arc must lead from e_b to the successor of e_a, unless the elements e_a and e_b are adjacent; the last conjunct states a similar condition for arcs from C_b's to C_a's.

To see that Φ works properly, consider first any word $W = a^n b^n$ in L. Then, $(W, R) \models \Phi$, where $R = \{(i, 2n - i + 1) | 1 \leq i \leq n\} \cup \{(2n - i + 1, i + 1) | 1 \leq i < n\}$. Conversely, suppose that $(W, R) \models \Phi$, where $W = c_1 c_2 \cdots c_m$ and each $c_i \in \{a, b\}$. By induction on $i \leq \lfloor m/2 \rfloor$, show that the following holds:

$$c_i = a,\ c_{m-i+1} = b,\ R(i, m - i + 1), \qquad \text{and} \qquad R(m - i + 1, i + 1).$$

Thus, if m is even, we have finished. Suppose then $m = 2n + 1$ for some n, so that the given word W is of the form $a^n c_{n+1} b^n$. In particular, we have $R(n, m - n + 1)$ and $R(n, n + 2)$. By Φ, $R(n, n + 2)$ implies $R(n + 2, n + 1)$, and $R(n + 2, n + 1) \wedge C_b(n + 2)$ implies $C_a(n + 1) \wedge R(n + 1, n + 1)$. However, $R(n + 1, n + 1)$ contradicts the asymmetry of R.

For (ii), we slightly modify the previous formula Φ by turning the existential quantifier $\exists z$ into a universal quantifier $\forall z$, and by replacing the last two conjuncts with

$$(R(x, y) \wedge C_a(x) \wedge Succ(x, z) \wedge z \neq y) \rightarrow R(y, z)$$

and

$$(R(x, y) \wedge C_b(x) \wedge Succ(z, x)) \rightarrow R(y, z),$$

respectively.

For (iii), observe that with first-order quantifier prefix $\forall \exists \forall$, it is easy to say that R describes a partitioning of the string in 2-element sets $\{e_1, e_2\}$, such that e_1 has color C_a and e_2 has color C_b:

$$\forall x \exists y \forall z [R(x, y) \wedge (C_a(x) \leftrightarrow C_b(y)) \wedge (R(x, z) \rightarrow R(z, x)) \wedge (R(x, z) \leftrightarrow z = y)].$$

Indeed, observe that R must be symmetric, and by the first and the last conjunct, we have $\forall x \exists ! y R(x, y)$. Moreover, it can be easily said with first-order prefix $\forall\exists\forall$ that a string W is of the form $a^n b^m$ (say that every C_b is followed by a C_b). $\square$

COROLLARY 3.2. *The ESO-prefix classes* $ESO(\forall\forall\exists)$, $ESO(\forall\forall\forall)$, *and* $ESO(\forall\exists\forall)$ *express some nonregular languages*.

In Section 10, we will derive by a more complicated proof even stronger results: the three ESO-prefix classes in Corollary 3.2 do not only express nonregular languages, they even express NP-complete languages.

Observe that the syntactic incomparability of ESO-prefix classes does not mean that their expressive capabilities over strings are incomparable. In particular, we show the following:

PROPOSITION 3.3. *Over strings*, $ESO(\forall\exists\forall)$ *reduces to* $ESO(\forall\forall\forall)$. *In other words*, *every language expressible in* $ESO(\forall\exists\forall)$ *is expressible in* $ESO(\forall\forall\forall)$.

PROOF. We begin with the following lemma.

LEMMA 3.4. *Let* P, Q *be* $(j + 1)$-*ary predicate symbols*, $j \geq 0$, *and let* $\mathbf{x}$ *be a* j-*tuple of individual variables. Then,* $\forall\mathbf{x}\exists y P(\mathbf{x}, y)$ *is over strings equivalent to* $\exists Q\forall\mathbf{x}\forall y\forall z.\alpha$ *for an appropriate quantifier-free formula* α.

PROOF OF LEMMA. Intuitively, $Q(\mathbf{x}, y)$ means that $P(\mathbf{x}, z)$ holds for some $z \leq y$. This can be expressed as follows:

$$\forall\mathbf{x}\forall y\forall z(Q(\mathbf{x}, min) \rightarrow P(\mathbf{x}, min)) \wedge (Succ(y, z) \rightarrow [Q(\mathbf{x}, z) \rightarrow (P(\mathbf{x}, z) \vee Q(\mathbf{x}, y))]);$$

we conjunct $Q(\mathbf{x}, max)$ to this and get the desired α. $\square$

To reduce ESO($\forall\exists\forall$) to ESO($\forall\forall\forall$), let

$$\Phi = \exists\mathbf{R}\forall x\exists y\forall z.\varphi,$$

where φ is quantifier-free. Seconder-order skolemization gives an equivalent

$$\Phi' = \exists\mathbf{R}\exists F[\forall x\exists y F(x, y) \wedge \forall x\forall y\forall z(F(x, y) \rightarrow \varphi(x, y, z))].$$

Now use the lemma and then convert the resulting formula into the prenex form. $\square$

(Note that this proposition can not be applied in the proof of Proposition 3.1, since it introduces additional predicate variables in the formula.)

4. *A Combinatorial Theorem on Hypergraphs*

In this section, we prove a result on hypergraphs. This result may be of independent interest and will be used in Sections 6 and 8.

We introduce the concept of $[e]$-transversal of a (directed) hypergraph, which is a key concept in the proofs of the main results in Sections 6 and 8. For understanding those proofs, it is necessary to be acquainted with the definitions of the present section.

An *r-uniform directed hypergraph* $H = (N, E)$ consists of an r-ary relation E over a finite set N, that is, $E \subseteq N^r$. (Note that other authors use the term *directed hypergraph*, for a different concept.) N is called the set of nodes and E the set of (hyper)edges. Whenever we use the term *hypergraph* in this paper, we actually mean uniform directed hypergraph. Directed (finite) graphs are a special case given by $r = 2$.

We denote by $Pos(a, e)$ the set of all *positions* at which a occurs in edge e. Let, for instance $e = \langle 5, 6, 4, 5\rangle$, then $Pos(5, e) = \{1, 4\}$, $Pos(6, e) = \{2\}$, $Pos(8, e) = \emptyset$, and so on.

By abuse of notation, we often write $|H|$ instead of $|E|$ for a hypergraph $H = (N, E)$. Moreover, for two hypergraphs $H = (N, E)$ and $H' = (N', E')$, we write $H \subseteq H'$ iff $N \subseteq N'$ and $E \subseteq E'$.

Let $H = (N, E)$ a hypergraph. The *degree* $deg_H(a)$ of a node $a \in N$ is the number of edges of H in which a occurs. This notion generalizes to sets W of vertices: If $W \subseteq N$, then the *degree* $deg_H(W)$ of set W is the number of edges $e \in E$ that *meet* W, that is, that have at least one component in W.

A *transversal* of a hypergraph $H = (N, E)$ is a set $T \subseteq N$ such that T meets all $e \in E$. A transversal is *minimal*, if it is of minimal cardinality.

Let us now introduce a more sophisticated concept of transversal, the *excluded edge transversal*.

Definition 4.1. Let $H = (N, E)$ be a hypergraph and $e \in E$ an edge. Then, an $[e]$-transversal of H is a subset T of N such that each edge $e' \in E - \{e\}$ has at least one component b such that $b \in T$ and $Pos(b, e') \not\subseteq Pos(b, e)$, that is, b occurs in e' at least at one position where it does not occur in e.

Note that each $[e]$-transversal of H is a transversal of $(N, H - \{e\})$, but not necessarily vice-versa. Furthermore, if T is an $[e]$-transversal of H, then every U such that $T \subseteq U \subseteq N$ is an $[e]$-transversal of H.

An $[e]$-transversal is *minimal*, if it contains a minimal number of elements. We denote by $\tau_e(H)$ the cardinality of a minimal $[e]$-transversal of H.

Example 4.1. Let $H = (N, E)$ with $N = \{1, 2, 3, 4, 5\}$ and $E = \{e_1, \ldots, e_6\}$:

e_1:	1	1	2	4	4
e_2:	1	1	1	4	4
e_3:	5	4	3	2	1
e_4:	2	2	2	3	3
e_5:	1	2	2	2	2
e_6:	4	4	2	4	4

Then $T = \{1, 2, 4\}$ is an $[e_1]$-transversal of H, but $T' = \{2, 4\}$ and $T'' = \{1, 2\}$ are not, even though T' and T'' are both transversals of $(N, E - \{e_1\})$. Note that T is even a minimal $[e_1]$-transversal of H. This can be seen as follows: Every $[e_1]$-transversal of H must contain 1 to meet e_2 correctly and 4 to meet e_6 correctly. In order to meet e_4 it must contain either 2 or 3. Therefore, it must contain at least three elements. T has three elements and is thus minimal. We have $\tau_{e_1}(H) = 3$.

Definition 4.2. Let $H = (N, E)$ be a hypergraph. Then $t(H)$ is defined by

$$t(H) = \frac{\sum_{e \in E} \tau_e(H)}{|H|}.$$

Thus, $t(H)$ is the *average* minimal $[e]$-transversal size of the hypergraph H. The goal of this section is to show that the asymptotic growth of $t(H)$ is superlinear in $\log |H|$.

Let us first state two simple lemmas.

LEMMA 4.1. *Let $H' \subseteq H$ and let e be an edge of H'. Then $\tau_e(H') \leq \tau_e(H)$.*

PROOF. It suffices to note that every $[e]$-transversal of H necessarily contains an $[e]$-transversal of H'. □

LEMMA 4.2. *Let $H = (N, E)$ and $H' = (N', E')$ be two hypergraphs. If $H' \subseteq H$, then $t(H') \cdot |H'| \leq t(H) \cdot |H|$.*

PROOF. We have

$$t(H') \cdot |H'| = \sum_{e \in E'} \tau_e(H') \leq \sum_{e \in E'} \tau_e(H) \leq \sum_{e \in E} \tau_e(H) = t(H) \cdot |H|.$$

Here, the first inequality follows from Lemma 4.1, as $H' \subseteq H$. □

We are now ready for the main result of this section.

THEOREM 4.3. *For every positive integer r, there is a monotone polynomial p such that every r-uniform hypergraph H satisfies $|H| < p(t(H))$.*

PROOF. We prove the statement by induction on r.

(Induction Base) If $r = 1$, then $|H| = t(H) + 1$, and the statement trivially holds.

(Induction Step) Suppose that p witnesses the claim for r, and let $H = (N, E)$ be an $r + 1$-uniform hypergraph. Without loss of generality, we assume that $|H| > 1$; by definition of $t(H)$, we then have $t(H) \geq 1$.

Since $t(H)$ is an average over all $\tau_e(H)$, there must exist an edge $e \in E$ such that $\tau_e(H) \leq t(H)$.

Let T be a minimal $[e]$-transversal of H. Then, $|T| \leq t(H)$. Since T meets all edges of H except possibly e, it follows that $deg_H(T) \geq |H| - 1 \geq |H|/2$.

Consequently, there exists an element $a \in T$ such that a occurs in at least $|H|/(2|T|)$ edges of H, that is, $deg_H(a) \geq |H|/(2|T|)$. Let A be the set of all edges of H containing a. We have $|A| \geq |H|/(2|T|)$.

The element a may occur in different positions in the edges in A. However, since there are only $r + 1$ positions, there must exist a position i such that a occurs in at least $|A|/(r + 1) \geq |H|/(2|T|(r + 1))$ elements of A at position i. Let $B \subseteq A$ be the set of all edges in A containing a in the ith position. Let $c := 2(r + 1)$. Note that c is a constant depending only on r and that $|B| \geq |H|/(c|T|) \geq |H|/(c \cdot t(H))$.

Let $H' = (N, B)$. Clearly, $H' \subseteq H$. Retain that

$$|H| \leq c \cdot |H'| \cdot t(H). \tag{1}$$

Define $G := (N, B_a)$ where

$$B_a = \{\langle a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{r+1}\rangle | \langle a_1, \ldots, a_{r+1}\rangle \in B\};$$

that is, B_a is obtained from B by dropping the ith column (which uniformly contains a in B).

Note that $|G| = |H'|$ and, as easily seen, $t(H') = t(G)$.

Since G is r-uniform, applying the assumption on p yields

$$|H'| = |G| < p(t(G)) = p(t(H')). \tag{2}$$

Furthermore,

$$\begin{aligned} |H| \cdot t(H') &\leq c \cdot |H'| \cdot t(H) \cdot t(H') && \text{by (1)} \\ &\leq c \cdot |H| \cdot t(H) \cdot t(H) && \text{by Lemma 4.2} \end{aligned}$$

so that

$$t(H') \leq c \cdot t(H) \cdot t(H). \tag{3}$$

Finally

$$\begin{aligned} |H| &\leq (c \cdot t(H)) \cdot |H'| && \text{by (1)} \\ &\leq (c \cdot t(H)) \cdot p(t(H')) && \text{by (2)} \\ &\leq (c \cdot t(H)) \cdot p(c \cdot t(H) \cdot t(H)) && \text{by (3) and the monotonicity of } p. \end{aligned}$$

It follows that the statement holds for $r + 1$, which concludes the induction and the proof of the theorem. □

Since any polynomial $p(n)$ is asymptotically dominated by 2^n and $t(H)$ is larger than some number d_r for all sufficiently large $|H|$, we obtain from the previous theorem the following result.

COROLLARY 4.4. *For each positive integer r, there exists a constant* $c_r \geq 0$ *such that for any r-uniform hypergraph H with* $|H| \geq c_r$, *the inequality* $t(H) > log|H|$ *holds*.

5. *Supports and Normal Forms*

In this section, we introduce some machinery which is used in the subsequent sections. In Section 5.1, we introduce concepts and notation. In Section 5.2, we prove some normal form theorems, which will be convenient in proofs.

5.1. SELECTORS, WITNESS FUNCTIONS, SUPPORTS, AND CONFLICTS. Recall that, in this paper, a term is an individual variable or an individual constant. Call a clause $\varphi(x_1, \ldots, x_n)$ *complete* (or a *complete type*) with respect to a given vocabulary and a set of terms if it is a maximal syntactically consistent conjunction of literals in the given vocabulary with variables in $\{x_1, \ldots, x_n\}$.

Definition 5.1 (*NF*1). Let A be an alphabet. An ESO($\forall\exists^*$) sentence Φ has the *normal form 1*, in short NF1, if its quantifier-free part is a disjunction $\varphi = \bigvee\Delta(\Phi)$ of clauses $\Delta(\Phi)$ such that each $\delta \in \Delta(\Phi)$ is complete for the vocabulary of Φ and for a set of individual variables (not necessarily all individual variables in Φ).

Definition 5.2. Let Φ be an NF1 sentence $\exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r.\varphi$ with quantifier-free part φ, and let W be a string with universe $U = U(W)$. A *support* for (W, Φ) is a pair $S = \langle \sigma, f \rangle$, where σ and f are as follows:

—σ is a map: $U \rightarrow \Delta(\Phi)$. Any such map is called a *selector function* for (W, Φ).
—$f = \langle f[1], \ldots, f[r] \rangle$ is an r-tuple of functions $f[i]: U \rightarrow U$. Any such r-tuple is called a *witness function* for (W, Φ).

Given a support $V = \langle \sigma, f \rangle$ for (W, Φ), we introduce the following notation:

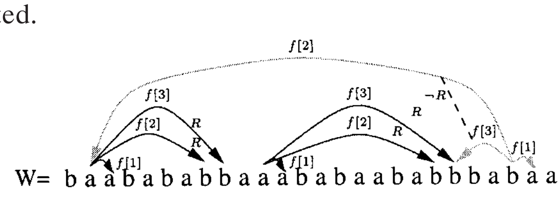
—For any $a \in U$, the *witness tuple* for a is the tuple $f(a)$, and *witset*(a, V) is the set $\{f[i](a) | 1 \leq i \leq r\}$ of *witness elements* for a.
—For any $A \subseteq U$, *wit*(A, V) is the set $\{f(a) | a \in A\}$ of the witness tuples for all elements in A, and *wit*(V) is the set *wit*(U, V) of all witness tuples according to V.
—For any $\delta \in \Delta(\Phi)$, *culp*(δ, V) is the set $\sigma^{-1}(\delta) = \{a \in U | \sigma(a) = \delta\}$ of elements ("culprits") assigned to δ in V, and *wit*(δ, V) is the set *wit*(*culp*$(\delta, V), V)$ of the witness tuples for all culprits for δ.
—For any $a \in U$, *lit*(a, V) is the collection of ground literals which results from replacing $x, y_1, \ldots, y_r$ with $a, f[1](a), \ldots, f[r](a)$ in $\sigma(a)$. If $A \subseteq U$, then *lit*$(A, V) = \bigcup\{$*lit*$(a, V) : a \in A\}$, *lit*$(\delta, V) =$ *lit*(*culp*$(\delta, V), V)$, and *lit*$(V) =$ *lit*(U, V).
—For any $a \in U$, a *free witness literal of a* is a literal in *lit*(a, V) that does not contain a. By *freelit*(a, V) we denote the set of free witness literals of a. If $A \subseteq U$, then *freelit*$(A, V) = \bigcup_{a \in A}$ *freelit*(a, V) and *freelit*$(V) =$ *freelit*(U, V).
—Finally, *lit*(W) is the set of all ground literals true in W.

Example 5.1. Consider a formula $\Phi = \exists R \forall x \exists y_1 \exists y_2 \exists y_3.(\varphi_1 \vee \varphi_2 \vee \cdots)$, where

$$\varphi_1 = C_a(x) \wedge C_a(y_1) \wedge C_b(y_2) \wedge C_b(y_3) \wedge Succ(x, y_1) \wedge Succ(y_2, y_3) \wedge R(x, y_2) \wedge R(x, y_3) \cdots$$

$$\varphi_2 = C_b(x) \wedge C_a(y_1) \wedge C_a(y_2) \wedge C_b(y_3) \wedge Succ(x, y_1) \wedge \neg R(y_2, y_3) \cdots$$

Let W be the string depicted below, and let $V = \langle \sigma, f \rangle$ be a support for W and Φ, such that $\sigma(2) = \varphi_1$, $\sigma(11) = \varphi_1$, and $\sigma(24) = \varphi_2$, and where f is defined on 2, 11, and 24 as depicted.



Then,

lit$(2, V) = \{C_a(2), C_a(3), C_b(8), C_b(9), Succ(2, 3), Succ(8, 9), R(2, 8), R(2, 9), \ldots\}$,

$lit(11, V) = \{C_a(11), C_a(12), C_b(20), C_b(21), Succ(11, 12), Succ(20, 21), R(11, 20), R(11, 21), \ldots\},$
$lit(24, V) = \{C_b(24), C_a(25), C_a(2), C_b(21), Succ(24, 25), \neg R(2, 21), \ldots\}.$

Hence, $lit(V) = \{R(2, 8), R(2, 9), R(11, 20), R(11, 21), \neg R(2, 21), \ldots\}$. The set $lit(W)$ contains the literals $C_b(1)$, $\neg C_a(1)$, $C_a(2)$, $\neg C_b(2), \ldots,$ $C_a(26)$, $\neg C_b(26)$, $Succ(1, 2)$, $Succ(2, 3)$, $\neg Succ(1, 3)$, etc; the literal $\neg R(2, 21)$ belongs to $freelit(V)$. □

A support V for (Φ, W) is called *locally consistent* if, for each $a \in U$, the union $lit(a, V) \cup lit(W)$ is consistent. A support V is called *consistent*, if the union $lit(V) \cup lit(W)$ is consistent; otherwise V is *inconsistent*.

