Annals of Pure and Applied Logic 48 (1990) 1–79
North-Holland

A UNIFORM METHOD FOR PROVING LOWER BOUNDS ON THE COMPUTATIONAL COMPLEXITY OF LOGICAL THEORIES

Kevin J. COMPTON*
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

C. WARD HENSON**
Department of Mathematics, University of Illinois, 1409 W. Green St., Urbana, IL 61801, USA

Communicated by Y. Gurevich
Received 16 April 1987; revised 7 December 1989

A new method for obtaining lower bounds on the computational complexity of logical theories is presented. It extends widely used techniques for proving the undecidability of theories by interpreting models of a theory already known to be undecidable. New inseparability results related to the well known inseparability result of Trakhtenbrot and Vaught are the foundation of the method. Their use yields hereditary lower bounds (i.e., bounds which apply uniformly to all subtheories of a theory). By means of interpretations lower bounds can be transferred from one theory to another. Complicated machine codings are replaced by much simpler definability considerations, viz., the kinds of binary relations definable with short formulas on large finite sets.

Numerous examples are given, including new proofs of essentially all previously known lower bounds for theories, and lower bounds for various theories of finite trees, which turn out to be particularly useful.

1. Introduction

In this paper we present a new method for obtaining lower bounds on the computational complexity of logical theories, and give several illustrations of its use. This method is an extension of widely used procedures for proving the recursive undecidability of logical theories. (See Rabin [53] and Eršov, Lavrov, Taimanov, and Taitslin [21].) One important aspect of this method is that it is based on a family of new inseparability results for certain logical problems, closely related to the well known inseparability result of Trakhtenbrot (as refined by Vaught), that no recursive set separates the logically valid sentences from those which are false in some finite model, as long as the underlying language has at least one non-unary relation symbol. By using these

* Research partially supported by NSF Grant DCR 86-05358.
** Research partially supported by NSF Grant DMS 83-02550 and the Deutsche Forschungsgemeinschaft.

inseparability results as a foundation, we are able to obtain *hereditary* lower bounds, i.e., bounds which apply uniformly to all subtheories of the theory.

The second important aspect of this method is that we use interpretations to transfer lower bounds from one theory to another. By doing this we eliminate the need to code machine computations into the models of the theory being studied. (The coding of computations is done once and for all in proving the inseparability results.) By using interpretations, attention is centered on much simpler definability considerations, viz., what kinds of binary relations on large finite sets can be defined using short formulas in models of the theory. This is conceptually much simpler than other approaches that have been proposed for obtaining lower bounds, such as the method of bounded concatenations of Fleischmann, Mahr, and Siefkes [27].

We will deal primarily with theories in first-order logic and monadic second-order logic. Given a set Σ of sentences in a logic L, we will consider the *satisfiability problem*

$$sat(\Sigma) = \{\sigma \in L \mid \sigma \text{ is true in some model of } \Sigma\}$$

and the *validity problem*

$$val(\Sigma) = \{\sigma \in L \mid \sigma \text{ is true in all models of } \Sigma\}.$$

A *hereditary* lower bound for Σ is a bound that holds for $sat(\Sigma')$ and $val(\Sigma')$ whenever $\Sigma' \subseteq val(\Sigma)$. If L is a first-order logic, define $inv(L)$ to be the set of sentences in L that are logically invalid, i.e., false in all models. If L is a monadic second-order logic, define $inv(L)$ to be the set of sentences false in all *weak models*. (See Section 2 definitions.)

The complexity classes used here are time-bounded classes for nondeterministic Turing machines and for the more general class of linear alternating Turing machines. In providing reductions between different decision problems, we are always able to give log-lin reductions. That is, our reduction functions can be computed by a deterministic Turing machine which operates simultaneously in log space and linear time. In particular, such functions have the property that the size of a value is bounded uniformly by a constant multiple of the size of the argument.

Let L_0 denote the first-order logic with a single, binary relation symbol. Let ML_0 denote the corresponding monadic second-order logic. Let $T(n)$ be a time resource bound which grows at least exponentially in the sense that there exists a constant d, $0<d<1$, such that $T(dn)/T(n)$ tends to 0 as n tends to ∞. (This condition is satisfied by the iterated exponential functions and other time resource bounds which arise most commonly in connection with the computational complexity of logical theories.) Let $sat_T(L_0)$ denote the set of sentences σ in L_0 such that σ is true in some model on a set of size at most $T(|\sigma|)$. (Here $|\sigma|$ denotes the length of σ.) Similarly define $sat_T(ML_0)$ for sentences of monadic second-order logic. The inseparability results which form the cornerstone of our method are as follows:

(a) For some $c > 0$, $sat_T(L_0)$ and $inv(L_0)$ cannot be separated by any set in $NTIME(T(cn))$.

(b) For some $c > 0$, $sat_T(ML_0)$ and $inv(ML_0)$ cannot be separated by any set in $ATIME(T(cn), cn)$.

($ATIME(T(cn), cn)$ denotes the set of problems recognized by alternating Turing machines in time $T(cn)$ making cn alternations along any branch.) In proving (b) we prove another interesting result. Let ML_+ be a monadic second-order logic with a ternary relation $PLUS$ and $sat_T(M\Sigma_+)$ be the set of sentences φ in this logic true in the model $\langle T(n), +\rangle$, where $n = |\varphi|$ and $+$ is the usual ternary addition relation on the set $T(n) = \{0, \ldots, T(n) - 1\}$. Then for some $c > 0$, $sat_T(M\Sigma_+)$ and $inv(ML_+)$ cannot be separated by any set in $ATIME(T(cn), cn)$.

We prove and discuss these results in Sections 3 and 4 respectively, In fact, we prove more: any problem separating $sat_T(L_0)$ and $inv(L_0)$ is a hard problem (under log-lin reductions) for the complexity class

$$\bigcup_{c>0} NTIME(T(cn)).$$

Any problem separating $sat_T(ML_0)$ and $inv(ML_0)$ is a hard problem for the complexity class

$$\bigcup_{c>0} ATIME(T(cn), cn).$$

In these results one can see a parallel between first-order logic and *NTIME*, on the one hand, and monadic second-order logic and linear alternating time, on the other. This parallel persists throughout the lower bounds for logical theories which we discuss here, and we feel that our point of view helps to explain why the complexity of some theories is best measured using *NTIME* while for others the best measure is linear alternating time.

In order to obtain lower bounds from these inseparability results, or to transfer lower bounds from one theory to another, we use interpretations. However, sometimes we require not just a single interpretation, but rather a sequence $\{I_n \mid n \geqslant 0\}$ of such interpretations. Not only do we require that each I_n define a sufficiently rich class of models when applied to the models of the theory under study, but also we require that the function taking n (in unary) to I_n should be log-lin computable.

As an example of how such interpretations are used, suppose Σ is a theory such that for each $n \geqslant 0$, I_n applied to models of Σ yields all the binary relations of size at most $T(n)$ (and perhaps others), for a given time resource bound T. It follows that for some constant $c > 0$, $sat(\Sigma)$ and $inv(L)$ cannot be separated by any set in $NTIME(cn))$. In general, it follows that Σ has a hereditary $NTIME(T(cn))$ lower bound. There is a corresponding result for the complexity classes

$ATIME(T(cn), cn)$ when the interpretations I_n interpret monadic second-order logic of binary relations on sets of size at most $T((n)$.

Upon establishing a lower bound for $sat(\Sigma)$ in this manner, we may then use interpretations of Σ to obtain lower bounds for other theories. Continuing the example above, assume I_n applied to the set of models of Σ yields all the binary relations of size at most $T(n)$. In fact, it will not be necessary to apply I_n to all models of Σ to obtain all binary relations of size at most $T(n)$, since there are only finitely many such binary relations. Suppose that $\mathscr{C}_n$ is a set of models of Σ such that I_n applied to $\mathscr{C}_n$ yields all the binary relations of size at most $T(n)$. If $\{I'_n \mid n \geq 0\}$ is another log-lin computable sequence of interpretations such that I'_n applied to models of a theory Σ' in a language L' yields all models of $\mathscr{C}_n$ (and perhaps others, some possibly not even models of Σ), then for some constant $c > 0$, $sat(\Sigma')$ and $inv(L')$ cannot be separated by any set in $NTIME(T(cn))$. Thus, we have developed a theory for establishing lower bounds of logical theories analogous to the theory for establishing NP-hardness results via polynomial time reductions.

The observation that I'_n applied to models of Σ' is allowed to yield models not satisfying Σ may seem unimportant, but in practice it results in significant simplifications in interpretations, especially compared to the method of establishing lower bounds by Turing machine encodings. There one must produce, for a nondeterministic Turing machine M with the appropriate running time, an efficient reduction from strings w to sentences φ_w in such a way that when M accepts w, φ_w is true in some model of Σ', *and* when M does not accept w, φ_w is true in no model of Σ'. Ensuring that a sentence is true in no model of Σ' in the case of nonacceptance can be cumbersome. Inseparability considerations eliminate the need for it.

Our method can give short, transparent lower bound results in many cases where Turing machine encodings are far from apparent. An example is the result of Compton, Henson, and Shelah [16] that the theory of almost all finite unary functions is not elementary recursive. The proof there, using the methods of this paper, is set forth in a short paragraph. A proof by Turing machine encodings would run to many pages, and possibly would never have been discovered at all. It seems likely that our methods will have to be used if sharp lower bounds are to be obtained for some of the important, more algebraic (and less directly combinatorial) theories which are known to be decidable. (See for example Problems 10.1, 10.2, 10.7, 10.9, 10.10.)

In making use of sequential families of interpretations there are certain technicalities regarding lengths of formulas which must be addressed. They can be illustrated by considering the formula φ' which results from a formula φ when one replaces every occurrence of a certain binary relation symbol P by a given formula ψ. If φ has many occurrences of P and if the length of ψ is of the same order of magnitude as the length of φ, then φ' may well be extremely long compared to φ. (This is precisely the kind of operation on formulas used as a reduction function between theories when one uses interpretations to obtain lower bounds.) This difficulty can be overcome if one uses sequences $\{I_n\}$ of interpretations in which either all formulas in each I_n are in prenex form, or are obtained by a certain kind of iterative process. In practice, the interpretation sequences used to transfer lower bounds from one logical problem to another can always be found satisfying one of these conditions.

We have used this approach to give a precise analysis of the computational complexity of various theories of finite trees. For each r let Σ_r denote the first-order theory of all finite trees of height r, and let $M\Sigma_r$ denote the corresponding monadic second-order theory. Also let Σ_∞ and $M\Sigma_\infty$ denote the corresponding theories of all finite trees. Let $\exp_m(n)$ be the m-times iterated exponential function (e.g., $\exp_2(n) = 2^{2^n}$) and let $\exp_\infty(n)$ be the tower of two function:

$$\exp_\infty(n) = \exp_n(1) = \left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} n \text{ times}.$$

Our results concerning the various theories of finite trees can be summarized as follows:

(a) For each $r \geqslant 4$ there are constants c and $d > 0$ such that $sat(\Sigma_r)$ is in

$$NTIME(\exp_{r-2}(dn))$$

but that $sat(\Sigma_r)$ and $val(\Sigma_r)$ are hereditarily not in

$$NTIME(\exp_{r-2}(cn)).$$

For $r = 3$ the upper bound is $NTIME(2^{dn^2})$ and the hereditary lower bound is $NTIME(2^{cn})$.

(b) There exist constants c and $d > 0$ such that $sat(\Sigma_\infty)$ is in

$$NTIME(\exp_\infty(dn))$$

but that $sat(\Sigma_\infty)$ and $val(\Sigma_\infty)$ are hereditarily not in

$$NTIME(\exp_\infty(cn)).$$

Hence, these problems are hereditarily not elementary recursive.

(c) For each $r \geqslant 1$ there are constants c and $d > 0$ such that $sat(M\Sigma_r)$ is in

$$ATIME(\exp_r(dn/\log n), dn)$$

but that $sat(M\Sigma_r)$ and $val(M\Sigma_r)$ are hereditarily not in

$$ATIME(\exp_r(cn/\log n), cn).$$

It is not hard to show that Σ_∞ and $M\Sigma_\infty$ are mutually interpretable, and hence have the same complexity. In any case, for such rapidly growing time resource bounds as $\exp_\infty(n)$, the difference between *NTIME* and *ATIME* has vanished.

In order to test our method for effectiveness and smoothness of use, we have used it to provide new proofs of essentially all previously known complexity lower bounds for first-order and monadic second-order theories. (We do not consider lower bounds for sentences with restricted quantification prefixes, as in Lewis [43] and Scarpellini [62]; nor do we consider lower bounds for nonclassical logics such as temporal logics, dynamic logics, logics of knowledge, and other modal logics. Many of these logics have decision problems complete in *deterministic* time classes. See, for example, Fischer and Ladner [25], Emerson and Halpern [18], and Halpern and Vardi [35].)

These new arguments avoid direct coding of machine computations, and are usually much simpler and more conceptual than the original arguments. We present many of these proofs, or at least sketches of them, in Section 8. In all cases our lower bounds are hereditary, and are expressed in terms of log-lin hardness for certain complexity classes, providing new complete problems for many of these *NTIME* and linear *ATIME* classes. In some cases we verify results which had been only announced, no published proof ever having appeared.

It is our hope that this systematic reorganization and simplification of the subject will stimulate the interests of many computer scientists and mathematicians, and they they will be inspired to investigate the many decidable theories for which no detailed complexity bounds have been found.

The organization of our paper is as follows: Section 2 contains various definitions and technical conventions. In Section 3 we present some technical machinery needed to handle the details about lengths of formulas which arise in complexity arguments. Sections 4 and contain the basic inseparability results for logical problems; these are the analogoues of the Trakhtenbrot–Vaught Theorem. In Sections 6 and 7 we discuss interpretations and set up the procedures by which they are used to obtain lower bounds for logical theories. Here too are proved the lower bounds for the various theories of finite trees which are treated here. Section 8 contains a lengthy series of applications of our method, yielding lower bounds for a wide variety of theories of independent interest. In Section 9 we obtain various upper bounds for problems treated here, in order to show that our method is capable of achieving sharp results. We present a selected list of open problems at the end of Section 10.

In this paper we have not discussed ways in which our method can be used to obtain lower bounds in terms of $SPACE(T(n))$ complexity classes. This is because there are so few known cases in which best possible lower bounds for logical theories are expressed in terms of space complexity classes. The exceptions are the *PSPACE*-complete theories such as those discussed in Stockmeyer [69] and Grandjean [31]. These are, in some sense, the least complex theories since Stockmeyer shows implicitly that if Σ is a logical theory with has at least one model with at least one nontrivial definable relation (i.e., the relation is true of some elements and false of others), then $sat(\Sigma)$ is log space, polynomial time hard for *PSPACE*. (Note that if equality is taken as a basic relation in the language, or is definable, then the hypothesis means simply that Σ has at least one model with two or more elements.) If Σ does not have any such model, then it is utterly trivial, and both $sat(\Sigma)$ and $val(\Sigma)$ are *LOGSPACE*. (We assume throughout that the vocabularies of logics are finite.) Possibly methods of this paper could be extended to obtain polynomial nondeterminstic time lower bounds for *PSPACE*-complete theories.

On the other hand, all 'natural' logical theories which are known to be decidable seem actually to be primitive recursive. Furthermore, among theories where a somewhat careful upper bound analysis has been carried out, decidable logical theories seem to fall into $NTIME(\exp_\infty(dn))$ for some constant d in all but a few cases. It would be nice to have an explanation (or a convincing refutation) of this phenomenon.

Reader, please do not be dismayed by the length of this paper. We believe that the method presented here is simple and can be mastered quickly. To get an overview of the applications of our method we advise looking over the results in Section 8 first. Not only does this give a summary of the main complexity lower bounds now known, but also we have tried to present these applications in such a way as to give an accessible exposition of how our methods are meant to be used, and the main technical points which arise in their use.

We would like to thank the referee for his extraordinarily careful reading of this paper and helpful comments.

2. Preliminaries

In this section we present the definitions and notations used throughout the paper.

All alphabets considered will be finite. The length of a string w is denoted $|w|$. The empty string is denoted ε.

We use the standard 'big oh' and 'little oh' notations throughout, as well as the 'big omega' notation: write $f(n) = \Omega(g(n))$ if $f(n) \geqslant kg(n)$ for some $k > 0$ on all large n.

All theories considered here either first-order or monadic second-order. For convenience we explicitly treat only relational languages in Sections 3–7; functions are handled by using their graphs as relations, and constants are treated as special unary relations. This restriction makes no difference as far as the lower bounds we obtain: sentences containing function and constant symbols can be transformed into equivalent sentences containing relation symbols with an increase in length of only a constant factor. (It is necessary to 'reuse' variables to accomplish this.)

We will assume for convenience that our languages contain equality. However, in many situations our methods work even without equality. We use equality mainly to keep formulas short while substituting other formulas for atomic formulas: equations are used for coding truth values. This can usually be done by other formulas, as long as they satisfy the right conditions of nontriviality. If one is willing to accept polynomial-time reductions, then equality is generally not necessary, but the lower bounds degrade slightly.

To specify a logic in this paper, we need only give its set of relation symbols and their associated arities (the *vocabulary* of the logic) and indicate whether the logic is first-order or monadic second-order. We formulate all of our logics using finitely many symbols, so that all terms and formulas are strings on a finite alphabet. In particular, a variable is represented by a symbol followed by a subscript in binary notation. Thus, to represent n distinct variables we need strings with total length about $n \log n$. (Logarithms will always have base 2.) To avoid subscripted subscripts we use lower case Latin letters t, u, x, y, z—possibly with subscripts—as *formal variables* to denote *actual variables* v_i. Monadic variables are represented by corresponding upper case letters.

The *power* of a model is the cardinality of its universe.

A *weak model* for a monadic second-order logic L is a pair $\langle \mathfrak{A}, \mathcal{F} \rangle$, where $\mathfrak{A}$ is a model for L and $\mathcal{F}$ is a collection of subsets of the universe of $\mathfrak{A}$. The truth value of a formula from L in $\langle \mathfrak{A}, \mathcal{F} \rangle$ is determined in the usual way except that monadic quantifiers range over the sets in $\mathcal{F}$ rather than the collection of all sets. *Throughout the paper, equivalence of monadic second-order formulas will mean equivalence on weak models.* This is stronger than the usual notion of equivalence.

The first-order and monadic second-order logics with vocabulary consisting just of a binary relation symbol P are central to our investigation. They will be denoted L_0 and ML_0 respectively.

We also study theories of finite trees; again the vocabulary consists just of one binary relation symbol which, in this case, interprets the successor (or parent-child) relation. Let L_t and ML_t denote the first-order and monadic second-order logics with this vocabulary. These are essentially the same as L_0 and ML_0; they differ only in the binary relation symbol used. However, it will be convenient to have a different notation for these logics when considering trees.

When considering finite trees we will often require the notion of a *primary subtree*. Such a subtree is formed by restricting to a set of vertices consisting of a child of the root and all its descendents. Thus, we may regard a tree as being formed by directing an edge from the root of the tree to the root of each of its primary subtrees.

The *depth* of a vertex in a tree is its distance from the root. The *height* of a vertex is the maximum distance to a leaf below it. Thus, the height of a tree is the maximum depth its vertices, which is also the height of the root.

As we noted in the introduction, we will consider problems of the form $sat(\Sigma)$ and $val(\Sigma)$. From a computational point of view, these two problems are complementary. That is, a sentence σ is in $sat(\Sigma)$ exactly when $\neg\sigma$ is *not* in $val(\Sigma)$. Hence, $sat(\Sigma)$ is a member of a particular complexity class if and only if $val(\Sigma)$ is a member of the corresponding co-complexity class. If Σ is a complete theory, then $sat(\Sigma) = val(\Sigma)$. When we are in a first-order logic, $val(\Sigma)$ is the deductive closure of Σ by the Gödel Completeness Theorem. There is no corresponding result for monadic second-order logic.

Often a logical theory is specified not by giving a set of axioms Σ, but by giving a class of models $\mathscr{C}$. In this situation we take Σ to be the set of sentences true in all members of $\mathscr{C}$. It is easy to verify in this case that $val(\Sigma) = \Sigma$ and $sat(\Sigma)$ is the set of sentences true in some member of $\mathscr{C}$.

If L is a first-order logic we define $inv(L)$ to be the set of sentences false in all models for L. This is just the complement of $sat(\emptyset)$. If L is a monadic second-order logic we define $inv(L)$ to be the set of sentences false in all weak models for L.

Given a time resource bound $T(n)$, let $sat_T(\Sigma)$ be the set of sentences φ true in some model of Σ of size at most $T(|\varphi|)$. Also, write $sat_T(L)$ for $sat_T(\emptyset)$. Let $sat_T^p(\Sigma)$ be the set of prenex sentences φ true in some model of Σ of size at most $T(|\varphi|)$.

Let $\mathfrak{A}$ be a model for a logic L, $\boldsymbol{m} = m_1, \ldots, m_k$ elements of $\mathfrak{A}$, and $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_k)$ a formula from L. Then $\varphi^{\mathfrak{A}}(\boldsymbol{x}, \boldsymbol{m})$ denotes the n-ary relation defined by

$$\varphi^{\mathfrak{A}}(\boldsymbol{a}, \boldsymbol{m}) \quad \Leftrightarrow \quad \mathfrak{A} \models \varphi[\boldsymbol{a}, \boldsymbol{m}]$$

for $\boldsymbol{a} = a_1, \ldots, a_n \in M$.

Interpretations of one class of models in another are fundamental to many parts of logic, and we use them extensively here. For example, to interpret a binary relation $\mathfrak{A}'$ (i.e., a model for the logic L_0) in a theory Σ from a logic L, we must produce formulas $\delta(x, \boldsymbol{u})$ and $\pi(x, y, \boldsymbol{u})$ from L so that for some model $\mathfrak{A}$ of Σ and some elements $\boldsymbol{m}$ of $\mathfrak{A}$, $\mathfrak{A}'$ is isomorphic to the structure

$$\mathfrak{B} = \langle \delta^{\mathfrak{A}}(x, \boldsymbol{m}), \pi^{\mathfrak{A}}(x, y, \boldsymbol{m}) \rangle$$

where we require that $\pi^{\mathfrak{A}}(x, y, \boldsymbol{m}) \subseteq \delta^{\mathfrak{A}}(x, \boldsymbol{m}) \times \delta^{\mathfrak{A}}(x, \boldsymbol{m})$. There is also a more general kind of interpretation that is often used, in which the domain of $\mathfrak{B}$ can be a set of k-tuples from $\mathfrak{A}$ (not just elements of $\mathfrak{A}$) and in which $\mathfrak{A}'$ is isomorphic to a quotient of $\mathfrak{B}$ by an equivalence relation definable in $\mathfrak{A}$.

Let φ be a formula in a logic L and D a unary relation symbol. By φ^D we mean the *relativization* of φ to D. This is formed by systematically replacing all subformulas $\forall y\, \psi$ and $\exists y\, \psi$ of φ with $\forall y\, (D(y) \rightarrow \psi)$ and $\exists y\, (D(y) \wedge \psi)$, respectively. If L is a monadic second-order logic, it is not necessary to relativize the set quantifiers since elements have already been restricted to D.

The complexity classes we use are defined by time resource bounds. A *time resource bound* T is a mapping from the nonnegative reals to the nonnegative reals such that for each $k > 0$, $T(kn)$ is dominated by some fully time constructible function on the integers; see Hopcroft and Ullman [37] for definitions. (Readers who need a primer in complexity theory may also wish to consult a recent survey article by Stockmeyer [70].) We will also require that

$T(n) \geqslant n$ and that for each $k \geqslant 1$, $T(kn) \geqslant k\ T(n)$. This last condition Machtey and Young [46] call *at least linear*. It says that when input length is increased by some factor, the allowed computation time increases by at least the same factor. It is included for technical reasons; we could get by with less, namely, that T be nondecreasing and for every l there should be a k such that $T(kn) \geqslant l\ T(n)$.

The iterated exponentials and tower of twos functions appear often as time resource bounds in the problems we consider. The iterated exponentials $\exp_n(n)$, where m is a non-negative integer, are defined by induction on m. Let $\exp_0(n) = n$ and $\exp_{m+1}(n) = 2^{\exp_m(n)}$. The tower of twos function $\exp_\infty(n)$ is defined to be

$$\exp_n(1) = \left.2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}\right\}{\scriptstyle n \text{ times}}.$$

Recall that a problem is *elementary recursive* if it is recognized in time $\exp_m(n)$ for some $m \geqslant 0$.

All of our bounds are for nondeterministic or alternating Turing machines. The set of problems recognized by nondeterministic Turing machines in time $T(n)$ is denoted $NTIME(T(n))$. With alternating Turing machines we will be concerned chiefly with the complexity classes $ATIME(T(n), cn)$, the set of problems recognized by an alternating Turing machine in time $T(n)$ making at most cn alternations. We will assume that alternating Turing machines have four types of states: universal, existential, accepting and rejecting. See Chandra, Kozen, and Stockmeyer [14] for the definition of acceptance by alternating Turing machines and a description of the computation trees associated with these machines.

We will sometimes say that a theory Σ has a hereditary $NTIME(T(cn))$ (or $ATIME(T(cn), cn)$) lower bound. By this we mean that there is a $c > 0$ such that for all $\Sigma' \subseteq \Sigma$, neither $sat(\Sigma')$ nor $val(\Sigma')$ is in $NTIME(T(cn))$ (respectively, $ATIME(T(cn), cn)$).

A log-lin reduction is a mapping computable in log space and linear time. In some sources this terminology is used for a log space computable, linearly bounded mapping, which is a weaker notion. (*Linearly bounded* means that output length is less than some constant multiple of input length.) It is not crucial for the applications presented here that our reductions be quite so restricted: polynomial time, linearly bounded reductions suffice. However, to obtain some results in the literature, such as the nondeterministic polynomial lower bounds in Grandjean [31], linear time reductions would be needed.

We encounter a technical problem with log-lin reductions: we do not know if they are closed under composition. To overcome this difficulty we define a stronger notion of *reset log-lin reduction*. A machine performing such reduction is a log space, linear time bounded Turing machine with work tapes, an input tape, and an output tape. It has the capability to reset the input tape head to the initial input cell on k moves during a computation, where k is fixed for all inputs; on all other moves the input tape head remains in place or moves one cell to the right.

It writes the output sequentially from left to right. Suppose that M' and M'' are two such machines using at most k' and k'' resets, respectively. We informally describe a machine M to compute the composition of the reductions computed by M' and M''. Imagine that the output tape of M' and the input tape of M'' have been removed. Instead, M' sends its output directly to M''. As M'' computes its output, it calls M' to supply it with a new symbol on those moves when the input head of M'' would have moved right. M' has only to resume its computation from the last call to supply this symbol. On those moves where M'' would have reset its input head, M' must begin its computation anew. Now the input head of M'' would have passed over each input cell at most $k''+1$ times during the computation, and to supply each symbol the input head of M' passes over each input cell at most $k'+1$ times. Thus, M resets its input head at most $(k'+1)(k''+1)-1$ times. Clearly, M is log space bounded. Since the part of M corresponding to M' is forced to begin its computation anew at most k'' times, it is easy to see that M is linear time bounded.

It is not difficult to show that the prenex formulas of a logic are closed under relativization up to reset log-lin reductions. That is, there is a reset log-lin reduction which takes formulas of the form φ^D, where φ is a prenex formula with no variable quantified more than once, to equivalent prenex formulas. We use this fact often. Unfortunately, we know of no way to eliminate duplicate quantifications of variables using reset log-lin reductions, but this can be accomplished easily with polynomial time, linearly bounded reductions.

A problem Σ is *hard* for a complexity class $\mathscr{C}$ via reductions from a class $\mathscr{S}$ if every problem $\Sigma' \in \mathscr{C}$ can be reduced to Σ by some $f \in \mathscr{S}$. That is, if A and A' are the alphabets for Σ and Σ' respectively, then f maps A'^* to A^* so that $w \in \Sigma'$ if and only if $f(w) \in \Sigma$. If, in addition, $\Sigma \in \mathscr{C}$, we say that Σ is *complete* for $\mathscr{C}$ via reductions from $\mathscr{S}$.

3. Reductions between formulas

One of our goals is to develop effective and easily used methods for transferring lower bounds from one problem to another. Our methods are based on interpretations between theories (or equivalently, between classes of models) and can be seen as an extension of the most widely used methods for proving the undecidability of logical theories; see Eršov, Lavrov, Taimanov, and Taitslin [21] and Rabin [53] for a discussion of undecidable theories from this point of view. To obtain complexity lower bounds for decidable theories we must use interpretations which have a somewhat more general form than those used in undecidability proofs, and there are certain technicalities about lengths of formulas which must be addressed in this more general setting. In this section we will develop the required machinery. The first-time reader may wish to skip the proofs in this section as they are somewhat tedious and only the statements of results will be used later.

A common method for proving that a theory Σ in a logic L is undecidable is to show that the theory Σ_0 of finite binary relations, formulated in the logic L_0, can be interpreted in Σ. In the simplest case this means that formulas $\delta(x, \boldsymbol{u})$ and $\pi(x, y, \boldsymbol{u})$ of L are given so that every finite binary relation can be obtained (up to isomorphism) in the form

$$\langle \delta^{\mathfrak{A}}(x, \boldsymbol{m}), \pi^{\mathfrak{A}}(x, y, \boldsymbol{m})\rangle$$

for $\mathfrak{A}$ a model of Σ and $\boldsymbol{m}$ a sequence of elements of $\mathfrak{A}$. The formulas δ and π are then used to define a reduction from formulas of L_0 to formulas of L, as follows: given a formula φ of L_0, replace every occurrence of an atomic formula $P(z, t)$ by the formula $\pi(z, t, \boldsymbol{u})$ and relativize every quantifier to the formula δ. (One must rewrite bound variables to avoid conflicts and make sure that $\boldsymbol{u}$ is a sequence of otherwise unused variables.) Call the resulting formula φ'. The reduction mapping $\varphi \mapsto \varphi'$ is then used to obtain undecidability results for Σ from corresponding results for Σ_0.

