Lecture Notes in Mathematics

Edited by A. Dold and B. Eckmann

718

Jeanne Ferrante
Charles W. Rackoff

The Computational Complexity of Logical Theories

Springer-Verlag
Berlin Heidelberg New York 1979

Authors

Jeanne Ferrante
Automatic Programming Group
IBM T. J. Watson Research Center
Box 218
Yorktown Heights, NY 10598
USA

Charles W. Rackoff
Department of Computer Science
University of Toronto
Toronto, Ontario
Canada M5S 1A7

AMS Subject Classifications (1970) 02B10, 02F10, 02G05, 69A20, 68A40

ISBN 3-540-09501-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-09501-2 Springer-Verlag New York Heidelberg Berlin

Library of Congress Cataloging in Publication Data
Ferrante, Jeanne, 1949- The computational complexity of logical theories. (Lecture notes in mathematics ; 718) Bibliography: p. Includes index. 1. Predicate calculus. 2. Computational complexity. I. Rackoff, Charles W., 1948- joint author. II. Title. III. Series: Lecture notes in mathematics (Berlin) ; 718.
QA3.L28 no. 718 [QA9.35] 510'.8s [511'.3] 79-15338

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.
2141/3140-543210

PREFACE

Since the early part of this century, logicians have been interested in decision procedures for theories in the predicate calculus. By a theory, we mean the set of sentences true about a particular structure, or the set of sentences true about every structure in a particular class of structures. One of the first decision procedures discovered was Löwenheim's procedure for validity of sentences in the monadic predicate calculus. Another early example is Presburger's procedure for the first-order theory of the integers under addition.

Although some people felt that some theories might be undecidable, it wasn't until the thirties that it was possible to carefully state, let alone prove, such conjectures. It was first necessary to have formal notions of computation, as provided by the work of people such as Church, Kleene, Post and Turing. The first undecidability result was Gödel's famous theorem that the first-order theory of the integers under the operations of addition and multiplication is not decidable. At the present time an enormous number of decidability and undecidability results are known; [ELTT65] is a good survey of these results.

For a long time, it was felt that one could say nothing stronger about a theory than that it was decidable. Although the concept of comparing the difficulty of undecidable (or nonrecursive) sets has existed for a long time, it was only recently that people have started investigating the relative decidability of recursive sets. It was in the sixties that the field of computational

complexity theory began. The goal was to look at a recursive set and ask how much of a particular resource (usually time or storage) was needed to decide the membership problem for that set. It was necessary of course to have a precise model of computation to work with. Many different models have been defined today, but we shall use the Turing machine because of its elegance and simplicity; most of the following theorems also hold for most other existing models of computation. Time and space on a Turing machine will be defined in a straightforward way in Chapter one. The time or space used for a particular machine will be expressed as a function of the length of input: a machine operates within time (or space) f(n) if on all inputs of length n which are accepted, the time (or space) used is at most f(n).

With this model, one can now look at previously existing algorithms (for instance decision procedures for logical theories) and ask how much time or space they use. One can use this model as a basis for comparing these algorithms, or for inventing new ones which are <u>better</u> in the very precise sense of using less time or space. One of the first of these <u>upper bound</u> results was Ferrante and Rackoff's theorem that the first order theory of the real numbers under addition can be decided by an algorithm operating in exponential space.

However, this model also gives us a precise way of saying that a particular set or theory <u>requires</u> a certain amount of time or space. Here one is saying not only that a particular algorithm

uses a lot of the resource, but that any algorithm for the membership problem must use a lot of the resource in question. These are called lower bound results. The techniques for showing that certain decidable sets require (for any algorithm) a lot of time or space to be decided, parallel earlier results showing that certain sets cannot be decided at all (by any algorithm). Whereas undecidability is usually proven by "arithmetization" of Turing machines, lower bounds are usually proven by efficient "arithmetization" of time or space bounded Turing machines, an idea having its beginnings in a paper by Stephen Cook.† The first lower bound result for logical theories was Albert Meyer's theorem that the second order theory of the integers under successor requires an enormous (non-elementary recursive) amount of time and space. This was done in 1973.

As with decidability and undecidability results, there exist today many different upper and lower complexity bounds for many different types of sets, and in particular for many different logical theories. The purpose of this work is to present certain of the theorems and proofs about complexity bounds for logical theories of the first-order predicate calculus. Although the subjects chosen are those that the authors have personally worked on, the proof methods are typical of those commonly used in this area.

Precise definitions of the concepts discussed here are presented in Chapter one. Chapters two through five contain upper bound results, and Chapters six through nine contain lower bound results.

---

† "The complexity of theorem proving procedures", Proc. 3rd Annual ACM Symposium on Theory of Computing, 151-158.