The following lemma is obvious.

LEMMA 5.1. *$W \models \Phi$ iff there exists a consistent support for Φ and W.*

Elements a, b of W *conflict* over a support V if the set $lit(a, V) \cup lit(b, V) \cup lit(W)$ contains some atom L together with its negation; such L is a *conflict* induced by a and b over V. We denote by $conf(a, b, V)$ the set of all conflicts induced by a and b over V. The following lemma is obvious:

LEMMA 5.2. *Let V be a support for (W, Φ). Then V is inconsistent if and only if there exists a pair $a, b \in U$ of elements conflicting over V, and V is locally inconsistent if and only if there exists an element $a \in U$ such that a conflicts with itself over V.*

5.2. FURTHER NORMAL FORMS

Definition 5.3 (*NF*2). An NF1 sentence $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \varphi$ is in *normal form* 2 (NF2), if and only if it fulfills the following conditions:

(a) If distinct variables u, v occur in a clause, then the clause contains the inequality literal $u \neq v$.
(b) For each clause δ, there exists a monadic predicate symbol P in $\mathbf{R}$ such that δ contains the literal $P(x)$ and the literals $\neg P(y_1), \ldots, \neg P(y_r)$.

LEMMA 5.3. *Let $G = (V, E)$ be a finite directed graph with a bound r on the out-degree $deg^+_G(v) = |\{v' \in V : (v, v') \in E\}|$ of each node v. Then G is $2r + 1$-colorable.*

PROOF. Induction on the number n of nodes. The case $n = 1$ is obvious. Suppose that the lemma has been proved for $n - 1$. Since r is the bound on the out-degree, the average in-degree is $\leq r$ and therefore the average (total) degree is $\leq 2r$. Hence, there is a node v of degree $\leq 2r$. By the induction hypothesis, the rest of the graph can be $(2r + 1)$-colored. The neighbors of v use at most $2r$ colors; an unused color can be used to color v. □

A simple compactness argument shows that the lemma holds also for infinite graphs, but we will not use that result here.

THEOREM 5.4. *Every NF1 sentence Φ can be transformed into an equivalent (over strings) NF2 sentence Φ^*.*

PROOF. Let Φ be as in the definition of NF1. To satisfy requirement (a), eliminate an equality $\chi = \xi$ or $\xi = \chi$, where χ precedes ξ in the canonical ordering of variables, remove that literal and substitute χ for ξ everywhere else in the clause.

To satisfy requirement (b), consider any string $W \models \Phi$ and any support V for (Φ, W). Create a directed graph G on $U(W)$ by linking every culprit x to each of its witness elements. Obviously, the out-degrees are bounded by r; by Lemma 5.3, G is $2r + 1$-colorable. Since this is true for *any* support V, the $(2r + 1)$-colorability follows from Φ itself. Introduce $2r + 1$ new monadic predicates $P_1, \ldots, P_{2r+1}$ and replace each clause $\delta \in \Phi$ by a collection of new clauses augmenting δ with all possible "colorings" of all individual variables in δ by means of the predicates P_i subject to the following restriction: x has a color different from the colors of all other variables. (Coloring of a variable χ with color predicate P_i means asserting $P_i(\chi)$ and $\neg P_j(\chi)$ for all $j \neq i$.) It is easy to see that the resulting sentence Φ^* is equivalent to Φ. □

Definition 5.4 (*NF*3). An NF2 sentence $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \varphi$ is in *normal form* 3 (NF3) if each clause contains all individual variables $x, y_1, \ldots, y_r$.

THEOREM 5.5. *For every sentence $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \varphi$, there exists an NF3 sentence Φ^* equivalent to Φ on strings of length at least $r + 1$.*

PROOF. The goal is achieved in three steps.

Step (1). If a clause δ of Φ does not contain all individual variables, augment it with literals $\chi \neq \xi$ where χ ranges over the variables in Φ and ξ ranges over the variables missing in δ. Let Φ_1 be the result. Clearly, Φ_1 satisfies the condition (a) from the definition of NF2.

Step (2). Use the second part of the proof of Theorem 5.4 to transform Φ_1 into Φ_2 that is in NF2.

Step (3). If a clause of Φ_2 is not complete with respect to the vocabulary of Φ, replace it with an equivalent disjunction of complete clauses. □

6. *Successor-free ESO*($\exists^*\forall\exists^*$) *Is Regular*

In this and the following two sections, we prove that ESO($\exists^*\forall\exists^*$) is regular. The proof is rather involved. In this section, we prove the regularity of the fragment of ESO($\forall\exists^*$) that does not use *Succ* or *min* or *max*. In Section 7, we prove the regularity of the fragment ESO($\forall\exists^*$) that does not use *min* or *max*. Finally, in Section 8, we prove the regularity of ESO($\exists^*\forall\exists^*$).

In Section 6.1, we introduce the semantic concept of *boundedness* and show that a sentence is regular if it is bounded. The main result of this section is then established in Section 6.2.

6.1. BOUNDED SENTENCES.

Let K be a positive integer. An NF1 sentence Φ is *K-bounded* if, for each $W \models \Phi$, there exists a consistent support V for (W, Φ) with $|wit(V)| \leq K$, so that the total number of witness tuples is $\leq K$. The sentence Φ is *bounded* if it is K-bounded for some K.

THEOREM 6.1. *If a successor-free NF*1 *sentence Φ* (*in which min*, *max do not occur*) *is bounded*, *then Φ is regular*.

PROOF. Suppose that an NF1 sentence Φ is K-bounded. We show that Φ is equivalent to a monadic second-order sentence, and thus, by Büchi's Theorem, Φ is regular.

Let Φ be of the form

$$\exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \bigvee_{\gamma \in \Delta} \gamma,$$

where $\mathbf{R}$ is a list of predicate variables and the γ are the clauses of $\Delta = \Delta(\Phi)$.

Note that whenever (W, Φ) has a consistent support, then there is also a consistent support V for (W, Φ) such that at most $K \cdot r$ elements of $U(W)$ appear in all witness tuples. Let $Z = \{z_1, \ldots, z_{K \cdot r}\}$ be a set of $K \cdot r$ fresh individual variables. Transform Φ to the following equivalent second-order sentence Φ':

$$\exists z_1 \cdots \exists z_{K \cdot r} \exists \mathbf{R} \forall x \bigvee_{\gamma \in \Delta} \bigvee_{\zeta_1, \ldots, \zeta_r \in Z} \gamma[y_1/\zeta_1, \ldots, y_r/\zeta_r],$$

where y_i/ζ_i means substitution of ζ_i for y_i. Let Δ' be the set of all clauses in Φ'. Now it is not hard to see how all predicates of arity >1 can be eliminated from Φ'. For notational simplicity, we assume that $\mathbf{R}$ consists of one binary predicate R. Note that each R-literal occurring in Φ' has all its arguments among $\{x\} \cup Z$. Replace each atom $R(\xi, \chi)$ by a new unary atom $R_{\xi,\chi}(x)$ or nullary atom $R_{\xi,\chi}$. That is, replace $R(z_i, z_j)$ by R_{z_i,z_j}, $R(z_i, x)$ by $R_{z_i,x}(x)$, $R(x, z_i)$ by $R_{x,z_i}(x)$, and $R(x, x)$ by $R_{x,x}(x)$. A clause $\gamma \in \Delta'$ yields a clause γ^*. Let $\mathbf{R}^*$ be the list of new monadic predicate symbols $R_{\xi,\chi}$ corresponding to the new atoms. Formula Φ' is then equivalent to

$$\exists z_1 \cdots \exists z_{K \cdot r} \exists \mathbf{R}^* \forall x \left(\psi \wedge \bigvee_{\gamma \in \Delta'} \gamma^* \right),$$

where $\forall x \psi$ asserts that the new predicates are properly correlated. The formula ψ is a conjunction of formulas like

$$z_2 = z_3 \rightarrow (R_{x,z_2}(x) \leftrightarrow R_{x,z_3}(x))$$

or

$$x = z_2 \rightarrow (R_{z_1,x}(x) \leftrightarrow R_{z_1,z_2}).$$

This final formula is monadic. $\square$

6.2. SUCCESSOR-FREE NF1 SENTENCES ARE REGULAR. In this section, we prove that every successor-free NF1 sentence Φ is regular. The crux of the proof is roughly described as follows: It is sufficient to show that Φ is bounded. To prove this, we take a string W such that $W \models \Phi$ and a support V for Φ and W having a minimal extent (i.e, assigning—in a precise sense—a minimal number of witness tuples). Since the number of witness tuples (for short, witnesses) in V is minimal, it is not possible to lump different witnesses together (i.e, we cannot replace the witness of one element by the witness of another element and thus decrease the number of witnesses). Since it is impossible to lump witnesses together, something must be responsible for this impossibility. Namely, for each element a and each witness w of another element different from its own witness w_0, there must exist at least one literal L induced by V that blocks the possibility of using w as witness for a instead of w_0. Such literals L are called *blockers*. For example, in Example 5.1, the literal $\neg R(2, 21)$ is a blocker, because it blocks the reuse of the witnesses of position 11 for position 2 in the string W.

We compute a lower bound on the number of necessary blockers. We note that the set of blockers for a particular element a having w_0 as witness corresponds to a $[w_0]$-transversal of the hypergraph H formed by all witnesses. By Corollary 4.4, the size of this transversal is on average at least $\log|H|$. Moreover, it will be shown that the sets of blockers corresponding to different elements are disjoint. Therefore, $|H| \cdot \log|H|$ is a lower bound on the total number of blockers necessary to ensure the minimality of V. However, any support V can induce only a linear number of literals (and thus blockers). By comparing the linear upper bound with the $|H| \cdot \log|H|$ lower bound, we conclude that the number of witnesses is constant; as a consequence, formula Φ is bounded.

THEOREM 6.2. *Every successor-free NF*1 *sentence* (*in which min*, *max do not occur*) *is regular*.

PROOF. Let $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r . \varphi$ be a successor-free NF1 sentence. By Theorem 5.5, we assume without loss of generality that Φ is in NF3.

We denote by ℓ the number of literals appearing in a clause of Φ; note that since Φ is in NF3, all clauses of φ have the same number of literals. For convenience, we write Δ as shorthand for $\Delta(\Phi)$.

We show that there exists a constant K such that if $W \models \varphi$, then there exists a consistent support V for Φ and W such that $|wit(V)| \leq K$. Thus Φ is bounded and by Theorem 6.1 regular.

Assume thus that $W \models \Phi$. Then let $V = \langle \sigma, f \rangle$ be a consistent support for Φ and W such that the sum

$$\sum_{\gamma \in \Delta} |wit(\gamma, V)|$$

is minimal over all consistent supports, that is, for no other consistent support V' of Φ and W this sum is smaller.

Let $\delta \in \Delta$ be a clause of φ, let $X = culp(\delta, V)$, that is, the set of elements in $U(W)$ that are mapped by σ to the clause δ, and let $q = |wit(\delta, V)|$.

Since Φ is in NF3, it is also in NF2 and thus it fulfills part (b) of Definition 5.3, and $\sigma(b) \neq \sigma(a)$ holds for all elements $a \in X$ and $b \in witset(a, V)$. Since all elements of X have the same image δ under σ, the following property is true:

Fact 6.3. $\forall a \forall b \in X, b \notin witset(a, V)$.

We define an equivalence relation $\sim$ on X as follows:

$$\forall u \forall v \in X : u \sim v \textit{ iff } f(u) = f(v).$$

Note that there are exactly $q = |wit(\delta, V)|$ equivalence classes in $X/\sim$. Denote by $Z_1, \ldots, Z_q$ the different equivalence classes of $X/\sim$.

For $a, b \in X$ we denote by $V[a \to \mathcal{W}(b)]$ the support obtained from V by assigning to a the witnesses of b instead of its own witnesses. Formally, $V[a \to$

$\mathcal{W}(b)] = \langle \sigma, f' \rangle$ where $f'(a) = f(b)$ and $f'(v) = f(v)$ for each $v \in U(W) \backslash \{a\}$. Observe that literals in $lit(V') \backslash lit(V)$ where $V' = V[a \to \mathcal{W}(b)]$ must involve a and apart from a only elements in $witset(b, V)$.

CLAIM 6.4. *$\forall Z \in X/\sim \; \exists a \in Z \; \forall b \in X \backslash Z$: $V[a \to \mathcal{W}(b)]$ is inconsistent.*

PROOF OF CLAIM. Assume the claim does not hold. Then there exist a set $Z \in X/\sim$ and a function $h : Z \to X \backslash Z$ such that for each $a \in Z$, $V[a \to \mathcal{W}(h(a))]$ is consistent. Let $Z = \{c_1, \ldots, c_k\}$. We show that the support $V^* = \langle \sigma, f^* \rangle$, defined by

$$V^* = V[c_1 \to \mathcal{W}(h(c_1))][c_2 \to \mathcal{W}(h(c_2))] \cdots [c_k \to \mathcal{W}(h(c_k))],$$

is a consistent support for Φ and W.

To prove this, suppose V^* is inconsistent. This inconsistency must be caused by two conflicting elements $a, b \in Z$. Indeed, let $\Lambda = lit(U(W) \backslash Z, V)$; then, $lit(V^*) = \Lambda \cup \bigcup_{a \in Z} lit(a, V^*)$, where $lit(a, V^*) = lit(h(a), V)[h(a)/a]$ is the set of literals $lit(h(a), V)$ in which $h(a)$ is uniformly replaced by a. Since $lit(V^*) \cup lit(W)$ is inconsistent but, by choice of h, $lit(V[a \to \mathcal{W}(h(a))]) \cup lit(W)$ and thus its subset $\Lambda \cup lit(a, V^*) \cup lit(W)$ is consistent, it follows that for some $a, b \in Z$, $lit(a, V^*) \cup lit(b, V^*) \cup lit(W)$ is not consistent, that is, $conf(a, b, V^*) \neq \emptyset$. Moreover, $conf(a, b, V^*)$ must contain two opposite literals L and $L' = \neg L$ that involve at least one of a, b. In fact, if both L, L' would involve neither a nor b, then $L, L' \in lit(h(a), V) \cup lit(h(b), V) \cup lit(W) \subseteq lit(V) \cup lit(W)$, which is a contradiction. Assume without loss of generality that both L and L' involve a. Then $L, L' \notin lit(b, V^*)$, because by Fact 6.3, a cannot occur in $f^*(b) = f(h(b))$ and thus a does not occur in $lit(b, V^*)$. Hence $L, L' \in lit(a, V^*) \cup lit(W)$, and thus $lit(a, V^*) \cup lit(W)$ is inconsistent. Since $lit(a, V^*) \cup lit(W) \subseteq lit(V[a \to \mathcal{W}(h(a))]) \cup lit(W)$, this implies that $V[a \to \mathcal{W}(h(a))]$ is inconsistent. However, $V[a \to \mathcal{W}(h(a))]$ is asserted to be consistent; it follows that V^* is consistent.

But by the definition of $\sim$ and V^*, we have $|wit(\delta, V^*)| = |wit(\delta, V)| - 1$ and thus $\Sigma_{\gamma \in \Delta} |wit(\gamma, V^*)| < \Sigma_{\gamma \in \Delta} |wit(\gamma, V)|$, which contradicts our minimality assumption on V. The claim is proved. □

Choose for each Z_i an element $a_i \in Z_i$ such that for each $b \in X \backslash Z_i$, the support $V[a_i \to \mathcal{W}(b)]$ is inconsistent. Let $w_i := f(a_i)$, for each $1 \leq i \leq q$, and let $V_{i,j} := V[a_i \to \mathcal{W}(a_j)]$. Note that by Fact 6.3, a_i does not occur in w_j, for each $1 \leq i, j \leq q$.

A conflict in $V_{i,j}$ must involve a literal $L'_{i,j}$ in

$$lit(V_{i,j}) \backslash lit(V) \subseteq lit(a_i, V_{i,j}).$$

Hence, the literal $L_{i,j}$ which is the opposite of $L'_{i,j}$ is in $lit(V)$. Since $L'_{i,j}$ is a new literal, it involves a_i. It is easy to see that $L'_{i,j}$ cannot be unary or an equality (otherwise, it would belong to $lit(V)$). Since $L'_{i,j} \notin lit(a_i, V)$, it follows that some b occurs in w_j in a position where it does not occur in w_i. Notice also that both $L'_{i,j}$ and $L_{i,j}$ contain apart from a_i only elements from w_j.

Fix such a literal $L_{i,j}$ and an element b as described for a_i and w_j and refer to them as $blocker(a_i, w_j)$ and $dart(a_i, w_j)$, respectively.

Let, for each $1 \leq i \leq q$,

$$B(a_i) = \{blocker(a_i, w_j) | 1 \leq j \leq q, j \neq i\}, \text{ and}$$
$$D(a_i) = \{dart(a_i, w_j) | 1 \leq j \leq q, j \neq i\}.$$

(Note that $D(a_i) \cap witset(a_j, V) \neq \emptyset$, for all $1 \leq i \neq j \leq q$.) For $1 \leq i \leq q$, we have

$$r \cdot |B(a_i)| \geq |D(a_i)|;$$

indeed, every element $e \in D(a_i)$ must occur in at least one $blocker(a_i, w_j) \in B(a_i)$, and at most r elements $e \in D(a_i)$ can occur in a single $blocker(a_i, w_j)$. Moreover, from Fact 6.3 and taking into account the constants occurring in $B(a_i)$, it follows that

$$B(a_i) \cap B(a_j) = \emptyset, \qquad \text{for all} \quad 1 \leq i \neq j \leq q.$$

Consequently,

$$\left| \bigcup_{i=1}^{q} B(a_i) \right| = \sum_{i=1}^{q} |B(a_i)| \geq \frac{1}{r} \sum_{i=1}^{q} |D(a_i)|.$$

Let H be the hypergraph $H = (U(W), \{w_i | 1 \leq i \leq q\})$. Then, it is easy to see that for $1 \leq i \leq q$, $D(a_i)$ is a $[w_i]$-transversal of H. Hence, we have $|D(a_i)| \geq \tau_{w_i}(H)$ and thus

$$\begin{aligned}\left| \bigcup_{i=1}^{q} B(a_i) \right| &= \sum_{i=1}^{q} |B(a_i)| \geq \frac{1}{r} \sum_{i=1}^{q} |D(a_i)| \\ &\geq \frac{1}{r} \sum_{i=1}^{q} \tau_{w_i}(H) = \frac{t(H) \cdot |H|}{r} = \frac{t(H) \cdot q}{r}.\end{aligned}$$

Denote by $vert(H)$ the set of all elements $b \in U(W)$ occurring in some edge of H and let $lit(H, V) = \bigcup_{b \in vert(H)} lit(b, V)$.

We show the following:

—For all $1 \leq i, j \leq q$ such that $i \neq j$, $blocker(a_i, w_j)$ belongs to $lit(H, V)$ or $freelit(V)$.

To verify this, first observe from the properties of $L_{i,j}$ that $blocker(a_i, w_j) \in lit(U(W) - \{a_i\}, V)$. It follows that if $blocker(a_i, w_j) \notin freelit(V)$, then $blocker(a_i, w_j) \in lit(b, V)$ for some $b \in U(W) \backslash \{a_i\}$ which occurs in w_j. This means $blocker(a_i, w_j) \in lit(H, V)$, however.