This kind of simple interpretation is not adequate for obtaining lower complexity bounds when Σ is a decidable theory. One works instead with a parameterized family of formulas $\{\varphi_n \mid n \geqslant 0\}$ from L_0 and uses a sequence of formula pairs $\{(\delta_n(x, \boldsymbol{u}), \pi_n(x, y, \boldsymbol{u})) \mid n \geqslant 0\}$ from L. In reducing the formulas φ_n to L, one proceeds as above, except that φ_n' is obtained from φ_n using δ_n and π_n. In complexity lower bound arguments, it is not only necessary that the function $\varphi_n \mapsto \varphi_n'$ should be efficiently computable, but also that $|\varphi_n'|$ should be linearly bounded in $|\varphi_n|$. If P occurs many times in φ_n and $|\pi_n|$ grows without bound as n increases, or if φ_n has many quantifiers and $|\delta_n|$ grows without bound as n increases, then the linear boundedness condition may not hold. However, in certain cases there are methods to efficiently replace φ_n' by an equivalent formula for which the linear boundedness condition does hold. Roughly speaking, we can do this when the formulas δ_n and π_n are all in prenex form, or are obtained by a certain kind of iterative procedure. The machinery developed here to accomplish this task is implicit in most complexity lower bound arguments for logical problems.

In order to describe this machinery, it is convenient to introduce an extension L^* of each logic L, in which explicit definitions are allowed. (L^* has no more expressive power than L, but properties can sometimes be expressed by shorter formulas in L^* than in L.) Continuing the example above, let φ_n^D denote a formula in which all quantifiers of φ_n have been relativized to a new unary relation symbol D. Then the extended lanugage L^* in this case would include a formula

$$[P(x, y) \equiv \pi_n(x, y, \boldsymbol{u})]\,[D(z) \equiv \delta_n(z, \boldsymbol{u})]\,\varphi_n^D$$

whose interpretation is exactly the same as that of φ_n', although its length is likely to be more under control. Here the equivalences in brackets are interpreted to mean that P is explictly defined by π_n and that D is explicitly defined by δ_n. The general problem, treated below in this section, is to find situations in which certain formulas of the extended language L^* can be efficiently reduced to equivalent formulas of L, *without a significant increase in the length of the formulas.* (In general, it is possible to find for each L^* formula of length n an equivalent L formula of length $O(n \log n)$; this is not good enough for sharp complexity bounds.)

Let L be either a first-order or monadic second-order logic. Define L^* as follows. Formulas of L^* may contain any of the symbols occurring in formulas of L and, in addition, *relation variables* S_i^j for each $i, j \geq 0$. In each case the arity of S_i^j is j and the subscript and superscript of S_i^j are expressed in binary notation. (If L is a monadic second-order logic we need two superscripts, the first denoting the arity of element arguments and the second denoting the arity of set arguments.) Subscripts and superscripts of relation variables contribute to the length of formulas in which they occur, just as element variable subscripts do. (However, superscripts may be ignored in asymptotic estimates of formula length because they are dominated in length by their corresponding arguments lists.) We define the set of formulas φ of L^* inductively, and at the same time define *free*(φ), the set of free variables in φ. An *atomic formula* φ of L^* is either an atomic formula of L or a formula $P(x_1, \ldots, x_j)$ where P denotes a relation variable—in the former *free*(φ) is the same as in L; in the latter, *free*$(\varphi) = \{P, x_1, \ldots, x_j\}$. More complex formulas φ may be constructed using the logical connectives and quantifiers appropriate to L; in these cases *free*(φ) is defined just as in L. The only other way to construct more complex formulas is by *explicit definition.* Let ψ and θ be formulas in L^*, P a relation variable which does not occur freely in θ, and $\boldsymbol{x} = x_1, \ldots, x_j$ a sequence of distinct element variables. Then φ given by

$$[P(\boldsymbol{x}) \equiv \theta]\ \psi$$

is also a formula of L^* and

$$\mathit{free}(\varphi) = ((\mathit{free}(\theta) - \{x_1, \ldots, x_j\}) \cup \mathit{free}(\psi)) - \{P\}.$$

The part of φ within brackets is an *explicit definition* which defines the interpretation of P in ψ. If θ is a prenex formula from L we will say that it is a *prenex definition.* The truth value of φ is the same as that of the second-order expresssion

$$\forall P\ ((\forall \boldsymbol{x}\ (P(\boldsymbol{x}) \leftrightarrow \theta)) \rightarrow \psi).$$

Notice that the truth value is consistent with the definition of *free*(φ). Notice also that the second-order expression above is equivalent to

$$\exists P\ ((\forall \boldsymbol{x}\ (P(\boldsymbol{x}) \leftrightarrow \theta)) \wedge \psi)$$

so that $\neg[P(\boldsymbol{x}) \equiv \theta]\ \psi$ is equivalent to $[P(\boldsymbol{x}) \equiv \theta] \neg \psi$. If *free*$(\varphi) = \emptyset$, then φ is a *sentence* of L^*.

We will let $sat^*(\Sigma)$ denote the set of sentences from L^* true in some model of Σ, $sat^*_T(\Sigma)$ denote the set of sentences φ from L^* true in some model of Σ of size at most $T(|\varphi|)$, $sat^*_T(L)$ denote the set of sentences φ from L^* true in some model of size at most $T(|\varphi|)$, and $inv^*(L)$ denote the set of sentences from L^* true in no model (or no weak model when L is a monadic second-order logic).

Introduction of explicitly defined relations is standard practice in mathematical discourse. Explicitly defined relations are also similar to nonrecursive procedures in programming languages.

Explicit definitions can be used to define reductions between satisfiability problems. To provide good lower bounds these reductions must be efficiently computable and linearly bounded. We will show, in fact, that there are reset log-lin reductions, defined on certain subsets of sentences from L^*, that take formulas to equivalent formulas in L. (Unfortunately, such reductions probably cannot be defined on the set of all sentences in L^*; with a little effort we can produce a polynomial time reduction which maps sentences in L^* of length n to equivalent sentences in L of length $O(n \log n)$.)

We inductively define positive and negative occurrences of a relation symbols Q in formulas from L^*. Q occurs positively in atomic formulas of the form $Q(\boldsymbol{x})$. Q occurs positively (negatively) in the formulas $\varphi \wedge \psi$ and $\varphi \vee \psi$ when it occurs positively (negatively) in either of the formulas φ or ψ. Q occurs positively (negatively) in the formula $\neg\varphi$ if it occurs negatively (positively) in the formula φ. Q occurs positively (negatively) in the formulas $\forall x\, \varphi$ and $\exists x\, \varphi$ if it occurs positively (negatively) in the formula φ. Q occurs positively (negatively) in the formula $[P(\boldsymbol{x}) \equiv \theta]\, \psi$, where P is not Q, if it occurs positively (negatively) in ψ, or if it occurs positively (negatively) in θ and P occurs positively in ψ, or if it occurs negatively (positively) in θ and P occurs negatively in ψ. We say that P occurs *only positively* in a formula if it does not occur negatively (in particular, it may not occur at all).

We inductively define an *iterative definition* $[P(\boldsymbol{x}) \equiv \theta]_n$ as follows. The iterative definition $[P(\boldsymbol{x}) \equiv \theta]_0$ is equivalent to the explicit definition $[P(\boldsymbol{x}) \equiv \iota]$, where ι is a sentence false in all models (or all weak models if L is a monadic second-order logic). The iterative definition $[P(\boldsymbol{x}) \equiv \theta]_{n+1}$ is equivalent to

$$[P(\boldsymbol{x}) \equiv [P(\boldsymbol{x}) \equiv \theta]_n\, \theta].$$

We make iterative definitions part of the syntax of L^*, but we require that the subscript n be written in unary notation so that the length of an iterative definition is of the same order ($O(n)$ and $\Omega(n)$) as the length of the nested explicit definitions it replaces. We call θ the *operator formula* for the iterative definition.

We can think of iterative definitions as approximations to *implicit definitions*. We will not formally define implicit definitions, since they do not figure directly in what follows, but an example should convey the idea. (See Moschovakis [51] for an account.) Consider a language containing just a binary relation symbol E

denoting the edge relation on graphs. The implicit definition

$$[P(x, y) \equiv (x = y \vee \exists z\ (P(x, z) \wedge E(z, y)))]$$

defines the path relation in each graph: $P(x, y)$ is the least relation satisfying the equivalence, so it holds precisely when there is a path between x and y. Now consider the related iterative defintion

$$[P(x, y) \equiv (x = y \vee \exists z\ (P(x, z) \wedge E(z, y)))]_n$$

which defines a relation $P(x, y)$ which holds precisely when the distance between x and y is at most $n - 1$ (when $n \geq 1$). Notice that this 'approximation' to the implicitly defined relation does not converge very rapidly. The iterative definition

$$[P(x, y) \equiv (x = y \vee E(x, y) \vee \exists z\ (P(x, z) \wedge P(z, y)))]_n$$

defines a relation $P(x, y)$ which holds precisely when the distance between x and y is at most 2^{n-1} (for $n \geq 1$), so this approximation to the path relation converges exponentially 'faster'. For an implicit definition to make sense, $\theta = \theta(P)$ should be monotone in P (i.e., for every structure $\mathfrak{A}$, if P and P' are relations on $\mathfrak{A}$ with $P \subseteq P'$, then $\theta^{\mathfrak{A}}(P) \subseteq \theta^{\mathfrak{A}}(P')$). Monotonicity can be guaranteed by requiring that P is positive in θ. No such restriction is needed for iterative definitions. In most of our applications P does occur positively and the iterative definitions approximate an implicit definition. Usually, the faster the convergence, the better the lower bounds obtained by our methods. We will see that the positivity of P in θ does have implications in lower bound results.

To show that we can efficiently transform iterative definitions into equivalent explicit definitions, we require the following theorem, which will also be used to show that certain sets of formulas in L^* can be efficiently transformed into equivalent formulas from L.

Theorem 3.1. *Let L be a first-order or monadic second-order logic and let L' be a logic, of the same type, whose vocabulary consists of the vocabulary of L together with relation symbols $P_1, \ldots, P_m$. There is a reset log-lin reduction taking each prenex formula of L' to an equivalent prenex formula of L' having at most one occurrence of each P_j.*

Proof. The proof follows an argument of Ferrante and Rackoff [24, pp. 155–157]. We must add some details, however, because they were not interested in obtaining a reset log-lin reduction. We adopt the same assumption they did there: we assume that L has a symbol for equality and that all structures have cardinality at least 2. We could dispense with this assumption at the cost of added complications.

We deal explicitly only with the case $m = 1$. It will be clear from the proof that the procedure can be iterated to treat $P_1, P_2, \ldots$ in succession.

We describe the action of our algorithm on φ, a prenex formula from L. First add a 0 bit to the end of every variable index occurring in φ. this will allow us to introduce variables of odd index without creating a conflict. Now φ is of the form

$$(Q_1x_1)(Q_2x_2)\cdots(Q_nx_n)\,\psi$$

where each Q_i is a quantifier and ψ is quantifier free. Let

$$P_1(x_{11},\ldots,x_{1l}),\ P_1(x_{21},\ldots,x_{2l}),\ldots,\ P_1(x_{k1},\ldots,x_{kl})$$

be all the subformulas of ψ containing P_1.

The idea is to replace each subformula $P_1(x_{i1},\ldots,x_{il})$ of ψ with a Boolean variable and stipulate with a formula containing just one occurrence of P_1 that each of these Boolean variables has the same truth value as the formula it replaces. Since we have no Boolean variable type, we instead replace each subformula $P_1(x_{i1},\ldots,x_{il})$ with an equation $v_1=v_{b(i)}$. We must ensure for each i that $b(i)$ is odd and greater than 1, that $b(i)$ is log-lin computable from $P_1(x_{i1},\ldots,x_{il})$ (with not resets), and that $b(i)\neq b(j)$ when $i\neq j$. To produce b satisfying these conditions suppose that $x_{i1},\ldots,x_{il}$ are formal variables denoting actual variables with subscripts $j_1,\ldots,j_l$ respectively. In the string $j_1\#j_2\#\cdots\#j_l$ replace every occurrence of 0 with 01, of 1 and 11, and of # with 10; let the result be $b(i)$. Let ψ' be the result of replacing each formula $P_1(x_{i1},\ldots,x_{il})$ in ψ by the formula $v_1=v_{b(i)}$. Now with a little effort we can see that φ is equivalent to

$$(Q_1x_1)\cdots(Q_nx_n)(\exists v_1, v_{b(1)},\ldots,v_{b(k)})(\forall y, y_1,\ldots,y_l)$$
$$\Big(\bigvee_{1\leq i\leq k}((y=v_{b(i)}\wedge y_1=xi_1\wedge\cdots\wedge y_l=x_{il})\rightarrow(v_1=y\leftrightarrow P_1(y)))\wedge\psi'\Big)$$

where y and $\mathbf{y}=y_1,\ldots,y_l$ denote variables with odd indices of odd length. It is not difficult to verify that this formula is reset log-lin computable from φ using two resets. □

Remark. The formula φ in the proof of Theorem 3.1 uses the symbol $\leftrightarrow$. If we require that formulas use only the Boolean connectives $\wedge$, $\vee$, and $\neg$, we must expand the subformula $v_1=y\leftrightarrow P_1(y)$ of φ' to obtain a formula in which P occurs twice, once positively and once negatively. It is easy to see that this is the best we can do. Suppose that the number of occurrences of P in such a formula could be reduced to one. If this occurrence were positive, then φ' would be monotone in P (i.e., truth is preserved when the interpretation of P is expanded). If this occurrrence were negative, then $\neg\varphi'$ would be monotone in P. But it is easy to produce a formula φ such that neither it nor its negation is monotone in P. However, in the case where P occurs only positively in φ, we can construct φ' in the proof of Theorem 3.1 using the subformula $c_1=y\rightarrow P_1(y)$ in place of $v_1=y\leftrightarrow P_1(y)$. For this case the theorem is true even if just the connectives $\wedge$, $\vee$, and $\neg$ are allowed. This is one of the advantages of using positive formulas.

Theorem 3.2. *Let L be a first-order or monadic second-order logic and l be a fixed positive integer. There is a reset log-lin reduction which taking each iterative definition of the form $[P(\boldsymbol{x}) \equiv \theta]_n$, where θ is a formula from L^* of length at most l, to an equivalent explicit definition.*

Proof. Since there are only finitely many formulas from L^* of length at most l, we may, given such formula θ, find an equivalent prenex formula θ' from L in constant time. Moreover, by Theorem 3.1 we may assume that P occurs in θ' just once, say in a subformula $P(\boldsymbol{y})$. Define formulas $\theta_n = \theta_n(\boldsymbol{x})$ by induction on n. Let θ_0 be a sentence false in all models (or all weak models if L is a monadic second-order logic). Form θ_{i+1} by substituting the variables $\boldsymbol{y}$ for corresponding free variables $\boldsymbol{x}$ in θ_i (perhaps changing other variables to avoid conflicts) and substituting the result for $P(\boldsymbol{y})$ in θ'. It is clear that $[P(\boldsymbol{x}) \equiv \theta]_n$ is equivalent to $[P(\boldsymbol{x}) \equiv \theta_n]$. If the substitution of variables has been done in a systematic way in the construction of θ_n, then it is clear that θ_n can be obtained from $[P(\boldsymbol{x}) \equiv \theta]_n$ by a reset log-lin reduction. □

Often we need to make several iterative definitions simultaneously. For example, Fischer and Rabin [26], in their lower bound proof for the theory of Real Addition, define sequences of formulas $\mu_n(x, y, z)$ and $\pi_n(x, y, z)$. Formula $\mu_n(x, y, z)$ holds precisely when x is a non-negative integer less than 2^{2^n} and $x \cdot y = z$; formula $\pi_n(x, y, z)$ holds precisely when x, y^x, and z are nonnegative integers less than 2^{2^n} and $y^x = z$. These definitions are simultaneous: the definition of π_{n+1}, for example, depends not only on π_n, but also on μ_n. Let us make the notion of simultaneous definition precise.

Let L be a first-order or monadic second-order logic and $\theta_1, \ldots, \theta_k$ be formulas from L^*. A *simultaneous iterative definition* is denoted

$$\begin{bmatrix} P_1(\boldsymbol{x}_1) \equiv \theta_1 \\ P_2(\boldsymbol{x}_2) \equiv \theta_2 \\ \vdots \quad\quad \vdots \\ P_k(\boldsymbol{x}_k) \equiv \theta_k \end{bmatrix}_n$$

Fix a structure $\mathfrak{A}$ for L and an assignment from $\mathfrak{A}$ to the free variables of this definition (defined in the obvious way). The simultaneous iterative definition assigns a relation from the universe of $\mathfrak{A}$ to each symbol $P_1, \ldots, P_k$. We define this assignment by induction on the *depth* n of the definition. When $n = 0$ it assigns the empty relation to each symbol. When $n > 0$ the assignment to θ_i is determined by letting the assignment for depth $n - 1$ interpret the free occurrences of $P_1, \ldots, P_k$ in θ_i. We can use simultaneous iterative definitions to augment the syntax of a logic in the same way we used iterative definitions. In particular, subscripts on definitions are expressed in unary notation.

The following theorem shows that simultaneous iterative definitions do not increase the expressiveness of a logic. Moreover, their use does not make for appreciably shorter expressions than use of ordinary iterative definitions. The theorem is proved along the same lines as similar results in Fischer and Rabin [26] and Ferrante and Rackoff [24, p. 159]. Moschovakis [51, p. 12] used similar ideas to prove an analogous theorem for simultaneous implicit definitions.

Theorem 3.3. *Let L be a first-order or monadic second-order logic and $P_1, \ldots, P_k$ be fixed relation variables. There is a reset log-lin reduction taking each formula φ of the form*

$$\begin{bmatrix} P_1(\boldsymbol{x}_1) \equiv \theta_1 \\ P_2(\boldsymbol{x}_2) \equiv \theta_2 \\ \vdots \quad\quad \vdots \\ P_k(\boldsymbol{x}_k) \equiv \theta_k \end{bmatrix}_n \psi$$

where ψ and $\theta_1, \ldots, \theta_k$ are formulas from L^* whose only free relation variables are $P_1, \ldots, P_k$, to an equivalent formula φ' of the form

$$[P(\boldsymbol{x}) \equiv \theta']_n \, \psi'$$

where θ' and ψ' are formulas from L^* whose only free relation variables is P. Moreover, if $P_1, \ldots, P_k$ occur only positively in each of the formulas $\theta_1, \ldots, \theta_k$, then we may arrange that P occurs only positively in θ.

Proof. As before, we assume that L has a symbol for equality and that all structures have cardinality at least 2. Again, we could dispense with these assumptions at the cost of added complications.

Without loss of generality, we may assume that the variable sequences $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ are mutually disjoint. Let $\boldsymbol{z}$ denote a sequence $z, z_1, \ldots, z_k$ of distinct variables disjoint from $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$. The idea of the proof is that one relation $P(\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ will code the relations $P_1(\boldsymbol{x}_1), \ldots, P_k(\boldsymbol{x}_k)$. To be more precise, the relation $P(\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ is equivalent to

$$\bigvee_{1 \le i \le k} (P_i(\boldsymbol{x}_i) \wedge z = z_i).$$

Thus, a particular $P_i(\boldsymbol{x}_i)$ can be extracted by writing

$$(\exists z, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \ldots, \boldsymbol{x}_k)\Big(P(\boldsymbol{z}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \wedge \bigwedge_{j \ne i} z \ne z_j\Big).$$

Call this formula $\delta_i(\boldsymbol{x})$ (or δ_i for short). Define θ' to be the L^* formula

$$[P_1(\boldsymbol{x}_1) \equiv \delta_1] \cdots [P_k(\boldsymbol{x}_k) \equiv \delta_k]\Big(\bigvee_{1 \le i \le k} (\theta_i \wedge z = z_i)\Big).$$

Notice that P is the only free relation variable in θ'. Let ψ' be the formula

$$[P_1(\boldsymbol{x}_1) \equiv \delta_1] \cdots [P_k(\boldsymbol{x}_k) \equiv \delta_k] \, \psi.$$

Now we can easily show that

$$\begin{bmatrix} P_1(\boldsymbol{x}_1) \equiv \theta_1 \\ P_2(\boldsymbol{x}_2) \equiv \theta_2 \\ \vdots \quad \vdots \\ P_k(\boldsymbol{x}_k) \;\; \theta_k \end{bmatrix}_n \psi$$

is equivalent to $[P(\boldsymbol{x}) \equiv \theta']_n \, \psi'$ by induction on n. □

Remark. Notice that in the proof of Theorem 3.3 formula ψ' is formed simply by inserting explicit definitions of fixed length before ψ. These definitions may be eliminated by replacing relation variables in ψ with their corresponding definitions. Now if ψ is a prenex formula or a member of a prescribed set of formulas (defined below), it is easy to arrange that ψ' is a formula of the same type.

We can now say precisely which kinds of definitions are used in the reductions described at the beginning of this section: they are prenex definitions and iterative definitions. It is useful, therefore, to have terminology to describe sets of formulas in L^* built up from prenex formulas using prenex and iterative definitions. We must place some restrictions on these sets to be able to efficiently translate them into equivalent formulas from L.

Let L be a first-order or monadic second-order logic. Let L' be the logic formed by adding relation variables $P_1, \ldots, P_k$ to the vocabulary of L, and l be a fixed positive integer. A *prescribed set of formulas* over L is a set of formulas of the form

$$[P_1(\boldsymbol{x}_1) \equiv \theta_1]_{n_1} \cdots [P_k(\boldsymbol{x}_k) \equiv \theta_k]_{n_k} \, \psi$$

where ψ is a prenex formula from L', and for each i either $n_i = 1$ and θ_i is a prenex formula from L' in which only $P_1, \ldots, P_{i-1}$ may occur as free relation variables (i.e., P_i has a prenex definition), or θ_i is a formula of length at most l from L^* in which only $P_1, \ldots, P_i$ may occur as free relation variables (i.e., P_i has an iterative definition in which the operator formula has bounded length). We place one further restriction on sets of prescribed formulas: each variable is quantified at most once in ψ and in each formula θ_i where P_i has a prenex definition. We impose this condition so that when we relativize all the formulas within a set to a unary relation symbol D, there is a reset log-lin reduction taking resulting formulas to equivalent formulas from another prescribed set of formulas. The condition is easy to satisfy in practice.

We now present our fundamental theorem for making reduction between formulas.

Theorem 3.4. *Let L be a first-order or monadic second-order logic. For each prescribed set of formulas over L there is a reset log-lin reduction taking each formula in the set to an equivalent formula in L.*

Proof. Fix a prescribed set of formulas over L. There are relation variable $P_1, \ldots, P_k$ as in the definition such that all formulas φ in the set are of the form

$$[P_1(\boldsymbol{x}_1) \equiv \theta_1]_{n_1} \cdots [P_k(\boldsymbol{x}_k) \equiv \theta_k]_{n_k} \psi,$$

where ψ is a prenex formula in which only $P_1, \ldots, P_k$ may occur as free relation variables, and for each i either $n_i = 1$ and θ_i is a prenex formula in which only $P_1, \ldots, P_{i-1}$ may occur as free relation variables, or θ_i is a formula of length at most l in which only $P_1, \ldots, P_i$ may occur as free relation variables.

At first glance it may seem that $P_1, \ldots, P_k$ are being defined simultaneously, but this is not the case. First P_1 is assigned a value by an iterative definition of depth n_1 which is substituted in the remaining definitions. Then P_2 is assigned a value by the next iterative definition of depth n_2 which is substituted in the remaining definitions, and so on. The proof combines this observation with the construction used in Theorem 3.3. As in that theorem, we will code the relations $P_1(\boldsymbol{x}_1), \ldots, P_k(\boldsymbol{x}_k)$ into a single relation $P(\boldsymbol{y})$ equivalent to

$$\bigvee_{1 \leq i \leq k} (P_i(\boldsymbol{x}_i) \wedge z = z_i)$$

where $\boldsymbol{y}$ is the variable sequence $z, z_1, \ldots, z_k, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$. As before, let $\delta_i(\boldsymbol{x}_i)$ be the formula

$$(\exists z, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_k)\Big(P(z, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \wedge \bigwedge_{j \neq i} z \neq z_j\Big).$$

To construct a formula φ' from L equivalent to φ we build, inductively, a sequence of formulas $\varphi_0(\boldsymbol{y}), \varphi_1(\boldsymbol{y}), \ldots, \varphi_k(\boldsymbol{y})$. Begin by taking φ_0 to be a sentence false in all models (or weak models, if L is a monadic second-order logic).

Suppose now that φ_{i-1} is given. Consider the simultaneous definition

$$\begin{bmatrix} P_1(\boldsymbol{x}_1) & \equiv P_1(\boldsymbol{x}_1) \\ \vdots & \vdots \\ P_{i-1}(\boldsymbol{x}_{i-1}) & \equiv P_{i-1}(\boldsymbol{x}_{i-1}) \\ P_i(\boldsymbol{x}_i) & \equiv \theta_i \\ P_{i+1}(\boldsymbol{x}_{i+1}) & \equiv P_{i+1}(\boldsymbol{x}_{i+1}) \\ \vdots & \vdots \\ P_k(\boldsymbol{x}_k) & \equiv P_k(\boldsymbol{x}_k) \end{bmatrix}$$

This definition simply defines $P_i(\boldsymbol{x}_i)$ to be θ_i and leaves the other relations unchanged. Use the construction in the proof of Lemma 3.3 to produce an equivalent definition $[P(\boldsymbol{y}) \equiv \eta_i(\boldsymbol{y})]$. Hence, η_i is

$$[P_1(\boldsymbol{x}_1) \equiv \delta_1] \cdots [P_k(\boldsymbol{x}_k) \equiv \delta_k]\, \psi_i$$

where ψ_i is the formula

$$(\theta_i \wedge z = z_i) \vee \Big(\bigvee_{j \neq i} (P_j(\boldsymbol{x}_j) \wedge z = z_j)\Big).$$

We claim that there is a reset log-lin reduction taking η_i to an equivalent prenex formula η_i' with just one subformula $P(\boldsymbol{u})$ in which P occurs. Whether P_i has a prenex definition, in which case θ_i is in prenex form, or an iterative definition, in which case θ_i is of bounded length, there is a simple reset log-lin reduction to convert ψ_i into prenex form. Apply the reduction given by Theorem 3.1 to the result to obtain an equivalent formula in which each of the symbols $P_1, \ldots, P_k$ occurs just once. In this formula, for each j, substitute $\delta_j(\boldsymbol{y}_j)$ for the subformula $P_j(\boldsymbol{y}_j)$. Convert to prenex form again by a reset log-lin reduction and apply the reduction of Theorem 3.1 one more time to obtain η' as desired. Notice that if P_i has an iterative definition, η_i has length less than some constant determined by l and the arities of $P_1, \ldots, P_k$.

If P_i has a prenex definition, form φ_i by substituting $\varphi_{i-1}(\boldsymbol{u})$ for $P(\boldsymbol{u})$ in η_i'. If P_i has an iterative definition we must make several substitutions. Beginning with φ_{i-1} replace free variables $\boldsymbol{y}$ with the corresponding variables $\boldsymbol{u}$ and substitute the result for $P(\boldsymbol{u})$ in η_i'. Repeat this operation n_i times. The resulting formula is φ_i.

In either case it is easy to see that φ_i is obtained by a reset log-lin reduction.

Since ψ is in prenex form, we can apply the reset log-lin reduction of Theorem 3.1 to obtain an equivalent prenex formula ψ' in which each of the symbols $P_1, \ldots, P_k$ occurs at most once. As before, there is a reset log-lin reduction to convert

$$[P_1(\boldsymbol{x}_1) \equiv \delta_1] \cdots [P_k(\boldsymbol{x}_k) \equiv \delta_k]\ \psi'$$

into a prenex formula with just one subformula $P(\boldsymbol{u})$ in which P occurs. Substitute $\varphi_k(\boldsymbol{u})$ for this subformula to obtain finally φ'. Repeated use of closure of reset log-lin reductions under composition shows that the mapping $\varphi \mapsto \varphi'$ is reset log-lin computable. □

Remark. Scrutiny of the preceding proof reveals two useful facts. First, if all the symbols P_i have prenex definitions we can arrange that φ' is in prenex form. Second, if we wish to restrict to formulas in which the only connectives are $\wedge$, $\vee$, and $\neg$, the theorem remains true providing P_i occurs only positively in θ_i when P_i has an iterative definition. To see this, observe that by the remark following Theorem 3.1 we can always ensure that the formulas η_i' each contain at most two occurrences of P. This is not a problem when P_i has a prenex definition because η_i' figures only once in the construction of φ_k and there are a bounded number of such definitions. When P_i has an iterative definition we can insure, again by the remark following Lemma 3.1, that P_i occurs at most once in η_i' since it occurs only positively in θ_i.

4. Inseparability results for first-order theories

Hereditary lower bound results have proofs similar to the classical hereditary undecidability results. Young [78], for example modified techniques used in the proof of the hereditary version of Gödel's Undecidability Theorem, which states that all subtheories of Peano Arithmetic are undecidable, to show that all subtheories of Presburger Arithmetic have an $NTIME(2^{2^{cn}})$ lower bound. Our starting point is another classical undecidability result—the Trakhtenbrot–Vaught Inseparability Theorem. Many hereditary undecidability results have been derived from this theorem.

Recall that L_0 is the first-order logic whose vocabulary contains just a binary relation symbol P. Let $fsat(L_0)$ be the set of sentences φ of L_0 true in some finite model, and $inv(L_0)$ the set of sentences of L_0 true in no model. The Trakhtenbrot–Vaught Inseparability Theorem states that $fsat(L_0)$ and $inv(L_0)$ are recursively inseparable: no recursive set contains one of these sets and is disjoint from the other. Trakhtenbrot [72] showed this for a first-order logic with sufficiently many binary relations in its vocabularly and Vaught [74, 75] reduced the number of binary relations to one. To see how this theorem gives hereditary undecidability results, suppose that for some theory Σ in a logic L there is a recursive reduction from the sentences of L_0 to the sentences of L that takes $fsat(L_0)$ into $sat(\Sigma)$ and $inv(L_0)$ into $inv(L)$. Clearly $sat(\Sigma)$ is not recursive since it separates the image of $fsat(L_0)$ from the image of $inv(L_0)$. Moreover, if $\Sigma' \subseteq val(\Sigma)$, then $sat(\Sigma) \subseteq sat(\Sigma')$ and $sat(\Sigma') \cap inv(L) = \emptyset$ so $sat(\Sigma')$ is not recursive either.