Now let us determine a lower bound for the number of blockers $blocker(a_i, w_j)$ which are in $freelit(V)$.

First observe that

$$|lit(H, V)| \leq \ell \cdot q \cdot r.$$

(Recall that ℓ is the number of literals in a clause.)

It follows that

$$\left| \bigcup_{i=1}^{q} B(a_i) \ \cap \ \mathit{freelit}(V) \right| \geq \frac{t(H) \cdot q}{r} - \ell \cdot q \cdot r;$$

thus,

$$\frac{t(H) \cdot q}{r} - \ell \cdot q \cdot r \leq |\mathit{freelit}(V)|.$$

Assume now without loss of generality that the $\delta \in \Delta$ we have chosen is such that $q = |\mathit{wit}(\delta, V)|$ is maximal over all clauses $\delta \in \Delta$. Then,

$$|\mathit{freelit}(V)| \leq |\Delta| \cdot q \cdot \ell.$$

From these bounds on $|\mathit{freelit}(V)|$, we derive that for maximal q, we have

$$\frac{t(H) \cdot q}{r} - \ell \cdot q \cdot r \leq |\Delta| \cdot q \cdot \ell,$$

whence

$$t(H) - \ell \cdot r^2 \leq |\Delta| \cdot \ell \cdot r.$$

If $q > c_r$, then $|H| > c_r$, and since H is r-uniform, by Corollary 4.4, we have $t(H) > \log|H| = \log q$. From this and the previous inequation we obtain $(\log q) - \ell \cdot r^2 \leq |\Delta| \cdot \ell \cdot r$, and thus $\log q \leq \ell \cdot r \cdot (r + |\Delta|)$, from which it follows that

$$q \leq 2^{\ell \cdot r \cdot (r+|\Delta|)}.$$

Let $k_0 = \mathit{max}\{c_r, 2^{\ell \cdot r \cdot (r+|\Delta|)}\}$. Then $q \leq k_0$.

Now let $K = k_0 \cdot |\Delta|$. Note that K is a constant that depends only on the formula Φ.

Since q is maximal, we have

$$|\mathit{wit}(V)| \leq \sum_{\gamma \in \Delta} |\mathit{wit}(\gamma, V)| \leq |\Delta| \cdot q \leq k_0 \cdot \Delta = K,$$

which means that $|\mathit{wit}(V)| \leq K$. This shows that Φ is bounded. □

7. *ESO*($\forall\exists^*$) *Is Regular*

In this section, we show how Theorem 6.2 can be lifted to the case where the successor predicate *Succ* is present, but *min* and *max* may not occur; we will deal with the constants *min* and *max* in the next section.

7.1. Local and Remote Witnesses. In this subsection, we extend the basic concepts and definitions from Section 5.1.

Let $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \varphi$ be a sentence (with *Succ*) in NF3. In the presence of successor, the witnesses $y_1, \ldots, y_r$ of a clause $\delta \in \Delta(\Phi)$ are split into two parts: the y_i which are (directly or transitively) connected to x via *Succ*, which we call the *local witnesses* of x, and all other y_i, which we call the *remote witnesses* of x.

Accordingly, if we have a consistent support $V = \langle \sigma, f \rangle$ for Φ and a string W, then $f(a)$, $witset(a, V)$, and $wit(a, V)$ for any $a \in U(W)$ are split into a local and a remote part, which is designated by index l and r, respectively.

More precisely, we have:

—$f_l(a)$ and $f_r(a)$ are the projections of $f(a)$ to the components which hold the local (respectively, remote) witness elements of a according to $\sigma(a)$; in particular, $f_r(a) = f(a)$ if δ has no local witnesses.

—$witset_l(a, V)$ (respectively, $witset_r(a, V)$) is the set of all elements occurring in $f_l(a)$ (respectively, $f_r(a)$).

—$wit_l(A, V) = \{ f_l(a) | a \in A\}$ and $wit_r(A, V) = \{ f_r(a) | a \in A\}$, for any $A \subseteq U(W)$.

—$wit_l(\delta, V) = wit_l(culp(\delta), V)$ and $wit_r(\delta, V) = wit_r(culp(\delta), V)$, for every $\delta \in \Delta(\Phi)$;

—$freelit_r(a, V)$ is the set of all literals $L \in lit(a, V)$ such that only elements from $witset_r(a, V)$ occur in L. In other words, neither a nor any local witness of a occurs in L.

Notice that $lit(W)$ now also contains all interpreted literals for *Succ* which are true in W. The definition of local consistency and consistency of a support V remains unchanged. We observe that the Lemmas 5.1 and 5.2 hold in the generalized setting.

7.2. Fourth Normal Form. For lifting the proof that ESO($\forall\exists^*$) sentences are regular to the case with successor, we have to deal with local and remote witness elements in an appropriate way.

Notice that all local witness elements are uniquely determined by the culprit. Therefore, only remote witness elements can be reused. As we will show, whenever $W \models \Phi$, then a consistent support V for Φ and W exists which uses only a constant number of remote witness elements (where the constant depends only on the formula Φ).

In order to deal with local witness elements, it is convenient to introduce a further normal form.

Definition 7.1. (*NF*4) An ESO($\forall\exists^*$) sentence Φ in NF3 has the normal form 4 (NF4), if it satisfies the following condition:

(c) For each clause $\delta \in \Delta(\Phi)$ and each local witness variable y_i, there exists a monadic predicate symbol Q in $\mathbf{R}$ such that $Q(y_i)$ is a conjunct of δ while for each variable y_j, $1 \leq j \neq i \leq r$ occurring in δ, $\neg Q(y_j)$ is a conjunct of δ.

Informally, condition (c) in the NF4 allows us to avoid locality conflicts when reusing witnesses; it prevents that local witness elements of $a \in culp(\delta, V)$ are witness elements of any $b \in culp(\delta, V)$ different from a.

Theorem 7.1. *For every NF*1 *sentence* $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r . \varphi$, *where* φ *is quantifier-free, there exists a sentence* Φ^* *in NF4 such that for each string* W *of length at least* $r + 1$, $W \models \Phi$ *iff* $W \models \Phi^*$.

PROOF. This can be established by following the proof of Theorem 5.5, by doing a little more coloring in the formula in the proof of Theorem 5.4. In the reduction to the graph coloring there, we add for each node b in the graph G on each occurrence of b as a local witness element $r - 1$ extra edges, directed from b to all witness fellows for this occurrence; that is, if $b \in witset_l(a, V)$, then we add arcs to every node $b' \in witset(a, V) \setminus \{b\}$.

Since an element b may only occur a bounded number of times as a local witness ($\leq c \cdot r$ for some constant c), the graph G that we construct this way has out-degrees bounded by $c' \cdot r^2$ for some constant c', and hence G is $(2c' \cdot r^2 + 1)$-colorable for some constant c'. Proceed then by asserting the coloring in the usual way. □

We now have the following lemma, which generalizes Fact 6.3 in the proof of Theorem 6.2.

LEMMA 7.2. *Let $X = culp(\delta, V)$ for some consistent support V for a string W and $\delta \in \Delta(\Phi)$ where Φ is a formula in NF4. Then, the following holds*:

$$\forall a \forall b \in X : a \notin witset(b, V) \wedge [a \neq b \Rightarrow witset_l(a, V) \cap witset(b, V) = \emptyset].$$

PROOF. From condition (b) of NF4 (a and b have the same color, and every element in $witset(b, V)$ has a color different from b's color), it follows $a \notin witset(b, V)$. Suppose that for some $a, b \in X$ an element $c \in witset_l(a, V) \cap witset(b, V)$ exists. Suppose c is colored red. By the coloring technique (condition (c) of NF4), all local witness elements of a have a different color, and all remote witness elements have a color different from all local witness colors. Since $b \in X$ and $c \in witset(b, V)$, c must be for b at the same local witness position as for a. But this means that $b = a$, a contradiction to the assumption. This proves the lemma. □

7.3. BOUNDED SENTENCES WITH SUCCESSOR. The concept of boundedness is generalized in the following way. An NF1 sentence Φ (with *Succ*) is *bounded*, if there exists a constant K such that for each W modeling Φ, there exists a consistent support V for Φ and W such that $|wit_r(V)| \leq K$, that is, the number of remote witness parts in V is at most K.

Theorem 6.1 is adapted as follows:

THEOREM 7.3. *Every NF1 sentence Φ (in which min and max do not occur) which is bounded is regular.*

PROOF. The proof is analogous to the proof of Theorem 6.1, but slightly more involved.

Let as in the proof of Theorem 6.1 be $Z = \{z_1, \ldots, z_k\}$ the fresh variables for the remote witness elements. Transform Φ to Φ':

$$\exists z_1 \cdots \exists z_k \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \bigvee_{\gamma \in \Delta(\Phi)} \bigvee_{\zeta_1, \ldots, \zeta_r \in Z} \gamma[\vartheta_1, \ldots, \vartheta_r],$$

where ϑ_i stands for y_i/ζ_i, if y_i is a remote witness variable for γ, and for y_i/y_i (i.e., replace y_i by itself) if y_i is a local witness variable for γ. Let us assume as in the proof of Theorem 6.1 that $\mathbf{R}$ consists of one binary predicate R (the general case is similar).

The replacement of binary atoms $R(\xi, \chi)$ by new monadic atoms $R_{\xi,\chi}(x)$ is done in a similar way, but for each local witness y_i from ξ, χ, we must record in $R_{\xi,\chi}$ the local position of y_i to x with respect to *Succ*; this can be handled if we assume without loss of generality that $y_1, y_3, \ldots, y_{2i+1}, \ldots$ (odd index) is the first, second, ith successor of x and $y_2, y_4, \ldots, y_{2i}, \ldots$ (even index) is the first, second, ith element before x.

For example, $R_{y_2,z_1}(x)$ then tells that the predecessor of x is in relation R to element z_1, and $R_{y_3,x}(x)$ that the second element after x is in relation R to x.

The formula ψ ensures the compatibility of the monadic predicates that we have introduced, and again consists of a conjunction of FO formulas for each pair of new predicates $R_{\xi,\chi}$ and $R_{\xi',\chi'}$.

For the previous two predicates, that formula would look like:

$$\forall x \forall y_1 \cdots \forall y_r \forall u \forall v_1 \cdots \forall v_r([Succ(y_2, x) \wedge Succ(u, v_1) \wedge Succ(v_1, v_3) \\ \wedge (y_2 = v_3) \wedge (z_1 = x)] \rightarrow [R_{y_2,z_1}(x) \leftrightarrow R_{y_3,x}(u)]);$$

(Variables $u, v_1, \ldots, v_r$ replace $x, y_1, \ldots, y_r$ for accessing the $R_{y_3,x}$.)

Repeating this process gives again a monadic formula Φ^*, and proves the theorem. $\square$

7.4. NF1 Sentences with Successor Are Regular. We now prove the generalization of Theorem 6.2 to the setting where successor literals occur in the formula. The proof is similar to the proof of Theorem 6.2 but somewhat more involved. The main differences caused by the successor literals are informally described as follows:

—For each fixed selection function σ, disjunct δ, and element a with $\sigma(a) = \delta$, the local witnesses of any support $V = \langle \sigma, f \rangle$ are unambiguously determined by δ. In other words, for any two consistent supports $V = \langle \sigma, f \rangle$ and $V' = \langle \sigma, f' \rangle$, it must hold that $f_l(a) = f'_l(a)$. If V is a consistent support, then, for different elements a and b, where $\sigma(a) = \sigma(b)$ and a, b have local witnesses, any support $V[a \rightarrow \mathcal{W}(b)]$ is inconsistent because of a local witness mismatch. Note that such an inconsistency is a *local inconsistency* and does by no ways imply the existence of blockers as in the successor-free case. It thus makes no sense to consider supports of type $V[a \rightarrow \mathcal{W}(b)]$. Instead, we will consider supports $V[a \rightarrow \mathcal{W}_r(b)]$, where the remote part $f_r(a)$ of a is replaced by the remote part $f_r(b)$ of b, while the local part of a remains unchanged. The aim is then to show that—for a minimal support—the inconsistency of all supports $V[a \rightarrow \mathcal{W}_r(b)]$ requires the existence of a certain number of blockers.

—In the proof of Theorem 6.2, we observed the following property of any support $V' = V[a \rightarrow \mathcal{W}(b)]$: Every literal in $lit(V')$ which was not already in $lit(V)$ must involve a and apart from a only elements in $witset(b, V)$. In analogy to this, we show the following property of any support $V' = V[a \rightarrow \mathcal{W}_r(b)]$ in the current setting with successors: Every literal in $lit(V')$ not contained in $lit(V)$ must involve some element from $witset_l(a, V) \cup \{a\}$ and also some element occurring in $f_r(b)$.

—Recall Fact 6.3 from the proof of Theorem 6.2 ($\forall a, b \in culp(\delta, V), b \notin witset(a, V)$). This fact was very useful to exclude certain types of conflicts, which may lead to inconsistent supports but do not imply the (desired) existence of blocking literals in *freelit*(V). In the successor case, such undesired conflicts may not involve the element a, but some element in $f_l(a)$, the local witness environment of a. For this reason, we will use a slightly stronger version of Fact 6.3, namely:

Fact 7.5. $\forall a, b \in culp(\delta, V)$,

$$a \notin witset(b, V) \wedge [a \neq b \Rightarrow witset_l(a, V) \ \cap \ witset(b, V) = \emptyset].$$

Fact 7.5 states that none of the local witnesses of a interferes with any of the witnesses of an element b that receives its witnesses by the same disjunct δ of φ.

—In analogy to the proof of Theorem 6.2, the main goal is to show that the inconsistency of $V[a \rightarrow \mathcal{W}_r(b)]$ for a large number of pairs a, b forces a large number of blockers to appear in *freelit*$_r(V)$. In presence of successor, for some pairs a, b, the inconsistency of $V[a \rightarrow \mathcal{W}_r(b)]$ still may be due to conflicting *Succ* literals and *not* to a blocker. However, by exploiting Fact 7.5, we are able to prove that this happens only in a limited number of cases, while still many pairs a, b effectively require a blocker.

—As in the proof of Theorem 6.2, not all blockers *blocker*(a_i, w_j) are good blockers in the sense that they belong to *freelit*$_r(V)$. In the successor-free case, we could show that the number of such bad blockers is small and that most blockers are good. Here we will show a similar result. However, due to the presence of local witnesses, the proof is slightly different.

Apart from these differences, the proof of the theorem below follows closely the proof of Theorem 6.2.

THEOREM 7.4. *Every NF*1 *sentence with possible occurrences of Succ is regular.*

PROOF. The proof is similar to the proof of Theorem 6.2, but now we have to deal with possible inconsistencies due to the successor.

Let $\Phi = \exists \mathbf{R} \forall x \exists y_1 \cdots \exists y_r \varphi$ be an NF1 sentence. By Theorem 7.1, we assume without loss of generality that Φ is in NF4.

We show that there exists a constant K such that if $W \models \Phi$, then there exists a consistent support V for Φ and W such that $|wit_r(V)| \leq K$. Thus, Φ is bounded and by Theorem 7.3 regular.

Assume thus that $W \models \Phi$. Then let $V = \langle \sigma, f \rangle$ be a consistent support for Φ and W such that the sum

$$\sum_{\gamma \in \Delta} |wit_r(\gamma, V)|$$

is minimal over all consistent supports, that is, for no other consistent support V' of Φ and W is this sum smaller.

Let $\delta \in \Delta$ be a clause of φ, let $X = culp(\delta, V)$, that is, the set of elements in $U(W)$ that are mapped in V by σ to the clause δ, and let $q = |wit_r(\delta, V)|$.

By Lemma 7.2, we have the following:

Fact 7.5. $\forall a \forall b \in X$:

$$a \notin witset(b, V) \wedge [a \neq b \Rightarrow witset_l(a, V) \ \cap \ witset(b, V) = \emptyset].$$

We define an equivalence relation $\sim$ on X as follows:

$$\forall a \forall b \in X : a \sim b \textit{ iff } \quad f_r(a) = f_r(b),$$

that is, the elements a and b that have the same remote witness parts are in the same class. Note that there are exactly $q = |wit_r(\delta, V)|$ equivalence classes in $X/\sim$. Denote by $Z_1, \ldots, Z_q$ the different equivalence classes of $X/\sim$.

For $a, b \in X$ we denote by $V[a \rightarrow \mathcal{W}_r(b)]$ the support obtained from V by assigning to a the remote witnesses of b instead of its own remote witnesses. Formally, if $V = \langle \sigma, f \rangle$, then $V[a \rightarrow \mathcal{W}_r(b)] = \langle \sigma, f' \rangle$ such that $f'(v) = f(v)$, for every $v \in U(W) \backslash \{a\}$, and $f'(a)$ is the (unique) tuple such that $f'_l(a) = f_l(a)$ and $f'_r(a) = f_r(b)$.

Compared to V, any new literal L in $lit(V[a \rightarrow \mathcal{W}_r(b)])$ must involve elements from $witset_l(a, V) \cup \{a\}$ and also elements from $f_r(b)$, that is, $witset_r(b, V)$. To see this, note that by the definition of $V[a \rightarrow \mathcal{W}_r(b)] =: V'$, clearly all new literals must be from $lit(a, V')$; since $lit(a, V')$ results from $lit(b, V)$ by substituting a and $wit_l(a, V)$ for b and $wit_l(b, V)$, respectively, and since, by Fact 7.5, neither a nor any local witness of a is a witness of b, any literal $L \in lit(a, V')$ which does neither contain a nor any element of $witset_l(a, V)$ belongs to $lit(b, V)$, and thus L is not new. Hence, all new literals of V' with respect to V must contain an element of $witset_l(a) \cup \{a\}$, and apart from a also some element in $witset_r(b, V)$.

Claim 7.6. $\forall Z \in X/\sim \; \exists a \in Z \; \forall b \in X \backslash Z : V[a \rightarrow \mathcal{W}_r(b)]$ *is inconsistent*.

Proof of Claim. Assume the claim does not hold. Then there exist a set $Z \in X/\sim$ and a function h: $Z \rightarrow X \backslash Z$ such that for each $a \in Z$, $V[a \rightarrow \mathcal{W}_r(h(a))]$ is consistent. Let $Z = \{c_1, \ldots, c_k\}$. We show that the support $V^* = \langle \sigma, f^* \rangle$, defined by

$$V^* = V[c_1 \rightarrow \mathcal{W}_r(h(c_1))][c_2 \rightarrow \mathcal{W}_r(h(c_2))] \cdots [c_k \rightarrow \mathcal{W}_r(h(c_k))],$$

is a consistent support for Φ and W.

To prove this, suppose V^* is inconsistent. This inconsistency must be caused by two conflicting elements $a, b \in Z$. Indeed, let $\Lambda = lit(U(W) \backslash Z, V)$; then, $lit(V^*) = \Lambda \cup \bigcup_{e \in Z} lit(e, V^*)$, where $lit(a, V^*) = lit(h(a), V)[h(a)/a]$ is the set of literals $lit(h(a), V)$ in which $h(a)$ is uniformly replaced by a. Since $lit(V^*) \cup lit(W)$ is inconsistent but, by choice of h, $lit(V[a \rightarrow \mathcal{W}_r(h(a))]) \cup lit(W)$ and thus its subset $\Lambda \cup lit(a, V^*) \cup lit(W)$ is consistent, it follows that for some $a, b \in Z$, $lit(a, V^*) \cup lit(b, V^*) \cup lit(W)$ is not consistent, that is, $conf(a, b, V^*) \neq \emptyset$.

Let $\{L, \neg L\} \in conf(a, b, V^*)$ be a conflict of a and b. Since every $V[c_i \rightarrow \mathcal{W}_r(h(c_i))]$ is locally consistent, also V^* is locally consistent, and thus we conclude that $a \neq b$ and $L, \neg L \notin lit(W)$.

Moreover, both L and $L' = \neg L$ must contain an element from either $S_a = witset_l(a, V) \cup \{a\}$ or from $S_b = witset_l(b, V) \cup \{b\}$; if they would contain neither, then $L, L' \in lit(V)$ would hold, which contradicts the consistency of V. Suppose then that L, L' contain an element from S_a. Then, $L, L' \notin lit(b, V^*)$, since, by Fact 7.5, no element from S_a can be a witness element of b. This is a contradiction. An analogous contradiction arises for assuming that L, L' contain an element from S_b. It follows that V^* is consistent.

But, by the definition of $\sim$ and V^*, we have $|wit(\delta, V^*)| = |wit(\delta, V)| - 1$ and thus $\Sigma_{\gamma \in \Delta} |wit(\gamma, V^*)| < \Sigma_{\gamma \in \Delta} |wit(\gamma, V)|$, which contradicts our minimality assumption on V. The claim is proved. $\square$

Choose for each Z_i an element $a_i \in Z_i$ such that for each $b \in X \backslash Z_i$, $V[a_i \to \mathcal{W}_r(b)]$ is inconsistent.

Note that, due to Fact 7.5, for every $1 \leq i \neq j \leq q$, no element from $witset_l(a_i, V) \cup \{a_i\}$ occurs in $f(a_j)$, that is, in $witset(a_j, V)$.

The inconsistency of $V[a_i \to \mathcal{W}_r(a_j)]$ implies that there is a pair of opposite literals $L_{i,j}$, $L'_{i,j} = \neg L_{i,j} \in lit(V[a_i \to \mathcal{W}_r(a_j)]) \cup lit(W)$. Since Φ has NF4, it follows that either

(i) the predicate symbol of $L_{i,j}$ is *Succ*, or

(ii) the predicate symbol of $L_{i,j}$ is some predicate P of arity > 1 different from *Succ* and equality.

Indeed, all monadic literals in $lit(V[a_i \to \mathcal{W}_r(a_j)])$ occur in $lit(V)$, and by Fact 7.5 a_i and its local witnesses are disjoint from a_j and its witnesses, which makes an equality conflict impossible.

Fact 7.5 tells us that for a particular j, only for a small number of i an $L_{i,j}$ as in (i) is possible. Namely, an $L_{i,j}$ as in (i) implies that the contiguous segment *Seg* constituted by $witset_l(a_i, V) \cup \{a_i\}$ is in $lit(a_i, V[a_i \to \mathcal{W}_r(a_j)])$ adjacent to some maximal contiguous segment *Seg′* constituted by elements in $witset_r(a_j, V)$; note that an overlap between *Seg* and *Seg′* is not possible. Since the number of maximal contiguous segments in $witset_r(a_j, V)$ is clearly bounded by r, for at most $2r$ elements a_i statement (i) applies. In the other cases, (ii) must be true.

Suppose that (ii) is true. Then, the following properties of $L_{i,j}$, $L'_{i,j}$ can be derived:

—both $L_{i,j}$ and $L'_{i,j}$ involve some element from $S_{a_i} = witset_l(a_i) \cup \{a_i\}$. For, otherwise $L_{i,j}$ and $L'_{i,j}$ would already have existed in $lit(V) \cup lit(W)$. Similarly,

—at least one of $L_{i,j}$, $L'_{i,j}$ must be from $lit(a_i, V[a_i \to \mathcal{W}_r(a_j)])$.

—$L_{i,j}$, $L'_{i,j} \notin lit(W)$. Immediate.

—not both $L_{i,j}$ and $L'_{i,j}$ appear in $lit(a_i, V[a_i \to \mathcal{W}_r(a_j)])$. This follows from Fact 7.5 and the local consistency of V.

Assume then without loss of generality that $L_{i,j} \notin lit(a_i, V[a_i \to \mathcal{W}_r(a_j)])$ and $L'_{i,j} \in lit(a_i, V[a_i \to \mathcal{W}_r(a_j)])$. Since $L_{i,j} \in lit(V[a_i \to \mathcal{W}_r(a_j)]) \backslash lit(W)$, it follows that $L_{i,j} \in lit(U(W) \backslash \{a_i\}, V)$ and thus $L_{i,j} \in lit(V)$. On the other hand, $L'_{i,j} \notin lit(V)$, for otherwise V would be inconsistent.

Let $w_j = f_r(a_j)$, for $j = 1, \ldots, q$. The literals $L_{i,j}$ and $L'_{i,j}$ have the following properties.

—$L_{i,j}$ and $L'_{i,j}$ must contain some element b from $witset_r(a_j, V)$ from some position of w_j at which in w_i no b is present. To see this, assume the contrary. Then since $\sigma(a_i) = \sigma(a_j) = \delta$, $L'_{i,j} \in lit(a_i, V)$ follows and thus $L_{i,j}$, $L'_{i,j} \in lit(V)$. Contradiction. (Observe that from NF4 of Φ and condition (a) of Definition 5.3, each element $v \in U(W)$ can occur at most once in any w_k, and hence $|Pos(v, w_k)| \leq 1$.)

—$L_{i,j}$, $L'_{i,j}$ contain apart from elements in S_{a_i} only elements from $witset_r(a_j)$. (Indeed, $L'_{i,j} = \neg L_{i,j}$ is in $lit(a_i, V[a_i \rightarrow \mathcal{W}_r(a_j)])$.

Call a pair a_i, a_j *Succ-consistent*, if $V[a_i \rightarrow \mathcal{W}_r(a_j)]$ does not have any inconsistency $L_{i,j}$, $L'_{i,j}$ of type (i).

For each $1 \le i, j \le q$ such that a_i, a_j are Succ-consistent, fix a literal $L_{i,j}$ and element b as described and refer to them as $blocker(a_i, w_j)$ and $dart(a_i, w_j)$, respectively.

Let, for each $1 \le i \le q$,

$$B(a_i) = \{blocker(a_i, w_j) | 1 \le j \ne i \le q,\ a_i, a_j \text{ are Succ-consistent}\}, \text{ and}$$

$$D(a_i) = \{dart(a_i, w_j) | 1 \le j \ne i \le q,\ a_i, a_j \text{ are Succ-consistent}\}.$$

Notice that $blocker(a_i, w_j)$ contains only elements from $witset_l(a_i, V) \cup \{a_i\}$ and from w_j (i.e., $witset_r(a_j)$). Moreover, $blocker(a_i, w_j)$ contains $dart(a_i, w_j)$ and some element from $witset_l(a_i, V) \cup \{a_i\}$.

For $1 \le i \le q$, we have

$$r \cdot |B(a_i)| \ge |D(a_i)|;$$

indeed, every element $e \in D(a_i)$ must occur in at least one $blocker(a_i, w_j) \in B(a_i)$, and at most r elements $e \in D(a_i)$ can occur in a single $blocker(a_i, w_j)$. Moreover, from Fact 7.5 and the elements occurring in $B(a_i)$, it follows that

$$B(a_i) \cap B(a_j) = \emptyset, \qquad \text{for all} \quad 1 \le i \ne j \le q.$$

Consequently,

$$\left| \bigcup_{i=1}^{q} B(a_i) \right| = \sum_{i=1}^{q} |B(a_i)| \ge \frac{1}{r} \sum_{i=1}^{q} |D(a_i)|. \tag{4}$$

Let H be the hypergraph $H = (U(W), \{w_i | 1 \le i \le q\})$, and $H_i \subseteq H$ the hypergraph $H_i = \{w_j | a_i, a_j \text{ are } Succ\text{-consistent}\}$, for $1 \le i \le q$.

It is easily verified that for every $1 \le i \le q$, the set $D(a_i)$ is a $[w_i]$-transversal of H_i.

Every $[w_i]$-transversal T of H_i can be extended to a $[w_i]$-transversal of H. (Indeed, consider $w_j \in H \backslash H_i$. Since $w_i = f_r(a_i) \ne f_r(a_j) = w_j$, there is a position where w_i and w_j differ; pick an element b_j in that position in w_j. By adding all b_j's to T, we obtain a $[w_i]$-transversal of H.) Thus, if we set $d_i = |H| - |H_i|$, we obtain the following inequation:

$$\tau_{w_i}(H_i) + d_i \ge \tau_{w_i}(H) \tag{5}$$

Indeed, we need to add at most d_i elements to a minimal $[w_i]$-transversal of H_i to obtain a $[w_i]$-transversal of H.

By the above considerations on the predicate in literal $L_{i,j}$, for every w_j there are at most $2r$ elements a_i such that a *Succ*-conflict between a_i and a_j is present in $V[a_i \to \mathcal{W}_r(a_j)]$. Hence, we obtain

$$\begin{aligned}\sum_{w_i \in H} |D(a_i)| &\geq \sum_{w_i \in H} \tau_{w_i}(H_i) \\ &\geq \sum_{w_i \in H} (\tau_{w_i}(H) - d_i) \\ &\geq \left(\sum_{w_i \in H} \tau_{w_i}(H) \right) - 2 \cdot q \cdot r \\ &= t(H) \cdot |H| - 2 \cdot q \cdot r.\end{aligned} \tag{6}$$

Thus, we obtain from (4) and (6) the following bound on the number of blocking literals:

$$\left| \bigcup_{i=1}^{q} B(a_i) \right| \geq \frac{t(H) \cdot q}{r} - 2 \cdot q. \tag{7}$$

We show the following:

—For every $1 \leq i \neq j \leq q$ such that a_i, a_j are *Succ*-consistent, either

(α) *blocker*$(a_i, w_j) \in$ *freelit*$_r(V)$, or
(β) there exist $b, b' \in U(W)$ such that b' occurs in *blocker*(a_i, w_j), $b' \in$ *witset*$_l(b) \cup \{b\}$, and *blocker*$(a_i, w_j) \in$ *lit*(b, V).

To verify this, recall that *blocker*$(a_i, w_j) \in$ *lit*(V); hence, if *blocker*$(a_i, w_j) \notin$ *freelit*$_r(V)$, then some $b \in U(W)$ must exist such that *blocker*$(a_i, w_j) \in$ *lit*$(b, V) \backslash$*freelit*$_r(V)$. Thus, some $b' \in$ *witset*$_l(b) \cup \{b\}$ exists such that b' occurs in *blocker*(a_i, w_j).

Let us check how many blockers can be covered by (β), for all $1 \leq i \neq j \leq q$. Denote by Λ the sets of all elements which occur in some *blocker*(a_i, w_j), where $1 \leq i \neq j \leq q$. Since *blocker*(a_i, w_j) contains only elements from *witset*$_l(a_i, V) \cup \{a_i\}$ and from w_j, that is, from *witset*$_r(a_j)$, we have that (by commuting the union terms):

$$\Lambda \subseteq \bigcup_{i \leq q} (\mathit{witset}_l(a_i) \cup \{a_i\} \cup \mathit{witset}_r(a_i)) = \bigcup_{i \leq q} (\mathit{witset}(a_i) \cup \{a_i\}).$$

Since $|\mathit{witset}(a_i)| \leq r$, it follows that $|\Lambda| \leq q(r+1)$.

Hence, there are at most $q(r+1)$ possible choices for b' as in (β). Fix b', and consider any possible b as in (β). Since $b' \in$ *witset*$_l(b) \cup \{b\}$, b and b' are close to each other; in particular, the distance between them cannot exceed r. Thus, for a fixed $b' \in \Lambda$, there are at most $2r + 1$ choices for a b such that $b' \in$ *witset*$_l(b) \cup \{b\}$ holds. It follows that there are at most $q(r+1)(2r+1)$ different b for all $b' \in \Lambda$. Now, since *blocker*(a_i, w_j) belongs to *lit*(b, V), and there are at most ℓ literals in *lit*(b, V), it follows that the overall number of literals *blocker*(a_i, w_j) of type (β) is at most $q(r+1)(2r+1)\ell$.

All other literals $blocker(a_i, w_j)$ must be covered by (α). It follows that

$$\left| \bigcup_{i=1}^{q} B(a_i) \cap freelit_r(V) \right| \geq \frac{t(H) \cdot q}{r} - 2 \cdot q - q(r+1)(2r+1) \cdot \ell.$$

Thus, we obtain the inequation

$$q \cdot \left(\frac{t(H)}{r} - c_0 \right) \leq |freelit_r(V)|, \tag{8}$$

where $c_0 = (2 + (r+1)(2r+1) \cdot \ell)$ depends only on Φ.

On the other hand, assume without loss of generality that the $\delta \in \Delta$ we have chosen is such that $q = |wit_r(\delta, V)|$ is maximal over all clauses $\delta \in \Delta$. Then,

$$|freelit_r(V)| \leq |\Delta| \cdot q \cdot \ell. \tag{9}$$

From (8) and (9), we derive that for maximal q, we have

$$\frac{t(H) \cdot q}{r} - c_0 \leq |\Delta| \cdot q \cdot \ell,$$

whence

$$t(H) - c_0 \cdot r \leq |\Delta| \cdot \ell \cdot r. \tag{10}$$

Observe that H is r'-uniform for some $r' \leq r$. If $q > c_{r'}$, then $|H| > c_{r'}$, and since H is r'-uniform, by Corollary 4.4, we have $t(H) > \log|H| = \log q$. From this and (10) we obtain $(\log q) - c_0 \cdot r \leq |\Delta| \cdot \ell \cdot r$, and thus $\log q \leq r \cdot (c_0 + |\Delta| \cdot \ell)$, from which it follows that

$$q \leq 2^{r \cdot (c_0 + |\Delta| \cdot \ell)}. \tag{11}$$

Let $k_0 = max\{c_r, 2^{r \cdot (c_0 + |\Delta| \cdot \ell)}\}$ and $K = k_0 \cdot |\Delta|$. Then, $q \leq k_0$, and both k_0 and K are constants only depending on formula Φ.

Since q is maximal, we have

$$|wit_r(V)| \leq \sum_{\gamma \in \Delta} |wit_r(\gamma, V)| \leq |\Delta| \cdot q \leq k_0 \cdot \Delta = K,$$

that is, $|wit_r(V)| \leq K$. This shows that Φ is bounded, which proves the theorem. $\square$

8. *ESO*($\exists^*\forall\exists^*$) *Is Regular*

The goal of this section is to finally show that the *full* class ESO($\exists^*\forall\exists^*$) is regular. To this aim, we first define some automata-theoretic concepts.

Definition 8.1. A *simple nondeterministic transducer* (*SNT*) is a tuple $T = (A, B, Q, E, q_{in}, Q_a)$, where

—A and B are finite alphabets, called the input alphabet and the output alphabet of T, respectively;

—Q is a finite set of states;

—$E \subseteq Q \times A \times B \times Q$ is a finite set of transitions, such that for each letter $a \in A$ and each state $q \in Q$, there exists some letter $b \in B$ and a state $q' \in Q$ such that the tuple $(q, a, b, q') \in E$.

—$q_{in} \in Q$ is the initial state.

—$Q_a \subseteq Q$ is the set of accepting states.

A *run* $\bar{\text{R}}$ of T on input string $W \in A^*$ is a sequence of length $|W|$ of tuples $t_i = (q_i, a_i, b_i, q_i') \in E$, $1 \leq i \leq |W|$, such that the following conditions are satisfied:

—$q_1 = q_{in}$,

—for $1 \leq i \leq |W|$, $a_i = W_i$ is the ith letter of W.

—for $1 \leq i \leq |W| - 1$, $q_{i+1} = q_i'$,

—$q'_{|W|} \in Q_a$.

The *output string* $\bar{\text{R}}(W)$ of such a run consists of the string $b_1 b_2 \cdots b_{|W|}$ in B^*.

The SNT T associates with any input string $W \in A^*$ a set of output strings $T(W) \subseteq B^*$, namely $T(W) = \{\bar{\text{R}}(W) | \bar{\text{R}}$ is a run of T on input $W\}$.

Definition 8.2. Let L and L' be languages over alphabets A and B, respectively. Then, L is *SNT reducible to* L', denoted by $L \preceq L'$, iff there exists an SNT T such that for each string $W \in A^*$, $W \in L$ iff $T(W) \cap L' \neq \emptyset$.

The following proposition is a simple special case of more general results on transductions that can be found, for example, in Eilenberg [1974], Berstel [1979] and Hopcroft and Ullman [1979] (in particular, see Hopcroft and Ullman [1979, Theorem 11.2, p. 276]).

PROPOSITION 8.1. *The class of regular languages is closed under SNT-reductions, that is, if* $L \preceq L'$ *and* L' *is regular, then also* L *is regular.*

THEOREM 8.2. *The class ESO*($\exists^*\forall\exists^*$) *is regular.*

PROOF. Let Φ be an ESO($\exists^*\forall\exists^*$) formula of the form $\exists \mathbf{P} \exists z_1 \cdots \forall z_k \forall x \exists y_1 \cdots \exists y_r . \varphi$ over an alphabet A, where φ is quantifier-free.

Denote $C = \{min, max, z_1, \ldots, z_k\}$ and $Terms = C \cup \{x, y_1, \ldots, y_r\}$. Without loss of generality, we assume that φ is a disjunction of complete types and that for all distinct $t, t' \in Terms$, each clause of φ asserts $t \neq t'$ and mentions all elements of *Terms* (cf. the proof of Theorem 5.4). Moreover, we assume without loss of generality that each clause δ of φ contains for each term $t \in Terms$ only one positive literal of the form $C_a(t)$, where C_a is in the signature (specifying the color of t). This literal is referred to as the *color qualification* of t in δ.

Let $L = \mathcal{L}(\Phi)$ be the language defined by Φ on A. Let $B = A \cup A \times C$ be an extension of the alphabet A. For any letter $e = (a, s) \in B$ such that $a \in A$ and $s \in C$, we will refer to s as the *label* of e.

Let T be an SNT with input alphabet A and output alphabet B operating as follows: For any string $W \in A^*$, T rejects W if $|W| < |Terms| = k + r + 3$; otherwise, T has all runs $\bar{\text{R}}$, which satisfy the following properties.

Denote the output for the letter $W_i \in A$ of W by $\bar{\text{R}}(W)_i$, $1 \leq i \leq |W|$, and denote the output for W by $\bar{\text{R}}(W)$.

—The first letter W_1 of W is transformed to $\bar{R}(W)_1 = (W_1, min)$, that is, the label *min* is attached to the first letter.
—The last letter $W_{|W|}$ of W is transformed to $\bar{R}(W)_{|W|} = (W_{|W|}, max)$, that is, the label *max* is attached to the last letter.
—There are exactly k distinct positions $1 < i_1, \ldots, i_k < |W|$ such that $\bar{R}(W)_{i_1} = (W_{i_1}, z_1)$, $\bar{R}(W)_{i_2} = (W_{i_2}, z_2), \ldots, \bar{R}(W)_{i_k} = (W_{i_k}, z_k)$. In other terms, exactly k positions different from the first and the last of W are labeled respectively with $z_1, \ldots, z_k$.
—For all positions $i \notin \{1, i_1, \ldots, i_k, |W|\}$, $\bar{R}(W)_i = W_i$.