Let $T(n)$ be a time resource bound. Recall that $sat_T(L_0)$ is the set of sentences φ in L_0 such that φ is true in a structure of power at most $T(|\varphi|)$. Our analogue of the Trakhtenbrot–Vaught Inseparability Theorem states that for T satisfying certain weak hypotheses, $sat_T(L_0)$ and $inv(L_0)$ are *$NTIME(T(cn))$-inseparable* for some $c > 0$. That is, no set in $NTIME(T(cn))$ contains one of these sets and is disjoint from the other. We show, in fact, that the result is true if we restrict to prenex sentences in L_0. Thus, using the reductions between formulas described in the previous section, we can obtain hereditary *NTIME* lower bounds for theories in much the same way that we obtain hereditary undecidability results. In the next section we prove an inseparability theorem which gives hereditary linear *ATIME* lower bounds.

Our result is a consequence of the following theorem.

Theorem 4.1. *Let $T(n)$ be a time resource bound and A be an alphabet. Given a problem $\Delta \subseteq A^*$ in $\bigcup_{c>0} NTIME(T(cn))$, there is a reset log-lin reduction taking each $w \in A^*$ to a prenex sentence φ_w of L_0 such that if $w \in \Delta$, then $\varphi_w \in sat_T(L_0)$ and if $w \notin \Delta$, then $\varphi_w \in inv(L_0)$. Moreover, each variable occurring in φ_w is quantified just once.*

Proof. Let M be a $T(cn)$ time bounded nondeterministic Turing machine that accepts Δ. We may assume that on all inputs, all runs of M eventually halt since we may incorporate into M a deterministic 'timer' which halts after some number of moves given by a fully time constructible function dominating $T(cn)$. To simplify notation we assume that M has just one tape. Extending to multitape Turing machines requires only minor modifications. Let m be the number of tape symbols used by M. We assume that one of the tape symbols not in A is a blank symbol, denoted #.

The proof has two parts. In the first we translate information about runs of M on input w into formulas φ'_w in a logic with a vocabulary consisting of $m+3$ binary relation symbols so that φ'_w satisfies the conditions of the theorem. In the second we transform sentences φ'_w into the desired sentences φ_w by combining $m+3$ binary relations into one.

Translating Turing machine runs into first-order sentences is an old idea in logic; see Turing [73], Büchi [13], and for a general discussion Börger [8, 9, 10]. Our translation of runs into sentences is standard except for some difficulties that must be overcome to obtain prenex sentences using reset log-lin reductions.

First we describe the intended meanings of the $m+3$ binary relation symbols constituting the vocabulary of the logic for sentences φ'_w. The symbol $\leqslant$ will interpret a discrete linear order with a least element. For convenience we use infix notation with this symbol. We call the least element with respect to this order 0 and denote the successor and predecessor of an element x by $x+1$ and $x-1$ respectively. In this way we identify elements of a model with consecutive nonnegative integers. We also have relation symbols *STATE, HEAD,* and SYM_a for each $a \in A$. $STATE(x, t)$ holds if M is in state x at time t. (States are ordered arbitrarily. We may assume that all models considered have at least as many elements as M has states since we can precede φ'_w with enough dummy quantifiers to insure that $T(|\varphi'_w|)$ exceeds the number of states.) $HEAD(x, t)$ holds if the read head of M scans the tape cell at position x at time t. $SYM_a(x, t)$ holds if the tape cell at position x contains symbol a at time t.

Let φ'_w be a prenex sentence asserting the following.

(a) Relation $\leqslant$ is a discrete linear order with a least element.

(b) Each tape cell contains precisely one symbol at each time. The read head scans prescisely once cell at a given time. M is in precisely one state at a given time.

(c) If $HEAD(x, t)$ does not hold and $SYM_a(x, t)$ holds, $SYM_a(x, t+1)$ also holds. If $HEAD(x, t)$ holds, then the values of $SYM_a(x, t+1)$, $HEAD(x \pm 1, t+1)$, and the element z making $STATE(z, t+1)$ true are determined by the values of $SYM_a(x, t)$, and the element y making $STATE(y, t)$ true, in accordance with the transition function of M.

(d) The read head initially scans cell 0. M begins in its initial state.

(e) If $STATE(x, t)$ holds, then t has a successor if and only if x is a final state. For some t there is a final state x such that $STATE(x, t)$ holds.

(f) The input tape initially contains w.

Notice that (f) is the only conjunct depending on w; the others are fixed. Thus, if we can express (f) as a prenex sentence obtainable from w by a reset log-lin reduction, it is a simple matter to produce a prenex sentence φ'_w equivalent to the conjunction of (a)–(f) obtainable from w by a reset log-lin reduction.

Suppose that $w = a_0a_1 \cdots a_{n-1}$ where each a_i is an input alphabet symbol. We cannot just say that there are positions $v_0, v_1, \ldots, v_{n-1}$ such that $v_0 = 0$, $v_{i+1} = v_i + 1$, and $SYM_{a_i}(v_i, 0)$ for $i < n$ because the combined length of variable indices is $\Omega(n \log n)$. We must define two relations *LEFT* and *RIGHT* to reduce the number of quantified variables.

Intuitively, *LEFT* and *RIGHT* interpret the left and right child relations for a binary tree τ on the first n elements $0, \ldots, n-1$ of the model. It is intended that $LEFT(x, y)$ holds precisely when $y = 2x$ and $RIGHT(x, y)$ holds precisely when $y = 2x + 1$. Then τ has root 0 and 0 is its own left child (so the notion of *tree* is interpreted somewhat loosely).

Using the machinery of the last section it is easy to give short formulas defining *LEFT* and *RIGHT* on long intervals. Let θ be the formula

$$\begin{aligned}&(x_1 = y_1 \wedge x_2 = y_2) \vee (x_1 + 1 = y_1 \wedge x_2 + 2 = y_2)\\ &\quad \vee \exists z_1, z_2\, (P(x_1, x_2, z_1, z_2) \wedge P(z_1, z_2, y_1, y_2)).\end{aligned}$$

Then the relation $P(x_1, x_2, y_1, y_2)$ given by the iterative definition

$$[P(x_1, x_2, y_1, y_2) \equiv \theta]_m$$

is true when $0 \leqslant y_1 - x_1 \leqslant 2^{m-1}$ and $2(y_1 - x_1) = y_2 - x_2$. Now $LEFT(x, y)$ is equivalent to $P(0, 0, x, y)$ and $RIGHT(x, t)$ is equivalent to $P(0, 1, x, y)$ on the interval $0, \ldots, 2^m + 1$, so we take $m = \lfloor \log(n-1) \rfloor$ to obtain *RIGHT* and *LEFT* on the interval $0, \ldots, n$. By Theorem 3.2 there are first-order formulas θ'_n and θ''_n defining *LEFT* and *RIGHT*; moreover, they are computable from the unary representation of m by a reset Turing machine in time $\log n$ and space $\log \log n$. By increasing time to $\log n \log \log n$ we can make θ'_n and θ''_n prenex formulas.

The height of τ is $h = \lceil \log n \rceil$. Now for every i such that $0 \leqslant i \leqslant h$ and every $j < 2^{h-i}$ define a quantifier free formula $\theta_{i,j}(x_0, \ldots, x_i)$ by induction on i. Roughly, $\theta_{i,j}(x_0, \ldots, x_i)$ says that if $(x_i, x_{i-1}, \ldots, x_0)$ is a path in τ from vertex $j = x_i$, then the symbol at position x_0 on the input tape is a_{x_0}. First $\theta_{0,j}(x_0)$ is $SYM_{a_j}(x_0, 0)$ when $j < n$ and some tautology (say $x_0 = x_0$) when $n \leqslant j \leqslant 2^h$. Next, $\theta_{i+1,j}(x_0, \ldots, x_{i+1})$ is the formula

$$(LEFT(x_{i+1}, x_i) \rightarrow \theta_{i,2j}(x_0, \ldots, x_i)) \wedge (RIGHT(x_{i+1}, x_i) \rightarrow \theta_{i,2j+1}(x_0, \ldots, x_i)).$$

By induction on i we can show that for $j \leqslant 2^{h-i}$ the sentence

$$\forall x_0, \ldots, x_i\, (x_i = j \rightarrow \theta_{i,j}(x_0, \ldots, x_i))$$

is true if and only if for every vertex $k < n$ which is an ith generation descendent of j, $SYM_{a_k}(k, 0)$ holds. Since 0 is a left child of itself, every vertex in τ is an hth generation descendent of 0, so the sentence

$$\forall x_0, \ldots, x_h\, (x_h = 0 \rightarrow \theta_{h,0}(x_0, \ldots, x_h))$$

says that $SYM_{a_k}(k, 0)$ holds when $k < n$; that is, it says w is written on the first n

cells of the input tape at time 0. Let ψ_w be a conjunction of this sentence and a sentence that says $SYM_{\#}(x, 0)$ holds for all $x > n$ (that is, that all the tape cells from position n onward are blank at time 0). We must show that there is a reset log-lin reduction taking w to ψ_w, from which it follows easily that there is a reset log-lin reduction taking w to φ'_w.

We describe the actions of a machine effecting such a reduction. For the conjunct asserting that cells from position n onward are blank this is straightforward so we need only show it for the conjunct asserting w is written on the first n cells.

We will suppose that induces of variables are in unary; thus, the formal variable x_i denotes the actual variable

$$v_{\underbrace{11\cdots1}_{i+1 \text{ times}}} .$$

First, w is read from the input tape while h cells are marked off on a work tape. One way to accomplish this is simply to keep a count on a work tape of the number of input tape cells scanned. Count in binary. Incrementing the count requires changing the low order 1-bits to 0 until encountering a 1, which is changed to 0. The work tape head is then returned to the lowest order bit to prepare for the next advance of the input head. It is not difficult to show that the time required to read the input tape and do all the increments is O(n).

Next the input head is reset. Now a simple algorithm utilizing a stack to keep track of subscripts will generate ψ_w. The maximal stack height is h. Formula ψ_w was defined in such a way that the information required from the input tape can be read off from left to right as the algorithm proceeds. Variable indices are easily computed from the stack height since they are in unary.

This computation clearly uses just log space. We need show that it takes just linear time. The time required is less than a constant multiple of the length of $\theta_{h,0}(x_0, \ldots, x_h)$; the length of this formula is in turn less than a constant multiple of the combined lengths of variable indices occurring within it. By induction on i, variable x_k occurs no more than $3 \cdot 2^{i-k}$ times in $\theta_{i,j}$ when $k < i$ and x_i occurs just twice. Hence, the combined lengths of variable indices occurring in $\theta_{h,0}$ is not more than

$$\sum_{k=0}^{h} 3(k+1)2^{h-k} = 3(2^{h+2} - h - 3) < 24n.$$

Thus, the computation requires just linear time. Notice that each variable in ψ_w is quantified just once so it is easily arranged that the same is true of φ'_w.

If necessary we can add dummy quantifiers at the beginning of φ'_w to make its length at least cn. Thus, if $w \in \Delta$, then φ'_w is true in some model of size at most $T(|\varphi'_w|)$. If $w \notin \Delta$, then φ'_w is true in no model. Suppose, on the contrary, that φ'_w is true in some model $\mathfrak{A}$. Consider the submodel of $\mathfrak{A}$ obtained by restricting to the elements which can be reached from 0 by finitely many applications to successor. The values of $STATE(x, t)$, $HEAD(x, t)$, and $SYM_a(x, t)$ on this submodel describe a run of M on input w. Since we have incorporated a timer which halts M on every run, this submodel is finite. But then the last element t in this submodel has no successor, so the state x for which $STATE(x, t)$ holds is final. Thus, M accepts w, a contradiction.

This completes the first part of the proof. We now show how to combine $m+3$ binary relations into one.

To simplify notation, let us rename the relation symbols $P_0, P_1, \ldots, P_{m+2}$. Suppose $\mathfrak{A}'$ is a model of φ'_w. Before describing how to transform φ'_w into φ_w, we describe the model $\mathfrak{A}$ of φ_w corresponding to $\mathfrak{A}'$. First form the disjoint union of the interpretations of $P_0, P_1, \ldots, P_{m+2}$ in $\mathfrak{A}'$. We have then relations $R_0, R_1, \ldots, R_{m+2}$ on disjoint domains $B_0, B_1, \ldots, B_{m+2}$. Their union is a single binary relation R on the domain $B = \bigcup_{i\leq m+2} B_i$. Now enlarge R so that $R \cap (B_0 \times B_i)$ is the natural bijection from B_0 to B_i when $1 \leq i \leq m+2$. Next, enlarge B by adding elements $b_0, b_1, \ldots, b_{m+2}$ and then add the pairs (b_i, b) to R for each $i \leq m+2$ and $b \in B_i$. Define $\mathfrak{A}$ to be $\langle B, R\rangle$. We see that if $b_0, b_1, \ldots, b_{m+2}$ are known, then the relations $R_0, R_1, \ldots, R_{m+2}$ can be recovered in $\mathfrak{A}$ and in fact we may use the natural isomorphisms from B_0 to B_i to define an isomorphic image of $\mathfrak{A}'$ with domain B_0.

Relativize the quantifiers of φ'_w to a unary relation symbol D, thereby forming $(\varphi'_w)^D$. This sentence contains relation symbols $D, P_0, P_1, \ldots, P_{m+2}$. Put the explicit definitions $[D(x) \equiv P(x_0, x)]$, $[P_0(x, y) \equiv P(x, y)]$ and for $1 \leq i \leq m+2$,

$$[P_i(x, y) \equiv \exists x', y'\, (P(x, x') \wedge P(y, y') \wedge P(x', y') \wedge P(x_i, x') \wedge P(x_i, y'))]$$

before this formula. Here $x_0, x_1, \ldots, x_{m+2}$ are new free variables whose intended interpretations in $\mathfrak{A}$ are $b_0, b_1, \ldots, b_{m+2}$. Now existentially quantify $x_0, x_1, \ldots, x_{m+2}$. There is a reset log-lin reduction that takes the resulting formula to an equivalent prenex formula φ_w. (Since each variable in φ'_w is quantified just once the relativizations may be pushed inward. Then since all of the explicit definitions are of fixed length, conversion to prenex form is straightforward.)

If M accepts w, then φ'_w is true in some model $\mathfrak{A}'$ of power at most $T(|\varphi'_w|)$. It follows that φ_w is true in some model $\mathfrak{A}$ of power at most

$$(m+3)(T(|\varphi'_w|)+1.$$

We can assume this quantity is less than $T(|\varphi_w|)$ by lengthening φ_w with dummy quantifiers if necessary and using the definition of time resource bound to infer that $(m+3)T(n) \leq T((m+3)n)$. If M does not accept w, then φ'_w has no models and hence neither does φ_w. Finally, it is easy to arrange that every variable in φ_w is quantified just once. □

Remark. Inspection of the construction of sentences φ_w in Theorem 4.1 reveals that for every constant $b > 0$ there are constants c_0 and c_1 such that $|\varphi_w| \leq$

$c_1|w| + c_0$ whenever $0 < c < b$. The constant c_1 depends only on the size of the input alphabet A and the constant b. (This will be useful in the proof of Theorem 4.3.) On the other hand, the constant c_0 depends on the particular Turing machine M whose runs φ_w describes.

Corollary 4.2. *Let $T_1(n)$ and $T_2(n)$ be time resource bounds such that*

$$NTIME(T_2(n)) - NTIME(T_1(n)) \neq \emptyset.$$

Suppose that $\lim_{n\to\infty} T_1(n)/n = \infty$. Then there is a constant $c > 0$ such that for each set Γ of satisfiable sentences with $sat_{T_2}(L_0) \subseteq \Gamma$,

$$\Gamma \notin NTIME(T_1(cn)).$$

Proof. Let Δ be an element of $NTIME(T_2(n)) - NTIME(T_1(n))$, where $\Delta \subseteq A^*$. By Theorem 4.1 there is a reset log-lin reduction taking each $w \in A^*$ to a sentence φ_w in L_0 so that Δ is mapped into $sat_{T_2}(L_0)$ and $A^* - \Delta$ is mapped into $inv(L_0)$. Suppose that this reduction takes at most time $b|w|$.

Let $c = 1/b$ and Γ be a set of satisfiable sentences containing $sat_{T_2}(L_0)$. Suppose that $\Gamma \in NTIME(T_1(cn))$. To decide if $w \in \Delta$ we will compute φ_w in time $b|w|$, and then determine if $\varphi_w \in \Gamma$ using a $T_1(cn)$ time-bounded nondeterministic Turing machine. Certainly $|\varphi_w| \leq b|w|$ so the composition of these two reductions takes time at most $b|w| + T_1(bc|w|)$. We know that $|w| \leq T_1(|w|)$ so this time is bounded above by $(b+1)T_1(|w|)$. Since $\lim_{n\to\infty} T_1(n)/n = \infty$ we can apply the Linear Speed Up Theorem (see Hopcroft and Ullman [37]) to show that $\Delta \in NTIME(T_1(n))$, a contradiction. Therefore, $\Gamma \notin NTIME(T_1(cn))$. □

Remark. We see from Corollary 4.2 that application of our results relies on the ability to separate nondeterministic time complexity classes. The strongest result in this direction for time resource bounds in the range bounded above by $\exp_\infty(n)$ is due to Seiferas, Fischer, and Meyer [64]. It says that if $T_2(n)$ is a time resource bound and $T_1(f(n+1)) \in o(T_2(f(n)))$ for some recursively bounded, strictly increasing function $f(n)$, then

$$NTIME(T_2(n)) - NTIME(T_1(n)) \neq \emptyset.$$

This theorem has interesting implications for us. Let $T(n)$ be a time resource bound. Take $T_1(n) = T(dn)$, where $0 < d < 1$, $T_2(n) = T(n)$, and $f(n) = n$. The Seiferas–Fischer–Meyer Theorem tells us that if $T(dn + d) = o(T(n))$, then

$$NTIME(T(n)) - NTIME(T(dn)) \neq \emptyset.$$

By taking a slightly smaller d the hypothesis may be simplified to $T(dn) = o(T(n))$. By Corollary 4.2 there is a constant $c > 0$ such that if Γ is a set of satisfiable sentences with $sat_T(L_0) \subseteq \Gamma$, then $\Gamma \notin NTIME(T(cn))$.

Most time resource bounds that occur as complexities of theories satisfy the hypothesis $T(dn) = o(T(n))$. Among them are the functions $\exp_r(n)$ when $r \geq 1$,

$2^{n/\log n}$, and 2^{n^k}. Powers do not satisfy this hypothesis, so when $T(n) = n^k$ where $k > 1$, we can only conclude that $\Gamma \notin NTIME(g(n))$ for all $g(n)$ such that $g(n) = o(n^k)$.

By considering just one time resource bound $T(n)$ we gain another advantage: we can prove a full-fledged inseparability theorem. This will allow us to obtain *NTIME* lower bounds for problems of the form $val(\Sigma)$, as well as problems of the form $sat(\Sigma)$.

Theorem 4.3. *If T is a time resource bound such that for some d between* 0 *and* 1, $T(dn) = o(T(n))$ *then there is a constant* $c > 0$ *such that* $sat_T(L_0)$ *and* $inv(L_0)$ *are* $NTIME(T(cn))$*-inseparable.*

Proof. Corollary 4.2 and the preceding remark show that there is a $c > 0$ such that if $sat_T(L_0) \subseteq \Gamma$ and $inv(L_0) \cap \Gamma = \emptyset$, then $\Gamma \notin NTIME(T(cn))$. (The corollary applies because $\lim_{n\to\infty} T(n)/n = \infty$ when $T(dn) = o(T(n))$.) We must show that there is a $c' > 0$ such that if $sat_T(L_0) \subseteq \Gamma$ and $inv(L_0) \cap \Gamma = \emptyset$, then $\bar{\Gamma}$, the set of sentences from L_0 not in Γ, is not in $NTIME(T(c'n))$.

Let b be a positive constant (say 1). By Theorem 4.1, if $\Delta \subseteq A^*$ is in $NTIME(T(c'n))$ for $0 < c' < b$, there is a reset log-lin reduction taking each $w \in A^*$ to a sentence φ_w of L_0 such that Δ is mapped into $sat_T(L_0)$ and $\bar{\Delta}$ is mapped into $inv(L_0)$. We know also from the remark following Theorem 4.1 that $|\varphi_w| \leq c_1 |w| + c_0$, where c_1 is a constant depending only on A and b, not on c' or Δ. Take c' small enough that $c'c_1 < c$.

Now consider a set Γ such that $sat_T(L_0) \subseteq \Gamma$ and $inv(L_0) \cap \Gamma = \emptyset$. We must show that $\bar{\Gamma} \notin NTIME(T(c'n))$ so suppose the contrary. We can take Δ in the previous paragraph to be $\bar{\Gamma}$ so there is a reset log-lin reduction mapping $\bar{\Gamma}$ into $sat_T(L_0) \subseteq \Gamma$ and Γ into $inv(L_0) \subseteq \bar{\Gamma}$. This reduction takes an input w to a sentence φ_w in time at most $b\,|w|$.

Thus, we can determine whether $w \in \Gamma$ by computing φ_w and determining in nondeterministic time $T(c'n)$ if $\varphi_w \in \bar{\Gamma}$. Hence,

$$\Gamma \in NTIME(bn + T(c'(c_0 n + c_1))) \subseteq NTIME(T(cn))$$

a contradiction. □

Remark. For simplicity, Theorem 4.3 was stated for sets $sat_T(L_0)$ and $inv(L_0)$. By Theorem 4.1 it remains true if we take instead $sat_T^p(L_0)$ and the set of prenex formulas in $inv(L_0)$. Similarly, Theorem 4.2 holds if we restrict to prenex sentences.

5. Inseparability results for monadic second-order theories

In this section we develop inseparability results for monadic second-order theories analogous to those for first-order theories in Section 4. The appropriate complexity classes here are the linear alternating time classes rather than nondeterministic time classes. Because linear alternating time classes are closed under complementation, we do not need a special argument like the one used in Theorem 4.3 to obtain inseparability results. Also, lower bounds are obtained by simple diagonalization, rather than more sophisticated results such as the theorem of Fischer, Meyer, and Seiferas used in the last section.

As before, inseparability results are closely related to satisfiability problems that are hard for certain complexity classes. The classes are of the form

$$\bigcup_{c>0} ATIME(T(cn), cn)$$

which is in many ways more natural than

$$\bigcup_{c>0} ATIME(T(cn), n).$$

If there is a reset log-lin reduction from a problem Γ to a class of the first form, then we may concude that Γ is also in the class. We know of no speed up theorem for alternations, so we cannot make the same claim for classes of the second form.

One of the main results of the section is Theorem 5.2, an analogue of Theorem 4.1. We could prove this result along the same lines as Theorem 1.4, but we obtain a somewhat sharper result if we appeal to a result of Lynch [45] relating nondeterministic time classes to the spectra of monadic second-order sentences. Lynch encodes Turing machine runs in a way different from the classical method used in the last section. Rather than explicitly accounting for symbols at each tape position and time in a machine run, he keeps track of just the symbol changed (not its position), the symbol which replaces it, and the direction of head movement at each time. If the underlying models have enough structure, it is possible to express derivability between instantaneous descriptions of nondeterministic Turing machines with just this information. Lynch shows, in particular, that this is the case if the underlying models have an addition relation $PLUS(x, y, z)$ which holds when $x + y = z$.

We begin, therefore, by considering the monadic second-order logic ML_+ whose vocabulary contains just a ternary relation symbol *PLUS*, and $M\Sigma_+$, the monadic second-order theory of addition on initial segments of the natural numbers. $M\Sigma_+$ can be axiomatized by a set of first-order sentences. Explicitly define a relation $\leqslant$ by

$$[x \leqslant y \equiv \exists z\, PLUS(x, z, y)].$$

Then $M\Sigma_+$ says that $\leqslant$ is a discrete linear order with least element 0 and

$$PLUS(x, y, z) \leftrightarrow ((y = 0 \wedge x = z) \vee PLUS(x, y - 1, z - 1)).$$

(The immediate predecessor of an element x is denoted $x - 1$; the immediate successor is denoted $x + 1$.) Note that even though $M\Sigma_+$ consists of first-order sentences, $sat_T(M\Sigma_+)$ is the set of *monadic second-order sentences* φ true in some model of $M\Sigma_+$ of size at most $T(|\varphi|)$.

Theorem 5.1. *Let $T(n)$ be a time resource bound and A an alphabet. Given a problem $\Delta \subseteq A^*$ in $\bigcup_{c>0} ATIME(T(cn), cn)$, there is a prescribed set Γ of sentences over ML_+, and a reset log-lin reduction each $w \in A^*$ to a sentence φ_w in Γ such that if $w \in \Delta$, then $\varphi_w \in sat_T^*(M\Sigma_+)$ and if $w \notin \Delta$, then $\varphi_w \in inv^*(ML_+)$.*

Proof. Fix $c > 0$; we may take c to be an integer. Let M be an alternating Turing machine that accepts Δ in time $T(cn)$ with at most cn alternations. As in Theorem 4.1 we may assume that on all inputs, all runs of M eventually halt: incorporate into M a deterministic 'timer' which halts after some number of moves given by a fully time constructible function dominating $T(cn)$. We assume for simplicity that M has just one tape; extending to multitape Turing machines requires only minor modifications. Let $a_1, \ldots, a_m$ be the tape symbols used by M.

Let $r = \{0, \ldots, r-1\}$ be a finite ordinal and $+$ be the usual addition relation on r, so that $\langle r, +\rangle$ is a model of $M\Sigma_+$. In this model we represent an instantaneous description of M by a sequence of sets $X_1, \ldots, X_{m+2} = \boldsymbol{X}$, where sets $X_1, \ldots, X_m$ partition r and X_{m+1} is a singleton set, and X_{m+2} is singleton set whose element is one of the states of M. The set of states is identified with an initial interval of r. We intend that the symbol a_i is at position x when $x \in X_i$, that the head scans position x when $x \in X_{m+1}$, and that M is in state x when $x \in X_{m+2}$. We will need to restrict to initial intervals in our models because when we consider truth in weak models at the end of the proof we may not be able to quantify over all subsets, but we can arrange to quantify over subsets of finite initial intervals. Let $ID(x, \boldsymbol{X})$ be the formula specifying that sets $X_1, \ldots, X_m$ partition the interval $[0, x]$, X_{m+1} and X_{m+2} are singleton sets contained in this interval, and the element in X_{m+2} is a state.

Recall that each state of M has one of four types: universal, existential, accepting, and rejecting. Let

$$UNIV(x, \boldsymbol{X}), \quad EXIS(x, \boldsymbol{X}), \quad ACC(x, \boldsymbol{X}), \quad \text{and} \quad REJ(x, \boldsymbol{X})$$

be formulas indicating that $ID(x, \boldsymbol{X})$ holds and the state for the instantaneous description represented by $\boldsymbol{X}$ is of the corresponding type.

Lynch [45] shows that for each nondeterministic Turing maching M' there is a monadic second-order formula $\eta_{M'}(\boldsymbol{X}, \boldsymbol{Y})$ that holds in $\langle r, +\rangle$ precisely when $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent instantaneous descriptions for M' and $\boldsymbol{Y}$ can be obtained from $\boldsymbol{X}$ within r or fewer moves of M' by a computation in which the head does not reach a tape position greater than or equal to r. We can regard the alternating Turing machine M as a nondeterministic Turing machine simply by ignoring state types. We also form the nondeterministic Turing machine M' by eliminating transitions out of all states in M except existential states and then ignoring state types, and

M'' by eliminating transitions out of all states in M except universal states and then ignoring state types. Let $\eta(x, \boldsymbol{X}, \boldsymbol{Y})$ be the formula

$$[D(y) \equiv y \leq x]\, \eta_M(\boldsymbol{X}, \boldsymbol{Y})^D.$$

Let $\eta_\exists(x, \boldsymbol{X}, \boldsymbol{Y})$ be the formula

$$[D(y) \equiv y \leq x](UNIV(x, \boldsymbol{X}) \wedge \eta_{M'}(\boldsymbol{X}, \boldsymbol{Y}))^D.$$

Let $\eta_\forall(x, \boldsymbol{X}, \boldsymbol{Y})$ be the formula

$$[D(y) \equiv y \leq x](EXIS(x, \boldsymbol{X}) \wedge \eta_{M''}(\boldsymbol{X}, \boldsymbol{Y}))^D.$$

That is, $\eta(x, \boldsymbol{X}, \boldsymbol{Y})$ expresses derivability between instantaneous descriptions on the interval $[0, x]$; $\eta_\exists(x, \boldsymbol{X}, \boldsymbol{Y})$ $(\eta_\forall(x, \boldsymbol{X}, \boldsymbol{Y}))$ expresses the same except that all states, excluding possibly the last, are existential (universal). Notice, in particular, that $\eta_\forall(x, \boldsymbol{X}, \boldsymbol{Y})$ holds of all $\boldsymbol{X}$. Let $TERM(x, \boldsymbol{X})$ be the formula

$$\forall \boldsymbol{Y}\, (\eta(x, \boldsymbol{X}, \boldsymbol{Y}) \rightarrow \boldsymbol{X} = \boldsymbol{Y})$$

and $TERM_\forall(x, \boldsymbol{X})$ be the formula

$$\forall \boldsymbol{Y}\, (\eta_\forall(x, \boldsymbol{X}, \boldsymbol{Y}) \rightarrow \boldsymbol{X} = \boldsymbol{Y}).$$

Let $\psi_m(x, \boldsymbol{X})$ be a prenex sentence asserting the following.

(a) Relation $\leq$ is a discrete linear order with a least element and a greatest element; every element other than the greatest has a successor.

(b) The relation $PLUS(x_0, x_1, x_2)$ holds if and only if either $x_0 = x_2$ and $x_1 = 0$ or $PLUS(x_0, x_1 - 1, x_2 - 1)$ holds.

(c) There is a set Y with no elements. For every set Y and element y there is a set $Y \cup \{y\}$.

(d) If y is an element such that for all $\boldsymbol{Y}$ satisfying $ID(y, \boldsymbol{Y})$, $TERM(y, \boldsymbol{X}, \boldsymbol{Y})$ implies $ACC(y, \boldsymbol{Y})$ or $REJ(y, \boldsymbol{Y})$, then $x \leq y$.

(e) The sequence $\boldsymbol{X}$ represents the instantaneous description for an input tape with w written on it, the head scanning the first position, and M in the initial state.

Each of these items can be expressed by a fixed sentence except the part of condition (e) concerning the input tape. The prenex formula expressing this part is constructed in the same way as in Theorem 4.1. As in that construction, it follows that there is a reset log-lin reduction taking w to ψ_w.