The different nondeterministic runs of T produce the set $T(W)$ consisting of all output strings fulfilling the above conditions. It is obvious that such a transducer T exists.

Let Φ' be the NF1 sentence (with possible occurrence of *Succ*) over alphabet B and the corresponding signature of the form

$$\exists \mathbf{P} \forall x \exists y_1 \cdots \exists y_r \exists z_{min} \exists z_{max} \exists z_1 \cdots \exists z_k . \varphi',$$

where φ' is obtained from φ as follows:

(1) Eliminate the constant *min* by replacing *min* everywhere with the variable z_{min} and replace the color qualification $C_a(min)$ of *min* in each clause δ by $C_{(a,min)}(z_{min})$;
(2) Eliminate the constant *max* by replacing *max* everywhere with the variable z_{max} and replace the color qualification $C_a(max)$ of *max* in each clause δ by $C_{(a,max)}(z_{max})$;
(3) Adjust all color qualifications of the z_i variables as follows: For each clause δ, and each $1 \leq i \leq k$, replace the color qualification $C_a(z_i)$ in δ by $C_{(a,z_i)}(z_i)$;
(4) Transform the so obtained formula into a disjunction of complete types.

Let $L' = \mathcal{L}(\Phi')$ be the language defined by Φ'. Since Φ' is in NF1, L' is regular.

CLAIM 8.3. *For each $W \in A^*$, $W \in L$ iff $T(W) \cap L' \neq \emptyset$.*

PROOF OF CLAIM. Assume first $W \in L$. Then, $W \models \Phi$, and hence there exist relations $\mathbf{P}$ and elements $a_1, \ldots, a_k$ in W such that

$$(W, \mathbf{P}, a_1 \ldots, a_k) \models \forall x \exists y_1 \cdots \exists y_r \varphi(z_1, \ldots, z_k).$$

Let $\bar{R}$ be the run of T which for $1 \leq i \leq k$ outputs $\bar{R}(W)_{a_i} = (W_i, z_i)$. By the construction of Φ', $\bar{R}(W) \models \Phi'$, hence $\bar{R}(W) \in L'$. Since $\bar{R}(W) \in T(W)$, it follows that $T(W) \cap L' \neq \emptyset$.

Conversely, assume that $T(W) \cap L' \neq \emptyset$. Then there is a run $\bar{R}$ of T such that $\bar{R}(W) \in L'$, and hence there exist relations $\mathbf{P}$ on W such that

$$(\bar{R}(W), \mathbf{P}) \models \forall x \exists y_1 \cdots \exists y_r \exists z_{min} \exists z_{max} \exists z_1 \cdots \exists z_k . \varphi'.$$

Let $a_1, \ldots, a_k$ be the (unique) positions of $\bar{R}(W)$ marked $z_1, \ldots, z_k$, respectively. Moreover, let $a_{min} = 1$ and $a_{max} = |W|$. By the construction of Φ' and the assumption about color qualifications in the disjuncts $\Delta(\Phi)$, it is obvious that

$$(\bar{\mathrm{R}}(W), \mathbf{P}, a_{min}, a_{max}, a_1, \ldots, a_k) \vDash \forall x \exists y_1 \cdots \exists y_r.\varphi'(z_{min}, z_{max}, z_1, \ldots, z_k),$$

and thus

$$(\bar{\mathrm{R}}(W), a_{min}, a_{max}) \vDash \exists \mathbf{P} \exists z_1 \cdots \exists z_k \forall x \exists y_1 \cdots \exists y_r.\varphi'(z_{min}, z_{max}),$$

whence

$$W \vDash \exists \mathbf{P} \exists z_1 \cdots \exists z_k \forall x \exists y_1 \cdots \exists y_r.\varphi,$$

and hence $W \vDash \Phi$, which proves that $W \in L$. □

In summary, we have shown that $L \leq L'$ via T. Since L' is regular, by Proposition 8.1, so is L. □

COROLLARY 8.4. *Over strings, ESO($\exists^*\forall\exists^*$) is equivalent to MSO and existential MSO.*

As already mentioned in the Introduction (Section 1), equivalence of ESO($\exists^*\forall\exists^*$) and existential MSO does not hold on finite ordered graphs. Indeed the property of disconnectivity of a finite ordered graph is expressible in MSO, while it is not expressible in ESO($\exists^*\forall\exists^*$) [Eiler and Gottlob 1998]. On the other hand, both existential MSO and ESO($\exists^*\forall\exists^*$) can express NP-complete graph properties.

9. *ESO($\exists^*\forall\forall$) Is Regular*

In this section, we show that ESO($\exists^*\forall\forall$) is the second maximal standard ESO-prefix class that is regular. There are no further such classes, and thus we obtain a complete characterization of those ESO-prefix classes which are regular.

THEOREM 9.1. *Over strings, every ESO($\exists^*\forall\forall$) sentence is equivalent to an MSO sentence.*

Theorem 9.1 together with Büchi's Theorem (Proposition 2.1) implies the desired result.

COROLLARY 9.2. *The class ESO($\exists^*\forall\forall$) is regular.*

In the rest of this section, we prove Theorem 9.1 under simplifying assumptions. A compact full proof is given in the appendix.

Let Φ be a ESO($\exists^*\forall\forall$) sentence $\exists \mathbf{R} \exists \mathbf{y} \forall x_1 \forall x_2.\varphi$ where φ is quantifier-free.

LEMMA 9.3. *The sentence Φ is equivalent to a disjunction of ESO($\exists^*\forall\forall$) sentences $\exists \mathbf{R} \exists \mathbf{y} \forall x_1 \forall x_2 \varphi_i$ such that each φ_i fixes (that is uniquely determines) the quantifier-free type of* $\mathbf{y}$.

(A quantifier-free type of $\mathbf{y}$ is a complete type of $\mathbf{y}$ on $=$ and the relations of a string.)

PROOF. Let $\psi_1(\mathbf{y}), \ldots, \psi_m(\mathbf{y})$ be the quantifier-free types of $\mathbf{y}$ consistent with φ. The desired $\varphi_i = \psi_i \wedge \varphi$. □

In the rest of the proof, we assume, without loss of generality, that φ fixes the quantifier-free type of $\mathbf{y}$, and that *min* and *max* do not occur in φ (they can be defined using additional variables y_{min} and y_{max} in $\mathbf{y}$ and adding $\neg Succ(x_1, y_{min})$ and $\neg Succ(y_{max}, x_1)$ to φ).

Let $y_1, \ldots, y_n$ be the constituents of $\mathbf{y}$. Without loss of generality, φ implies that all $n + 2$ individual variables take different values. Indeed, if φ implies that two distinct constituents of $\mathbf{y}$ have the same values, then one of the constituents can be eliminated.

Without loss of generality, restrict attention to strings of length $\geq n + 2$. If $\psi(\mathbf{y}, x_1, x_2)$ is quantifier-free, let $\dot\forall x_1 x_2 \psi(x_1, x_2)$ assert that $\psi(\mathbf{y}, x_1, x_2)$ holds for all values x_1, x_2 such that

$$x_1 \neq x_2, \qquad \bigwedge_i x_1 \neq y_i, \qquad \text{and} \qquad \bigwedge_i x_2 \neq y_i.$$

Call this universal quantifier *strict*.

Lemma 9.4. *Φ is equivalent to a sentence of form $\exists \mathbf{R} \exists \mathbf{y} \dot\forall x_1 x_2 \psi$ where ψ is quantifier-free.*

Proof. Replace $\forall x_1 \forall x_2 \varphi(\mathbf{y}, x_1, x_2)$ with $\dot\forall x_1 x_2 \psi(\mathbf{y}, x_1, x_2)$, where ψ is the conjunction of formulas

$$\bigwedge_{i,j} \varphi(\mathbf{y}, y_i, y_j), \qquad \bigwedge_i \varphi(\mathbf{y}, x_1, y_i),$$
$$\bigwedge_i \varphi(\mathbf{y}, y_i, x_2), \qquad \text{and} \qquad \varphi(\mathbf{y}, x_1, x_2). \quad \square$$

We therefore will assume that the universal quantifier in Φ is strict.

We illustrate the rest of the proof on the example where $\mathbf{R}$ contains only unary predicates and one binary predicate E. Thus, $\Phi = \exists \mathbf{U} \exists E \exists \mathbf{y} \dot\forall x_1 x_2 \varphi_1$ where $\mathbf{U}$ is a tuple of unary predicates.

Lemma 9.5. *Without loss of generality, φ contains no E-atoms with at most one universal variable.*

Proof. We illustrate the proof on the example where $\mathbf{y}$ contains only one constituent y. In that case, there are seven possible E-atoms with at most one universal variable:

$$E(x_1, x_1), E(x_2, x_2), E(x_1, y), E(x_2, y), E(y, x_1), E(y, x_2), E(y, y).$$

Using fresh unary predicate variables $E^{*,*}$, $E^{*,y}$, $E^{y,*}$ and a nullary predicate variable $E^{y,y}$ replace the seven atoms in φ with atoms

$$E^{*,*}(x_1), E^{*,*}(x_2), E^{*,y}(x_1), E^{*,y}(x_2), E^{y,*}(x_1), E^{y,*}(x_2), E^{y,y},$$

respectively. Let ψ be the resulting formula and let

$$\Psi = \exists \mathbf{U} \exists E \exists \mathbf{P} \exists y \dot\forall x_1 x_2 \psi.$$

We check that Ψ is equivalent to Φ. First suppose that a string W models Φ and fix the values of all existential variables so that the expanded structure W^*

models $\dot\forall x_1 x_2 \psi$. Let b be the value of y. Set

$$E^{*,*}(x) := E(x, x)$$

$$E^{*,y}(x) := E(x, b)$$

$$E^{y,*}(x) := E(b, x)$$

$$E^{y,y} := E(b, b).$$

Clearly $W^* \models \exists P \exists Q \exists E \exists \mathbf{y} \dot\forall x_1 x_2 \psi$ and therefore $W \models \Psi$.

Secondly suppose that a string W models Ψ and fix the values of all existential variables so that the expanded structure W^* models $\dot\forall x_1 x_2 \psi$. Let b be the value of y. Notice that ψ says nothing about E on pairs (x, x), (b, x), (x, b). Redefine E on these pairs with respect to the following recipe:

$$E(x, x) := E^{*,*}(x)$$

$$E(x, b) := E^{*,y}(x)$$

$$E(b, x) := E^{y,*}(x)$$

$$E(b, b) := E^{y,y}.$$

The modified W^* satisfies φ. Hence, $W \models \Phi$. □

Remark 9.1. The mnemonic names $E^{*,*}$, $E^{*,y}$ and $E^{y,*}$ ease transition to the case when $\mathbf{y}$ has several constituents or $\mathbf{R}$ has several predicates of arity >1. If $\mathbf{y}$ has two constituents y_1, y_2 but the binary predicate E still is the only nonunary predicate, the new unary predicates are

$$E^{*,*},\ E^{*,y_1},\ E^{*,y_2},\ E^{y_1,*},\ E^{y_2,*}$$

and the new nullary predicates are

$$E^{y_1,y_1},\ E^{y_1,y_2},\ E^{y_2,y_1},\ E^{y_2,y_2}.$$

If $\mathbf{y}$ has only one constituent y and a ternary predicate R is the only nonunary predicate in $\mathbf{R}$, then the new unary predicates are

$$R^{*,*,*},\ R^{y,*,*},\ R^{*,y,*},\ R^{*,*,y},\ R^{y,y,*},\ R^{y,*,y},\ R^{*,y,y}$$

and the only new nullary predicate is $R^{y,y,y}$. □

In the rest of the proof, we assume that φ does not contain E-atoms with at most one universal variable. The only E-atoms that can appear in φ are $E(x_1, x_2)$ and $E(x_2, x_1)$.

Let P and Q be fresh nullary predicate variables, and let $\varphi'(\mathbf{y}, x_1, x_2)$ (respectively, $\varphi''(\mathbf{y}, x_1, x_2)$) be obtained from $\varphi(\mathbf{y}, x_1, x_2)$ by replacing $E(x_1, x_2)$ with P (respectively, Q) and replacing $E(x_2, x_1)$ with Q (respectively, P).

LEMMA 9.6. *The formula $\exists E \dot\forall x_1 x_2 \varphi(\mathbf{y}, x_1, x_2)$ is logically equivalent to the MSO formula.*

$$\dot\forall x_1 x_2 \exists P \exists Q [(\varphi'(\mathbf{y}, x_1, x_2) \wedge \varphi''(\mathbf{y}, x_2, x_1)].$$

PROOF. Let $\alpha(\mathbf{y}) = \dot{\forall} x_1 x_2 \varphi(\mathbf{y}, x_1, x_2)$ and $\beta(\mathbf{y})$ be the MSO formula above.

First, we suppose that $(M, E) \models \alpha(\mathbf{b})$, where M is any model. We check that $M \models \beta(\mathbf{b})$. Let a_1, a_2 be distinct elements of M that do not occur in $\mathbf{b}$, and set $P := E(a_1, a_2)$, $Q := E(a_2, a_1)$. Since $(M, E) \models \varphi(\mathbf{b}, a_1, a_2)$, $M \models \varphi'(\mathbf{b}, a_1, a_2)$. Since $(M, E) \models \varphi(\mathbf{b}, a_2, a_1)$, $M \models \varphi''(\mathbf{b}, a_2, a_1)$. Thus, $M \models \exists P \exists Q[\varphi'(\mathbf{b}, a_1, a_2) \wedge \varphi''(\mathbf{b}, a_1, a_2)]$. Since a_1, a_2 were arbitrary distinct elements outside of $\mathbf{b}$, $M \models \beta(\mathbf{b})$.

Secondly suppose that $M \models \beta(\mathbf{b})$ and order the elements of M. For all elements $a_1 < a_2$ in M that do not occur in $\mathbf{b}$, choose (the values of) P and Q such that $M \models \varphi'(\mathbf{b}, a_1, a_2) \wedge \varphi''(\mathbf{b}, a_2, a_1)$. Set $E(a_1, a_2) := P$ and $E(a_2, a_1) := Q$. Define the remaining values of E arbitrarily.

We check that $(M, E) \models \alpha(\mathbf{b})$. Indeed, let $a_1 < a_2$ be distinct elements of M outside of $\mathbf{b}$. By the definition of E, $M \models \varphi'(\mathbf{b}, a_1, a_2) \wedge \varphi''(\mathbf{b}, a_2, a_1)$. It follows that (M, E) models both $\varphi(\mathbf{b}, a_1, a_2)$ and $\varphi(\mathbf{b}, a_2, a_1)$. Thus, $(M, E) \models \alpha(\mathbf{b})$. Hence, $M \models \exists E \alpha(\mathbf{b})$. $\square$

It follows that the sentence Φ is equivalent to an MSO sentence. This establishes Theorem 9.1 (see appendix for a full proof).

By Corollary 9.2 and the results in the previous sections, we thus obtain the following exhaustive characterization of the regular ESO-prefix classes.

THEOREM 9.7

(*i*) *ESO*$(\exists^*\forall\exists^*)$ *and ESO*$(\exists^*\forall\forall)$ *are the only maximal regular standard ESO-prefix classes.*

(*ii*) *The unique maximal (general) regular ESO-prefix class is given by ESO*$(\exists^*\forall\exists^*) \cup$ *ESO*$(\exists^*\forall\forall) =$ *ESO*$(\exists^*\forall(\forall \cup \exists^*))$.

(*iii*) *There are three minimal nonregular ESO-prefix classes, which are the standard ESO-prefix classes ESO*$(\forall\forall\forall)$, *ESO*$(\forall\forall\exists)$, *and ESO*$(\forall\exists\forall)$.

10. *A Dichotomy Theorem for Model Checking*

In this section, we establish a result which is rather unexpected: For any ESO-prefix class, model checking is either possible by a DFA (and thus in constant space), or it is NP-complete. This shows that there is a (provably) huge gap in the computational complexity of different ESO-prefix classes.

THEOREM 10.1. *Model checking for ESO*(Q) (*i.e.*, *given a string W, decide whether $W \models \Phi$ where Φ is fixed*) *is NP-complete*, *for every* $Q \in \{\forall\forall\forall, \forall\forall\exists, \forall\exists\forall\}$. *Moreover*, *NP-hardness holds for sentences whose list of second-order variables consists of a single binary predicate variable*.

PROOF. Clearly, the problem is in NP. For the hardness part, we show that SAT can be reduced to model checking for $\Sigma^1_1(Q)$.

We first show this for $Q = \forall\forall\forall$, and then by adaptations of the proof for the other prefixes. For making the proof more intelligible, we first show that SAT is expressible by formulas with monadic and two predicate variables R, R'; later, we will show how to get rid of all predicate variables except for R.

We choose a string encoding of SAT instances, which are collections of clauses $\mathcal{C} = \{C^1, \ldots, C^m\}$ on propositional variables $p_1, \ldots, p_n$, as follows.

The alphabet is $A = \{\mathtt{0}, \mathtt{1}, \mathtt{+}, \mathtt{-}, \mathtt{[}, \mathtt{]}, \mathtt{(}, \mathtt{)}\}$. We encode the variables p_i, $1 \leq i \leq n$, by binary strings of length $\lceil \log n \rceil$. Each string encoding p_i is enclosed by parentheses '(',')'. The polarity of a literal $p_i / \neg p_i$ is represented by the letters '+' or '−', respectively, which immediately follows the closing parenthesis ')' of the encoding of p_i. A clause is encoded as a sequence of literals which is enclosed in square brackets '[',']'. We assume without loss of generality that $\mathscr{C} \neq \emptyset$ and that each clause $C^i \in \mathscr{C}$ contains at least one literal.

For example, the clause set $\mathscr{C} = \{\{p, q, \neg r\}, \{\neg p, \neg q, r\}\}$ is encoded by the following string:

$$[(\mathtt{00}) + (\mathtt{01}) + (\mathtt{10}) -][(\mathtt{00}) - (\mathtt{01}) - (\mathtt{10}) +].$$

Here, the propositional variables p, q, r are encoded by the binary strings 00, 01, 10, respectively.

This encoding is somewhat redundant but very intuitive. It is evident that such an encoding can be obtained from any standard representation of SAT in logspace.

In what follows, we will use the formulas

$$eqcol(x, y) = \bigvee_{\ell \in A} (C_\ell(x) \wedge C_\ell(y)), \tag{12}$$

$$varenc(x) = C_{(}(x) \vee C_{\mathtt{0}}(x) \vee C_{\mathtt{1}}(x) \vee C_{)}(x), \tag{13}$$

which state that the string has at positions x and y the same letter from A and that x is a letter of a variable encoding, respectively.

Consider the $\Sigma_1^1(\forall\forall\forall)$ formula

$$\Phi = \exists G \exists V \exists R \exists R' \forall x \forall y \forall z . \varphi,$$

where G and V are unary, R' and R are binary, and φ is the conjunction of the following quantifier-free formulas:

$\varphi_G = \varphi_{G,1} \wedge \varphi_{G,2} \wedge \varphi_{G,3}$,

where

$$\varphi_{G,1} = (C_{[}(x) \rightarrow \neg G(x)) \wedge (C_{]}(x) \rightarrow G(x)),$$

$$\varphi_{G,2} = (Succ(x, y) \wedge \neg C_{[}(y) \wedge \neg C_{)}(y)) \rightarrow (G(y) \leftrightarrow G(x)),$$

$$\varphi_{G,3} = (C_{)}(y) \wedge Succ(x, y) \wedge Succ(y, z)) \rightarrow$$
$$(G(y) \leftrightarrow [G(x) \vee (V(y) \wedge C_{+}(z)) \vee (\neg V(y) \wedge C_{-}(z))]),$$

and

$$\varphi_V = (C_{)}(x) \wedge C_{)}(y) \wedge R(x, y)) \rightarrow (V(x) \leftrightarrow V(y)),$$

$$\varphi_R = [R(x, y) \rightarrow (eqcol(x, y) \wedge varenc(x))] \wedge [(C_{(}(x) \wedge C_{(}(y)) \rightarrow R(x, y)],$$
$$\wedge [(\neg C_{(}(x) \wedge Succ(z, x)) \rightarrow (R(x, y) \leftrightarrow (R'(z, y) \wedge eqcol(x, y)))], \tag{14}$$

$$\varphi_{R'} = Succ(z, y) \rightarrow [R'(x, y) \leftrightarrow (R(x, z) \wedge \neg C_{)}(z))].$$

The intuition behind this formula is as follows: The predicate V assigns a truth value to each occurrence of a variable in the represented clause set $\mathscr{C}$, which is given by the value of V at the closing parenthesis of this occurrence. The clause set $\mathscr{C}$ is satisfiable, precisely if there exists such a V assigning every occurrence of the same variable the same truth value, such that every clause is satisfied. This property is checked by the use of G, R, and R'.

The predicate G is used for checking whether each clause $C \in \mathscr{C}$ is satisfied by the assignment V. To this end, the predicate G is set to false at the '[' marking the beginning of C, and set to true at the ']' marking the end of C by formula $\varphi_{G,1}$; the formulas $\varphi_{G,2}$ and $\varphi_{G,3}$ propagate the value of G from a position x in the clause representation to the successor position y, where the value switches from false to true if y marks the sign of a literal which is satisfied by V; the conjunct $\neg C_{[}$ in φ_G prohibits the transfer of G from the end of C to the beginning of the next clause, for which G must be initialized to false.

The predicate R is used to identify the closing parentheses ')' of the representations of occurrences of the same variables. For positions x and y at which the string W has letter ')', the predicate $R(x, y)$ is true precisely if x and y mark the end of the same variable name. This is used in the formula φ_V, which then simply states that V assigns every occurrence of a variable p in $\mathscr{C}$ the same truth value.

The purpose of the formulas φ_R and $\varphi_{R'}$ is to ensure that $R(x, y)$ has for positions x and y which mark the ends of occurrences of the same variable the desired meaning. This is accomplished in an inductive way. $R(x, y)$ intuitively expresses the following: x and y have the same color and must be part of the encodings $o_x = (\cdots)$ and $o_y = (\cdots)$ of variable occurrences in $\mathscr{C}$; moreover, x and y are at the same distance from the beginnings of these encodings. By reference to the predicate R', $R(x, y)$ furthermore expresses that these properties also hold for the pair (x^-, y^-), where x^- (respectively, y^-) is the predecessor of x in o_x (respectively, of y in o_y). The predicate R' is an auxiliary predicate since we may not introduce additional first-order variables, which are needed in the natural statement of the inductive property of R.

The formulas φ_R and $\varphi_{R'}$ thus effect that if $R(x, y)$ is true for positions x and y with letter ')' in the string, then x and y mark the end of the same variable.

We establish the following fact. Let for each $x \in W$ such that $W \models \mathit{varenc}(x)$ be $bv(x) = \max\{y | y \leq x,\ W \models C_{(}(y)\}$ the position of the closest '(' in W preceding or identical to x. Moreover, denote by $\mathit{reldist}(x)$ the distance between $bv(x)$ and x and by $\mathit{preds}(x) = \{y \in W | bv(x) \leq y < x\}$ the set of all elements y preceding x in the encoding of the variable to which x contributes.

FACT 10.2. *Suppose that for relations R, R' we have $(W, R, R') \models \forall x \forall y \forall z.\varphi_R \wedge \varphi_{R'}$. Then, for all elements $x, y \in W$, $R(x, y)$ holds iff*

(1) $W \models \mathit{eqcol}(x, y)$,
(2) $W \models \mathit{varenc}(x)$,
(3) $\mathit{reldist}(x) = \mathit{reldist}(y)$, *and*
(4) *for all $x' \in \mathit{preds}(x)$, $y' \in \mathit{preds}(y)$, $\mathit{reldist}(x') = \mathit{reldist}(y')$ implies $\mathit{eqcol}(x', y')$.*

PROOF. For every elements $x, y \in W$ such that $W \models \neg\,\mathit{eqcol}(x, y)$ or $W \models \neg\,\mathit{varenc}(x)$, we clearly have $\neg R(x, y)$ by formula φ_R. Thus, it suffices to consider all elements $x, y \in W$ such that $W \models \mathit{eqcol}(x, y) \wedge \mathit{varenc}(x)$, (and thus $W \models \mathit{varenc}(y)$).

For such x, y, we prove the claimed equivalence by induction on $n = \max\{\mathit{reldist}(x), \mathit{reldist}(y)\}$.

For $n = 0$, it is easy to see that the equivalence holds.

Consider thus $n > 0$. Then, we have $\mathit{reldist}(x)$, $\mathit{reldist}(y) > 0$ and $W \models \neg C_{(}(x)$. Suppose that $R(x, y)$ holds. Then, by part (14) of formula φ_R, for the predecessor x^- of x in W, $R'(x^-, y)$ holds (note that x^- must exist). By the formula $\varphi_{R'}$, we have that for the predecessor y^- of y in W, $R(x^-, y^-)$ is true. From the induction hypothesis, we thus easily conclude that (3) and (4) hold if $R(x, y)$ holds.

Conversely, suppose that (1)–(4) hold. Then, by the induction hypothesis, we have that $R(x^-, y^-)$ holds where x^- and y^- are as previously. Since clearly $W \not\models C_{)}(y^-)$, by $\varphi_{R'}$ we have that $R'(x^-, y)$ is true, and by part (14) of formula φ_R we thus have that $R(x, y)$ is true. This concludes the induction step and thus the proof of Fact 10.2. $\square$

On the other hand, for every word W encoding a SAT instance, we can define relations R and R' such that $(W, R, R') \models \forall x, y, z.\ \varphi_R \wedge \varphi_{R'}$. Indeed, include in R all tuples (x, y) which satisfy (1)–(4) in Fact 10.2, and include in R' all tuples (x, y^+) such that (x, y) is in R, $W \models \mathit{Succ}(y, y^+)$, and $W \models \neg C_{)}(y)$. For these R and R' we can check that $(W, R, R') \models \forall x, y, z. \varphi_R \wedge \varphi_{R'}$.

CLAIM 10.3. *For a string W encoding a clause set $\mathscr{C}$, $\mathscr{C}$ is satisfiable iff $W \models \Phi$.*

PROOF OF CLAIM. Suppose that $\mathscr{C}$ is satisfiable. Then, there exists an assignment V of truth values to all occurrences of propositional atoms in $\mathscr{C}$ such that all occurrences of the same variables receive the same value and every clause is satisfied. We thus can readily define relations V, G, R and R' such that $(W, G, V, R, R') \models \forall x \forall y \forall z \varphi$, and thus $W \models \Phi$; on the segment encoding a clause $C \in \mathscr{C}$ in W, G is false until the end marker ')' of the first encoding of a variable in C such that V makes the corresponding literal true, and G is true from there onwards.

On the other hand, suppose that $W \models \Phi$. Hence, there exist relations G, V, R, and R' such that $(W, G, V, R, R') \models \forall x \forall y \forall z \varphi$. Define a truth assignment τ to the variables p in $\mathscr{C}$ by $\tau(p) = V(x)$, where $x \in W$ is any end marker ')' of the encoding of an occurrence of p in $\mathscr{C}$.

From Fact 10.2, it follows that if $R(x, y)$ holds for elements $x, y \in W$ such that $W \models C_{)}(x) \wedge C_{)}(y)$, then x and y mark the ends of the occurrences of the same variable. Hence, by formula φ_V, the end markers $x_1(p), \ldots, x_{k_p}(p)$ of the encodings of all occurrences of variable p in $\mathscr{C}$ have the same value in V. From formula φ_G, it thus follows that for every clause $C \in \mathscr{C}$, there must exist a literal $L \in C$ such that τ makes L true. Hence, $\mathscr{C}$ is satisfiable. This proves the claim. $\square$

This concludes the proof for the class ESO(Q) where $Q = \forall\forall\forall$. For the remaining two classes ($Q = \forall\forall\exists$, $Q = \forall\exists\forall$), we slightly adjust Φ. Let C'_+ be a fresh monadic predicate variable, and define

$$\varphi_+ = \mathit{Succ}(x, y) \rightarrow ((C'_+(x) \leftrightarrow C_+(y));$$

intuitively, φ_+ states that C'_+ is a left shift of C_+ by one position in the string (where the value of $C'_+(max)$ is not specified).

Modify φ now to a formula φ' as follows.

—add φ_+ as a conjunct;

—replace $\varphi_{G,3}$ by the following formula $\varphi'_{G,3}$:

$$\varphi'_{G,3} = (C_{)}(y) \wedge Succ(x, y)) \rightarrow$$
$$[G(y) \leftrightarrow (G(x) \vee (V(y) \wedge C'_+(y)) \vee (\neg V(y) \wedge \neg C'_+(y)))];$$

—rewrite φ_R to φ'_R by moving $Succ(z, x)$ in the third conjunct from the premise to the consequent, that is,

$$\varphi'_R = [R(x, y) \rightarrow (eqcol(x, y) \wedge varenc(x))] \wedge$$
$$[(C_{(}(x) \wedge C_{(}(y)) \rightarrow R(x, y)] \wedge [(\neg C_{(}(x) \wedge$$
$$(x \neq min)) \rightarrow (Succ(z, x) \wedge (R(x, y) \leftrightarrow (R'(z, y) \wedge eqcol(x, y))))];$$

—and, replace $\varphi_{R'}$ by the formula

$$\varphi'_{R'} = (x \neq min) \rightarrow [Succ(z, x) \wedge (R'(y, x) \leftrightarrow (R(y, z) \wedge \neg C_{)}(z)))].$$

Observe that in the formula φ', the variable z only occurs in φ'_R and in $\varphi'_{R'}$.

Let

$$\Phi_{\forall\forall\exists} = \exists C'_+, G, V, R, R' \forall x \forall y \exists z.\varphi',$$
$$\Phi_{\forall\exists\forall} = \exists C'_+, G, V, R, R' \forall x \exists z \forall y.\varphi'.$$

Then, for every string W which encodes a clause set $\mathscr{C}$,

$$W \models \Phi \Leftrightarrow W \models \Phi_{\forall\forall\exists} \Leftrightarrow W \models \Phi_{\forall\exists\forall}$$

Indeed, the formula $\forall x \forall y \exists z \varphi'(x, y, z)$ is true in an expanded structure (W, C'_+, G, V, R, R'), iff for all elements $a, b \in W$, $\varphi'(a, b, f(a))$ is satisfied, where $f(a)$ is the predecessor of a in W, if $a \neq min$, and $f(min)$ is arbitrary (observe that then $R(a, b)$ is false if $a = min$ or $b = min$). Therefore, the existential quantifier on z only depends on x and can be moved before $\forall y$.

Thus, NP-hardness of model-checking holds for the prefix-classes ESO($\forall\forall\forall$), ESO($\forall\exists\forall$), and ESO($\forall\forall\exists$).

To conclude the proof of the theorem, we show first how we can get rid of the binary predicate variable R', and then how we can eliminate all monadic predicate variables.

10.1. GETTING RID OF R'. Recall that $\Phi = \exists G \exists V \exists R \exists R' \forall x \forall y \forall z.\varphi$. For any relations G, V, R, R' on a string W such that $(W, G, V, R, R') \models \forall x \forall y \forall z.\varphi$, we observe the following:

(1) For all elements $a, b \in U(W)$, $R(a, b)$ and $R'(a, b)$ never hold simultaneously; therefore, there is enough "space" in the relation R for packing the tuples of R' into R and discarding R', provided we can distinguish both groups of tuples from each other.

(2) From Fact 10.2 it follows, in particular, that for elements $a, b \in W$, $R(a, b)$ holds only if $reldist(a) \equiv reldist(b) \pmod 2$. On the other hand, as easily seen, $R'(a, b)$ only holds if $reldist(a) \not\equiv reldist(b) \pmod 2$. This allows us to distinguish possible tuples in R from possible tuples in R'. Thus, if we set $R := R \cup R'$, any tuple (a, b) of the new R stems from the old R, if $reldist(a) \equiv reldist(b) \pmod 2$, and from R', if $reldist(a) \not\equiv reldist(b)$ (mod2).

(3) The property $reldist(x) \equiv reldist(y) \pmod 2$ is clearly a regular property. It can be logically defined via a monadic predicate P such that

$$P(x) \leftrightarrow (reldist(x) \equiv 0 \pmod 2).$$

In fact, if predicate P is available, then $reldist(x) \equiv reldist(y) \pmod 2$ if and only if $P(x) \leftrightarrow P(y)$.

Based on these observations, it is clear that the following formula Φ^* is equivalent to Φ over SAT instances:

$$\Phi^* = \exists P \exists G \exists V \exists R \forall x \forall y \forall z (\varphi_P(x, y) \wedge \varphi^*(x, y, z)),$$

where $\varphi_P(x, y)$ defines the predicate P by

$$\varphi_P(x, y) = (C_((x) \rightarrow P(x)) \wedge$$
$$[(Succ(x, y) \wedge varenc(x) \wedge varenc(y)) \rightarrow (P(y) \leftrightarrow \neg P(x))],$$

and φ^* results from φ by replacing any atom $R(\chi, \xi)$ with $R(\chi, \xi) \wedge (P(\chi) \leftrightarrow P(\xi))$ and replacing any atom $R'(\chi, \xi)$ with $R(\chi, \xi) \wedge \neg (P(\chi) \leftrightarrow P(\xi))$.

From the sentences $\Phi_{\forall\forall\exists}$ and $\Phi_{\forall\exists\forall}$, we can remove R' in an analogous manner, obtaining sentences $\Phi^*_{\forall\forall\forall}$ and $\Phi^*_{\forall\exists\forall}$.

10.2. Getting Rid of Monadic Predicate Variables. Note that the only monadic predicate variables in the formulas Φ^*, $\Phi^*_{\forall\forall\exists}$, and $\Phi^*_{\forall\exists\forall}$ are C'_+, G, V, and P. Moreover, for every choice of relations (G, V, P, R) witnessing that $W \models \Phi^*$ (respectively, (C'_+, G, V, P, R) witnessing that $W \models \Phi^*_{\forall\forall\exists}$ or $W \models \Phi^*_{\forall\exists\forall}$), the constants *min* and *max* do not occur in any tuple of R, if we strip off possible such tuples arising from the packing of R' into R by adding $R'(x, y) \models varenc(x)\ varenc(y)$ in the very beginning. Thus, it is possible to pack the monadic predicates into R as follows.

Replace in Φ^*, $\Phi^*_{\forall\forall\exists}$, and $\Phi^*_{\forall\exists\forall}$ uniformly $C'_+(\chi)$ by $R(min, \chi)$, $G(\chi)$ by $R(max, \chi)$, $V(\chi)$ by $R(\chi, min)$, and $P(\chi)$ by $R(\chi, max)$. By doing this and some minor adjustments, we obtain formulas $\Phi^\dagger$, $\Phi^\dagger_{\forall\forall\exists}$, and $\Phi^\dagger_{\forall\exists\forall}$ which contain R as single predicate variable and which are over SAT instances equivalent to Φ^*, $\Phi^*_{\forall\forall\exists}$, and $\Phi^*_{\forall\exists\forall}$, respectively. $\square$

Corollary 10.4. *Let $\mathfrak{Q} \subseteq \{\exists, \forall\}^*$ such that $\mathfrak{Q} \cap \exists^*\forall(\exists^+\forall\{\exists, \forall\}^* \cup \forall\{\exists, \forall\}^+) \neq \emptyset$. Then, model checking for ESO($\mathfrak{Q}$) is NP-complete.*

From Theorems 9.7 and 10.1, we obtain the following dichotomy theorem.

Theorem 10.5 (Dichotomy Theorem for Model Checking). *Let $\mathfrak{Q} \subseteq \{\forall, \exists\}^*$ be any prefix set. Then,* (*i*) *ESO*($\mathfrak{Q}$) *expresses either only regular languages, or also some NP-complete languages; equivalently,*

(*ii*) *model-checking for ESO*($\mathfrak{Q}$) *is either feasible by a finite-state automaton (and thus in constant space and linear time), or NP-complete.*

Observe that this dichotomy is provably strict and does not, like other dichotomy theorems, for example, Schaefer's result on SAT [Schaefer 1978] rely on widely believed assumptions such as P $\neq$ NP. Furthermore, not every ESO-prefix class which expresses some NP-complete languages can express all of NP. In fact, there is a unique minimal ESO-prefix class that captures NP.

PROPOSITION 10.6. *The unique minimal ESO-prefix class* (*closed under subprefixes*) *capturing NP over strings is ESO*($\forall^*$).

PROOF (SKETCH). As shown in Leviant [1989] and Eiter et al. [1996], this class captures NP over arbitrary finite ordered structures. Any (syntactically) smaller class C must have a constant bound k on the number of universal FO-quantifiers. But then model-checking for C is feasible in NTIME(n^k) on a RAM, since all "existential data" can be guessed nondeterministically. (On a TM, the exponent may be slightly higher but still constant because it may be necessary to check the consistency between several consecutive guesses.) By well-known time hierarchy theorems, it follows that C cannot express all of NP. $\square$

11. *Finite Satisfiability*

In this section, we prove some results concerning the satisfiability of ESO($\mathfrak{Q}$) sentences Φ over strings, that is, deciding whether $\mathcal{L}(\Phi) \neq \emptyset$. Recall from the introduction that the satisfiability problem for ESO($\mathfrak{Q}$) is a reformulation of the satisfiability problem for FO($\mathfrak{Q}$).

It turns out that the separation of regular and nonregular ESO-prefix classes also gives us a precise characterization of those classes for which satisfiability over strings is decidable. In fact, the regular and the decidable classes coincide.

THEOREM 11.1. *Let $\mathfrak{Q}$ be any prefix set. Then, over finite strings, the satisfiability problem for ESO*($\mathfrak{Q}$) *is decidable if and only if ESO*($\mathfrak{Q}$) *is regular.*

PROOF. The proofs of Theorem 8.2 (regularity of ESO($\exists^*\forall\exists^*$)) and Theorem 9.1 (regularity of ESO($\exists^*\forall\forall$)) are constructive, in the sense that a finite state automaton recognizing $\mathcal{L}(\Phi)$ can be effectively constructed from a given sentence Φ. Hence, from Theorem 9.7 and the fact that the emptiness problem of finite automata is decidable, it follows that for every regular class ESO($\mathfrak{Q}$), satisfiability over finite strings is decidable.

Conversely, it suffices to show that satisfiability over finite strings is undecidable for all classes ESO(Q) where $Q \in \{\forall\forall\forall, \forall\forall\exists, \forall\exists\forall\}$. This can be established by a simple encoding of the question whether a given domino problem has a periodic solution, which is up to minor adaptations identical to the encoding of domino problems into formulas of the Kahr–Moore–Wang class in the proof of Börger et al. [1997, Theorem 3.1.9]. For the reader's convenience, we describe the reduction here.