Consider any weak model $\mathfrak{A}$ in which conditions (a)–(c) hold. For the moment, let us suppose that the element x from the universe of $\mathfrak{A}$ is finite, i.e., a finite distance from the least element 0. (By condition (a), this is a well defined notion.) Condition (b) ensures that $PLUS$ restricted to the interval $[0, \ldots, x]$ in $\mathfrak{A}$ is the usual addition relation. Condition (c) ensures that quantification over subsets of $\{0, \ldots, x\}$ in the weak model $\mathfrak{A}$ is quantification over all subsets of $\{0, \ldots, x\}$. Thus, for finite x, the formulas $\eta(x, \boldsymbol{X}, \boldsymbol{Y})$, $\eta_\exists(x, \boldsymbol{X}, \boldsymbol{Y})$, and $\eta_\forall(x, \boldsymbol{X}, \boldsymbol{Y})$ have interpretations in $\mathfrak{A}$ corresponding to computations of M as described above.

Now in $\mathfrak{A}$ consider x and X satisfying conditions (d) and (e). (We no longer stipulate that x is finite.) Since all runs of M on input w halt, say within k moves, we see that every such x is at most distance k from 0. (Note that this is the case even if there is no y as described in (d).) Thus such an x will be finite and the observations of the previous paragraph pertain.

Let $\theta(x, X)$ be the formula

$$\begin{aligned}&ACC(x, X) \vee (EXIS(x, X) \wedge \exists Y\, (\eta_{\exists}(x, X, Y) \wedge P(x, Y)))\\ &\quad \vee (UNIV(x, X) \wedge \forall Y\, ((\eta_{\forall}(x, X, Y)) \wedge TERM_{\forall}(x, Y)) \rightarrow P(x, Y))).\end{aligned}$$

Let φ_w be the sentence

$$[P(x, X) \equiv \theta(x, X)]_{cn+1}\, \exists x\, \exists X\, (\psi_w(x, X) \wedge P(x, X))$$

where $n = |w|$. Clearly, there is a prescribed set of sentences Γ containing every φ_w. It it also clear that if $w \in \Delta$, then φ_w is true in some model of size at most $T(|\varphi_w|)$ because in such models P interprets acceptance by M for $x = T(cn)$.

If $w \notin \Delta$, then φ_w is true in no model. Suppose on the contrary that φ_w is true in $\mathfrak{A}$. Then in $\mathfrak{A}$ there are x and X such that $\psi_w(x, X)$ and $P(x, X)$ hold, where P is given by the implicit definition

$$[P(x, X) \equiv \theta(x, X)]_{cn+1}.$$

By the remarks above, x must be finite, and hence P describes an accepting computation of M on input w, contradiction. □

The next theorem, the analogue of Theorem 4.1, follows from the previous theorem and a result of Kaufmann and Shelah [39].

Theorem 5.2. *Let $T(n)$ be a time resource bound and A an alphabet. Given a problem $\Delta \subseteq A^*$ in $\bigcup_{c>0} ATIME(T(cn), cn)$, there is a prescribed set Γ of sentences over ML_0, and a reset log-lin reduction taking each $w \in A^*$ to a sentence φ_w in Γ such that if $w \in \Delta$, then $\varphi_w \in sat_T^*(ML_0)$ and if $w \notin \Delta$, then $\varphi_w \in inv^*(ML_0)$.*

Proof. By the previous theorem we need only show that there is a formula $\pi(x, y, z)$ from ML_0 such that for each finite ordinal $n = \{0, \ldots, n-1\}$ there is a binary relation R on n such that $\pi^{\mathfrak{A}}(x, y, z)$ is an addition relation on n, where $\mathfrak{A} = \{n, R\rangle$. Kaufmann and Shelah [39] prove a much stronger result: there is a formula $\pi(x, y, z)$ such thaat for almost every binary relation R on n, $\pi^{\mathfrak{A}}(x, y, z)$ is an addition relation on n, where $\mathfrak{A} = \langle n, R\rangle$.

For the sake of completeness, we sketch a proof of the simpler result that there is a formula that codes an addition relation on some binary relation of each finite power.

First suppose that the vocabulary for the logic has three binary relation symbols P_1, P_2, P_3, rather than just one, and that they interpret binary relations R_1, R_2, R$_3$ on m. To simplify the proof we assume that $m = r^3$. It is easy to specify a formula $\psi(X)$ saying the relations R_1, R_2, R_3 restricted to $m \times X$ are functions, respectively denoted f_1, f_2, f_3, and that $(f_1(x), f_2(x), f_3(x))$ ranges over each triple in X^3 precisely once as r ranges over m. Thus $|X| = r$ and we have defined a bijection between m and X^3. Since we can quantify over subsets of m, we can quantify over ternary relations on X when $\psi(X)$ holds. Therefore, we can, without much trouble, define an addition relation on X. Also, we can extend this relation to define addition *modulo* r. But then it is easy to define addition on m using the bijection between m and X^3.

Using the construction in the proof of Theorem 4.1, the three binary relations R_1, R_2, R_3 on a set of size $m = r^3$ can be coded as a single binary relation on a set of size $n = 3(m + 1)$. This set has three disjoint subsets of size m on which addition and addition *modulo* m can be coded. It is not difficult now to define addition on all of n. Thus, there is a binary relation on n from which an addition relation can be defined when n is of the form $3(r^3 + 1)$. With a little effort this construction can be made to work for arbitrary n. □

We now state an analogue of Corollary 4.2.

Corollary 5.3. *Let* $T_1(n)$ and $T_2(n)$ *be time resource bounds such that*

$$ATIME(T_2(n), n) - ATIME(T_1(n), n) \neq \emptyset.$$

Suppose that $\lim_{n\to\infty} T_1(n)/n = \infty$. *Then there is a constant* $c > 0$ *such that for each set* Γ *of satisfiable sentences with* $sat_{T_2}(ML_0) \subseteq \Gamma$,

$$\Gamma \notin ATIME(T_1(cn), cn).$$

The proof is the same as for Corollary 4.2. Note, however, that we rely on a result that says the Linear Speed Up Theorem applies to the alternating Turing machines. We must also use Theorem 3.4 to obtain a reset log-lin reduction from a prescribed set of sentences over ML_0 to equivalent sentences in ML_0.

We also have an analogue for Theorem 4.3.

Theorem 5.4. *If* T *is a time resource bound such that for some* d *between* 0 *and* 1, $T(dn) = o(T(n))$, *then there is a constant* $c > 0$ *such that* $sat_T(ML_0)$ *and* $inv(ML_0)$ *are* $ATIME(T(cn), cn)$*-inseparable.*

Proof. The proof is much simpler than that of Theorem 4.3. We can separate $ATIME(T(n), n)$ and $ATIME(T(dn), dn)$ using a straightforward diagonalization, so we do not appeal to the more difficult methods used in separating *NTIME* classes. Then we use the previous corollary to show that for some $c > 0$, if $sat_T(ML_0) \subseteq \Gamma$ and $inv(ML_0) \cap \Gamma = \emptyset$ then $\Gamma \notin ATIME(T(cn), cn)$. Since $ATIME(T(cn), cn)$ is closed under complementation, we have that $sat_T(ML_0)$ and $inv(ML_0)$ are $ATIME(T(cn), cn)$-inseparable. □

Remark. By Theorem 5.1, Theorems 5.3 and 5.4 hold with $sat_T(M\Sigma_+)$ and $inv(ML_+)$ in place of $sat_T(ML_0)$ and $inv(ML_0)$.

It is also important to note that even though we reduced the prescribed sets of sentences in these theorems to equivalent sets of monadic second second-order sentences, it is the prescribed sets which are used to obtain lower bound results. For example, in the proof of Theorem 5.4 we actually showed that there is a prescribed set Γ of sentences over ML_0 such that $sat_T^*(ML_0) \cap \Gamma$ and $inv^*(ML_0) \cap \Gamma$ are $ATIME(T(cn), cn)$-inseparable. In Section 6 and 7 we will find lower bounds for various theories Σ from logics L by finding a reset log-lin reduction from Γ to Γ', a prescribed set of sentences over L, so that $sat_T^*(ML_0) \cap \Gamma$ is mapped into $sat^*(\Sigma) \cap \Gamma'$ and $inv^*(ML_0) \cap \Gamma$ is mapped into $inv^*(L) \cap \Gamma'$. Thus, for some $c > 0$ $sat^*(\Sigma) \cap \Gamma'$ and $inv^*(L) \cap \Gamma'$ are $ATIME(T(cn), cn)$-inseparable. Then by Theorem 3.4, $sat(\Sigma)$ and $inv(L)$ are $ATIME(T(cn), cn)$-inseparable.

6. Tools for *NTIME* lower bounds

We present several useful tools for establishing *NTIME* lower bounds for theories by interpreting models from classes of known complexity. We begin with some definitions regarding interpretations of classes of models and give a general outline of how interpretations are used to obtain lower bounds. Theorem 6.2, a specific instance of the method, follows from the results in Section 4. It tells how to obtain lower bounds by interpreting binary relations. We then show in Theorem 6.3 how to interpret binary relations in finite trees of bounded height. As a consequence we obtain hereditary lower bounds for theories of finite trees of bounded height and a tool for obtaining further lower bounds by interpreting classes of these trees in other theories. We obtain similar results for classes of finite trees of unbounded height in Theorem 6.6 and its corollaries.

Let Σ be a theory in a logic L' and $\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots$ be classes of models for a logic L whose vocabulary consists of relation symbols $P_1, \ldots, P_k$. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ be sequences of distinct variables with the length of $\boldsymbol{x}_i$ equal to the arity of P_i. Suppose that there are formulas $\delta_n(x, \boldsymbol{u}), \pi_n^1(\boldsymbol{x}_1, \boldsymbol{u}), \ldots, \pi_n^k(\boldsymbol{x}_k, \boldsymbol{u})$ from L' which are reset log-lin computable from n (expressed in unary notation) so that for each $\mathfrak{A} \in \mathscr{C}_n$ there is a model $\mathfrak{A}'$ of Σ and elements $\boldsymbol{m}$ and $\mathfrak{A}'$ with

$$\langle \delta_n^{\mathfrak{A}'}(x, \boldsymbol{m}), \pi_n^{1\mathfrak{A}'}(\boldsymbol{x}_1, \boldsymbol{m}), \ldots, \pi_n^{k\mathfrak{A}'}(\boldsymbol{x}_k, \boldsymbol{m}) \rangle$$

isomorphic to $\mathfrak{A}$. The parameter sequence $\boldsymbol{u}$ is allowed to grow as a function of n. The sequence $\{I_n \mid n \geqslant 0\}$ where

$$T_n = (\delta_n, \pi_n^1, \ldots, \pi_n^k)$$

is called an *interpretation* of the classes $\mathscr{C}_n$ in Σ. The interpretation is a *simple interpretation* if the formulas $\delta_n, \pi_n^1, \ldots, \pi_n^k$ are fixed with respect to n; this is the situation traditionally found in logic. The interpretation is a *prenex interpretation* if the formulas $\delta_n, \pi_n^1, \ldots, \pi_n^k$ are all in prenex form. The interpretation is an *iterative interpretation* if the formulas $\delta_n, \pi_n^1, \ldots, \pi_n^k$ are given by iterative definitions. By this we mean that there are formulas

$$\delta(x, \boldsymbol{u}, D),\ \pi^1(\boldsymbol{x}_1, \boldsymbol{u}_1, P_1), \ldots, \pi^k(\boldsymbol{x}_k, \boldsymbol{u}, P_k)$$

and integer functions $f, g_1, \ldots, g_k$ which are reset log-lin computable (using unary notation) such that δ_n is the formula given by the iterative definition

$$[D(x) \equiv \delta]_{f(n)}$$

and π_n^i is the formula given by the iterative definition

$$[P_i(x) \equiv \pi^i]_{g_i(n)}$$

as in Theorem 3.2. Notice that we may regard simple interpretations as special cases of either prenex or iterative interpretations.

There is a slightly more general point of view toward interpretations which is sometimes useful. Suppose $\mathscr{C}_0' \subseteq \mathscr{C}_1' \subseteq \mathscr{C}_2' \subseteq \cdots$ are classes of models of Σ such that for each $n \geqslant 0$ and $\mathfrak{A} \in \mathscr{C}_n$, the model $\mathfrak{A}'$ in the above definition can be found in $\mathscr{C}_n'$. Then we will say that we have an interpretation of the classes $\mathscr{C}_n$ in the classes $\mathscr{C}_n'$. Sometimes we will not mention Σ at all when discussing interpretations in this context. In that case we must say whether we intend the classes $\mathscr{C}_n'$ to be models for a first-order or for a monadic second-order logic. Thus, we will say that there is an interpretation of the classes $\mathscr{C}_n$ in the first-order (or monadic second-order) classes $\mathscr{C}_n'$.

Interpretations, inseparability, and prescribed sets are the cornerstones of our method. Suppose we have an interpretation of classes $\mathscr{C}_n$ in classes $\mathscr{C}_{kn}'$ for some nonnegative integer k. (In most cases k is 1 but occasionally we need a larger value.) Suppose also that there is a prescribed set Γ of formulas over L such that

$$\{\varphi \in \Gamma \mid \varphi \text{ is realized in some } \mathfrak{A} \in \mathscr{C}_n \text{ where } |\varphi| = n\}$$

and $inv^*(L)$ are $NTIME(T(cn))$-inseparable for some $c > 0$. (A formula is *realized* in $\mathfrak{A}$ if it is true in $\mathfrak{A}$ for some assignment to its free variables). Now map each formula φ in Γ to the formula φ' given by

$$[D(x) \equiv \delta_n]\,[P_1(\boldsymbol{x}_1) \equiv \pi_n^1] \cdots [P_k(\boldsymbol{x}_k) \equiv \pi_n^k]\ \varphi^D$$

where $n = |\varphi|$. By adding dummy quantifiers in the right places we can ensure that $|\varphi'| \geqslant kn$. If φ is realized in some model in $\mathscr{C}_n$, then φ' is realized in some model in $\mathscr{C}_{kn}'$. If φ is true in no model, then φ' is true in no model. (There is a minor point which should be addressed here. To be completely rigorous we should require that for all models $\mathfrak{A}'$ that $\delta_n^{\mathfrak{A}'}(x) \neq \emptyset$ since certain formulas in $inv^*(L)$ may become true when relativized to an empty relation. For example, consider the sentence $\forall x(x \neq x)$. We can always meet this requirement by replacing $\delta_n(x)$ with the formula $\delta_n(x) \vee \forall x \neg \delta_n(x)$ so we can ignore this point in subsequent discussions.) When we have a prenex interpretation, or an iterative interpretation and the definitions in φ' are replaced by the appropriate iterative definitions, the sentences φ' all belong to some prescribed set Γ' of formulas over L'. (If the interpretation is iterative, then by definition the parameter sequence cannot grow with n.) We have now that

$$\{\varphi' \in \Gamma' \mid \varphi' \text{ is realized in some } \mathfrak{A}' \in \mathscr{C}'_n \text{ where } |\varphi'| = n\}$$

and $inv^*(L')$ are $NTIME(T(cn))$-inseparable for some $c > 0$. It follows by Theorem 3.4 that $sat(\Sigma)$ and $inv(L')$ are $NTIME(T(cn))$-inseparable for some $c > 0$ so Σ has a hereditary $NTIME(T(cn))$ lower bound.

Now let us broaden the definition of interpretation to cover instances where the formulas δ_n and $\pi_n^1, \ldots, \pi_n^k$ contain free relation variables which also receive prenex or iterative definitions. For example, if these formulas contain a free unary relation variable Q we would write

$$[Q(x) \equiv \theta_n]\,[D(x) \equiv \delta_n]\,[P_1(\boldsymbol{x}_1) \equiv \pi_n^1] \cdots [P_k(\boldsymbol{x}_k) \equiv \pi_n^k]\,\varphi^D$$

for φ', the formulas θ_n having the same sort of restrictions as δ_n and $\pi_n^1, \ldots, \pi_n^k$. In this case we would have

$$I_n = (\theta_n, \delta_n, \pi_n^1, \ldots, \pi_n^k)$$

as the elements of our interpretation. If these formulas have all prenex definitions we could use Theorem 3.1 to rewrite δ_n and $\pi_n^1, \ldots, \pi_n^k$ so that Q occurs just once and substitute θ_n for occurrences of Q. For iterative definitions, we know of no similar substitution which eliminates Q and keeps the length of φ' linearly bounded in n. Fortunately, such a substitution is not needed and is even undesirable. By taking a top down approach to the construction of interpratations, building complex relations from simpler relations, we make our task easier and exposition clearer.

We extend our definitions of *simple, prenex,* and *iterative* interpretation to this more general situation. (In principle, some definitions could be prenex and others iterative, but this does not seem to occur in practice.)

We see that hereditary lower bounds are obtained using interpretations to transfer inseparability results from one prescribed set of formulas to another. One of the advantages of this method is that by establishing lower bounds in this manner, we also establish tools for proving further lower bounds. In the situation described above, if we have another prenex or iterative interpretation of classes $\mathscr{C}'_n$ in classes $\mathscr{C}''_n$ of models of Δ, then we can use $\mathscr{C}'_n$ and Γ' in place of $\mathscr{C}_n$ and Γ to establish a lower bound for Δ. Compare with the well known methods for establishing NP-hardness of a problem by reducing to a problem already known to be NP-hard. After the first lower bound or hardness result has been proved one should never again have to code Turing machines.

It is worth noting what happens when the interpretation used to establish a lower bound is not prenex or iterative. In that case we do not know that there is a reset log-lin reduction taking formulas φ' defined above to equivalent formulas in L'. As we mentioned in Section 3, the shortest equivalent formula in L' we know how to obtain has length $\Omega(n \log n)$ in the worst case. We can only conclude that $sat(\Sigma)$ and $inv(L')$ are $NTIME(T(cn/\log n))$-inseparable for some $c > 0$ so Σ has a hereditary $T(cn/\log n)$ lower bound. Successive applications of such interpretations give increasingly worse bounds. After k interpretations the lower bound would be $T(cn/(\log n)^k)$ rather than $T(cn)$. In the case where the formulas in Γ are in prenex form already we do not have this loss in the lower bound, but even then, when the interpretation is not prenex or iterative, we cannot use the classes $\mathscr{C}_n$ to obtain further lower bounds without a subsequent loss. We have introduced prenex and interative interpretations to avoid these losses. In our experience prenex and iterative interpretations not only achieve sharp lower bounds, but also are easy to manage and occur quite naturally in applications.

Within this framework we can also accommodate the more general kind of interpretation in which the domain of the interpreted model is not a subset of $\mathfrak{A}'$, but a set of k-tuples from $\mathfrak{A}'$, and the equality relation is interpreted by an equivalence relation definable in $\mathfrak{A}'$. We have found this to be necessary for only two theories treated here and so we avoided stating these definitions in the fullest generality. However, it would not have been difficult to introduce these features explicitly. (See Examples 8.7 and 8.8.)

Although we have emphasized inseparability results, we should not lose sight of the fact that the starting point for our reductions, Theorem 4.1, is a hardness result: every set Γ of satisfiable sentences with $sat_T(L_0) \subseteq \Gamma$ is hard for the complexity class

$$\bigcup_{c>0} NTIME(T(cn))$$

via reset log-lin reductions and for the class

$$\bigcup_{c>0} NTIME(T(n^c))$$

via polynomial time reductions. Thus, all our inseparability results can be reformulated as hardness results. We summarize the previous discussion and make this point precise in the following theorem.

Theorem 6.1. *Let $\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots$ be classes of models such that for some prescribed set Γ of formulas over a first-order logic L,*

$$\{\varphi \in \Gamma \mid \varphi \textit{ is realized in some } \mathfrak{A} \in \mathscr{C}_n \textit{ where } |\varphi| = n\}$$

and $inv^*(L)$ are $NTIME(T(cn))$-inseparable for some $c > 0$. Let $\mathscr{C}'_0 \subseteq \mathscr{C}'_1 \subseteq \mathscr{C}'_2 \subseteq \cdots$ be classes of models of a theory Σ in a logic L'. If there is a prenex or iterative interpretation of the classes $\mathscr{C}_n$ in the classes $\mathscr{C}'_{kn}$ for some nonegative integer k, then the following are true.

(i) *The sets $sat(\Sigma)$ and $inv(\Sigma)$ are $NTIME(T(cn))$-inseparable for some $c > 0$.*

(ii) *If for some d between* 0 *and* 1, $T(dn) = \mathrm{o}(T(n))$, *then* Σ *has a hereditary* $NTIME(T(cn))$ *lower bound.*

(iii) *For each* $\Sigma' \subseteq val(\Sigma)$, $sat(\Sigma')$ *and* $val(\Sigma')$ *are both hard for the complexity class*

$$\bigcup_{c>0} NTIME(T(cn))$$

via reset log-lin reductions.

(iv) *For each* $\Sigma' \subseteq val(\Sigma)$, $sat(\Sigma')$ *and* $val(\Sigma')$ *are both hard for the complexity class*

$$\bigcup_{c>0} NTIME(T(n^c))$$

via polynomial time reductions.

(v) *There is a prescribed set* Γ' *of sentences over* L' *such that*

$$\{\varphi' \in \Gamma' \mid \varphi' \text{ is realized in some } \mathfrak{A}' \in \mathscr{C}_n' \text{ where } |\varphi'| = n\}$$

and $inv^*(L')$ *are* $NTIME(T(cn))$*-inseparable for some* $c > 0$.

Usually when we apply our method we state the result as in (ii) for brevity, but the reader should be aware that all of the conclusions hold.

The following results is an immediate consequence of the above theorem and Theorem 4.3. It is one of the most useful tools for establishing *NTIME* lower bounds.

Theorem 6.2. *Let* $T(n)$ *be a time resource bound such that for some d between* 0 *and* 1, $T(dn) = \mathrm{o}(T(n))$. *Let* $\mathscr{C}_n$ *be the class of binary relations (i.e., structures for* L_0*) on sets of size at most* $T(n)$ *and* Σ *a theory in a logic L. If there is an interpretation of the classes* $\mathscr{C}_n$ *in* Σ, *then* Σ *has a hereditary* $NTIME(T(cn))$ *lower bound.*

The first application of this result is to first-order theories of finite trees of bounded height.

Recall that we express first-order properties of trees in the logic L_t whose vocabulary contains just a binary relation symbol which interprets the parent-child relation. Let Σ_r be the theory of finite trees of height at most r and Σ_∞ be the theory of finite trees of arbitrary height. Define $M\Sigma_r$ and $M\Sigma_\infty$ similarly for the monadic second-order logic ML_t.

Let $\mathscr{T}_k$ be the class of finite trees of height k. We inductively define certain restricted classes of finite trees in which the classes $\mathscr{C}_n$ in Theorem 6.2 are interpreted. For each $m > 0$ let $\mathscr{T}_0^m$ be the class $\mathscr{T}_0$ (all of whose elements are isomorphic). The class $\mathscr{T}_{k+1}^m$ consists of those trees whose primary subtrees are all in $\mathscr{T}_k^m$, but such that no more than m primary subtrees may be in the same isomorphism class. Clearly, $\mathscr{T}_k^m \subseteq \mathscr{T}_k$ and $\mathscr{T}_0^m \subseteq \mathscr{T}_1^m \subseteq \mathscr{T}_2^m \subseteq \cdots$. Also, if $\mathscr{T}_k^m$

contains t isomorphism types, then $\mathcal{T}^m_{k+1}$ contains at most $(m+1)^t$ isomorphism types.

From the following theorem we obtain hereditary lower bounds for theories of finite trees of bounded height and another useful tool for obtaining other lower bounds.

Theorem 6.3. *Let $\mathscr{C}_n$ be the class of binary relations on a set of size* $\exp_{r-2}(n)$ *and $m = m(n)$ be the least integer such that $m \log m \geqslant n$. Then there is a prenex interpretation of the classes $\mathscr{C}_n$ in the first-order classes $\mathcal{T}^m_r$ when $r > 3$ and in the first-order classes $\mathcal{T}^{2m}_r$ when $r = 3$.*

Proof. Note that $m = \mathrm{O}(n/\log n)$.

First, consider the case $r > 3$. Define $\varphi_m(x, y)$ to be a formula, with free relation variable Q, which says that for all children $t_1, \ldots, t_m$ of x, there are children $u_1, \ldots, u_m$ of y such that $Q(t_i, u_i)$ holds for $1 \leqslant i \leqslant m$, and

$$\bigwedge_{1 \leqslant i,j \leqslant m} t_i = t_j \leftrightarrow u_i = u_j.$$

We wish to write φ_m as a prenex formula which can be computed from n (in unary) by a reset log-lin reduction. Unfortunately, the displayed formula has length $\Omega(n^2)$ so we replace it with

$$(\forall t, t', u, u')\Big(\Big(\bigvee_{1 \leqslant i \leqslant m} (t = t_i \wedge u = u_i)$$

$$\wedge \bigvee_{1 \leqslant i \leqslant m} (t' = t_i \wedge u' = u_i)\Big) \rightarrow (t = t' \leftrightarrow u = u')\Big)$$

which is reset log-lin computable from n. Hence, we can write φ_m as a prenex formula which is reset log-lin computable from n.

When $k > 0$, the iterative definition

$$[Q(x, y) \equiv \varphi_m(x, y) \wedge \varphi_m(y, x)]_k$$

defines an equivalence relation on the vertices of height less than k. For $k = 1$, Q is an equivalence relation on the leaves, which are precisely the vertices of height 0. This relation makes all leaves equivalent. Increasing k to 2, we must extend the relation to vertices of height 1. These are the vertices adjacent to a leaf and all of whose children are leaves. Two such vertices are equivalent if they either have the same number of children or both have at least m children. In general, for larger k we extend the relation to vertices of height $k - 1$ leaving the relation unchanged for lower heights. Two vertices of height $k - 1$ are equivalent if for each equivalence class represented among their children (on which the relation has already been defined), they either have the same number of children in the class or both have at least m children in the class. When $k = r$ we have an equivalence relation on the set of all vertices in a tree of height r except the root.

By Theorem 3.2 there is a reset log-lin reduction taking the iterative definition

$$[Q(x, y) \equiv \varphi_m(x, y) \wedge \varphi_m(y, x)]_r$$

to an equivalent explicit definition

$$[Q(x, y) \equiv \psi_r^m(x, y)].$$

Moreover, since r is fixed we can arrange that ψ_r^m is a prenex formula. We will say that two vertices x and y in a tree of height r have the same *ψ_r^m-type* if $\psi_r^m(x, y)$ holds.

Define $\delta_n(r)$ to be a prenex formula that says x is a child of the root, x has at least one child, and no two distinct children of x have the same ψ_r^m-type. Define $\pi_n(x, y)$ to be a prenex formula that says $\delta(x)$ and $\delta(y)$ hold and there is a child z of the root coding (x, y). By *z coding* (x, y) we mean that for every ψ_r^m-type, if x and y have no child of that type, then neither does z; if x has a child of that type but y does not, then z has precisely two children of that type; if x has no child of that type but y does, then z has precisely three children of that type; and if x and y both have a child of that type, then z has at least four children of that type. (For this coding to work, m must be at least 4, but this is not really a problem because if $m < 4$, then $n < 5$ and we can easily formulate interpretations of $\mathscr{C}_n$ in $\mathscr{T}_r^m$ when $n < 5$.) Both $\delta_n(x)$ and $\pi_n(x, y)$ are reset log-lin computable from n.

Now is is easy to show by induction on k that if x and y are vertices of height at most k, where $k < r - 1$, and two subtrees formed by restricting to x and its descendents, and to y and its descendents, are nonisomorphic trees in $\mathscr{T}_k^m$, then x and y have different ψ_r^m-types. We know that

$$|\mathscr{T}_2^m| = (m + 1)^{m+1} > 2^n$$

and that if $|\mathscr{T}_k^m| = t$, then $|\mathscr{T}_{k+1}^m| = (m + 1)^t$ so $|\mathscr{T}_k^m| > \exp_{k-1}(n)$ when $k \geq 2$. Thus, for vertices of height $r - 2$ there are at least $\exp_{r-3}(n)$ ψ_r^m-types. If x is a child of the root satisfying $\delta_n(x)$, its children are of height at most $r - 2$ and it has either 0 or 1 children of each possible ψ_r^m-type. Thus, it is possible to distinguish between as many as $\exp_{r-2}(n)$ vertices x satisfying $\delta_n(x)$. Clearly, if $\delta_n(x)$ and $\delta_n(y)$ hold, (x, y) can be coded by some child of the root. It is easy to see that every binary relation on a set of size at most $\exp_{r-2}(n)$ is isomorphic to $\langle \delta_n^{\mathfrak{A}}(x), \pi_n^{\mathfrak{A}}(x, y)\rangle$ for some tree $\mathfrak{A}$ in $\mathscr{T}_r^m$. This concludes the case $r > 3$.

Now we consider the case $r = 3$. The construction just given shows that every binary relation on a set of size at most $2^m = 2^{O(n/\log n)}$ is isomorphic to $\langle \delta_n^{\mathfrak{A}}(x), \pi_n^{\mathfrak{A}}(x, y)\rangle$ for some tree $\mathfrak{A}$ in $\mathscr{T}_3^m$. We must work harder to remove the $\log n$ denominator in the exponent; to do this we must interpret $\mathscr{C}_n$ in $\mathscr{T}_3^{2m}$.

We begin by specifying formulas $\theta_m(x, y)$ and $\eta_m(x, y)$ which will define equivalence relations on vertices of height 2. The formula $\theta'_m(x, y)$ says that for all children $t_1, \ldots, t_m$ of x with at least 1 but no more than m children, there are children $u_1, \ldots, u_m$ of y with at least 1 but no more than m children, such that $\psi_2^m(t_i, u_i)$ holds for $1 \leq i \leq m$, and $t_i = t_j \leftrightarrow u_i = u_j$ holds for $1 \leq i,\ j \leq m$. The formula $\eta'_m(x, y)$ says that for all children $t_1, \ldots, t_m$ of x with more than m children, there are children $u_1, \ldots, u_m$ of y with more than m children, such that $\psi_2^{2m}(t_i, u_i)$ holds for $1 \leqslant i \leqslant m$, and $t_i = t_j \leftrightarrow u_i = u_j$ holds for $1 \leqslant i,\ j \leqslant m$. Using the same argument as above we can say that $\theta_m(x, y)$ is a prenex formula which is reset log-lin computable from n and equivalent to $\theta'_m(x, y) \wedge \theta'_m(y, x)$, and similarly for $\eta_m(x, y)$.