A *domino system* is a triple $\mathfrak{D} = \langle D, H, V\rangle$ where D is a finite set, whose elements are called *dominoes*, and $H, V \subseteq D \times D$ are binary relations. $\mathfrak{D}$ *tiles*

$\mathbb{N} \times \mathbb{N}$, where $\mathbb{N}$ denotes the nonnegative integers, if there is a *tiling* $\tau: \mathbb{N} \times \mathbb{N} \to D$ such that, for each $i, j \in \mathbb{N}$,

(i) If $\tau(i, j) = d$ and $\tau(i + 1, j) = d'$, then $(d, d') \in H$; and
(ii) If $\tau(i, j) = d$ and $\tau(i, j + 1) = d'$, then $(d, d') \in V$.

A domino system admits a periodic tiling of $\mathbb{N} \times \mathbb{N}$, if there are integers $h, v > 0$ such that for all $i, j \in \mathbb{N}$, $\tau(i, j) = \tau(i + h, j)$ and $\tau(i, j) = \tau(i, j + v)$. We note the following lemma.

LEMMA 11.2. *Whether a given domino system $\mathfrak{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N}$ is undecidable.*

Note that Berger [1966] proved that the unconstrained domino problem is undecidable. The lemma is an immediate consequence of the stronger result that the classes of domino systems that do not admit tilings of $\mathbb{N} \times \mathbb{N}$ and that admit periodic tilings of $\mathbb{N} \times \mathbb{N}$ are recursively inseparable (see Börger et al. [1997, Theorem 3.1.7]).

We describe an encoding of the question whether a domino system has a periodic tiling into a formula from ESO($\forall\exists\forall$); for the other classes, it is similar. For each domino $d \in D$, use a binary predicate variable P_d, and let $\mathbf{P}$ denote the tuple of all these variables. Define the formula Φ as $\exists \mathbf{P} \forall x \exists x' \forall y . \varphi(x, x', y)$, where

$$\varphi(x, x', y) = \varphi_s(x, x') \wedge \bigwedge_{d \neq d'} \neg (P_d(x, y) \wedge P_{d'}(x, y)) \wedge \bigvee_{(d,d') \in H} (P_d(x, y) \wedge P_{d'}(x', y)) \wedge \bigvee_{(d,d') \in V} (P_d(y, x) \wedge P_{d'}(y, x'))$$

and

$$\varphi_s(x, x') = Succ(x, x') \vee (x = max \wedge x' = min).$$

Intuitively, the formula φ_s expands a given finite string W periodically, by considering the first position in W as the successor of the last position. This is good enough for modeling a periodic tiling in which the horizontal period h and the vertical period v coincide; as argued below, imposing the restriction that $h = v$ must hold does not affect the answer of whether $\mathfrak{D}$ admits a periodic tiling. The second conjunct of φ states that at most one domino can be at any position, while the last two conjuncts express that some domino must be there and the tiling is admissible.

The formula Φ is satisfiable on finite strings over any fixed alphabet A, if and only if $\mathfrak{D}$ admits a periodic tiling of $\mathbb{N} \times \mathbb{N}$.

Indeed, if τ is a periodic tiling of $\mathbb{N} \times \mathbb{N}$ with horizontal and vertical periods h and v, respectively, then consider any string W on A whose length $|W|$ is the least common multiple of h and v. Define for all P_d in $\mathbf{P}$ relations on W by $P_d(i, j) \leftrightarrow \tau(i - 1, j - 1) = d$. Then, $(W, \mathbf{P}) \models \forall x \exists x' \forall y . \varphi(x, x', y)$, and hence Φ is satisfiable on finite strings over A.

Conversely, if for relations $\mathbf{P}$ on a string W over A, we have that $(W, \mathbf{P}) \models \forall x \exists x' \forall y . \varphi(x, x', y)$, then the map $\tau: \mathbb{N} \times \mathbb{N} \to D$ defined by $\tau(i, j) = d$

iff $(W, \mathbf{P}) \models P_d(i' + 1, j' + 1)$, where $i' = i \bmod |W|$ and $j' = j \bmod |W|$, is a periodic tiling of $\mathbb{N} \times \mathbb{N}$. $\square$

From this theorem we immediately get a corollary on the finite satisfiability of first order formulas over strings. Note that, as already mentioned in the introduction, in the context of satisfiability, FO($\mathcal{Q}$) formulas may contain free predicate variables besides those occurring in the signature of the input string (i.e., *Succ*, C_a, etc.).

COROLLARY 11.3. *Over finite strings, for any prefix set $\mathcal{Q}$ either*

(*i*) *the satisfiability problem for FO($\mathcal{Q}$) is decidable and ESO($\mathcal{Q}$) defines only regular languages, or*

(*ii*) *the satisfiability problem for FO($\mathcal{Q}$) is undecidable and ESO($\mathcal{Q}$) defines some NP-complete language.*

12. *Capturing REG and Closure under Complementation*

In this section, we investigate ESO-prefix classes that capture the class of regular languages, and that are closed under complementation.

The following lemma is well known; for an illustrative proof, see, for example, Kolaitis and Papadimitriou [1990].

LEMMA 12.1. *Every formula $\exists \mathbf{R} \exists y_1 \cdots \exists y_k.\varphi$, where φ is quantifier-free, is equivalent to a formula $\exists y_1 \cdots \exists y_k.\psi$, where ψ is quantifier-free.*

A similar but weaker lemma holds if we allow one universal FO quantifier before the quantifier-free part.

LEMMA 12.2. *Over strings, every formula $\Phi = \exists \mathbf{R} \exists y_1 \cdots \exists y_k \forall x.\varphi$, where φ is quantifier-free, is equivalent to a finite disjunction of first-order formulas $\exists y_1 \cdots \exists y_k \forall x.\psi$, where ψ is quantifier-free.*

PROOF (SKETCH). The proof follows the lines of the proof of Theorem 9.1. Φ can be assumed to be a disjunction of formulas $\Phi' = \exists \mathbf{R} \exists y_1 \cdots \exists y_{k'} \forall x.\varphi'$, where $k' \leq k$ and φ' fixes the quantifier-free type of the variables y_i such that they are all different from each other and from *min* and *max*. It suffices to prove the claim for a formula Φ of this form to obtain the result.

Predicates $R \in \mathbf{R}$ of arity > 1 can be replaced in Φ by monadic predicates, so that we may assume that all predicates in Φ have arity ≤ 1. Moreover, the quantifier-free part of Φ can be rewritten to a DNF $\bigvee_j \delta_j$ of types δ_j each of which extends the quantifier-free type of the variables y_i by fixing the location of x with respect to the variables y_i and *min* and *max* such that if $x = y_i$ for some y_i (respectively, $x = min$, $x = max$) occurs in δ, then x does not occur in any other literal of δ.

Remove then all clauses δ_i which contain an opposite pair of literals, remove from the remaining types all literals $R(x)$, $\neg R(x)$ where R is any (monadic) predicate from $\mathbf{R}$, and remove atoms $R(c)$ where $c \in \{min, max, y_1, \ldots, y_k\}$ by using fresh nullary predicate variables. It is not hard to see that the resulting formula (which contains no predicate variables of arity > 0) is equivalent to a disjunction of formulas $\exists y_1 \cdots \exists y_k \forall x.\psi$, where ψ is quantifier-free. This proves the lemma. $\square$

THEOREM 12.3. *Over strings, every language expressible in ESO*($\exists^*\forall$) *is expressible in FO*($\exists^*\forall$).

PROOF. By Lemma 12.2, it remains to show that any disjunction $\varphi = \bigvee_{i=1}^{k} \varphi_i$ of FO sentences $\varphi_i = \exists \mathbf{y}_i \forall x \cdot \psi_i$, where ψ_i is quantifier-free, is over strings equivalent to some FO($\exists^*\forall$) sentence ψ. Let $\mathbf{z} = z_1, \ldots, z_k$ be fresh FO variables, and suppose the $\mathbf{y}_i$ are pairwise disjoint; the desired ψ is

$$\psi = \exists \mathbf{z} \exists \mathbf{y}_1 \cdots \exists \mathbf{y}_k \forall x \Big[\bigvee_i \psi_i \wedge \bigvee_i (z_i = min) \wedge \bigwedge_i ((min \neq max \wedge z_i = min) \rightarrow \psi_i) \Big].$$

Intuitively, each z_i serves as a switch; if $min \neq max$ and $z_i = min$, then the formula φ_i is selected as a witness for the truth of φ. This is checked by the rightmost conjunct of ψ. The conjunct $\bigvee_i (z_i = min)$ enforces that at least one switch is on. However, for the extremal case where $min = max$ the rightmost disjunct does not work; this case is taken care of by the conjunct $\bigvee_i \psi_i$. □

By exploiting the previous lemmata, the following proposition is easily proved.

PROPOSITION 12.4. *Let $A = \{a, b\}$. Then,* (*i*) *$L_1 = a^*$ is not expressible in ESO*($\exists^*$); (*ii*) *$L_2 = a^*b^*$ is not expressible in ESO*($\exists^*\forall$); (*iii*) *$L_3 = \{a, b\}^+\{ba, ab\}\{a, b\}^+$ is not expressible in ESO*($\{\exists, \forall\}$).

PROOF. Lemma 12.1, part (i) is easy; suppose a formula $\exists \mathbf{y}.\varphi(\mathbf{y})$, where φ is quantifier-free, expresses L_1. Then, for every $W \in a^*$, there is some tuple $\mathbf{c}$ for $\mathbf{y}$ on W such that $W \models \varphi(\mathbf{c})$. For W large enough, choose an element b in W which is not coincident and not adjacent to min, max and any element in $\mathbf{c}$, and change the color of x. The resulting string $W' \models \varphi(\mathbf{c})$; this contradicts that $\exists \mathbf{y}.\varphi(\mathbf{y})$ expresses L_1.

Part (ii) is shown analogously. Suppose L_2 were definable in ESO($\exists^*\forall$). Then, from Lemma 12.2, it follows that L_2 were definable by a finite disjunction $\varphi = \bigvee_{i=1}^{d} \varphi_i$ of formulas $\varphi_i = \exists y_1 \cdots \exists y_k \forall x \psi_i$, for some $k \geq 0$, where each φ_i is quantifier-free.

Hence, for any string of form $W = a^n b^n$, there is some φ_i such that $W \models \varphi_i$, that is, there are elements $a_1, \ldots, a_k$ in W such that $(W, a_1, \ldots, a_k) \models \forall x \psi_i$. For W large enough (choose $n > 3k + 2$), there are elements x_a and x_b of colors C_a and C_b in W, respectively, such that x_a and x_b are not adjacent or coincident to any of $min, a_1, \ldots, a_k, max$. Exchange the colors of x_a and x_b, and let W' be the resulting string. It is easy to see that $(W', a_1, \ldots, a_k) \models \forall x \cdot \psi_i$, and thus $W' \models \varphi$. This contradicts that φ defines L_2.

To prove part (iii), suppose first L_3 were definable in ESO($\exists$). By Lemma 12.1, L_3 would be definable by a sentence $\exists x \varphi(x)$, where φ is quantifier-free. Consider the string $W = aaba$, which is in L_3; it is easily seen that if $W \models \exists x \varphi(x)$, then either $a^4 \models \exists x \varphi(x)$ or $ab^2a \models \exists x \varphi(x)$; this contradicts that $\exists x \cdot \varphi(x)$ defines L_3.

Suppose then L_3 were definable in ESO($\forall$). By Lemma 12.2, L_3 were definable by a disjunction $\varphi = \bigvee_{i=1}^{n} \forall x \varphi_i(x)$ of universal formulas $\forall x \varphi_i(x)$, where $\varphi_i(x)$ is quantifier-free. Consider $W = a^2baa^2$. Then, $W \models \forall x \varphi_i(x)$ for some i. Change the color of the third letter to C_a. Then, for the resulting $W' = a^6$, we have $W' \models \forall x \varphi_i(x)$, since the third and fourth letter of W' are indiscernible for φ_i and φ_i is true for the fourth letter of W. This contradicts that φ defines L_3. $\square$

We thus establish the following result:

THEOREM 12.5. *Let $\mathcal{Q}$ be any nontrivial prefix set such that ESO($\mathcal{Q}$) is regular. Then, the following statements are equivalent*:

(*i*) *ESO($\mathcal{Q}$) is closed under complementation*;
(*ii*) *ESO($\mathcal{Q}$) captures REG*;
(*iii*) $\mathcal{Q} \cap \exists^*\forall\{\forall, \exists\}^+ \neq \emptyset$.

PROOF

(i) $\Rightarrow$ (ii). Suppose that ESO($\mathcal{Q}$) is closed under complementation. By Proposition 2.2, for establishing (ii) it suffices to show that $\mathcal{Q}$ contains some prefix Q which contains $\forall\exists$ or $\forall\forall$ as a substring. We show that such a Q must exist in $\mathcal{Q}$.

Suppose first that $\mathcal{Q}$ contains any prefix Q' such that $|Q'| > 1$. Observe that the complement $\overline{L_2} = \{a, b\}^*ba\{a, b\}^*$ of the language L_2 in part (ii) of Proposition 12.4 is clearly expressible both in ESO($\exists\exists$) and in ESO($\exists\forall$), while L_2 is neither expressible in ESO($\exists\exists$) nor in ESO($\exists\forall$). Since ESO($\mathcal{Q}$) is closed under complementation, it follows that in this case, $\mathcal{Q}$ contains some Q having $\forall\exists$ or $\forall\forall$ as substring.

Suppose now that every prefix $Q \in \mathcal{Q}$ has length ≤ 1. We derive a contradiction. Since $\mathcal{Q}$ is nontrivial, it follows that either $\exists$ or $\forall$ must belong to $\mathcal{Q}$, Suppose $\forall \in \mathcal{Q}$. Since the complement $\overline{L_3} = \{\lambda, a, b\} \cup \{a, b\}(a^* \cup b^*)\{a, b\}$ of the language L_3 in part (iii) of Proposition 12.4 is easily expressed in ESO($\forall$) and ESO($\mathcal{Q}$) is, by hypothesis, closed under complementation, it follows that some prefix Q' such that $|Q'| > 1$ must belong to $\mathcal{Q}$; this is a contradiction. Hence, $\forall \notin \mathcal{Q}$, and consequently $\{\exists\} \subseteq \mathcal{Q} \subseteq \{\lambda, \exists\}$ (recall that λ is the empty prefix). Clearly, the complement $\overline{L_1} = \{a, b\}^*b\{a, b\}^*$ of the language L_1 in part (i) of Proposition 12.4 is expressible in ESO($\exists$), while L_1 is not; this is a contradiction. This proves (i) $\Rightarrow$ (ii).

(ii) $\Rightarrow$ (iii). Suppose that ESO($\mathcal{Q}$) captures REG, and assume that $\mathcal{Q} \cap \exists^*\forall\{\exists, \forall\}^+ = \emptyset$. Then, since ESO($\mathcal{Q}$) is regular, it follows from Theorem 9.7 that $\mathcal{Q} \subseteq \exists^*\forall$. By part (ii) of Proposition 12.4, a^*b^* is not expressible in ESO($\exists^*\forall$); hence, ESO($\mathcal{Q}$) does not capture REG, which is a contradiction.

(iii) $\Rightarrow$ (i). From Proposition 2.2 and the hypothesis, it follows that ESO($\mathcal{Q}$) captures REG; it is well known that REG is closed under complementation. $\square$

COROLLARY 12.6. *ESO($\exists^*\forall$) is the unique maximal regular ESO-prefix class that does not capture REG.*

Note that Theorem 12.3, Corollary 12.6 and the results by Immerman [1987] imply that model checking for regular ESO-prefix classes which do not capture REG is in low levels of AC^0.

From the dichotomy theorem for model checking (Theorem 10.5), we thus obtain the following dichotomy theorem for closure under complementation.

THEOREM 12.7 (DICHOTOMY THEOREM FOR CLOSURE UNDER COMPLEMENTATION). *Let $\mathfrak{Q} \subseteq \{\exists, \forall\}^*$ be any nontrivial prefix set. Then, under the assumption NP ≠ co-NP, the following statements are equivalent*:

(*i*) *ESO*($\mathfrak{Q}$) *is closed under complementation.*
(*ii*) *ESO*($\mathfrak{Q}$) *captures REG.*
(*iii*) $\mathfrak{Q} \cap \exists^*\forall\{\forall, \exists\}^+ \neq \emptyset$ *and* $\mathfrak{Q} \cap \exists^*\forall(\exists^+\forall\{\exists, \forall\}^* \cup \forall\{\exists, \forall\}^+) = \emptyset$.

13. *Conclusion and Further Work*

In this paper, we have investigated the expressive power and complexity of the ESO-prefix classes over strings. We succeeded to settle some of the major questions on this issue, giving a complete picture of the regular ESO-prefix classes, their relationship to MSO, the complexity of ESO prefix classes, and a characterization of those classes that are closed under complementation.

Some of the results were rather unexpected and required novel and involved combinatorial proof arguments. In particular, this applies to the fact that ESO($\exists^*\forall\exists^*$), that is, the ESO-standard prefix class in which the first-order parts of sentences are from the Ackermann class with equality, and the proof of this result appears to be rather difficult.

Our results show several dichotomy properties for ESO-prefix classes over strings. In particular, model checking for a class ESO($\mathfrak{Q}$) is either feasible by a finite state automaton (and therefore in constant space), or it is NP-complete; a regular ESO($\mathfrak{Q}$) is either equivalent to MSO, or its expressive power is first-order (in fact, restricted to Σ_2^0); and, either ESO($\mathfrak{Q}$) is not closed under complementation (assuming NP ≠ co-NP), or it captures the class REG of regular languages. In summary, with respect to any of these criteria, each ESO-prefix class ESO($\mathfrak{Q}$) exhibits either a very good or an extremely bad behavior. Moreover, this matches observations on similar good/bad dichotomies of prefix classes which have been noted earlier in different contexts, for example, 0–1 laws for ESO-prefix classes [Kolaitis and Vardi 1990]. For a discussion, see Fagin's [1993, p. 20].

Three particular related research issues deserve further investigation:

—The scope of the present paper are finite strings. However, infinite strings or ω-words are another important area of research. In particular, Büchi has shown that an analogue of his theorem (Proposition 2.1) also holds for ω-words [Büchi 1962]. For an overview of this and many other important results on ω-words, we refer the reader to the excellent survey paper [Thomas 1990]. In this context, we have started to investigate which of the results of the present paper survive for ω-words. For some results, such as Theorem 9.1 this is obviously the case since no finiteness assumption on the input word structures was made in the proof. For the generalization of other results, such as Theorem 8.2, further research is needed.

—We are investigating which of our results survive in case a linear order $<$ is available on the word structures, and in case the successor relation *Succ* is replaced by such a linear order. As already mentioned in Section 2, for full ESO $<$ is tantamount to *Succ* because of interdefinability. However, this is not so for many of the limited ESO-prefix classes. Preliminary results suggest that most of the results in this paper carry over to the $<$ case.

—A further issue is the extension of our results to the fragments $\Sigma_k^1(\mathcal{Q})$, $k > 1$ of second-order logic, and to all of SO($\mathcal{Q}$).