Since θ_m and η_m define equivalence relations on the set of vertices of height 2 we can speak of the θ_m-type and η_m-type of a vertex x of this height. Define $v_i(x)$ to be the minimum of m and the number of children of x with precisely i children. Define $v_i^+(x)$ to be the minimum of m and the number of children of x with at least i children. The θ_m-type of x is precisely determined by the values $v_1(x), \ldots, v_m(x)$ and the η_m-type by the values $v_{m+1}(x), \ldots, v_{2m-1}, v_{2m}^+(x)$. We see then that the θ_m-type and the η_m-type of a vertex are independent, and that there are $m^{m+1} > 2^n$ θ_m-types and the same number of η_m-types.

Let $\delta_n(x)$ be a prenex formula that says x is a child of the root and $v_0(x) = 0$. Let $\pi_n(x, y)$ be a prenex formula that say $\delta_n(x)$ and $\delta_n(y)$ hold and there is a child z of the root such that $v_0(z) > 1$ and $\theta_m(x, z)$ and $\eta_m(z, y)$ hold.

It is easy to arrange that $\delta_n(x)$ and $\pi_n(x, y)$ are reset log-lin computable from n. Each binary relation on a set of size at most 2^n is isomorphic to $\langle \delta_n^{\mathfrak{A}}(x), \pi_n^{\mathfrak{A}}(x, y)\rangle$ for some tree $\mathfrak{A}$ in $\mathcal{T}_3^{2m}$. □

Corollary 6.4. *Let $r \geqslant 3$. Σ_r has a hereditary $NTIME(\exp_{r-2}(cn))$ lower bound.*

Corollary 6.5. *Let $r \geqslant 3$ and Σ be a theory in a logic L. If there is an interpretation of the classes $\mathcal{T}_r^{n/\log n}$ in Σ, then Σ has a hereditary $NTIME(\exp_{r-2}(cn))$ lower bound.*

Remark. For each $r \geqslant 3$ there is a constant $d > 0$ such that every tree in $\mathcal{T}_r^{n/\log n}$ has at most $\exp_{r-2}(dn)$ vertices. Hence, we can view Corollary 6.5 as a significant improvement over Theorem 6.2 for obtaining $NTIME(\exp_{r-2}(cn))$ lower bounds: rather than interpreting all binary relations on sets of size $\exp_{r-2}(cn)$ we need only interpret all trees of height r on sets of this size. In applications it is often much more natural to interpret trees than binary relations. See also Theorems 7.5 and 7.8.

We next prove results similar to Theorem 6.3 and Corollaries 6.4 and 6.5 for finite trees of unbounded height.

Theorem 6.6. *Let $\mathscr{C}_n$ be the class of binary relations on a set of size $\exp_\infty(n-3)$. Then there is a prenex interpretation of the classes $\mathscr{C}_n$ in the first-order classes $\mathcal{T}_n^2$.*

Proof. Recall from the proof of Theorem 6.3 the formula ψ_r^m which defines an equivalence relation on the set of all vertices in trees of depth r except the root.

In that proof r was fixed, so we could assume that ψ_r^m was in prenex form, and m increased with n. In this proof we fix $m = 2$ and consider formulas ψ_n^2. We see now that ψ_n^2 has an iterative definition. By induction on k, there can be as many as $\exp_\infty(k)$ ψ_n^2-types among vertices of height $k < n$ in a tree in $\mathcal{T}_n^2$.

We define δ_n and π_n in essentially the way as in Theorem 6.3. The only difference is that when a vertex z coded a pair (x, y) there, z could have up to four children of the same type. Here, in trees from $\mathcal{T}_n^2$, we have at most two, so we refer to types of grandchildren rather than children. Thus, we interpret binary relations on sets of size $\exp_\infty(n-3)$ rather than $\exp_\infty(n-2)$. □

Corollary 6.7. *Σ_∞ has a hereditary NTIME$(\exp_\infty(cn))$ lower bound.*

Corollary 6.8. *If there is an interpretation of the classes $\mathcal{T}_n^2$ in a theory Σ, then Σ has a hereditary NTIME$(\exp_\infty(cn))$ lower bound.*

7. Tools for linear *ATIME* lower bounds

The theorems in this section are counterparts of those in the last section. In order to obtain linear alternating time lower bounds for logical theories we must introduce a stronger form of interpretability which we call *monadic interpretability*. Theorems 7.2 and 7.3 tell how to obtain lower bounds by monadic interpretation of addition relations and binary relations. We then show, in Theorems 7.4 and 7.6 that binary relations have monadic interpretations in certain classes of trees of bounded height. From these results we obtain useful tools for establishing linear *ATIME* lower bounds, and lower bounds for monadic second-order theories of trees of bounded height.

Suppose Σ is a theory in a logic L' and $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \ldots$ are classes of models for a monadic second-order logic ML whose vocabulary consists of relation symbols $P_1, \ldots, P_k$. Suppose that there are formulas $\delta_n(x, \boldsymbol{u})$, $\pi_n^1(\boldsymbol{x}_1, \boldsymbol{u}), \ldots, \pi_n^k(\boldsymbol{x}_k, \boldsymbol{u})$, and $\sigma_n(x, \boldsymbol{t}, \boldsymbol{u})$ in L', reset log-lin computable from n, so that for each $\mathfrak{A} \in \mathcal{C}_n$ there is a model $\mathfrak{A}'$ of Σ and elements $\boldsymbol{m}$ in $\mathfrak{A}'$ with

$$\langle \delta_n^{\mathfrak{A}'}(x, \boldsymbol{m}), \pi_n^{1\mathfrak{A}'}(\boldsymbol{x}_1, \boldsymbol{m}), \ldots, \pi_n^{k\mathfrak{A}'}(\boldsymbol{x}_k, \boldsymbol{m})\rangle$$

isomorphic to $\mathfrak{A}$ and the sets

$$\delta_n^{\mathfrak{A}'}(x, \boldsymbol{m}) \cap \sigma_n^{\mathfrak{A}'}(x, \boldsymbol{p}, \boldsymbol{m})$$

range over all subsets of $\delta_n^{\mathfrak{A}'}(x, \boldsymbol{m})$ as $\boldsymbol{p}$ ranges over $\mathfrak{A}'$. The parameter sequence $\boldsymbol{u}$ is allowed to grow as a function of n but $\boldsymbol{t}$ must remain fixed. The sequence $\{I_n \mid n \geqslant 0\}$ where

$$I_n = (\delta_n, \pi_n^1, \ldots, \pi_n^k, \sigma_n)$$

is called a *monadic interpretation* of the classes $\mathcal{C}_n$ in Σ. We define *simple,* prenex, and *iterative* monadic interpretations similarly to definitions in the last section. We also define the notion of *monadic interpretation* of classes $\mathscr{C}_n$ in classes $\mathscr{C}'_n$ as in the last section.

Evidently, a monadic interpretation of classes $\mathscr{C}_n$ is nothing more than an interpretation where the models in $\mathscr{C}_n$ are regarded as models for a monadic second-order logic. Note that if we have an interpretation of classes $\mathscr{C}_n$ in a theory Σ in some monadic second-order logic L', then we can automatically extend to a monadic interpretation by taking $\sigma_n(x, X)$ to be the formula $x \in X$ where X is a new set variable.

The framework for obtaining linear alternating time lower bounds is essentially the same as before. Suppose we have a monadic interpretation of classes $\mathscr{C}_n$ in classes $\mathscr{C}'_{kn}$ for some nonnegative integer k and that there is a prescribed set Γ of formulas over ML such that

$$\{\varphi \in \Gamma \mid \varphi \text{ is realized in some } \mathfrak{A} \in \mathscr{C}_n \text{ where } |\varphi| = n\}$$

and $inv^*(ML)$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$. Now given formula φ in ML we form $\varphi^\dagger$ as follows. Replace each monadic quantification $\forall X$ or $\exists X$ with a quantification $\forall \boldsymbol{t}_X$ or $\exists \boldsymbol{t}_X$, where $\boldsymbol{t}_X$ is a variable sequence of the same length as $\boldsymbol{t}$, uniquely determined by X, and not in conflict with the other variables in φ. (We may need to change other indices to avoid conflicts.) Introduce a relation variable S and replace each atomic formula $x \in X$ by $S(x, \boldsymbol{t}_X)$. We can easily arrange that there be a reset log-lin reduction taking φ to $\varphi^\dagger$. Now map each formula φ in Γ to the formula φ' given by

$$[D(x) \equiv \delta_n]\,[P_1(\boldsymbol{x}_1) \equiv \pi_n^1] \cdots [P_k(\boldsymbol{x}_k) \equiv \pi_n^k]\,[S(x, \boldsymbol{t}) \equiv \sigma_n]\,(\varphi^D)^\dagger$$

where $n = |\varphi|$. As before, it is easy to arrange that $|\varphi'| \geqslant kn$. If φ is realized in some model in $\mathscr{C}_n$, then φ' is realized in some model in $\mathscr{C}'_{kn}$; if φ is true in no weak model, then φ' is true in no model or weak model (depending on whether L' is first-order or monadic second-order). When we have a prenex interpretation, or an iterative interpretation and the definitions in φ' are replaced by the appropriate iterative definitions, the sentences φ' all belong to some prescribed set Γ' of formulas over L'. We have now that

$$\{\varphi' \in \Gamma' \mid \varphi' \text{ is realized in some } \mathfrak{A}' \in \mathscr{C}'_n \text{ where } |\varphi'| = n\}$$

and $inv^*(L')$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$. Then $sat(\Sigma)$ and $inv(L')$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$ so Σ has a hereditary $ATIME(T(cn), cn)$ lower bound.

As before we allow a more elaborate definition of monadic interpretations where formulas δ_n, $\pi_n^1, \ldots, \pi_n^k$, and σ_n may contain free relation variables which also receive prenex or iterative definitions.

We summarize these remarks in the following theorem which parallels Theorem 6.1.

Theorem 7.1. *Let $\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots$ be classes of models such that for some prescribed set Γ of formulas over a monadic second-order logic L,*

$$\{\varphi \in L \mid \varphi \text{ is realized in some } \mathfrak{A} \in \mathscr{C}_n \text{ where } |\varphi| = n\}$$

and $inv^*(L)$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$. Let $\mathscr{C}'_0 \subseteq \mathscr{C}'_1 \subseteq \mathscr{C}'_2 \subseteq \cdots$ be classes of models of a theory Σ in a logic L'. If there is a prenex or iterative monadic interpretation of the classes $\mathscr{C}_n$ in the classes $\mathscr{C}'_{kn}$ for some nonnegative integer k, then the following are true.

(i) *The sets $sat(\Sigma)$ and $inv(\Sigma)$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$.*

(ii) *If for some d between 0 and 1, $T(dn) = o(T(n))$, then Σ has a hereditary $ATIME(T(cn), cn)$ lower bound.*

(iii) *For each $\Sigma' \subseteq val(\Sigma)$, $sat(\Sigma')$ and $val(\Sigma')$ are both hard for the complexity class*

$$\bigcup_{c>0} ATIME(T(cn), cn)$$

via reset log-lin reductions.

(iv) *For each $\Sigma' \subseteq val(\Sigma)$, $sat(\Sigma')$ and $val(\Sigma')$ are both hard for the complexity class*

$$\bigcup_{c>0} ATIME(T(n^c), n^c)$$

via polynomial time reductions.

(v) *There is a prescribed set Γ' of sentences over L' such that*

$$\{\varphi' \in \Gamma' \mid \varphi' \text{ is realized in some } \mathfrak{A}' \in \mathscr{C}'_n \text{ where } |\varphi'| = n\}$$

and $inv^*(L')$ are $ATIME(T(cn), cn)$-inseparable for some $c > 0$.

The following theorems are immediate consequences of the preceding theorem and Theorems 5.1, 5.2, and 5.4.

Theorem 7.2. *Let $T(n)$ be a time resource bound such that for some d between 0 and 1, $T(dn) = o(T(n))$. Let $\mathscr{C}_n$ be the class of addition relations on sets of size at most $T(n)$ and Σ a theory in a logic L. If there is a monadic interpretation of the classes $\mathscr{C}_n$ in Σ, then Σ has a hereditary $ATIME(T(cn), cn)$ lower bound.*

Theorem 7.3. *The previous theorem holds with binary relations in place of addition relations.*

From the following theorem we obtain a useful tool for obtaining lower bounds, but not the best lower bounds for monadic second-order theories of trees of bounded height.

Theorem 7.4. *Let $\mathscr{C}_n$ be the class of binary relations on a set of size* $\exp_{r-1}(n)$. *Then there is an iterative monadic interpretation of the classes $\mathscr{C}_n$ in the monadic second-order classes $\mathscr{T}_r^{2^n}$ when $r > 2$ and in the monadic second-order classes $\mathscr{T}_r^{2^{2n}}$ when $r = 2$.*

Proof. The proof is very similar to the proof of Theorem 6.3. First, consider the case $r > 2$.

We iteratively define a relation $Q'(X, Y)$ which says that $|X| = |Y| \leq 2^n$. Write $Z = X \mathbin{\dot\cup} Y$ as an abbreviation for $Z = X \cup Y \wedge X \cap Y = \emptyset$. Let $\rho(X, Y)$ be the formula

$$\begin{aligned}(X = \emptyset \wedge Y = \emptyset) &\vee (|X| = 1 \wedge |Y| = 1)\\ &\vee (\exists X_1, X_2, Y_1, Y_2)(X = X_1 \mathbin{\dot\cup} X_2 \wedge Y = Y_1 \mathbin{\dot\cup} Y_2\\ &\qquad\qquad \wedge Q'(X_1, Y_1) \wedge Q'(X_2, Y_2)).\end{aligned}$$

The iterative definition $[Q'(X, Y) \equiv \rho]_{n+1}$ gives the desired relation.

Now let $\varphi(x, y)$ be a formula, with free relation variable Q, which says that if X is a subset of the set of children of x, and $Q(x_1, x_2)$ holds for all $x_1, x_2 \in X$, then there is a set Y which is a subset of the set of children of y such that $Q'(X, Y)$ holds, and $Q(y_1, y_2)$ holds for all $y_1, y_2 \in Y$, and $Q(x_1, y_1)$ holds for all $x_1 \in X$ and $y_1 \in Y$.

Consider a tree in $\mathscr{T}_r^{2^n}$. When $k > 0$, the iterative definition

$$[Q(x, y) \equiv \varphi(x, y) \wedge \varphi(y, x)]_k$$

defines an equivalence relation on the vertices of height less than k. For $k = 1$, Q is an equivalence relation making all leaves equivalent. Increasing k to 2, two vertices of height 1 are equivalent if they have the same number of children. (In Theorem 6.3 this was true only up to $O(n/\log n)$ children. This is the reason we get an additional level of exponentiation here.) For larger k extend to vertices of height $k - 1$ leaving the relation unchanged at lower heights. Two vertices of height $k - 1$ are equivalent if for each equivalence class represented among their children (on which the relation has already been defined), they either have the same number of children in the class or both have at least 2^n children in the class. There is a reset log-lin reduction taking the iterative definition

$$[Q(x, y) \equiv \varphi(x, y) \wedge \varphi(y, x)]_r$$

to an equivalent explicit definition

$$[Q(x, y) \equiv \psi_r^n(x, y)].$$

Moreover, since r is fixed this can be expressed as a simple definition. The rest of the theorem proceeds as in the proof of Theorem 6.3 except that ψ_r^n is used in place of ψ_r^m, $\mathscr{T}_r^{2^n}$ is used in place of $\mathscr{T}_r^m$, and we automatically have a monadic interpretation since we are interpreting in a monadic second-order theory.

Let us consider the case $r = 2$. Our formulas will now have parameters U, V, W, and Z. Let $\theta_2^n(x, y, U)$ and $\eta_2^n(x, y, V)$ be defined as $\psi_2^n(x, y)$ above except that rather than comparing the number of children of vertices x and y of height 1, $\theta_2^n(x, y, U)$ compares the number of children of x and y in U, and $\theta_2^n(x, y, V)$ compares the number of children of x and y in V. Now let $\delta_n(x)$ be the formula $x \in Z$. Let $\pi_n(x, y)$ be a prenex formula that asserts the following.

(a) $x \in Z$ and $y \in Z$.

(b) If $x \neq y$ there is a $t \notin Z$ such that $\theta_2^n(x, y)$ holds and $\eta_2^n(t, y)$ holds.

(c) If $x = y$, then $x \in W$.

Every binary relation on a set of size at most 2^n is interpreted in some tree in $\mathcal{T}_2^{2^{2n}}$. For every pair (x, y) in the relation with $x \neq y$ there is at least one $t \notin Z$ such that $\theta_m(x, t)$ and $\eta_m(t, y)$ hold. □

Corollary 7.5. *Let $r \geqslant 2$ and Σ be a theory in a logic L. If there is a monadic interpretation of the classes $\mathcal{T}_r^{2^n}$ in Σ, then Σ has a hereditary*

$$ATIME(\exp_{r-1}(cn), cn)$$

lower bound.

Remark. For each $r \geqslant 2$ there is a constant $d > 0$ such that every tree in $\mathcal{T}_r^{2^n}$ has at most $\exp_{r-1}(dn)$ vertices, so we can view Corollary 7.5 as an improvement over Theorem 7.3 for obtaining $ATIME(\exp_{r-1}(cn), cn)$ lower bounds. It is instructive to compare this result with Corollary 6.5. It often happens that an interpretation of the classes $\mathcal{T}_r^{n/\log n}$ in a theory can be modified slightly to obtain a monadic interpretation of the classes $\mathcal{T}_{r-1}^{2^n}$, even when the theory is first-order. This explains, in part, why *NTIME* lower bounds can often be pushed up to linear *ATIME* lower bounds.

We now prove another theorem about monadic interpretations of classes of trees of bounded height. We shall see in Section 9 that this theorem gives the best lower bounds for monadic second-order theories of trees of bounded height.

Theorem 7.6. *Let $\mathscr{C}_n$ be the class of binary relations on a set of size $\exp_r(\lfloor n/\log n \rfloor)$. Then there is a prenex monadic interpretation of the classes $\mathscr{C}_n$ in the monadic second-order classes $\mathcal{T}_r^{\exp_2(\lfloor n/\log n \rfloor)}$ when $r > 1$ and in the monadic second-order classes $\mathcal{T}_r^{\exp_1(\lfloor 2n/\log n \rfloor)}$ when $r = 1$.*

Proof. First, consider the case $r > 1$.

Formulas will contain parameters $\boldsymbol{X} = X_1, X_2, \ldots, X_m$, where $m = 1 + \lfloor n/\log n \rfloor$. This is the first case we have encountered where the parameter sequence grows with n. Let Q' be a binary relation variable and $\theta_m(x, y)$ a prenex formula that says x and y are both leaves and

$$\bigwedge_{1 \leqslant i \leqslant m} x \in X_i \leftrightarrow y \in X_i.$$

Q' will be given by the prenex definition $[Q'(x, y) \equiv \theta_m]$. $Q'(x, y)$ is obviously an equivalence relation on the vertices of height 0.

Now let $\varphi(x, y)$ be a formula (with free relation variable Q) that says x is not a leaf and for every child t of x there is a child u of y such that $Q(t, u)$ holds. Now the iterative definition

$$[Q(x, y) \equiv Q'(x, y) \vee (\varphi(x, y) \wedge \varphi(y, x))]_k$$

defines an equivalence relation on the vertices of heights less than k. For $k = 1$, Q is identical to the equivalence relation Q'. For larger k we extend the relation to vertices of height $k-1$ by specifying that two such vertices are equivalent if precisely the same equivalence classes are represented among their children. When $k = r$ we have an equivalence relation on the set of all vertices in a tree of depth r except the root.

The iterative definition

$$[Q(x, y) \equiv Q'(x, y) \vee (\varphi(x, y) \wedge \varphi(y, x))]_r$$

can be converted to a prenex definition

$$[Q(x, y) \equiv \psi_r(x, y)]$$

of fixed length since r is constant. By Theorem 3.1 the sequence

$$[Q'(x, y) \equiv \theta_m][Q(x, y) \equiv \psi_r(x, y)]$$

of prenex definitions can be replaced by a single prenex definition

$$[Q(x, y) \equiv \psi_r^m(x, y)]$$

where $\psi_r^m(x, y)$ is reset log-lin computable from n. We will say that two vertices x and y in a tree of height r have the same ψ_r^m*-type* if $\psi_r^m(x, y)$ holds.

Now it is easy to show by induction on k that there is a tree of height r and sets $X_1, \ldots, X_m$ of leaves in this tree such that there are at least $1 + \exp_{k+1}(\lfloor n/\log n \rfloor)$ ψ_r^m-types among vertices of height k when $k < r$: every nonempty set of ψ_r^m-types of vertices of height $k-1$ determines a distinct ψ_2^m-type for a vertex of height k. More is required to see that there is such a tree in $\mathcal{T}_r^{\exp_2(\lfloor n/\log n \rfloor)}$. When $k = 0$ there is no problem. Consider the case $k = 1$. For $i = 1, 2, \ldots, \exp_2(\lfloor n/\log n \rfloor)$ let τ_i be the tree of height 1 with precisely i leaves. Each of these trees is in $\mathcal{T}_1^{\exp_2(m)}$. Without much trouble we can choose, from the leaves of each τ_i, sets $X_1, \ldots, X_m$ so that the roots of τ_i and τ_j have different ψ_r^m-types when $i \neq j$. Further, for τ_1, $X_1, \ldots, X_m$ can be chosen in at least two ways. Thus, there are at least $1 + \exp_2(\lfloor n/\log n \rfloor)$ ψ_r^m-types among vertices of height 1.

From the set of trees $\{\tau_i \mid 1 \leq i \leq \exp_2(m)\}$ choose a nonempty subset and form a tree of height 2 by directing edges from a new vertex to the roots of trees in this subset. We can form at least $1 + \exp_2(\lfloor n/\log n \rfloor)$ such trees, each one in $\mathcal{T}_2^{\exp_2(\lfloor n/\log n \rfloor)}$. Moreover, if we carry along the subsets $X_1, \ldots, X_m$ in each subtree τ_i, each root has a distinct ψ_r^m-type. Continue for each $k<r$, forming at least $1+\exp_{k+1}(\lfloor n/\log n\rfloor)$ trees in $\mathcal{T}_k^{\exp_2(\lfloor n/\log n\rfloor)}$, with subsets $X_1, \ldots, X_m$, so that each root has a distinct ψ_r^m-type.

The interpretation of binary relations on sets of size $\exp_r(\lfloor n/\log n\rfloor)$ now uses precisely the same construction used in the proofs of Theorems 6.3 and 7.4.

Let us consider the case $r=1$. Our formulas will now have parameters $\boldsymbol{X}=X_1, \ldots, X_m$, $\boldsymbol{Y}=Y_1, \ldots, Y_m$, and Z and W. Define $\theta_m(x, y)$ as above and $\eta_m(x, y)$ in the same way except that $\boldsymbol{Y}$ is used in place of $\boldsymbol{X}$. These formulas define independent equivalence relations on leaves so we can speak of the θ_m- and η_m-type of a leaf. The relations are independent and of index at most 2^m. The rest of the proof then follows as the in case $r=2$ in Theorem 7.4. □

Corollary 7.7. *Let $r\geqslant 1$. $M\Sigma_r$ has a hereditary $ATIME(\exp_r(cn/\log n), cn)$ lower bound.*

Corollary 7.8. *Let $r\geqslant 2$ and Σ be a theory in a logic L. If there is a monadic interpretation of the classes $\mathcal{T}_r^{\exp_2(n/\log n)}$ in Σ, then Σ has a hereditary*

$$ATIME(\exp_r(cn/\log n), cn)$$

lower bound.

The case $r=1$ is worth stating separately. Observe that trees of height 1 do not really have much structure. We can regard them as sets with one distinguished point, the root. Therefore, we state the result in terms of interpretations of sets rather than trees.

Corollary 7.9. *Let $\mathcal{C}_n$ be the class of sets of size at most $2^{n/\log n}$ and Σ a theory in a logic L. If there is a monadic interpretation of the classes $\mathcal{C}_n$ in Σ, then Σ has a hereditary $ATIME(2^{cn/\log n}, cn)$ lower bound.*

This concludes our survey of tools for establishing lower bounds. The next section contains many examples of their application.

8. Applications

In this section we use the methods developed in earlier sections to give a representative sample of arguments for known lower bounds of theories. We believe that the details given here justify the claim that every known lower bound for a theory can be obtained in this way, with simpler, more conceptual proofs. In particular, there is no further need to code Turing machine computations. Moreover, in almost all cases our approach gives technical improvements on known results: we always obtain hereditary lower bounds; these bounds hold for both $sat(\Sigma)$ and $val(\Sigma)$, in contrast to most published *NTIME* lower bounds which are for just $sat(\Sigma)$; and the reductions used are reset log-lin reductions rather than polynomial time, linearly bounded reductions. In a few cases we obtain qualitative improvements in the bounds. The most significant improvement is that we always obtain inseparability results.

To simplify the statement of results and avoid repetition, when we say a theory Σ has a hereditary $NTIME(T(cn))$ lower bound, we intend each of the statements (i)–(iv) in Theorem 6.1. When we say a theory Σ has a hereditary $ATIME(T(cn), cn)$ lower bound, we intend each of the statements (i)–(iv) in Theorem 7.1. For convenience we will also use functional notation rather than relations in some examples.

Example 8.1. *The first-order theory of finite linear orders with an added unary predicate.*

We show that this theory has a hereditary lower bound of

$$NTIME(\exp_\infty(cn))$$

by iteratively interpreting the classes $\mathcal{T}_n^2$ and applying Corollary 6.8. In fact, we interpret the classes $\mathcal{T}_n$ consisting of the finite trees of height n.

We denote the linear order by $\leq$ and the predecessor and successor of an element x by $x-1$ and $x+1$ (when they exist). Let $x<y$ be the formula $x \leq y \wedge x \neq y$ and $LAST(x)$ the formula asserting that x is the last element in the order. Let P be the unary relation symbol. We can identify each finite model $\mathfrak{A} = \langle m, R\rangle$ of this theory (where $m = \{0, \ldots, m-1\}$) with a string $a_0a_1 \cdots a_{m-1}$ of 0's and 1's. We stipulate that $a_i = 1$ if and only if $P(i)$ holds.

Representing a finite tree as a string of 0's and 1's is straightforward. Now with each vertex x in a tree of height n associate a string that begins $0^{n-r+1}1$, where r is the depth of x, followed by the strings associated with each child y of x. The tree is represented by the string associated with the root. For example, the tree of Fig. 1 is represented by the string

$$0001001010100100101.$$

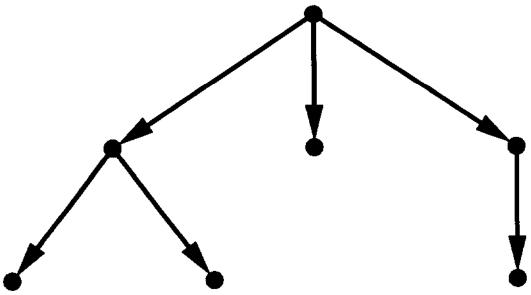

Fig. 1. Tree example.

In general, τ may have many representations, since the children of each vertex are ordered arbitrarily. Thus, the tree above is also represented by the string

$$0001001001010010101.$$

However, τ can be easily interpreted within each of its representations $\mathfrak{A}$.

Let $\delta_n(x)$ be the formula $P(x)$ (indicating that x is a position where 1 occurs). Let $\theta(x, y)$ be the following formula with free relation variable Q:

$$\begin{aligned}&((x = 0 \vee P(x-1)) \wedge (y = 0 \vee P(y-1)))\\&\quad \vee (\neg P(x-1) \wedge \neg P(y-1) \wedge Q(x-1, y-1)).\end{aligned}$$

Then $[Q(x, y) \equiv \theta]_{n+1}$ defines a relation $Q(x, y)$ which holds precisely when the number of consecutive 0's preceding position x and the number of consecutive 0's preceding position y are equal and at most n. Let $\pi_n(x, y)$ be the formula

$$x < y \wedge P(x) \wedge P(y) \wedge \neg P(x-1) \wedge Q(x-1, y) \wedge \forall z\, (x < z < y \rightarrow \neg Q(x, z)).$$

Thus, $\pi_n(x, y)$ holds precisely when there are 1's at positions x and y, x precedes y, there is exactly one more 0 preceding x then preceding y (but no more than n 0's preceding y), and there is no position between x and y which has as many 0's preceding as x has. Now a tree τ of height n represented by a linear order $\mathfrak{A}$ is isomorphic to $\langle \delta_n^{\mathfrak{A}}(x), \pi_n^{\mathfrak{A}}(x, y)\rangle$ when Q is given by the iterative definition $[Q(x, y) \equiv \theta]_{n+1}$.

Remark. Note that for some $d > 0$ each tree in $\mathcal{T}_n^2$ has at most $\exp_\infty(dn)$ vertices so for some $d' > 0$ each representation $\mathfrak{A}$ of a tree in this class has at most $\exp_\infty(d'n)$ elements. From Theorem 6.1 we see that we have a tool for obtaining further lower bounds. If there is a prenex or iterative interpretation of the classes $\mathcal{C}_n$ consisting of linear orders of length at most $\exp_\infty(n)$ with added unary predicates in a theory Σ, then Σ has a hereditary lower bound of $NTIME(\exp_\infty(cn))$.

Example 8.2. *The first-order theory of all linear orders.*

We show that this theory has a hereditary lower bound of

$$NTIME(\exp_\infty(cn))$$

by a simple interpretation of the models in Example 8.1.

Consider a finite linear order $\langle n, \leqslant\rangle$ together with a unary relation R on n interpreting the predicate symbol P. We will represent $\langle n, \leqslant, R\rangle$ by a linear order $\mathfrak{A} = \langle S, \leqslant\rangle$ formed by replacing each $i \in n$ with a copy of the closed unit interval $[0, 1]$ if i is not in R and with a single point followed by a copy of the closed unit interval $[0, 1]$ if i is in R; the order is extended in the obvious way.

Now we interpret $\langle n, \leqslant, R\rangle$ in $\mathfrak{A}$ as follows. Let $\delta(x)$ be the first-order formula saying that x has no immediate successor and x is either the least element or has an immediate predecessor. Clearly, $\delta(x)$ picks out all the left endpoints of the unit intervals in $\langle S, \leqslant\rangle$. There are precisely n elements in $\delta^{\mathfrak{A}}(x)$. Let $\pi(x)$ be the first-order formula saying that x has not immediate successor but does have an immediate predecessor y such that either y has no immediate predecessor or is the least element. Thus, $\pi(x)$ picks out the left endpoints of the unit intervals associated with elements i in R. Thus, $\langle n, \leqslant, R\rangle$ is isomorphic to $\langle \delta^{\mathfrak{A}}(x), \leqslant^{\mathfrak{A}}(x, y), \pi^{\mathfrak{A}}(x)\rangle$.

Remark. Robertson [59] and Stockmeyer [68] showed independently that the first-order theory of finite linear orders with an added unary predicate is not elementary recursive. They obtained an

$$NTIME(\exp_\infty(c \log n))$$

lower bound, but not in the hereditary form. Stockmeyer obtained the same bound for the first-order theory of linear orders.

Stockmeyer shows in the same paper that if $\langle A, \leqslant\rangle$ is any infinite linear order, then the theory of the models $\langle A, \leqslant, R\rangle$, where R is an added unary relation, is not elementary recursive. A simple interpretation of Example 8.1 in this theory shows that it has a hereditary $NTIME(\exp_\infty(cn))$ lower bound.

Of course, from Example 8.1 also follows the result of Meyer [49] that the weak monadic second-order theory of the natural numbers with successor is not elementary recursive and, hence, the same result for other monadic second-order theories, including the monadic second-order theory of two successors studied by Rabin [54]. In all cases we have hereditary $NTIME(\exp_\infty(cn))$ lower bounds.

Another result that can be immediately obtained in this way is the following strengthening of a result announced in Mayer [48].

Example 8.3. *The first-order theory of* $\langle \{0, 1\}^*, r_0, r_1, \leqslant\rangle$, *the binary tree formed by taking the prefix order* $\leqslant$ *on* $\{0, 1\}^*$, *together with successor functions* r_0 *and* r_1. (*Thus,* $u \leqslant w$ *precisely when there is a* v *such that* $uv = w$, *and* $r_0(w) = w0$, $r_1(w) = w1$.)

We show that this theory has a hereditary lower bound of

$$NTIME(\exp_\infty(cn))$$

by a simple interpretation of the classes in Example 8.1.

Let $\mathfrak{A} = \langle \{0, 1\}^*, r_0, r_1, \leqslant\rangle$. We produce formulas $\delta(x, w)$, $\lambda(x, y, w)$, and $\pi(x, w)$ such that

$$\langle \delta^{\mathfrak{A}}(x, w), \lambda^{\mathfrak{A}}(x, y, w), \pi^{\mathfrak{A}}(x, w)\rangle$$

includes all models $\langle n, \leqslant, R\rangle$ in Example 8.1 as w ranges over $\{0, 1\}^*$. The formula $\delta(x, w)$ is $x \leqslant w \wedge x \neq w$. The formula $\lambda(x, y, w)$ is

$$\delta(x, w) \wedge \delta(y, w) \wedge x \leqslant y.$$

The formula $\pi(x, w)$ is $\delta(x, w) \wedge r_1(x) \leqslant w$.

Example 8.4. *The first-order theory of finite unary functions.*

A unary function is a model $\langle B, f\rangle$ where f is a function from B to itself. We show that this theory has a hereditary lower bound of

$$NTIME(\exp_\infty(cn))$$

by a simple interpretation of the classes $\mathcal{T}_n^2$ and applying Corollary 6.8. In fact, just as in Example 8.1, we interpret the classes $\mathcal{T}_n$ consisting of the finite trees of height n.

Let $\delta(x)$ be the formula $x = x$ and $\varphi(x, y)$ be the formula

$$x = f(y) \wedge x \neq y.$$

Clearly, every finite tree is isomorphic to a model $\langle \delta^{\mathfrak{A}}(x), \pi^{\mathfrak{A}}(x, y)\rangle$ for some unary function $\mathfrak{A}$. $\mathfrak{A}$ is formed from the tree by mapping every vertex other than the root to its parent and the root to itself.

Remark. A lower bound of $NTIME(\exp_\infty(c \log n))$ for the first-order theory of unary functions was announced by Meyer [48], but no proof has ever been published.

A simple interpretation in the opposite direction (interpreting unary functions in trees) appears in Korec and Rautenberg [40]. Thus, the theory of finite unary functions and the theory of finite trees have the same complexity up to a constant factor in the argument.

Our next example gives another family of useful theories which are not elementary recursive: the theories of pairing functions. Lower bounds for these theories were first given by Rackoff [56] (see Ferrante and Rackoff [24]). Their treatment shows that these theories are in fact *hereditarily* not elementary recursive. (To obtain this from the result stated by Ferrante and Rackoff, we must use the fact that the theory of pairing functions is finitely axiomatizable.)

Example 8.5. *The theory of any pairing function.*

A pairing function is a model $\mathfrak{A} = \langle B, f\rangle$ where f is a one-to-one binary function on B. We show that the theory of any pairing function has a hereditary lower bound of

$$NTIME(\exp_\infty(cn))$$

by iteratively interpreting the classes $\mathscr{C}_n$ of linear orders of length $\exp_\infty(n)$ with an added unary predicate. These classes were discussed in Example 8.1.

The idea behind the interpretation is to have elements of $\mathfrak{A}$ represent sequences of length $\exp_\infty(n)$. One way to do this, say for an element a_ε, is to find a pair (a_0, a_1) such that $f(a_0, a_1) = a_\varepsilon$. (There is at most one such pair, since f is a pairing function.) Then find a quadruple $(a_{00}, a_{01}, a_{10}, a_{11})$ such that $f(a_{00}, a_{01}) = a_0$ and $f(a_{10}, a_{11}) = a_1$ and repeat until we have a sequence $(a_w \mid w \in \{0, 1\}^m)$, where $m = \exp_\infty(n-1)$. (The order on $\{0, 1\}^m$ is the lexicographic order.) We could then say that a_ε represents this sequence. We may not be able to carry out this construction for every element a_ε because f may not be onto, but certainly every sequence of length $\exp_\infty(n)$ is represented by some element, and every element represents at most one sequence of length $\exp_\infty(n)$.

We may regard this construction as giving a labeling of vertices in the full binary ordered tree of height m. The root is labeled a_ε, its left and right children are labeled a_0 and a_1, respectively, and so on. Unfortunately, the construction has two limitations that make it unacceptable for our purposes: a branch in the tree may have several vertices with the same label; and two different branches may be labeled identically. We must modify the construction to overcome these difficulties. Rather than taking a_{w0} and a_{w1} so that $f(a_{w0}, a_{w1}) = a_w$, we will take b and c so that $f(b, c) = a_w$ and then take a_{w0} and a_{w1} so that $f(a_{w0}, a_{w1}) = c$. That is, a_{w0} and a_{w1} are chosen so that $\exists x f(x, f(a_{w0}, a_{w1})) = a_w$ holds. Clearly, a_{w0} and a_{w1} are uniquely determined by a_w. It is still true that every element represents at most one sequence of length $\exp_\infty(n)$, but now a sequence of length $\exp_\infty(n)$ may be represented by many elements.

We claim that with this modified construction, every sequence of length $\exp_\infty(n)$ is represented by some element such that no branch in the associated tree has duplicate labels and no two branches are labeled identically. Consider a sequence

$$(a_w \mid w \in \{0, 1\}^m)$$

where $m = \exp_\infty(n-1)$. We can find elements a_w for each $w \in \{0, 1\}^k$ with $k < m$ such that $\exists x f(x, f(a_{w0}, a_{w1})) = a_w$ always holds; whenever v is a prefix of w, $a_v \neq a_w$; and $a_{w0} \neq a_{w1}$. The elements are selected in a bottom up fashion, i.e., working from the leaves of the tree up to the root. Suppose that elements a_v are already known when w lexicographically precedes v and we wish to find a_w. Since f is a pairing function, $f(x, f(a_{w0}, a_{w1}))$ takes distinct values as x ranges over the model. Thus, we can choose x so that $a_w = f(x, f(a_{w0}, a_{w1}))$ is not equal to a_v when w lexicographically precedes v since there are only finitely many such v. Following this procedure we eventually reach a_ε and all labels in the tree are distinct.

Let $LEFT(x, y)$ be the forula $(\exists t, u)\, f(t, f(y, u)) = x$ and $RIGHT(x, y)$ be the formula $(\exists t, u)\, f(t, f(u, y)) = x$. Thus, $LEFT(x, y)$ holds when y is the left child of x in our tree, and similarly for $RIGHT(x, y)$. Let $SUCC(x, y)$ be the formula $LEFT(x, y) \vee RIGHT(x, y)$.

We wish to specify three first-order formulas. Formula $\delta_n(x, u)$ should say that x in an element in a sequence of length $\exp_\infty(n)$ which is represented by u; that the elements of this sequence are distinct; and that no branch in the associated tree has duplicate labels. Formula $\lambda_n(x, y, u)$ should say that $\delta_n(x, u)$ and $\delta_n(y, u)$ hold, and that either $x = y$ or in the sequence represented by u, x occurs before y. Formula $\sigma_n(x, x', u, u')$ should say that $\delta_n(x, u)$ and $\delta_n(x', u')$ both hold, and that x occupies the same position in the sequence represented by u as x' does in the sequence represented by u'.

Now suppose that we already have relations

$$Q_0(x, u), \quad Q_1(x, y, u), \quad \text{and} \quad Q_2(x, x', u, u')$$

that hold under the same conditions as $\delta_n(x, u)$, $\lambda_n(x, y, u)$, and $\sigma_n(x, x', u, u')$, only for sequences of length $\exp_\infty(n-1)$ rather than $\exp_\infty(n)$. (If $n = 0$, take Q_0, Q_1, and Q_2 to be empty relations.)

Suppose $n > 0$. Using Q_0 and Q_1, we can say, in regard to a sequence of distinct elements of length $\exp_\infty(n-1)$ represented by v, that a particular element is first; also, that a particular element last; also, that one element occurs before another; and that one element occurs immediately before another. Hence, we can say that v represents a *branch* of length $\exp_\infty(n-1)$ from an element u. By this we mean v represents a sequence of distinct elements of length $\exp_\infty(n-1)$ such that when x is the first element of the sequence, $SUCC(u, x)$ holds, and when x occurs immediately before y in the sequence, $SUCC(x, y)$ holds.

Now define formulas δ, λ, and σ in terms of Q_0, Q_1, and Q_2. The formula $\delta(x, u)$ says that if Q_0 is empty, then $x = u$, and if Q_0 is not empty, the following hold.

(a) There is an element v representing a branch of length $\exp_\infty(n-1)$ from u with x as the last element.

(b) For every element y occurring in a branch of length $\exp_\infty(n-1)$ from u, if y is not the last element in the branch, then there are distinct elements z and z' such that $LEFT(y, z)$ and $RIGHT(y, z')$ hold. This ensures that the binary tree is full and that no two branches in the tree are labeled identically.

(c) If v and v' represent different branches of length $\exp_\infty(n-1)$ from u, then the last elements of these two branches are distinct. (It is easy to tell if v and v' represent different branches; this happens if and only if there are lements $y \neq y'$ such that $Q_2(y, y', v, v')$ holds.)

The formula $\lambda(x, y, u)$ says that if Q_1 is empty, then $x = y = u$, and if Q_1 is not empty, then either $x = y$, or the following hold.

(a) There are branches from u, represented by v and v' and with last elemengs x and y, of length $\exp_\infty(n-1)$.

(b) There is an element z such that if $LEFT(z, t)$ and $RIGHT(z, t')$ hold, then $Q_2(t, t', v, v')$ holds. This guarantees that the branch from u to x is to the left of branch from u to y.

The formula $\sigma(x, x', u, u')$ says that if Q_2 is empty then $x = u$ and $x' = u'$, and if Q_2 is not empty, then the following hold.

(a) There is a branch from u, represented by v and with last element x. There is another from u', represented by v' and with last element x', of length $\exp_\infty(n-1)$.

(b) If $Q_2(z, z', v, v')$ holds, z occurs immediately before t in the branch represented by v, and z' occurs immediately before t' in the branch represented by v', then $LEFT(z, t)$ holds just in case $LEFT(z', t')$ holds. This says that the same sequence of lefts and rights leads from u to x as from u' to x'.

Clearly, δ, λ, and σ are first-order expressible in terms of Q_0, Q_1, and Q_2. The simultaneous iterative definition

$$\begin{bmatrix} Q_0(x, u) & \equiv \delta \\ Q_1(x, y, u) & \equiv \lambda \\ Q_2(x, x', u, u') & \equiv \sigma \end{bmatrix}_{n+1}$$

gives the desired formulas δ_n, λ_n and σ_n. By Theorem 3.3, this simultaneous iterative definition can be replaced by just an iterative definition..

For some element u, $\langle \delta^{\mathfrak{A}}(x, u), \lambda^{\mathfrak{A}}(x, y, u)\rangle$ is a linear order of length $\exp_\infty(n)$. Then as u' ranges over the elements in $\mathfrak{A}$, $\delta^{\mathfrak{A}}(x, u) \cap \delta^{\mathfrak{A}}(x, u')$ ranges over all unary predicates on $\delta^{\mathfrak{A}}(x, u)$. Thus, we can interpret every linear order of length $\exp_\infty(n)$ with an added unary predicate.

This is the first example we have encountered where the relations being iteratively defined do not occur positively in the operator formulas. Hence, according to the remark following Theorem 3.1, we cannot guarantee that the lower bounds obtained here hold if only the connectives $\wedge$, $\vee$, and $\neg$ are allowed in formulas. However, with slightly more effort we can overcome this difficulty and use only formulas where the defined relation symbols occur only positively. We have not done so to simplify exposition.

Remark. It is a long-standing open question whether the first-order theory of the free group F_k on $k \geqslant 2$ generators is decidable. Semenov [65] observed that this theory is at least not elementary recursive. He showed that it is possible to give a first-order definition of a pairing function on F_k, from which it follows that the theory of F_k has a hereditary $NTIME(\exp_\infty(cn))$ lower bound.

This use of pairing functions is a quick way to show that many decidable theories are hereditarily no elementary recursive. For example, consider the first-order theory of the model $\langle \mathbb{N}, +, 2^x\rangle$, where $\mathbb{N}$ is the set of nonnegative integers. Semenov [66] showed that this theory is decidable. Observe that the function

$$f(x, y) = 2^{2x} + 2^{2y+1}$$

is a pairing function on this model, so the theory has a hereditary $\exp_\infty(cn)$ lower bound. Variations on the same argument apply to the other models $\langle \mathbb{N}, +, g(x)\rangle$ treated by Semenov [66] when $g(x)$ grows rapidly enough. In particular, when $g(x+1) \geqslant 2g(x)$, the argument gives a hereditary $\exp_\infty(cn)$ lower bound.

This completes our discussion of theories which are not elementary recursive. Note that in all cases treated here, there is an iterative interpretation of the classes $\mathcal{T}_n^2$ in the theory treated. (Sometimes, there is an iterative interpretation of the classes $\mathcal{T}_n^2$ into classes $\mathcal{C}_n$ and another iterative interpretation of the classes $\mathcal{C}_n$ in the theory; the chain of interpretations may be even longer. Nonetheless, these situations still quality as interpretations of $\mathcal{T}_n^2$.) It seems likely that almost all 'natural' theories that are hereditarily not elementary recursive interpret finite trees (in the same sense that almost all 'natural' theories that are undecidable interpret finite binary relations). Of course, an interpretation of trees can be traced back, in Theorem 6.3, to an interpretation of binary relations on sets of size $\exp_\infty(n)$, and thence, in Theorem 4.1, to a coding of Turing machines with a $\exp_\infty(n)$ time resource bound. However, the Turing machine encodings lie very far below the surface and are extremely complicated compared to the original interpretation of trees.

We now treat various elementary recursive theories for which sharp lower bounds are known.

Example 8.6. *The first-order theory of models $\langle \mathbb{N}, S, P\rangle$, where $\mathbb{N}$ is the nonnegative integers, S is the successor function on $\mathbb{N}$, and P is an arbitrary unary relation.*

We show that this theory has a hereditary lower bound of

$$NTIME(\exp_2(cn))$$

by iteratively interpreting the classes $\tau_4^{n/\log n}$ and applying Corollary 6.5.

We use the same coding of trees into strings of 0's and 1's that was used in Example 8.1. Given a tree $\tau \in \mathcal{T}_4^{n/\log n}$, let $a_0 a_1 \cdots a_{m-1}$ be a string that represents it and P a unary relation on $\mathbb{N}$ such that $P(i)$ holds if and only if either $i < m$ and $a_i = 1$, or i is m or $m+1$. We want to recover τ in $\langle \mathbb{N}, S, P\rangle$. The problem is that we have only a successor function rather than the linear order we had in Example 8.1. When we examine the role played by the linear order there, however, we see that we did not need the full linear order. We needed only enough of the order to say $x < y$ when y is a child of x, and that there is no position between x and y which has as many 0's preceding as x has when y is a child of x. Moreover, we did not need to make either statement when x is the root because there are no other positions in the coding which have as many 0's preceding as x has. Thus, we needed only to determine if $x \leqslant y$ when x and y belong to the same primary subtree. For $\tau \in \mathcal{T}_4^{n/\log n}$ every primary subtree is in $\mathcal{T}_3^{n/\log n}$. It is not difficult to see that for some d, the trees in $\mathcal{T}_3^{n/\log n}$ are all represented by strings of length at most 2^{dn}. Thus, if we can define a relation which holds when $x \leqslant y$ and $y - x \leqslant 2^{dn}$ we are done. Let θ be the formula $x = y \vee S(x) = y \vee \exists z\, (Q(x, z) \wedge Q(z, y))$. The iterative definition $[Q(x, y)] \equiv [\theta]_{dn}$ gives the desired relation.

Remark. This proof shows that if there is a prenex or iterative interpretation of the classes $\mathcal{C}_n$ consisting of successor relations of length at most $\exp_2(n)$ with added unary predicates in a theory Σ, then Σ has a hereditary lower bound of $NTIME(\exp_2(cn))$. The same argument shows that if there is a prenex or iterative interpretation of the classes $\mathcal{C}'_n$ consisting of successor relations of length at most 2^n with added unary predicates in a theory Σ, then Σ has a hereditary lower bound of $NTIME(2^{cn})$.

Example 8.7. *The theory of one-to-one unary functions.*

We show that this theory has a hereditary lower bound of

$$NTIME(2^{cn})$$

by iteratively interpreting successor relations of length at most 2^n with an added unary predicate (as discussed in the preceding remark).

Let $\mathscr{C}_n$ be the class of finite one-to-one unary functions $\langle B, f\rangle$ such that the cycle decomposition has, for each k with $1 \leqslant k \leqslant 2^n$, at least one and not more than two cycles of length k. We regard each such model as coding a unary prediate P on $\{1, 2, \ldots, 2^n\}$, the presence of a single k-cycle indicating that $\neg P(k)$ holds and the presence of two k-cycles indicating that $P(k)$ holds.

Define the *distance* between elements x and y on the same cycle to be the least i such that $f^i(x) = y$. For elements x, y, and z on the same k-cycle, we say that y is *between* x and z if the distance from x to z is the sum of the distances from x to y and y to z. Now it is a simple exercise to give a simultaneous iterative definition

$$\begin{bmatrix} Q_1(x, y, z) & \equiv \theta_1 \\ Q_2(x, y, x', y') & \equiv \theta_2 \end{bmatrix}_n$$

such that $Q_1(x, y, z)$ holds precisely when the distance from x to y and the distance from y to z are at most 2^n and y is between x and z, and $Q_2(x, y, x', y')$ holds precisely when the distance from x to y is the same as the distance from x' to y' but not more than 2^n.

Assume that no cycle has length more than 2^n. For Q_1 and Q_2 as given above, let $\eta_n(x, y)$ be the formula $Q_2(f(x), x, f(y), y)$ .It is easy to see that η_n defines an equivalence relation that holds precisely when x and y lie on cycles of the same length. Let $\pi_n(x)$ be the formula

$$\exists z\, (\neg Q_1(f(x), z, x) \wedge Q_2(f(x), x, f(z), z)).$$

Thus, $\pi_n(x)$ holds precisely when there are at least two cycles with the same length as the cycle containing x. Let $\sigma_n(x, y)$ be the formula

$$Q_2(f(x), x, f(f(y)), y).$$

We see that $\sigma_n(x, y)$ holds precisely when x lies on a cycle of length one less than the cycle containing y. These formulas interpret successor relations on set of size at most 2^n with an added unary predicate. The interpretation differs from interpretations discussed previously, however, in that we must take a quotient by the equivalence relation η_n, rather than interpret the domain as a subset.

Remark. The theories in Examples 8.6 and 8.7 are treated by Ferrante and Rackoff [24]. They give a matching upper bound for the former and an upper bound of $NTIME(2^{dn^2})$ for the latter. (Example 8.7 first appeared in Ferrante [23].) The inseparability, hereditary, and hardness results presented here are new.

Example 8.8. *The first-order theory of* $\langle \{0, 1\}^*, r_0, r_1\rangle$, *the binary tree on* $\{0, 1\}^*$ *together with successor functions* r_0 *and* r_1 *(described in Example* 8.3).

We show that this theory has a hereditary lower bound of

$$ATIME(2^{cn}, cn)$$

by a monadic interpretation of the classes $\mathscr{T}_2^{2^n}$ and applying Corollary 7.5. Each tree in $\mathscr{T}_2^{2^n}$ has at most 2^{dn} vertices, for some $d > 0$; therefore, each such tree can be represented by a linear order of length 2^{kn} with added unary predicate, as in Example 8.1.

Thus, we need only give a monadic interpretation of models

$$\langle \{1, \ldots, 2^{kn}\}, \leqslant\rangle$$

in $\langle \{0, 1\}^*, r_0, r_1\rangle$. (The unary predicate comes automatically, since we have a monadic interpretation.) Specifying this monadic interpretation is straightforward: an integer l in $\{1, \ldots, 2^{kn}\}$ is represented by the equivalence class of strings of length l; $\leqslant$ is given by the prefix order; and subsets of $\{1, \ldots, 2^{kn}\}$ are represented by strings w via

$$\{l \mid \text{for some } v \text{ of length } l,\ r_1(v) \text{ is a prefix of } w\}.$$

To complete the construction it suffices to give iterative definitions for formulas $\lambda_n(u, v)$ saying that for some $l \leqslant 2^{kn}$, u and v both have length l, and formulas $\pi_n(u, v)$ saying that for some $l \leqslant 2^{kn}$, u has length l and is a prefix of v. This is done analogously to the iterative definition of linear order on intervals of length 2^{dn} in Example 8.6. We leave the details to the reader.

Remark. Example 8.8 was first treated by Vogel [76]. He also showed that for some $d > 0$, $ATIME(2^{dn}, n)$ is an upper bound for this theory. The hereditary lower bound can be derived from Vogel's result because the theory is finitely axiomatizable.

In Examples 8.7 and 8.8 the equality relation is interpreted by an equivalence relation rather than true equality. Correspondingly, models $\mathfrak{A}$ are interpreted by *quotients* of defined substructures, rather than the substructures themselves. This kind of interpretation is common in undecidability proofs (see examples in Rabin [53] and Eršov, Lavrov, Taimanov, and Taitslin [21]) and requires no changes in the proofs in Sections 6 and 7. In Example 8.8 the equivalence relation could be avoided by representing an integer l by an string of l 0's. However, in Example 8.7 the use of an equivalence relation in place of equality seem unavoidable.

Example 8.9. *Any extension* Σ, *with an infinite model, of the first-order theory of Boolean algebras.*

We show that Σ has a hereditary lower bound of

$$ATIME(2^{2n/\log n}, cn)$$

by giving a monadic prenex interpretation of sets of size up to $2^{cn/\log n}$ and applying Corollary 7.9.

Fix $\mathfrak{B}$, an infinite model of Σ. Let $m = \lceil n/\log n \rceil$ and $l \leq 2^m$. Define $\delta_n(x, \boldsymbol{u})$ to be a prenex formula with parameters $\boldsymbol{u} = u_1, \ldots, u_m$ that says x in an *atom relative to* $\boldsymbol{u}$. By this we mean that for each i such that $1 \leq i \leq m$, either $x \subseteq u_i$ or $x \cap u_i = \emptyset$, and x is a maximal element with this property (with respect to the relation $\subseteq$ on $\mathfrak{B}$). Clearly we may take $\delta(x, \boldsymbol{u})$ to be reset log-lin computable. By appropriately choosing elements $\boldsymbol{a}$ in $\mathfrak{B}$, we can arrange that $\delta_n^{\mathfrak{B}}(x, \boldsymbol{a})$ has precisely l elements. Let $\sigma_n(x, \boldsymbol{u}, v)$ be the formula $\delta_n(x, \boldsymbol{u}) \wedge x \subseteq v$. As b ranges over the elements of $\mathfrak{B}$, $\sigma_n^{\mathfrak{B}}(x, \boldsymbol{a}, b)$ ranges over all subsets of $\delta^{\mathfrak{B}}(x, \boldsymbol{a})$.

Remark. Lower and upper bounds for the theories of Boolean algebras were first obtained by Kozen [41]. Note that the apparent discrepancy between Kozen's results and the results presented here arises because Kozen regards all variables as having unit length, while we take the length of subscripts into account.

Example 8.10. *The pure logic L having infinitely many monadic predicates.*

We show that this theory has a hereditary lower bound of

$$NTIME(2^{cn/\log n}).$$

Unlike the other bounds given in this section, this bound is obtained by direct interpretation of binary relations. Also, this is the only *NTIME* lower bound we consider which somehow involves the expression $n/\log n$. We can show that there is a prenex interpretation of binary relations on sets of size $2^{cn/\log n}$ and then apply Theorem 6.2. The interpretation is essentially the same as the interpretation in Theorem 7.6 for the case $r = 1$. The only difference is that we no longer have a monadic interpretation because we cannot quantify over the monadic predicates.

Remark. This result was proved by Meyer and Rackoff (see Rackoff [55]) and a matching upper bound was given by Lewis [43].

We now turn to discussion of the fundamental lower bound results obtained by Fischer and Rabin [26] and improved somewhat by Bruss and Meyer [12] and Berman [7].

Example 8.11. *The first-order theory of real addition.*

This is the first-order theory of the model $\langle \mathbb{R}, + \rangle$, where $\mathbb{R}$ is the set of real numbers. We obtain a herediary lower bound of

$$ATIME(2^{cn}, cn)$$

by giving a monadic iterative interpretation of sets of size 2^n with an addition relation and applying Theorem 7.2. Fischer and Rabin give a simultaneous iterative definition of formulas $\mu_n(x, y, z)$, $\pi_n(x, y, z)$. Formula $\mu_n(x, y, z)$ holds precisely when x is nonnegative integer less than 2^{2^n} and $x \cdot y = z$; formula $\pi_n(x, y, z)$ holds precisely when x, y^x, and z are nonnegative integers less than 2^{2^n} and $y^x = z$.

From these formulas we can define directly formulas $\sigma_n(x, u)$ that hold when x is a nonnegative integer less than 2^n, u is a nonnegative integer less than 2^{2^n}, and the $(x+1)$st bit in the binary representation of u is 1. Thus, as a ranges over elements of $\mathbb{R}$, $\sigma_n^{\mathbb{R}}(x, a)$ includes all subsets of $\{0, \ldots, 2^n - 1\}$. We thereby obtain a monadic interpretation of $\langle \{0, \ldots, 2^n - 1\}, + \rangle$.

Remark. The same argument will work if the theory of real addition is replaced by any extension of the theory of semigroups that has, for each n, a model $\langle B, \circ \rangle$ with an element a whose powers $a^0, a^1, \ldots, a^n$ are distinct. Any such theory has an $ATIME(2^{cn}, cn)$ hereditary lower bound.

Example 8.12. *Presburger arithmetic, the first-order theory of addition on the natural numbers.*

This is the first-order theory of the model $\langle \mathbb{N}, + \rangle$, where $\mathbb{N}$ is the set of nonnegative integers. We obtain a hereditary lower bound of

$$ATIME(2^{2^{cn}}, cn)$$

by giving a monadic iterative interpretation of sets of size 2^{2^n} with an addition relation and applying Theorem 7.2. As in Fischer and Rabin [26], we use divisibility to push the definition of the multiplication and exponentiation relations $\mu_n(x, y, z)$ and $\pi_n(x, y, z)$ in the previous example up to intervals of length $g(n)$, where $g(n) \geq \exp_3(dn)$; this estimate comes from the Prime Number Theorem. We then obtain a monadic interpretation as in the previous example.

Remark. Evidently, the basic ideas used in Examples 8.11 and 8.12 are already found in Fischer and Rabin [26]. However, we obtain several benefits from our approach. First, by using the machinery of the previous sections we avoid technicalities concerning coding sequences and Turing machine computations which Fischer and Rabin needed to address. Second, by emphasizing inseparability instead of just complexity, we obtain hereditary lower bounds. Finally, by observing that monadic quantification is implicit in the interpretations, we see why the appropriate lower bounds are *ATIME* bounds rather than an *NTIME* bounds.

The linear alternating time lower bounds in Example 8.11 and 8.12 were proved by Berman [7]. A slightly weaker hereditary $NTIME(2^{cn})$ lower bound for real addition was obtained by Ferrante and Rackoff [24], and a hereditary $NTIME(2^{2^{cn}})$ lowr bound for Presburger arithmetic was obtained by Young [78].

The full strength of the Prime Number Theorem is not needed in Example 8.12. A much simpler result, due to Tchebychef, suffices; see Theorem 7 of Hardy and Wright [36]. This illustrates a common phenomenon in lower bound results for a theories. Crude arguments often suffice. Sophisticated mathematics and an intimate knowledge of the theory under consideration are rarely needed. In this respect, upper bound results may be much more difficult.

Note that since we have monadic quantification on sets of size 2^{cn} in a model of real addition (or any extension of the theory of semigroups as described in the remark after Example 8.11) we have a monadic interpretation of the classes $\mathcal{T}_2^{2^n}$ from the same representation of trees used Example 8.1. Similarly, since we have monadic quantification on sets of size $2^{2^{cn}}$ in a model of Presburger arithmetic, we have a monadic interpretation of the classes $\mathcal{T}_3^{2^n}$. These facts will be useful in the next two examples.

Fischer and Rabin [26] announced two other lower bounds: a lower bound of $NTIME(\exp_3(cn))$ for the first-order theory of integer multiplication; and a lower bound of $NTIME(2^{2^{cn}})$ for the theory of finite Abelian groups. They did not supply proofs (but mentioned the key idea of encoding sequences of integers as exponents in a prime decomposition). To our knowledge, no proofs have ever been published. In the next two examples we sketch proofs of the stronger $ATIME$ versions of these results.