Appendix A

THEOREM 9.1. *Over strings, every ESO*($\exists^*\forall\forall$) *sentence is equivalent to an MSO sentence.*

PROOF. Let Φ be a formula $\exists\mathbf{R}\exists\mathbf{y}\forall x_1\forall x_2.\varphi$, where $\mathbf{y} = y_1, \ldots, y_n$ is a vector of FO variables and φ is quantifier-free. Without loss of generality, we assume that *min* and *max* do not occur in φ (they can be defined using additional variables y_{min} and y_{max} in $\mathbf{y}$ and adding $\neg Succ(x_1, y_{min})$ and $\neg Succ(y_{max}, x_1)$ to φ).

Clearly, it suffices to consider only strings W such that $|U(W)| \geq n + 2$. Therefore, by a similar argument as in Lemma 9.4, we may assume without loss of generality that φ is a DNF formula

$$\varphi(\mathbf{y}, x_1, x_2) = (x_1 = x_2) \vee \bigvee_{i,j} (x_i = y_j) \vee \bigvee_{\delta_k} \delta_k,$$

such that every clause δ_k is a complete type over $T = \{x_1, x_2, y_1, \ldots, y_n\}$ which includes for every pair ξ, χ of different variables in T the literal $\xi \neq \chi$.

We show that all predicates in $R \in \mathbf{R}$ of arity >1 can be removed from Φ by introducing new monadic predicates $\mathbf{R^*}$.

First, we can easily remove all occurrences of atoms $R(\chi_1, \ldots, \chi_m)$ from φ such that at most one of x_1, x_2 occurs in them as follows: Replace $R(\chi_1, \ldots, \chi_m)$,

—if x_1 occurs in it, by a unary atom $R^{\chi'_1, \ldots, \chi'_m}(x_1)$, where $\chi'_i = *$ if $\chi_i = x_1$ and $\chi'_i = \chi_i$ otherwise, for $i = 1, \ldots, m$;
—if x_2 occurs in it, by a unary atom $R^{\chi'_1, \ldots, \chi'_m}(x_2)$, where $\chi'_i = *$ if $\chi_i = x_2$ and $\chi'_i = \chi_i$ otherwise, for $i = 1, \ldots, m$;
—otherwise, by a nullary predicate $R^{\chi_1, \ldots, \chi_m}$.

For example, $R(y_1, x_2, y_2, x_2)$ is replaced by $R_{y_1,*,y_2,*}(x_2)$, and $R(y_1, y_3)$ by R_{y_1,y_3}. Intuitively, $R_{\chi'_1, \ldots, \chi'_m}(x_i)$ represents the atom $R(\chi_1, \ldots, \chi_m)[*/x_i]$.

Let $\mathbf{R^*}$ be the list of all predicates variables that we have introduced for all $R \in \mathbf{R}$ of arity >1, and let

$$\Phi^* = \exists\mathbf{R^*}\exists\mathbf{R}\exists\mathbf{y}\forall x_1\forall x_2 \cdot \varphi^*,$$

where $\varphi^* = \bigvee_{\delta \in \Delta(\Phi)} \delta^*$ and δ^* is the clause obtained from δ by the above replacements. Notice that in δ^*, every literal with a predicate $R \in \mathbf{R}$ of arity >1 contains both x_1 and x_2.

Over any string W such that $|U(W)| \geq n + 2$, Φ is equivalent to Φ^*. To see this, observe that if Φ^* is true, then any interpreted atom $R^{\chi'_1, \ldots, \chi'_m}(e)$ occurring in some satisfied interpreted clause $\delta^*(\mathbf{y}, a_1, a_2)$ of φ represents an interpreted atom $R(a_1, \ldots, a_m)$. By the inequality literals in $\delta^*(\mathbf{y}, a_1, a_2)$, different atoms $R^{\chi'_1, \ldots, \chi'_m}(e)$ occurring in satisfied clauses $\delta^*(\mathbf{y}, a_1, a_2)$ represent different R-atoms; moreover, there is no conflict between any R-atom, represented by such an $R^{\chi'_1, \ldots, \chi'_m}(e)$ or by a nullary predicate $R^{\chi_1, \ldots, \chi_m}$, and any actual interpreted R-literal occurring in some interpreted clause δ^* which is satisfied.

It remains to remove all occurrences of atoms $R(\chi_1, \ldots, \chi_m)$ from Φ^* where $R \in \mathbf{R}$ and both x_1, x_2 occur among $\chi_1, \ldots, \chi_m$.

For each pair $\gamma^*, \delta^* \in \Delta(\Phi^*)$ such that the conjunction

$$C_{\gamma^*,\delta^*}(\mathbf{y}, x_1, x_2) = \gamma^*(\mathbf{y}, x_1, x_2) \wedge \delta^*(\mathbf{y}, x_2, x_1)$$

does not contain a pair of opposite literals, let $d_{\gamma^*,\delta^*}(\mathbf{y}, x_1, x_2)$ be the conjunction which results from C after removal of all literals that involve an $R \in \mathbf{R}$ of arity >1; denote by Γ the collection of all such d_{γ^*,δ^*}. (Notice that for different γ^*, δ^*, the clauses d_{γ^*,δ^*} are not necessarily different.)

Let $\mathbf{R}^1$ be the monadic relations in $\mathbf{R}$, and let

$$\Phi' = \exists \mathbf{R^*} \mathbf{R}^1 \exists \mathbf{y} \forall x_1 \forall x_2 . \varphi',$$

where $\varphi' = \bigvee_{\alpha \in \Gamma} \alpha(\mathbf{y}, x_1, x_2)$; notice that Φ' is a monadic SO formula.

CLAIM 13.1. *Over any string W such that $|U(W)| \geq n + 2$, $W \models \Phi' \leftrightarrow \Phi^*$.*

Hence, $W \models \Phi' \leftrightarrow \Phi$, which proves the result.

To prove the claim, suppose first that $W \models \Phi^*$. Then, relations $\mathbf{R^*}$, $\mathbf{R}$ and a tuple $\mathbf{b}$ of elements in W exist such that for every pair a_1, a_2 of elements in W, disjuncts $\gamma, \delta \in \Delta(\Phi)$ exist such that

$$(W, \mathbf{R^*}, \mathbf{R}, \mathbf{b}) \models \gamma^*(\mathbf{b}, a_1, a_2) \wedge \delta^*(\mathbf{b}, a_2, a_1),$$

and thus

$$(W, \mathbf{R^*}, \mathbf{R}, \mathbf{b}) \models d_{\gamma^*,\delta^*}(\mathbf{b}, a_1, a_2),$$

where $d_{\gamma^*,\delta^*} \in \Gamma$. This means $(W, \mathbf{R^*}, \mathbf{R}^1, \mathbf{b}) \models \forall x_1 \forall x_2 . \varphi'$, and hence $W \models \Phi'$.

Now suppose $W \models \Phi'$. Then, for suitable relations $\mathbf{R^*}$, $\mathbf{R}^1$ on W and a tuple $\mathbf{b}$ of elements in W, we have

$$(W, \mathbf{R^*}, \mathbf{R}^1, \mathbf{b}) \models \forall x_1 \forall x_2 . \varphi',$$

and thus for each pair a_1, a_2 of elements in W, there is some clause $d_{\gamma^*,\delta^*} \in \Gamma$, denoted by σ_{a_1,a_2}, such that $(W, \mathbf{R^*}, \mathbf{R'}, \mathbf{b}) \models \sigma_{a_1,a_2}(a_1, a_2)$.

We define extensions for the $R \in \mathbf{R}$ of arity > 1 as follows. For elements a_1, $a_2 \in U(W)$ such that $a_1 \leq a_2$ in W according to *Succ*, we have $\sigma_{a_1,a_2} = d_{\gamma^*,\delta^*}$ for some $\gamma^*, \delta^* \in \Delta(\Phi^*)$. Let $C_{\gamma^*,\delta^*}(\mathbf{b}, x_1, x_2)$ be the conjunction from which d_{γ^*,δ^*} results in the construction of Φ'. Then, $C(\mathbf{b}, a_1, a_2)$ does not contain any pair of opposite literals. Include in R every tuple $(e_1, \ldots, e_m)$ such that a positive literal $R(e_1, \ldots, e_m)$ occurs in $C(\mathbf{b}, a_1, a_2)$. Notice that if $(e_1, \ldots, e_m)$ is added to R, then there are no elements $a'_1 < a'_2$ such that $a'_1 \neq a_1$ or $a'_2 \neq a_2$ for which $\neg R(e_1, \ldots, e_m)$ occurs in the satisfied conjunction $C_{\gamma^{*\prime},\delta^{*\prime}}(\mathbf{b}, a'_1, a'_2)$, where $\sigma_{a'_1,a'_2} = d_{\gamma^{*\prime},\delta^{*\prime}}$. Indeed, the literals involving both a'_1 and a'_2 are different from those involving both a_1 and a_2. Then,

$$(W, \mathbf{R^*}, \mathbf{R}, \mathbf{b}) \models \gamma^*(\mathbf{b}, a_1, a_2) \wedge \delta^*(\mathbf{b}, a_2, a_1);$$

consequently,

$$(W, \mathbf{R^*}, \mathbf{R}, \mathbf{b}) \models \forall x_1 \forall x_2 \bigvee_{\delta^* \in \Delta(\Phi^*)} \delta^*(\mathbf{b}, x_1, x_2),$$

which means $W \models \Phi^*$. This proves the equivalence of Φ^* and Φ' over W and concludes the proof of the theorem. □

Remark. In the above proof, neither the finiteness of the universe nor the particular features of word structures (such as colors, succ) are used. Therefore, Theorem 9.1 generalizes to arbitrary structures.

ACKNOWLEDGMENTS. The authors thank E. Grädel and E. Rosen for comments concerning decidability aspects of FO-prefix classes over finite strings; they proposed Corollary 11.3 and suggested a reduction from domino problems for proving the undecidability part. We also thank F. Neven as well as the referees for a very close reading of the manuscript and their numerous comments. Furthermore, we are grateful to H. Veith and H. Habiger for useful comments. We acknowledge the support by the Austrian Science Fund Project NZ29-1NF and the National Science Foundation (NSF) grant CCR 95-04375.

REFERENCES

ABITEBOUL, S., HULL, R., AND VIANU, V. 1995. *Foundations of Databases*. Addison-Wesley, Reading, Mass.

AJTAI, M. 1983. Σ_1^1 Formulae on Finite Structures. *Ann. Pure Appl. Logic 24*, 1–48.

BASIN, D., AND KLARLUND, N. 1995. Hardware verification using monadic second-order logic. In *Proceedings of the 7th International Conference on Computer Aided Verification* (*CAV'95*). Lecture Notes in Computer Science, vol. 939. Springer-Verlag, New York, pp. 31–41.

BERGER, R. 1966. The undecidability of the domino problem. *Mem. AMS 66*.

BERSTEL, J. 1979. *Transductions and Context-Free Languages*. Teubner, Stuttgart.

BÖRGER, E., GRÄDEL, E., AND GUREVICH, Y. 1997. *The Classical Decision Problem*. Springer, Berlin-Heidelberg, Germany.

BÜCHI, J. R. 1962. On a decision method in restricted second-order arithmetic. In *Proceedings of the International Congress on Logic, Methodology and Philosophy of Science*, E. Nagel et al., eds, Stanford University Press, Stanford, Calif., pp. 1–11.

BÜCHI, J. R. 1960. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grund. Math. 6*, 66–92.

DE ROUGEMONT, M. 1987. Second order and inductive definability on finite structures. *Z. Math. Logik Grund. Math. 33*, 47–63.

DURAND, A., LAUTEMANN, C., AND SCHWENTICK, T. 1998. Subclasses of Binary NP. *J. Logic Comput. 8*, 2, 189–207.

EBBINGHAUS, H.-D., AND FLUM, J. 1995. Finite Model Theory. In *Perspectives in Mathematical Logic*. Springer-Verlag, New York.

EILENBERG, S. 1974. *Automata, Languages, and Machines*. Academic Press, New York.

EITER, T., AND GOTTLOB, G. 1998. On the expressiveness of frame satisfiability and fragments of second order logic. *J. Symb. Logic 63*, 1, 73–82.

EITER, T., GOTTLOB, G., AND GUREVICH, Y. 1996. Normal forms for second-order logic over finite structures, and classification of NP optimization problems. *Ann. Pure Appl. Logic 78*, 111–125.

FAGIN, R. 1974. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of Computation*, R. M. Karp, ed. American Mathematics Society, Providence, R.I., pp. 43–74.

FAGIN, R. 1975. Generalized monadic spectra. *Z. Math. Logik Grund. Math. 21*, 89–96.

FAGIN, R. 1993. Finite-model theory—A personal perspective. *Theoret. Comput. Sci. 116*, 1, 3–31.

GRÄDEL, E. 1991. The expressive power of second-order Horn logic. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science* (*STACS '91*). Lecture Notes in Computer Science, vol. 480, Springer-Verlag, New York, pp. 466–477.

GRÄDEL, E. 1992. Capturing complexity classes with fragments of second order logic. *Theoret. Comput. Sci*. 101, 5–57.

GRANDJEAN, E. 1985. Universal quantifiers and time complexity of random access machines. *Math. Syst. Theory 13*, 171–187.

GUREVICH, Y. 1988. Logic and the challenge of computer science. In *Trends in Theoretical Computer Science*, E. Börger, ed. chapter 1. Computer Science Press, Rockville, Md.

HENRIKSEN, J. G., JENSEN, J., JØRGENSEN, M., KLARLUND, N., PAIGE, B., RAUHE, T., AND SANDHOLM A. 1996. Mona: Monadic second-order logic in practice. In *Proceedings of Tools and Algorithms for the Construction and Analysis of Systems, First International Workshop, TACAS'95*. Lecture Notes in Computer Science, vol. 1019. Springer-Verlag, New York, pp. 89–110.

HOPCROFT, J., AND ULLMAN, J. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Mass.

IMMERMAN, N. 1986. Relational queries computable in polynomial time. *Inf. Cont. 68*, 86–104.

IMMERMAN, N. 1987. Languages that capture complexity classes. *SIAM J. Comput. 16*, 760–778.

IMMERMAN, N. 1999. *Descriptive Complexity*. Springer-Verlag, New York.

KLARLUND, N. 1998. Mona & Fido: The Logic-Automaton Connection in Practice. In *Proceedings of the Conference on Computer Science Logic* (*CSL '97*), Mogens Nielsen, Wolfgang Thomas, eds. Lecture Notes in Computer Science, vol. 1414. Springer-Verlag, New York, pp. 311–326.

KOLAITIS, P., AND PAPADIMITRIOU, C. 1990. Some computational aspects of circumscription. *J. ACM 37*, 1 (Jan.), 1–15.

KOLAITIS, P., AND PAPADIMITRIOU, C. H. 1991. Why not negation by fixpoint? *J. Comput. Syst. Sci. 43*, 125–144.

KOLAITIS, P., AND THAKUR, M. N. 1995. Approximation properties of NP minimization classes. *J. Comput. Syst. Sci. 50*, 391–411.

KOLAITIS, P., AND VARDI, M. Y. 1990. 0-1 Laws and decision problems for fragments of second-order logic. *Inf. Comput. 87*, 302–338.

LAUTEMANN, C., SCHWENTICK, T., AND THÉRIEN, D. 1995. Logics for context-free languages. In *Proceedings of the Conference on Computer Science Logic* (*CSL '94*). Leszek Pacholski, Jerzy Tiuryn, eds. Lecture Notes in Computer Science, vol. 933, Springer-Verlag, New York, pp. 205–216.

LEIVANT, D. 1989. Descriptive characterizations of computational complexity. *J. Comput. Syst. Sci. 39*, 51–83.

LYNCH, J. F. 1992. The quantifier structure of sentences that characterize nondeterministic time complexity. *Comput. Complex. 2*, 40–66.

MCNAUGHTON, R., AND PAPERT, S. 1971. *Counter-Free Automata*. MIT Press, Cambridge, Mass.

OLIVE, F. 1998. A conjunctive logical characterization of nondeterministic linear time. In *Proceedings of the Conference on Computer Science Logic* (*CSL '97*), Mogens Nielsen, Wolfgang Thomas, eds. Lecture Notes in Computer Science, vol. 1414, Springer-Verlag, New York, pp. 360–372.

PACHOLSKI, L., AND SZWAST, W. 1991. On the 0-1 law for the existential second-order minimal Gödel sentences with equality. In *Proceedings of the 6th Annual Symposium on Logic in Computer Science* (*LICS '91*). IEEE Computer Science Press, Los Alamitos, Calif., pp. 290–285.

PANCONESI, A., AND RANJAN, D. 1993. Quantifiers and approximation. *Theoret. Comput. Sci. 107*, 145–163.

PAPADIMITRIOU, C. H., AND YANNAKAKIS, M. 1991. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci. 43*, 425–440.

PIN, J.-E. 1986. *Varieties of Formal Languages*. North Oxford, London and Plenum, New York.

PIN, J.-E. 1994. Logic on words. *Bull. EATCS 54*, 145–165.

PIN, J.-E. 1996. Semigroups and automata on words. *Ann. Math. Artif. Int. 16*, 343–384.

ROSEN, E. 1999. An existential fragment of second order logic. *Arch. Math. Logic 38*, 217–234.

SACCA, D. 1997. The expressive powers of stable models for bound and unbound DATALOG queries. *J. Comput. Syst. Sci. 54*, 3, 441–464.

SCHAEFER, T. J. 1978. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on the Theory of Computing* (*STOC '78*) (San Diego, Calif., May 1–3). ACM, New York, pp. 216–226.

SCHLIPF, J. 1995. The expressive powers of logic programming semantics. *J. Comput. Syst. Sci. 51*, 1, 64–86.

SCHWENTICK, T. 1994. Graph connectivity and monadic NP. In *Proceedings of the IEEE Symposium on Foundations of Computer Science* (*FOCS '94*). IEEE Computer Science Press, Los Alamitos, Calif., pp. 614–622.

SCHWENTICK, T. 1995. Graph connectivity, monadic NP and built-in relations of moderate degree. In *Proceedings of the 22nd International Colloquium on Automata, Languages, and Programming*

(*ICALP '95*), Zoltán Fülöp and Ferene Gicseg, Eds. Lecture Notes in Computer Science, vol. 944. Springer-Verlag, New York, pp. 405–416.

STOCKMEYER, L. J. 1977. The polynomial-time hierarchy. *Theoret. Comput. Sci. 3*, 1–22.

STRAUBING, H. 1994. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, Berlin, Boston, Basel.

STRAUBING, H., THERIEN, D., AND THOMAS, W. 1995. Regular languages defined with generalized quantifiers. *Inf. Computat. 118*, 289–301.

THOMAS, W. 1990. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, chapter 4, J. van Leeuwen, ed. Elsevier Science Publishers B.V. (North-Holland).

THOMAS, W. 1996. Languages, automata, and logic. In *Handbook of Formal Language Theory*, vol. III, G. Rozenberg and A. Salomaa, eds. Springer-Verlag, New York, pp. 389–455.

TRAKHTENBROT, B. A. 1961. Finite automata and the logic of monadic predicates. *Dokl. Akad. Nauk SSSR 140*, 326–329.

VARDI, M. Y. 1982. The complexity of relational query languages. In *Proceedings of the 14th ACM Symposium on the Theory of Computing* (*STOC '82*) (San Francisco, Calif., May 5–7). ACM, New York, pp. 137–146.

RECEIVED APRIL 1988; REVISED MARCH 1999; ACCEPTED JUNE 1999