Example 8.13. *The first-order theory of multiplication on the positive integers.*

This is the first-order theory of the model $\langle \mathbb{I}, \cdot \rangle$, where $\mathbb{I}$ is the set of positive integers. We obtain a hereditary lower bound of

$$ATIME(\exp_3(cn), cn)$$

by giving a monadic iterative interpretation of the classes $\mathcal{T}_4^{2^n}$ and applying Corollary 7.5.

Let p_i be the ith prime number. Observe that $\langle \mathbb{I}, \cdot \rangle$ is isomorphic to a direct sum of countably many copies of $\langle \mathbb{N}, + \rangle$ by the mapping that takes the sequence $(a_1, a_2, \ldots)$, where a_i is 0 for all large i, to $p_1^{a_1} \cdot p_2^{a_2} \cdots$. Moreover, each direct summand (i.e., set of powers of some prime p_i) can be defined. But we saw in Example 8.12 and the previous remark that there is a monadic interpretation of the classes $\mathcal{T}_3^{2^n}$ in $\langle \mathbb{N}, + \rangle$. The idea here is that we interpret a tree from $\mathcal{T}_4^{2^n}$ in $\langle \mathbb{I}, \cdot \rangle$ by directing edges from a new root to the roots of trees interpreted in each direct summand.

Clearly, we can specify a first-order formula $\alpha(x, y, z, t)$ that says t is a prime, x, y, and z are powers of t, and $x \cdot y = z$. Let $\delta_n(x, u')$, $\pi_n(x, y, u')$, and $\sigma_n(x, u', v')$ be the formulas giving a monadic iterative interpretation of $\mathcal{T}_3^{2^n}$ in $\langle \mathbb{N}, + \rangle$. We saw in Example 8.12 that such formulas exist. (If we substitute $\alpha(x, y, z, t)$ for all occurrences of $x + y = z$ we obtain formulas $\delta_n'(x, t, u')$, $\pi_n'(x, y, t, u')$, and $\sigma_n'(x, t, u', v')$ in the language of $\langle \mathbb{I}, \cdot \rangle$.) For each fixed prime t, these formulas give a monadic iterative interpretation of $\mathcal{T}_3^{2^n}$ in $\langle \mathbb{I}, \cdot \rangle$. For $\delta_n'(x, t, u')$ to be satisfied, it is necessary that x and u' be powers of t. Thus, except possibly for 1, which is a power of every prime, there are no elements common to the interpretations of $\delta_n'(x, t, u')$ and $\delta_n'(x, t', u'')$ when t and t' are distinct primes.

Let $\eta(t, u', u)$ be a formula that says t is a prime and u' is the largest power of t dividing u. Define $\delta_n''(x, u)$ to be the formula

$$x = 1 \vee \exists t, u' \, (\eta(t, u', u) \wedge \delta_n'(x, t, u')).$$

Define $\pi_n''(x, y, u)$ to be the formula

$$\begin{aligned}&(x = 1 \wedge \exists t, u' \, (\eta(t, u', u) \wedge \delta_n'(y, t, u') \wedge \forall z \neg \pi_n'(z, y, t, u')))\\ &\quad \vee (\exists t, u' \, (\eta(t, u', u) \wedge \pi_n'(x, y, t, u')))\end{aligned}$$

The first disjunct gives an edge from 1, the root of the tree, to the root of each primary subtree; the second disjunct gives the edges in the primary subtrees. Using δ_n'' and π_n'' we can interpret each tree from $\mathcal{T}_4^{2^n}$ by taking a disjoint collection of trees in $\mathcal{T}_3^{2^n}$ and 1 as a new root—we need only choose u suitably. Define $\sigma_n''(x, u, v, w)$ to be the formula

$$(x = 1 \wedge w = 1) \vee \exists t, u', v' \, (\eta(t, u', u) \wedge \eta(t, v', v) \wedge \sigma_n'(x, t, u', v')).$$

By varying v and w we obtain all subsets of $\delta_n''(x, u)$, so we have a monadic interpretation.

Remark. An upper bound of $ATIME(\exp_3(dn), dn)$ for the first-order theory of integer multiplication can be obtained from the treatment of this theory in Ferrante and Rackoff [24]. The original reference for an upper bound on the first-order theory of integer multiplication is Rackoff [56].

As our last example we consider the first-order theory of finite Abelian groups. Lo [44] has given an extensive treatment of upper bounds for theories of Abelian groups. He states there bounds in terms of the classes $SPACE(2^{2^{dn}})$, but it is clear that his analysis gives $ATIME(2^{2^{dn}}, dn)$ upper bounds. We derive a matching lower bound, not just for the theory of finite Abelian groups, but also for the theory of finite cycle groups.

Example 8.14. *The first-order theory of finite cyclic groups.*

We obtain a hereditary lower bound of

$$ATIME(2^{2^{cn}}, cn)$$

by giving a monadic iterative interpretation of the classes $\mathcal{T}_3^{2^n}$ and applying Theorem 7.5. We use a device similar to the one used in Example 8.13. Now rather than regarding the positive integers with multiplication as a direct sum of copies of $\langle \mathbb{N}, + \rangle$, we regard a finite cyclic group as a direct sum of cyclic groups whose orders are prime powers.

Let $C(l)$ be the cyclic group of order l. We know from the remarks following Example 8.12 that there is a $d > 0$ such that when $l \geqslant 2^{2^{dn}}$, there is a monadic iterative interpretation of $\mathcal{T}_2^{2^n}$ in $C(l)$. More precisely, there are formulas $\delta_n(x, t', u')$, $\pi_n(x, y, t', u')$, and $\sigma_n(x, t', u', v)$ given by iterative definitions such that if u' is a generator of $C(l)$, then each tree in $\mathcal{T}_2^{2^n}$ is isomorphic to $\langle \delta_n^{C(l)}(x, t', u'), \pi_n^{C(l)}(x, y, t', u')\rangle$ for some t' in $C(l)$, and $\sigma_n^{C(l)}(x, t', u', v')$ includes all subsets of $\delta_n^{C(l)}(x, t', u')$ as v' ranges over elements of $C(l)$. (We need to mention the generator u of $C(l)$ explicitly in these formulas because there is no preferred generator; this necessitates only a minor modification of the formulas in Example 8.11.)

Let $p_1, p_2, \ldots, p_k$ be the prime numbers less than $2^{2^{dn+1}}$ and m_i the largest power of p_i less than $2^{2^{dn+1}}$. Then since $p_i \cdot m_i \geqslant 2^{2^{dn+1}}$ and $p_i < m_i$, we know that $m_i \geqslant (2^{2^{dn+1}})^{\frac{1}{2}} = 2^{2^{dn}}$ so that there is a monadic iterative interpretation of $\mathcal{T}_2^{2^n}$ in each $C(m_i)$ as described above.

By Tchebychef's theorem (Theorem 7 of Hardy and Wright [36]) we know that k, the number of primes less than $2^{2^{dn+1}}$, is at least $2^{2^{dn}}$ for sufficiently large n. By taking d large enough, we can insure that k is greater than the maximum number of primary subtrees in each tree of $\mathcal{T}_3^{2^n}$. Now take $m = m_1 \cdot m_2 \cdots m_k$ so that $C(m) = C(m_1) \oplus C(m_2) \oplus \cdots \oplus C(m_k)$. We see that $m \geqslant (2^{2^{dn}})^k \geqslant 2^{2^{2^{dn}}}$. We need to show that we can combine the monadic iterative interpretations of $\mathcal{T}_2^{2^n}$ in the direct summands $C(m_i)$ to obtain a monadic iterative interpretation of $\mathcal{T}_3^{2^n}$ in $C(m)$. To do this, we will show that we can define the decomposition of $C(m)$ into the factor subgroups $C(m_i)$.

Let $\beta(x, y, t, u)$ be the following formula with free relation variable Q:

$$\begin{aligned}&(x = 0 \wedge y = 0) \vee (x = t \wedge y = u) \vee (\exists x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)\\ &\quad(Q(x_1, y_1, t, u) \wedge Q(x_2, y_2, x_1, y_1) \wedge Q(x_3, y_3, t, u)\\ &\qquad \wedge Q(x_4, y_4, t, u) \wedge (x = x_2 + x_3 + x_4) \wedge (y = y_2 + y_3 + y_4)).\end{aligned}$$

Then by induction on n the relation $Q(x, y, t, u)$ given by the iterative definition

$$[Q(x, y, t, u) \equiv \beta]_n$$

is true precisely when there is an integer j in the range $0 \leqslant j < 2^{2^{n-1}}$ such that $x = jt$ and $y = ju$. Notice that we are using an idea first exploited by Fischer and Rabin [26] in their proof of a lower bound for Example 8.11: each integer $j < 2^{2^n}$ can be written $j = j_1 j_2 + j_3 + j_4$, where $j_1, j_2, j_3, j_4 < 2^{2^{n-1}}$.

For the remainder of the proof we will assume that $Q(x, y, t, u)$ is given by the iterative definition

$$[Q(x, y, t, u) \equiv \beta]_{dn+2}.$$

Fix a generator u of $C(m)$. We can specify that $x = ju$ for some integer j in the interval

$$I = \{j \mid 0 \leqslant j < 2^{2^{dn+1}}\}$$

by writing $Q(x, x, u, u)$. Since $m \geqslant 2^{2^{2^{dn}}}$, the values ju are distinct for $j \in I$. Let us identify I with the set of elements ju such that $j \in I$. The group operation restricted to I defines integer addition. We can also define multiplication: if

$x_1 = j_1u$ and $x_2 = j_2u$ then $y = j_1j_2u$ precisely when $Q(y, x_1, x_2, u)$ holds. Therefore, we can say that $s \in I$ is prime and also that $s \in I$ is a prime power.

Let $\alpha(x, y, z, s, u)$ be a first-order formula that says $s \in I$ is the largest power of some prime in I, x, y, and z are annihilated by s (i.e., $Q(0, s, x, u)$, $Q(0, s, y, u)$, and $Q(0, s, z, u)$ hold), and $x + y = z$. Hence, if $\alpha(x, y, z, s, u)$ holds, then $s = m_i$ for some $i \leqslant k$ and $x, y, z \in C(m_i)$. In other words, $\alpha(x, y, z, s, u)$ defines addition on the direct summands $C(m_i)$.

Now every element t in $C(m)$ can be uniquely expressed as a sum

$$t_1 + t_2 + \cdots + t_k$$

where each t_i is an element of $C(m_i)$. We can specify a formula $\eta(s, t', t, u)$ that says $t' = t_i$ when s is m_i. We say simply that t' is annihilated by s and $t - t'$ is divisible by s:

$$Q(0, s, t', u) \wedge \exists z\, Q(t - t', s, z, u).$$

Thus, we can define the decomposition of $C(m)$ into its factor subgroups. In particular, since u can be expressed as a sum $u_1 + u_2 + \cdots + u_k$, where u_i is a generator of $C(m_i)$, the formula $\eta(s, u', u, u)$ picks a unique generator for each factor subgroup as s ranges over maximal prime powers in I.

The rest of the proof proceeds as in Example 8.13. For example, we form $\delta'_n(x, s, t', u', u)$ by substituting $\alpha(x, y, z, s, u)$ for each occurrence of $x + y = z$ in $\delta_n(x, t', u')$. Then $\delta''_n(x, t, u)$ is the formula

$$x = 0 \vee \exists s, t', u'\, (\eta(s, t', t, u) \wedge \eta(s, u', u, u) \wedge \delta'_n(x, s, t', u', u)).$$

We define $\pi''_n(x, y, t, u)$ and $\sigma''_n(x, t, u, v, w)$ similarly.

9. Upper bounds

In this section we give upper bounds that show that most of the lower bounds obtained in Sections 4–7 are the best possible.

First we give upper bounds for $sat_T(L_0)$, $sat^p_T(L_0)$, and $sat^*_T(L_0)$. Recall from Theorem 4.3 that when $T(dn) = o(T(n))$ for some d between 0 and 1, these sets are not in $NTIME(T(cn))$ for some $c > 0$.

Proposition 9.1. *Let T be a time resource bound. Then*

$$sat_T(L_0) \in NTIME(T(n)^{n+2}),$$
$$sat^p_T(L_0) \in NTIME(T(n)^{(1+\varepsilon)n/\log n}), \quad \textit{for each } \varepsilon > 0,$$
$$sat^*_T(L_0) \in NTIME(nT(n)^{n+2}).$$

Proof. To determine whether a sentence φ from L_0 is in $sat_T(L_0)$, nondeterministically generate a finite binary relation. We give a nondeterministic recursive procedure that determines whether φ is true in this relation.

If $\varphi \in sat_T(L_0)$, then it holds in some binary relation $\mathfrak{A}$ on a set of size at most $T(n)$. The representation of this relation requires at most $T(n)^2$ bits. We will show that our recursive procedure halts within time $cT(n)^{n+2}$ on $\mathfrak{A}$.

The procedure tests the subformulas of φ and combines results to produce an answer. We may assume that all negations have been pushed inward so that only atomic formulas are negated. It is clear how the procedure works if φ is a conjunction or disjunction. If φ begins with an existential quantifier, an element of $\mathfrak{A}$ is nondeterministically assigned as the value of the quantified variable and the enclosed formula is checked. If φ begins with a universal quantifier, then each element of $\mathfrak{A}$ is assigned in turn to the quantified variable and the enclosed formula is checked. When an atomic formula is reached it can be determined in time $O(T(n)^2)$ whether it is true in $\mathfrak{A}$ for the assignment values at this point. Since for each of at most n universal quantifiers, $T(n)$ values are generated, the total time is $O(T(n)^2 T(n)^n)$, as claimed.

Suppose φ is in prenex normal form. For each $\varepsilon > 0$, whenever η is sufficiently large there are at most $(1+\varepsilon)n/\log n$ universal quantifiers in φ, so determining whether a prenex formula is in $sat^p_T(L_0)$ is in $NTIME(T(n)^{(1+\varepsilon)n/\log n})$. If φ is in L_0^*, then the same sort of procedure is used, except that when a relation variable is encountered, it is necessary to jump to its definition (this may take n moves), compute its value by calling our recursive procedure, and return. Total time, then, is $O(nT(n)^2 T(n)^n)$ because the tree of recursive procedure calls has height at most n and branches at most n times at each vertex; at the leaves, there is a cost of $T(n)^2$ moves to evaluate atomic formulas; at the vertices corresponding to relation variable references there is a cost of $O(n)$ moves to find definition.

For all three bounds we must use the Linear Speed Up Theorem (see Hopcroft and Ullman [37]) to eliminate constants in front of the time bounds. □

We see that if $T(n)^{n+2} = O(T(dn))$ for some $d > 0$, then $sat_T(L_0) \in NTIME(T(dn))$, so we have essentially the same upper and lower bounds. Similar remarks pertain in the other cases.

Proposition 9.2. *Let T be a time resource bound. Then*

$$sat_T(ML_0),\ sat^p_T(ML_0),\ \textit{and}\ sat^*_T(ML_0)$$

are in $ATIME(T(n)^2, n)$.

Proof. Given a sentence φ of length n in ML_0, nondeterministically generate a binary relation. We use alternation to determine whether φ holds in the relation. If $\varphi \in sat_T(ML_0)$, then it holds in some binary relation $\mathfrak{A}$ on a set of size at most $T(n)$. For each set quantifier encountered it is necessary to generate $T(n)$ bits to assign a value to the quantified variable. There are $O(n)$ such variables so this part of the computation takes time $O(nT(n))$. This time is dominated by the $O(T(n)^2)$ time needed to generate $\mathfrak{A}$ and verify atomic formulas. If φ is a sentence in ML_0^*, we use the same procedure except that when a subformula with a relation variable is encountered, the value of the subformula is guessed and verified using alternation. □

Notice that if $T(n) = O(T(cn)^{\frac{1}{2}})$ for some $c > 0$, then for some $d > 0$, $sat_T(ML_0) \in ATIME(T(dn), n)$, so again we have essentially the same upper and lower bounds.

We now turn to upper bounds for theories of finite trees. To determine if a sentence φ of length n is in $sat(\Sigma_r)$, we will show that it suffices to nondeterministically generate a tree in $\mathcal{T}_r^m$, where $m \log m \geqslant n$, and verify that φ holds in this tree. In fact, we prove a somewhat stronger result: given a tree of height r or less, there is a tree in $\mathcal{T}_r^m$ satisfying precisely the same sentences of length n or less. Our proof uses Ehrenfeucht games. Observe that a sentence of length n can have at most m distinct variables (i.e., variables with different subscripts). Therefore, we will use the formulation of Ehrenfeucht games for logics with a bounded number of variables. These games were introduced for infinitary logics by Barwise [3] and later used by Immerman [38] to obtain lower bounds for queries on finite relational structures.

Given two structures $\mathfrak{A}$ and $\mathfrak{B}$ for a first-order logic L, write $\mathfrak{A} \equiv_n^m \mathfrak{B}$ to indicate that $\mathfrak{A}$ and $\mathfrak{B}$ satisfy precisely the same sentences from L to quantifier rank at most n and with at most m distinct variables. The game used to characterize $\equiv_n^m$ is played for n moves between players I and II on a pair of structures $\mathfrak{A}$ and $\mathfrak{B}$. Each player begins with m pebbles. On the first move player I places a pebble on an element of $\mathfrak{A}$ (or $\mathfrak{B}$) and player II responds by placing a corresponding pebble on an elemnt of $\mathfrak{B}$ (respectively, $\mathfrak{A}$). On each remaining move player I has two options: he may place an unplayed pebble on an element of $\mathfrak{A}$ (or $\mathfrak{B}$), in which case player II places a corresponding pebble on an element of $\mathfrak{B}$ (respectively, $\mathfrak{A}$); or he may remove one of the pebbles on $\mathfrak{A}$ (or $\mathfrak{B}$) and replay it (not necessarily on the same structure), in which case player II removes the corresponding pebble from $\mathfrak{B}$ (respectively $\mathfrak{A}$) and replays it in response to the move of player I. Player II wins if the mapping from the set of elements of $\mathfrak{A}$ covered by pebbles at the end of the game to corresponding elements in $\mathfrak{B}$ is an isomorphism between substructures of $\mathfrak{A}$ and $\mathfrak{B}$; otherwise, player I wins.

The basic result concerning this game is that player II has a winning strategy if and only if $\mathfrak{A} \equiv_n^m \mathfrak{B}$. We use this fact to prove two simple lemmas.

We will assume henceforth that in addition to the binary edge relation, trees also have a unary relation that is true only at the root. This is a technical convenience to force player II to pebble a root whenever player I pebbles a root.

For a tree $\mathfrak{A}$ and a vertex x in $\mathfrak{A}$, let $\mathfrak{A}_x$ be the subtree of $\mathfrak{A}$ whose set of vertices consists of x and all of its descendents.

Lemma 9.3. *Suppose $\mathfrak{A}$ is a tree and x is a vertex of $\mathfrak{A}$. Let $\mathfrak{A}'$ be the result of replacing $\mathfrak{A}_x$ in $\mathfrak{A}$ by another tree $\mathfrak{B}$. If $\mathfrak{A}_x \equiv_n^m \mathfrak{B}$ then $\mathfrak{A} \equiv_n^m \mathfrak{A}'$.*

Proof. We know the player II has a winning strategy for the m pebble game of length n on $\mathfrak{A}_x$ and $\mathfrak{B}$. Player II uses this as part of a winning strategy for the m pebble game of length n on $\mathfrak{A}$ and $\mathfrak{A}'$. Whenever player I pebbles an element in $\mathfrak{A}-\mathfrak{A}_x$ or $\mathfrak{A}'-\mathfrak{B}$, player II responds by pebbling the same element; whenever player I pebbles an element in $\mathfrak{A}_x$ or $\mathfrak{B}$, player II responds by pebbling the element dictated by the strategy for $\mathfrak{A}_x$ and $\mathfrak{B}$. Notice that if player I pebbles the root of $\mathfrak{A}_x$, player II pebbles the root of $\mathfrak{B}$ (and vice versa) so that adjacency to element in $\mathfrak{A}-\mathfrak{A}_x$ will be the same. This is a winning strategy for player II so $\mathfrak{A} \equiv_n^m \mathfrak{A}'$. □

Lemma 9.4. *Let $\mathfrak{A}$ and $\mathfrak{B}$ be trees. Suppose that for each isomorphism type, the set of primary subtrees of $\mathfrak{A}$ and the set of primary subtrees of $\mathfrak{B}$ either contain the same number of trees of that type, or both contain at least m trees of that type. Then $\mathfrak{A} \equiv_n^m \mathfrak{B}$ for every $n > 0$.*

Proof. It is easy to determine a winning strategy for player II. If player I pebbles the root of $\mathfrak{A}$ (or $\mathfrak{B}$), player II pebbles the root of $\mathfrak{B}$ (respectively, $\mathfrak{A}$). If player I pebbles an element in a primary subtree $\mathfrak{A}'$ of $\mathfrak{A}$ and no other elements of $\mathfrak{A}'$ have been pebbled, player II responds by pebbling the corresponding element in a primary subtree $\mathfrak{B}' \cong \mathfrak{A}'$ of $\mathfrak{B}$ where no other elements have been pebbled. Player II responds similarly if player I pebbles an element in a primary subtree $\mathfrak{B}'$ of $\mathfrak{B}$ and no other elements of $\mathfrak{B}'$ have been pebbled. If player I chooses an element in a primary subtree $\mathfrak{A}'$ of $\mathfrak{A}$ where some elements have already been pebbled, these elements correspond to elements already pebbled in some primary subtree $\mathfrak{B}' \cong \mathfrak{A}'$ of $\mathfrak{B}$. The isomorphism determines the response of player II. Player II responds similarly if player I chooses from a primary subtree of $\mathfrak{B}$ where elements have already been pebbled. Because no more than m elements in a structure are pebbled at any time, it is easy to see that this strategy can always be carried out for $\mathfrak{A}$ and $\mathfrak{B}$ satisfying the hypotheses of the lemma. □

Theorem 9.5. *Given a finite tree $\mathfrak{A}$ of height at most r, there is a tree $\mathfrak{B} \in \mathcal{T}_r^m$ such that $\mathfrak{A} \equiv_n^m \mathfrak{B}$ for all $n \geq 0$.*

Proof. Modify $\mathfrak{A}$ in the following manner. For each nonleaf vertex x of depth $r-1$, consider all children y of x and subtrees $\mathfrak{A}_y$; for each isomorphism type, if more than m subtrees $\mathfrak{A}_y$ are isomorphic, delete enough of them so that there are precisely m. Continue this modification procedure for vertices of depth $r-2$, $r-3$, and so on, up to the root. Call the resulting tree $\mathfrak{B}$. It is clear from the two preceding lemmas that every time we delete subtrees in this process, we obtain a tree in the same $\equiv_n^m$-class as $\mathfrak{A}$. Thus, $\mathfrak{A} \equiv_n^m \mathfrak{B}$. It is evident that $\mathfrak{B} \in \mathcal{T}_r^m$. □

Remark. With slight modifications, this proof shows that for any tree $\mathfrak{A}$ of height r or less, there is a tree $\mathfrak{B} \in \mathcal{T}_r^m$ such that $\mathfrak{A} \equiv_r^m \mathfrak{B}$ for all $m \geq 0$.

We have, as an immediate consequence of the preceding theorem, upper bounds for theories of finite trees.

Corollary 9.6. *For $r > 3$, there is a $d > 0$ such that*

$$sat(\Sigma_r) \in NTIME(\exp_{r-2}(dn)).$$

Also, $sat(\Sigma_3) \in NTIME(2^{dn^2})$ for some $d > 0$.

Proof. When $r \geqslant 3$ and m is the least integer such that $m \log m \geqslant n$, each tree in $\mathcal{T}_r^m$ has at most $\exp_{r-2}(cn)$ vertices. Thus, the time required to nondeterministically generate a tree in $\mathcal{T}_r^m$ and determine whether a sentence φ of length n holds in this tree is $\exp_{r-2}(cn)^n$. This function is dominated by $\exp_{r-2}(dn)$ for some $d > 0$ when $r > 3$ and by $\exp_{r-2}(dn^2)$ when $r = 3$. □

Remark. This theorem gives matching upper bounds for the lower bounds obtained in Corollary 6.4 except for the case $r = 3$. There the lower bound is $NTIME(2^{cn})$ and the upper bound is $NTIME(2^{dn^2})$. Ferrante and Rackoff [24] get precisely the same bounds for the theory of one-to-one functions (cf. Examples 8.7). It is more satisfying to say that $sat(\Sigma_3)$ is a complete problem for

$$NEXPTIME = \bigcup_{k>0} NTIME(2^{n^k})$$

via polynomial time reductions, according to Theorem 6.1(iv).

Both $sat(\Sigma_1)$ and $sat(\Sigma_2)$ are in *PSPACE* since the number of vertices in trees in $\mathcal{T}_1^m$ and $\mathcal{T}_2^m$ is polynomially bounded in n, where m is the least integer such that $m \log m \geqslant n$. (Recall that by Theorem 9.5 for every finite tree $\mathfrak{A}$ of height at most r, there is a tree $\mathfrak{B} \in \mathcal{R}_r^m$ such that $\mathfrak{A} \equiv_n^m \mathfrak{B}$.) Every first-order theory with a model of power greater than 1 is hard for *PSPACE* (via log space reductions), so we know fairly precisely the complexity of these theories.

Write $\mathfrak{A} \equiv_n^m \mathfrak{B}$ to indicate that $\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same monadic second-order sentences of quantifier rank n with at most m variables. To obtain upper bounds for monadic second-order theories of finite trees we must introduce Ehrenfeucht games characterizing the relation $\approx_n^m$.

In such a game, players I and II play for n moves on structures $\mathfrak{A}$ and $\mathfrak{B}$. Let $P_1, P_2, \ldots, P_m$ be unary relation symbols not in the language of $\mathfrak{A}$ and $\mathfrak{B}$. During each move of the game one of the symbols P_i will be assigned a pair of sets. This pair contains a subset of $\mathfrak{A}$ and a subset of $\mathfrak{B}$. Initially, each of these symbols is assigned the empty set for $\mathfrak{A}$ and the empty set for $\mathfrak{B}$. On each move player I picks a relation symbol P_i. The previous assignment to P_i is forgotten. Player I assigns a subset of $\mathfrak{A}$ (or $\mathfrak{B}$). Player II responds by assigning a subset of $\mathfrak{B}$ (respectively $\mathfrak{A}$) to P_i. Whenever player I picks a singleton set, player II must respond with a singleton set. (Singleton set moves correspond to element quantifiers.) Now suppose that at the end of the game symbols $P_1, P_2, \ldots, P_m$ are assigned subsets $R_1, R_2, \ldots, R_m$ of $\mathfrak{A}$ and subsets $S_1, S_2, \ldots, S_m$ of $\mathfrak{B}$. If the set

$$\{(x, y) \mid \text{for some } i,\ R_i = \{x\} \text{ and } S_i = \{y\}\}$$

is an isomorphism between substructures of

$$\langle \mathfrak{A}, R_1, \ldots, R_m \rangle \quad \text{and} \quad \langle \mathfrak{B}, S_1, \ldots, S_m \rangle$$

then player II wins; otherwise, player I wins.

The basic result concerning this game is that player II has a winning strategy if and only if $\mathfrak{A} \approx_n^m \mathfrak{B}$.

We will sometimes say that an Ehrenfeucht game is played on structures $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$, where $R_1, \ldots, R_m$ are subsets of $\mathfrak{A}$ and $S_1, \ldots, S_m$ are subsets of $\mathfrak{B}$. By this we mean that the symbols $P_1, \ldots, P_m$ initially have $R_1, \ldots, R_m$ and $S_1, \ldots, S_m$ assigned to them. The game then proceeds as before. We will say that $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$ are *n-equivalent* if player II has a winning strategy for games of length n on this pair of structures. Clearly, n-equivalence is an equivalence relation.

Using monadic second-order Ehrenfeucht games we can show that $\equiv_n^m$ can be replaced by $\approx_n^m$ in Lemma 9.3. In fact, we can show more.

Lemma 9.7. *Suppose that $\mathfrak{A}$ is a finite tree with subsets $R_1, \ldots, R_m$. Let x be a vertex of $\mathfrak{A}$ and $\langle \mathfrak{A}', R_1', \ldots, R_m' \rangle$ the structure obtained by replacing the substructure $\langle \mathfrak{A}_x, R_1 \cap \mathfrak{A}_x, \ldots, R_m \cap \mathfrak{A}_x \rangle$ of $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ by another structure $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$, where $\mathfrak{B}$ is a tree. If*

$$\langle \mathfrak{A}_x, R_1 \cap \mathfrak{A}_x, \ldots, R_m \cap \mathfrak{A}_x \rangle \quad \textit{and} \quad \langle \mathfrak{B}, S_1, \ldots, S_m \rangle$$

are n-equivalent, then so are $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, R_1', \ldots, R_m' \rangle$.

We would also like to prove a result like Lemma 9.4, but this is not so simple. We must prove results like Lemma 9.4 and Theorem 9.5 simultaneously. To do this we need more complicated sets of trees.

Definition. Define functions $f(m, n, r)$ and $g(m, n, r)$ as follows.

$$f(m, n, 0) = 2^m, \qquad g(m, n, 0) = 2^{n+1}, \qquad g(m, 0, r) = 2$$

and

$$f(m, n, r+1) = 2^m(g(m, n, r) + 1)^{f(m,n,r)}$$

$$g(m, n+1, r) = f(m, n, r)g(m, n, r).$$

Functions f and g are defined for all integers $m,\ n,\ r \geqslant 0$ because we can repeatedly use the last equation to obtain

$$g(m, n, r+1) = f(m, n-1, r+1)f(m, n-2, r+1) \cdots f(m, 0, r+1)$$

and then replace each factor $f(m, i, r+1)$ by $2^m(g(m, i, r) + 1)^{f(m,i,r)}$.

Now define classes $\mathcal{U}_r^{m,n}$ of trees of height at most r. $\mathcal{U}_0^{m,n}$ contains all trees of height 0. $\mathcal{U}_{r+1}^{m,n}$ consists of all trees whose primary subtrees come from $\mathcal{U}_r^{m,n}$, no more than $g(m, n, r)$ primary subtrees coming from the same isomorphism class. Define classes $\mathcal{V}_r^{m,n}$ of structures $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$, where $\mathfrak{A}$ is a tree of height at most r and each R_i is a subset of $\mathfrak{A}$. $\mathcal{V}_0^{m,n}$ contains all structures $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$, where $\mathfrak{A}$ is a tree of height 0. $\mathcal{V}_{r+1}^{m,n}$ consists of all structures $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ such that substructures formed by restricting to primary subtrees of $\mathfrak{A}$ all come from $\mathcal{V}_r^{m,n}$, no more than $g(m, n, r)$ such substructures coming from the same isomorphism class. Notice that $|\mathcal{V}_r^{m,n}| = f(m, n, r)$.

Theorem 9.8. *Given a finite tree $\mathfrak{A}$ of height at most r and an integer $n > 0$, there is a tree $\mathfrak{B} \in \mathcal{U}_r^{m,n}$ such that $\mathfrak{A} \approx_n^m \mathfrak{B}$.*

Proof. We prove a more general assertion. Given a structure $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$, where $\mathfrak{A}$ is a tree of height at most r and $R_1, \ldots, R_m$ are subsets of $\mathfrak{A}$, there is a structure $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle \in \mathcal{V}_r^{m,n}$ such that $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$ are n-equivalent.

The proof is by induction on r. For $r = 0$, the assertion is obvious. Assume that $\mathfrak{A}$ has height $r > 0$ and that the assertion holds for trees of lesser height. Consider the substructures of $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ formed by restricting to primary subtrees. By the induction hypothesis, each such substructure is n-equivalent to some structure in $\mathcal{V}_{r-1}^{m,n}$; for each n-equivalence class replace the substructures in that class by a structure from $\mathcal{V}_{r-1}^{m,n}$ in the same class with the provision that if there are more than $g(m, n, r-1)$ substructures in the class we first eliminate enough of them to make their number precisely $g(m, n, r-1)$. In this way we form a structure $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle \in \mathcal{V}_r^{m,n}$. We show that $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$ are n-equivalent.

Fix an n-equivalence type τ. Let $\langle \mathfrak{A}', R_1', \ldots, R_m' \rangle$ be the union of substructures of $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ of type τ formed by restricting to primary subtrees of $\mathfrak{A}$. Define the substructure $\langle \mathfrak{B}', S_1', \ldots, S_m' \rangle$ of $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$ similarly. Thus, $\mathfrak{A}'$ and $\mathfrak{B}'$ are forests, and each substructure of $\langle \mathfrak{A}', R_1', \ldots, R_m' \rangle$ or $\langle \mathfrak{B}', S_1', \ldots, S_m' \rangle$ formed by restricting to a tree in the forest is of type τ. Moreover, $\mathfrak{A}'$ and $\mathfrak{B}'$ either contain the same number of trees or both contain at least $g(m, n, r-1)$ trees. We claim that $\langle \mathfrak{A}', R_1', \ldots, R_m' \rangle$ and $\langle \mathfrak{B}', S_1', \ldots, S_m' \rangle$ are n-equivalent. From this claim it follows easily that $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ and $\langle \mathfrak{B}, S_1, \ldots, S_m \rangle$ are n-equivalent because player II can combine the winning strategies on the pairs of substructures.

We establish the claim by induction on n. The case $n = 0$ is clear. (Notice, however, that it is crucial that $g(m, 0, r-1) = 2$ because S_i' must be assigned a nonsingleton set whenever R_i' is assigned a nonsingleton set.) Suppose that $n > 0$ and that the claim is true for smaller values.

Player I will begin by assigning a subset of one of the forests—say subset R_i'' of

$\mathfrak{A}'$—to a relation symbol P_i. This gives a new structure $\langle \mathfrak{A}', R_1'', \ldots, R_m'' \rangle$, where $R_j'' = R_j'$ when $i \neq j$. By the induction hypothesis (for the induction on r), every substructure formed by restricting this structure to a tree in $\mathfrak{A}'$ is $(n-1)$-equivalent to some structure in $\mathcal{V}_{r-1}^{m,n-1}$. Hence, there can be at most $f(m, n-1, r-1)$ $(n-1)$-equivalence classes represented among such substructures. Player II responds by assigning a subset of S_i'' of $\mathfrak{B}'$ to P_i to obtain a structure $\langle \mathfrak{B}', S_1'', \ldots, S_m'' \rangle$, where $S_j'' = S_j'$ when $i \neq j$. She does this in such a way that for every $(n-1)$-equivalence type τ', $\langle \mathfrak{A}', R_1'', \ldots, R_m'' \rangle$ and $\langle \mathfrak{B}', S_1'', \ldots, S_m'' \rangle$ either have the same number of substructures of type τ' formed by restricting to trees in $\mathfrak{A}'$ and $\mathfrak{B}'$, or both have at least $g(m, n-1, r-1)$ such substructures of type τ'. Player II can always make such a response because $\mathfrak{A}'$ and $\mathfrak{B}'$ either have the same number of trees or both have at least $g(m, n, r-1) = f(m, n-1, r-1)g(m, n-1, r-1)$ trees.

By the induction hypothesis (for the induction on n)

$$\langle \mathfrak{A}', R_1'', \ldots, R_m'' \rangle \quad \text{and} \quad \langle \mathfrak{B}', S_1'', \ldots, S_m'' \rangle$$

are $(n-1)$-equivalent so $\langle \mathfrak{A}', R_1', \ldots, R_m' \rangle$ and $\langle \mathfrak{B}', S_1', \ldots, S_m' \}$ are n-equivalent. □

Theorem 9.9. *For each $r > 1$ there is a $d > 0$ such that*

$$sat(M\Sigma_r) \in ATIME(\exp_r(dn/\log n), n).$$

Proof. It is easy to show by induction that for each $r \geq 1$ there is a $c > 0$ such that

$$f(m, n, r) \leq \exp_{r+1}(c(m + \log n)),$$
$$g(m, n, r) \leq \exp_{r+1}(c(m + \log n)).$$

If we let $h(m, n, r)$ be the maximum number of vertices of any structure in $\mathcal{U}_r^{m,n}$ (or $\mathcal{V}_r^{m,n}$), we see that

$$h(m, n, 0) = 1,$$
$$h(m, n, r+1) = h(m, n, r) f(m, n, r) g(m, n, r) + 1.$$

For each $r \geq 2$ there is a $c > 0$ such that $h(m, n, r) \leq \exp_r(c(m + \log n))$.

When $r \geq 2$ we can determine if a sentence φ from ML_t is in $sat(M\Sigma_r)$ by nondeterministically generating a tree in $\mathcal{U}_r^{m,n}$, where $m \log m \geq n$, and using alternation to verify that φ holds in this tree. This can be done in $ATIME(\exp_r(dn/\log n), n)$.

When $r = 1$ we must be a little more careful because a tree in $\mathcal{U}_1^{m,n}$ may have $O(2^n)$ vertices. However a tree of height 1 has almost no structure. Suppose we have chosen m subsets $R_1, \ldots, R_m$ from a tree $\mathfrak{A}$ of height 1. Let $+R_i$ be R_i and $-R_i$ be the complement of R_i. For $d = (d_1, \ldots, d_m) \in \{+, -\}^m$ let $d \cdot R = d_1R_1 \cap d_2R_2 \cap \cdots \cap d_mR_m$. We need only keep track of which sets R_i contain the root of $\mathfrak{A}$ and the values $|d \cdot R|$ for each $d \in \{+, -\}$. Thus, $\langle \mathfrak{A}, R_1, \ldots, R_m \rangle$ can be represented in space $O(2^m \log n)$. Using this kind of representation, the argument above shows that $sat(M\Sigma_1) \in ATIME(2^{dn/\log n}, n)$. □

10. Open problems

We close with a list of problems, mostly concerned with lower bounds for theories arising in algebra. We have selected only a small number of problems from the large number of theories whose complexities deserve to be investigated.

Problem 10.1. *Determine the complexity of first-order theories of finite fields.*

The first-order theory of finite fields, and several related theories, were shown to be decidable by Ax [1] in a paper which has proved to be of great mathematical influence. Later Fried and Sacerdote [28] gave a primitive recursive decision procedure, but it is not known if any of these theories is elementary recursive.

It is not difficult to show that these theories have a hereditary lower bound of $ATIME(\exp_2(cn), cn)$. The method is to give a monadic interpretation of the classes of binary relations on sets of size at most $\exp_2(n)$, and then apply Theorem 7.3. We give a brief sketch of the argument. Let $\mathcal{F}$ be any infinite field which is a model of the theory of finite fields and in which one has the coding of finite sets used by Duret [17]. That is, given any two disjoint finite sets $A, B \subseteq \mathcal{F}$, there is an element $w \in \mathcal{F}$ such that if $a \in A$ then $a + w$ is a square in $\mathcal{F}$ and if $b \in B$ then $b + w$ is not a square in $\mathcal{F}$. Construct by iteration formulas $\alpha_n(x, \boldsymbol{u})$ such that for each n there is a choice of parameters $\boldsymbol{u}$ so that $\alpha_n(x, \boldsymbol{u})$ is true in $\mathcal{F}$ of $\exp_2(n)$ values of x. For example, if $\mathcal{F}$ has characteristic 0, then $\alpha_n(x)$ can be constructed as in Fisher and Rabin [26] so that $\alpha_n(x)$ holds in $\mathcal{F}$ exactly when x is one of the integers $0, \ldots, \exp_2(n) - 1$. Alternatively, one could use formulas $\alpha_n(x, y)$ asserting that x is an $\exp_2(n)$th root of y. Now consider the formulas $\beta_n(x, t, \boldsymbol{u})$ given by

$$\exists y, z\ (\alpha_n(y, \boldsymbol{u}) \wedge \alpha_n(z, \boldsymbol{u}) \wedge x = y + zt).$$

For an appropriate choice of t in $\mathcal{F}$, the mapping $(y, z) \mapsto y + zt$ in one-to-one on $\alpha_n^{\mathcal{F}}(x, \boldsymbol{u})$. This together with the coding of finite sets gives every binary relation on sets of size at most $\exp_2(n)$. The coding of finite sets also gives every subset of the universe so we have the required monadic interpretation.

Problem 10.2. *Determine the complexity of the first-order theory of linearly ordered Abelian groups.*

This theory was shown to be decidable by Gurevich [32], with later improvements in Gurevich [33]. There is a simple interpetation of the first-order theory of linear orders in this theory, so it has a hereditary $NTIME(\exp_\infty(cn))$ lower bound. (See Example 8.2.) On the other hand, a primitive recursive decision procedure for the theory was given by Gurevich [33]. It would be interesting to improve either of these bounds.

Problem 10.3. *Determine the complexity of first-order theories of valued fields.*

Ax and Kochen [2] and Eršov [20] proved the decidability of various first-order theories of valued fields, including some power series fields and the fields of p-adic numbers $\mathbb{Q}_p$. Brown [11] obtained an elementary recursive upper bound for the first-order theory of 'almost all' of the fields $\mathbb{Q}_p$—that is, the set of sentences true in $\mathbb{Q}_p$ for all but finitely many p. Very little is known about lower bounds for this theory or about the other related theories covered by the Ax–Kochen–Eršov work.

Problem 10.4. *Determine the complexity of the first-order theory of Boolean algebras with several distinguished filters.*

Eršov [19] proved decidability of the first-order theory of Boolean algebras with a distinguished filter. A recent paper by Touraille [71] presents some results on the elimination of quantifiers for this theory, but does not show decidability. Rabin [54] showed decidability of the theory of Boolean algebras with quantification over filters by giving an interpretation in the monadic second order theory of two successors. This gives an upper bound of $NTIME(\exp_\infty(dn))$, but nothing is known about lower bounds.

Problem 10.5. *Determine the complexity of the first-order theory of the lattice of closed subsets of the Cantor set.*

Rabin [54] proved that this theory is decidable by interpreting it in the monadic second-order theory of two successors. As in the previous problem, this gives an upper bound of $NTIME(\exp_\infty(dn))$. It is not known if this theory is elementary recursive and no nontrivial lower bounds are known. A more explicit analysis of this theory has been given by Gurevich [34].

Problem 10.6. *Determine the complexity of the first-order theory of l_∞, the ring of bounded sequences of real numbers.*

Cherlin [15] gave a very explicit and difficult decision procedure for this theory, but its complexity has not been analyzed. It should be possible to extract an upper bound from Cherlin's work. While it seems unlikely to us that this theory is elementary recursive, there are no good lower bounds known.

Problem 10.7. *Determine the complexity of the theory of pairs of torsion-free abelian groups, and of the theory of a vector space V with k distinguished subspaces* ($k = 1, 2, 3, 4$).

The theory of pairs of torsion-free abelian groups was proved decidable by Kozlov and Kokorin [42]; see also Schmitt [63]. For $k \geqslant 5$, the theory T_k of vector spaces with k distinguished subspaces (over some fixed field $\mathcal{F}$) is undecidable; see Baur [4] or Slobodskoi and Fridman [67]. Here the vector space is equipped with + and a fmaily of unary scalar multiplication functions, one for each element of $\mathcal{F}$. If $k \leqslant 4$, however, the theory T_k was shown to be decidable by Baur [5]. See Prest [52] for a discussion of how these theories are related to the representation theory of finite dimensional algebras over $\mathcal{F}$. The theories T_k are stable and hence the undecidability of T_5 could not be proved by the usual means of interpreting arithmetic or finite graphs. There does not seem to be any corresponding *a priori* impediment to using the methods of this paper to obtain lower bounds on the complexity of T_1, T_2, T_3, or T_4.

Problem 10.8. *Determine the complexity of the first-order theory of real closed fields and the theory of the first-order theory of algebraic closed fields.*

These theories are, respectively, the first-order theory of the real numbers and the first-order theory of the complex numbers. Good upper and lower bounds are known for these theories, but the gap has not been completely closed. Berman's $ATIME(2^{cn}, n)$ lower bound for real addition is the best bound known for the theory of real closed fields. We discussed this bound in Example 8.11. By the remarks following the example, we have the same lower bound for the theory of algebraic closed fields. The best upper bound at present for the theory of real closed fields is $SPACE(2^{dn})$; this was proved by Ben-Or, Kozen, and Reif [6]. This bound holds as well for the theory of algebraic closed fields since there is a simple interpretation of the complex number in the real numbers. For the same reason, any lower bound for the theory of algebraic closed fields would hold for the theory of real closed fields.

Robinson [61] showed that if A is the field of real algebraic numbers, then the first-order theory of $\langle \mathbb{R}, +, \cdot, A \rangle$ is also decidable. It would be interesting to know if this theory has a somewhat higher complexity than the theory of real closed fields.

Problem 10.9. *Determine the complexity of the first-order theory of differentially closed fields.*

Robinson [60] proved the decidability of this theory, but essentially nothing more is known about its complexity. See Wood [77] for a fuller discussion of this mathematically interesting theory.

Problem 10.10. *Is elementary recursiveness of a theory preserved under product and sheaf constructions?*

Decidability of first-order theories is preserved by many general constructions, such as products (Feferman and Vaught [22]) and sheaf constructions (Macintyre [47]). Some upper bound results for weak products are presented in Ferrante and Rackoff [24], where the question is raised whether, for every model $\mathfrak{A}$ whose first-order theory is elementary recursive, the first-order theory of the weak direct product $\mathfrak{A}^{\omega}$ is also elementary recursive. The same type of question for other product and sheaf constructions is also open and worth investigating. (See Chapter 5 of Ferrante and Rackoff [24].)

Problem 10.11. *Give nontrival lower bounds for mathematically interesting theories whose decidability is still open.*

Examples include the first-order theories of the field of rational functions over the complex numbers; the real exponential field $\langle \mathbb{R}, +, \cdot, \exp \rangle$; the field of meromorphic functions; and many others. It may be possible to show that some of these theories are not elementary recursive, just as Semenov [65] did for the theory of free groups. (See the remarks following Example 8.5.)

Problem 10.12. *Is there a 'natural' decidable theory which is not primitive recursive?*

Problem 10.13. *Is there a 'natural' decidable theory with a lower bound of the form* $NTIME(\exp_\infty(f(n)))$ *where* $f(n)$ *is not linearly bounded?*

Problem 10.14. *Determine the complexities of fragments of theories with given prefix structures.*

There has been some interesting work done in this area. (See, for example, Robertson [59], Reddy and Loveland [58], Fürer [30], and Scarpellini [62].)

In certain cases the methods of this paper should give results under these restrictions. This is not likely to be true where iterative interpretations are used, since iterative, definitions almost always introduce an unbounded number of alternations of quantifiers. However, where prenex interpretations are used in conjunction with Theorem 6.1 and Corollary 6.1, it seems clear that complexity results for sentences with specific limitations on prefix structure can be obtained.

Other syntactic limitations can also be imposed on decision problems and have been widely studied in the setting of the decidable/undecidable distinction. For example, in algebraic theories, one may pay attention to the degree and number of variables of occurring polynomials.

In decision problems (and, more generally, complexity problem) the refinement mentioned above, especially limitations to a simpler and more intelligible prefix structure, often reflect restriction to mathematically more interesting and significant problems (as has been emphasized to us by G. Kreisel). Thus, the undecidability of Hilbert's tenth problem is far more interesting than the undecidability of arithmetic; the undecidability of the word problem for finitely presented groups is far more interesting than the undecidability of the theory of groups. One can hope for and expect to see a similar kind of increasing matrurity in the study of complexity of decidable problems, not only at the level of *NP*-complete or *PSPACE*-complete problems (where it already exists to some extent), but also at higher levels of complexity.

Problem 10.15. *Characterize the PSPACE-complete theories.*

We noted in Section 1 that every theory with a model of power greater than 1 is *PSPACE*-hard. Thus, the *PSPACE*-complete theories are, in some sense, the simplest nontrivial theories. A number of different theories have been shown to be *PSPACE*-complete. (See Stockmeyer [69], Ferrante and Rackoff [24], and Grandjean [31].) It would be interesting to have model theoretic characterization of these theories.

Problem 10.16. *If one substitutes 'tree' for 'binary relation' in the definitions of* $sat_T(L_0)$ *and* $sat_T(ML_0)$*, do Theorems* 6.1 *and* 7.1 *still hold for* $T(n) \leqslant \exp_\infty(cn)$?

An affirmative answer would give a generalization of Corollaries 6.5, 7.5, and 7.8.

Problem 10.17. *Use the techniques of this paper to derive a lower bound for the emptiness problem for *-free regular expressions.*

The proof that this problem is not elementary recursive is one of the more difficult results in Stockmeyer [68]. McNaughton and Papert [50] show that the first-order theory of finite linear orders with an added unary predicate (Example 8.1) can be reduced to this problem, but it is not clear that an elementary recursive reduction can be found. Fürer [29] give a lower bound of

$$NTIME(\exp_\infty(cn/\log^* n)^2))$$

for the emptiness problem for *-free regular expressions. It would be interesting to know if this could be strengthened to a lower bound of

$$NTIME(\exp_\infty(cn)).$$

References

[1] J. Ax, The elementary theory of finite fields, Ann. of Math. 88 (1968) 239–371.

[2] J. Ax and S. Kochen, Diophantine problems over local fields III: decidable fields, Ann. of Math. 83 (1966) 437–456.

[3] J. Barwise, On Moschovakis closure ordinals, J. Symbolic Logic 42 (1977) 292–296.

[4] W. Baur, Undecidability of the theory of abelian groups with a subgroup, Proc. Amer. Math. Soc. 55 (1976) 125–128.

[5] W. Baur, On the elementary theory of quadruples of vector spaces, Ann. Math. Logic 19 (1980) 243–262.

[6] M. Ben-Or, D. Kozen and J. Reif, the complexity of elementary algebra and geometry, J. Comput. System Sci. 32 (1986) 251–264.

[7] L. Berman, The complexity of logical theories, Theoret. Comput. Sci. 11 (1980) 71–77.

[8] E. Börger, Decision problems in predicate logic, in: G. Lolli, G. Longo, A. Marcja, eds., Logic Colloquium '82 (North-Holland, Amsterdam, 1984) 263–301.

[9] E. Börger, Spektralproblem and completeness of logical decision problems, in E. Börger, G. Hasenjaeger, D. Rödding, eds., Logics and Machines: Decision Problems and Complexity-Lecture Notes in Comput. Sci. 171 (Springer, Berlin, 1984) 333–356.

[10] E. Börger, Berechenbarkeit, Komplexität, Logic (Vieweg, Wiesbaden, 1985).

[11] S.S., Brown, Bounds on transfer principles for algebraically closed and discretely valued fields, Mem. Amer. Math. Soc. 204 (1978).

[12] A. Bruss and A. Meyer, On time-space classes and their relation to the theory of real addition, Theoret. Comput. Sci. 11 (1980) 59–69.

[13] J.R. Büchi, Turing machines and the Entscheidungsproblem, Math. Ann. 148 (1962) 201–213.

[14] A. Chandra, D. Kozen and L. Stockmeyer, Alternation, J. Assoc. Comput. Mach. 28 (1981) 114–133.

[15] G. Cherlin, Rings of continuous functions: decision problems, in: Model Theory of Algebra and Arithmetic, Lecture Notes in Math. 834 (Springer, Berlin, 1980) 44–91.

[16] K.J. Compton, C.W. Henson, and S. Shelah, Nonconvergence, undecidability, and intractability in asymptotic problems, Ann. Pure Appl. Logic 36 (1987) 207–224.

[17] J.-L. Duret, Les corps pseudo-finis ont la propriété d'indépendence, C. R. Acad. Sci. Paris Sér. A-B 290 (1980) A981–A983.

[18] E.A. Emerson and J. Halpern, Decision procedures and expressiveness in the temporal logic of branching time, J. Comput. System Sci. 30 (1985) 1–24.

[19] Y. Eršov, Decidability of the elementary theory of relatively complemented distributive lattices and the theory of filters, Algebra i Logika 3 (1964) 17–38. (Russian).

[20] Y. Eršov, On the elementary theory of maximal normed fields, Soviet Math. Dokl. 6 (1965) 1390–1393 (English translation).

[21] Y. Eršov, I.A. Lavrov, A.D. Taimanov and M.A. Taitslin, Elementary theories, Russian Math, Surveys 20 (1965) 35–100 (English translation).

[22] S. Feferman and R. Vaught, The first-order properties of products of algebraic systems, Fund. Math. 47 (1959) 57–103.

[23] J. Ferrante, Some upper and lower bounds on decision procedures in logic, Doctoral thesis, Massachusetts Institute of Technology, Cambridge, MA (1974).

[24] J. Ferrante and C. Rackoff, The Computational Complexity of Logical Theories, Lectures Notes in Math. 718 (Springer, Berlin, 1979).

[25] M.J. Fischer and R. Ladner, Propositional dynamic logic of regular programs, J. Comput. System Sci. 18 (1979) 194–211.

[26] M.J. Fischer and M. Rabin, Super-exponential complexity of Presburger arithmetic, in: R.M. Karp, ed., Complexity of Computation, SIAM-AMS Proc. 7 (Amer. Math. Soc. New York, 1974) 27–42.

[27] K. Fleischmann, B. Mahr and D. Siefkes, Bounded concatenation theory as a uniform method for proving lower complexity bounds, in: R. Gandy, M. Hyland, eds., Logic Colloquium '76, (North-Holland, Amsterdam, 1977) 471–490.

[28] M. Fried and G. Sacerdote, Solving Diophantine problems over all residue class field of a number field and all finite fields, Ann. of Math. 104 (1976) 203–233.

[29] M. Fürer, Nicht-elementare untere Schranken in der Automaten-theorie, Doctoral Thesis, ETH, Zurich (1978).

[30] M. Fürer, The complexity of Presburger arithmetic with bounded quantifier alternation, Theoret. Comp. Sci. 18 (1982) 105–111.

[31] E. Grandjean, Complexity of the first-order theory of almost all finite structures, Inform. and Control 57 (1983) 180–204.

[32] Y. Gurevich, Elementary properties of ordered Abelian groups, Amer. Math. Soc. Transl. 46 (1965) 165–192.

[33] Y. Gurevich, Expanded theory of ordered abelian groups, Ann. Math. Logic 12 (1977) 193–228.

[34] Y. Gurevich, Crumbly spaces, in; Proc. 1979 Intl. cong. Logic, Methodology and Philosophy of Science (North-Holland, Amsterdam, 1982) 179–191.

[35] J. Halpern and M. Vardi, The complexity of reasoning about knowledge and time, in; Proc. 18th Ann. ACM Symp. on Theory of Computing (Assoc. Comput. Mach., New York, 1986) 304–315.

[36] G.H. Hardy and E.M. Wright, theory of Numbers (Oxford University Press, London, 1964).

[37] J. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages and Computation (Addison-Wesley, Reading, MA, 1979).

[38] N. Immerman, Relational queries computable in polynomial time, Inform. and Control 68 (1986) 86–104.

[39] M. Kaufmann and S. Shelah, On random models of finite power and monadic logic, Discrete Math. 54 (1983) 285–293.

[40] I. Korec and W. Rautenberg, Model interpretability into trees and applications, Arch. Math. Logik Grundlag. 17 (1975/76) 97–104.

[41] D. Kozen, Complexity of Boolean algebras, Theoret. Comp. Sci. 10 (1080) 221–247.

[42] G.T. Kozlov and A.I. Kokorin, Elementary theory of abelian groups without torsion, with a predicate selecting a subgroup, Algebra and Logic 8 182–190 (English translation).

[43] H.R. Lewis, Complexity results for classes of quantificational formulas, J. Comput. System Sci. 21 (1980) 304–315.

[44] Libo Lo, On the computational complexity of the theory of abelian groups, Ph.D. Thesis, University of Michigan, Ann Arbor, MI (1984).

[45] J.F. Lynch, Complexity classes and finite models, Math. Systems Theory 15 (1982) 127–144.

[46] M. Machtey and P. Young, Introduction to the General Theory of Algorithms (North-Holland, New York, 1978).

[47] A. Macintyre, Model completeness for sheaves of structures, Fund. Math. 81 (1973) 73–89.

[48] A. Meyer, The inherent computational complexity of theories of ordered sets, in: Proc. 1974 Intl. Cong. of Mathematicians, Vancouver, B.C., Canada (1974) 477–482.

[49] A. Meyer, Weak monadic second order theory of successor is not elementary recursive, in: Logic Colloquium (Boston 1972–73), Lecture Notes in Math. 453 (Springer, Berlin, 1975) 132–154.

[50] R. McNaughton and S. Papert, Counter Free Automata (MIT Press, Cambridge, MA, 1971).

[51] Y. Moschovakis, Elementary Induction on Abstract Structures (North-Holland, Amsterdam, 1974).

[52] M. Prest, Model theory and representation type of algebras, in: F.R. Drake and J.K. Truss, eds., Logic Colloquium '86 (North-Holland, Amsterdam, 1988) 219–260.

[53] M. Rabin, A simple method for undecidability proofs and some applications, in: Proc. 1964 Intl. Cong. Logic, Methodology and Philosophy of Science (North-Holland, Amsterdam, 1964) 58–68.

[54] M. Rabin, Decidability of second order theories and automata on infinite tress, Trans. Amer. Math. Soc. 141 (1969) 1–35.

[55] C. Rackoff, The complexity of theories of the monadic predicate calculus, Research Report no. 136, INRIA, Roquencourt, France (1975).

[56] C. Rackoff, The computational complexity of some logical theories, Doctoral Thesis, Massachusetts Institute of Technology, Cambridge, MA (1975).

[57] C. Rackoff, On the complexity of the theories of weak direct powers, J. Symbolic Logic 41 (1976) 561–573.

[58] C.R. Reddy, and D. Loveland, Presburger arithmetic with bounded quantifier alternation, in: Proc. 10th Ann. ACM Symp. on theory of Computing (Assoc. Comput. Mach., New York, 1978) 320–325.

[59] E.L. Robertson, Structure of complexity in the weak monadic second-order theories of monadic second-order theories of the natural numbers, in: Proc. 6th Ann. ACM Symp. on Theory of Computing (Assoc. Comput. Mach., New York, 1974) 161–171.

[60] A. Robinson, On the concept of a differentially closed field, Bull. Res. Council Israel 8F (1959) 113–128.

[61] A. Robinson, Solution of a problem of Tarski, Fund. Math. 47 (1959) 179–204.

[62] B. Scarpellini, Complexity of subcases of Presburger arithmetic, Trans. Amer. Math. Soc. 284 (1984) 203–218.

[63] P. Schmitt, The elementary theory of torsion free abelian groups with a predicate specifying a subgroup, Z. Math. Logik Grundlag. Math. 28 (1982) 323–329.

[64] J. Seiferas, M. Fischer and A. Mayer, Separating nondeterministic time complexity classes, J. Assoc. Comput. Mach. 25 (1978) 146–167.

[65] A.L. Semenov, An interpretation of free algebras in free groups, Soviet Math. Dokl. 21 (1980) 952–955 (English translation).

[66] A.L. Semenov Logical theories of one-place functions on the set of natural numbers, Math. USSR-Izv. 22 (1984) 587–618 (English Translation).

[67] A.M. Slobodskoi and E.I. Fridman, Theories of abelian groups with predicates specifying a subgroup, Algebra and Logic 14 (1976) 353–355 (English translation).

[68] L. Stockmeyer, The complexity of decision problems in automata and logic, Doctoral Thesis, Massachusetts Institute of Technology, Cambridge, MA (1974).

[69] L. Stockmeyer, The polynomial-time hierarchy, Theoret. Comp. Sci. 3 (1977) 1–22.

[70] L. Stockmeyer, Classifying the computational complexity of problems, J. Symbolic Logic 52 (1987) 1–43.

[71] A. Touraille, Élimination des quantificateurs dans la théorie élémentaire des algébres de Boole munies d'une famille d'idéaux distingués, C. R. Acad. Sci. Paris, Ser. I 300 (1985) 125–128.

[72] B.A. Trakhtenbrot, The impossibility of an algorithm for the decision problem for finite models, Dokl. Akad. Nauk SSSR 70 (1950) 569–572.

[73] A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc. (2) 42 (1937) 230–265. Correction, *ibid.* 43 (1937) 544–546.

[74] R. Vaught, Sentences true in all constructive models, J. Symbolic Logic 25 (1960) 39–58.

[75] R. Vaught, On a theorem of Cobham concerning undecidable theories, in: Proc. 1960 Intl. Cong. Logic, Phil. and Methodology of Sci. (Stanford University Press, Standford, CA, 1962) 14–25.

[76] H. Vogel, Turing machines with linear alternation, theories of bounded concatenation and the decision problem of first-order theories, Theoret. Comp. Sci. 23 (1983) 333–337.

[77] C. Wood, The model theory of differential fields revisited, Israel J. Math. 25 (1976) 331–352.

[78] P. Young, Gödel theorems, exponential difficulty and undecidability of arithmetic theories: an exposition, in: A. Nerode, R. Shore, eds., Recursion Theory, Proc. Symp. Pure Math. 42 (Amer. Math. Soc., New York, 1985) 503–522.