ACKNOWLEDGEMENTS

The authors would like to thank Albert Meyer for his help and encouragement, both during their studies and since. Most of these results would not have been obtained without his ideas and suggestions. The authors are also grateful to Teresa Miao for the typing of this manuscript.

While at MIT, the authors were supported by NSF grant GJ-34671. The first author would also like to thank the Tufts University Faculty Awards Committee. The second author thanks the Department of Computer Science of the University of Toronto, as well as the National Research Counc of Canada.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

CHAPTER 1

INTRODUCTION AND BACKGROUND

Section 1: Introduction

The significance of the distinction between decidable and undecidable theories has been blurred by recent results of Meyer and Stockmeyer [Mey75,MS72,SM73,Sto74] and Fischer and Rabin [FiR74] who have shown many of the decidable theories known to logicians cannot be decided by any algorithm whose computational complexity grows less than exponentially with the size of sentences to be decided. In some cases even larger lower bounds have been established. In this volume we investigate the computational complexity of a number of different logical theories. This investigation has two parts: obtaining both upper and lower bounds on the amount of time or space needed. Upper bounds are obtained by exhibiting an efficient algorithm for deciding the theory; lower bounds are obtained by showing all possible algorithms for deciding the theory must use a certain amount of time or space.

Most of our upper bound results use the technique of Ehrenfeucht games [Ehr61] to show that quantifiers ranging over all elements of a structure of a theory may be restricted to small finite subsets of the structure. In this way, truth or validity of sentences becomes decidable trivially by exhaustive search. Upper bounds on the size of the finite set in turn yield upper bounds on the space and time required to perform the search. The finite set to which a quantifier can be restricted depends on the depth of the quantifier, and these sets are obtained by analyzing the information that formulas of a specific quantifier-depth can convey in a particular theory.

In addition, we use the technique of Ehrenfeucht games to derive some general results about the theories of weak direct powers of structures. We obtain general results relating the complexities of theories to the complexities of their weak direct powers and direct products, thereby obtaining computational versions of results of Mostowski [Mos52] and Feferman and Vaught [FV59]. In particular, we show that the theory of the weak (or strong) direct product of a structure is elementary recursive if (but not only if) the theory of the structure is elementary recursive and if another condition holds; this other condition says roughly that not too many sets of k-tuples can be defined in the structure with quantifier depth n formulas.

Most of our lower bound results are obtained by efficient arithmetization of Turing machines which run in bounded time or space. This parallels the results in recursive function theory where undecidability is proved from arithmetization of all Turing machines.

We summarize the specific results of the volume in Table 1. NTIME(t(n)) (DTIME(t(n))) is the class of sets accepted by a non-deterministic (deterministic) Turing machine within time t(n) on inputs of length n; NSPACE(s(n)) and DSPACE(s(n)) are defined similarly for space s(n). (For formal definitions of the classes NTIME, DTIME, NSPACE and DSPACE, see Definition 3.2).

One interesting observation is that wildly varying "jumps" in complexity occur when a monadic predicate is added to a first-order theory. Contrast, for example, the theory of a 1-1 unary function and the theory of one successor, and their respective theories with a monadic predicate symbol added. Both of these should be compared to the first-order theory

First-order Theory of	Upper Bound	Lower Bound
1-1 Unary Function	$\in$ NTIME(2^{cn^2}) and NSPACE(2^{cn}) (for some c)	$\notin$ NTIME($2^{c'n}$) for some c' > 0
1-1 Unary Function with a monadic predicate	$\in$ NTIME($2^{2^{cn}}$) (for some c)	$\notin$ NTIME($2^{2^{c'n}}$) for some c' > 0
One Successor	$\in$ DSPACE(n^2)	$\notin$ NSPACE(s(n)) for all s(n) = o(n)
One Successor with a monadic predicate	$\in$ NTIME($2^{2^{cn}}$) (for some c)	$\notin$ NTIME($2^{2^{c'n}}$) for some c' > 0
<N,< >	$\in$ DSPACE(n^2)	$\notin$ NSPACE(s(n)) for all s(n) = o(n)
Two Successors with the equal length predicate	$\in$ DSPACE(2^{cn}) (for some c)	(even without the equal length predicate) $\notin$ NTIME($2^{c'n}$) for some c' > 0
Well-Order	$\in$ DSPACE(n^3)	$\notin$ NSPACE(s(n)) for all s(n) = o(n)
Lexicographical Order	$\in$ DSPACE(n^2)	$\notin$ NSPACE(s(n)) for all s(n) = o(n)

TABLE 1

First-Order Theory of	Upper Bound	Lower Bound
$\langle Z,+,\leq,0\rangle$	$\in$ DSPACE($2^{2^{cn}}$) (for some c)	$\notin$ NTIME($2^{2^{c'n}}$) for some c' > 0 (Fischer & Rabin) [FiR74]
$\langle R,+,\leq,0\rangle$	$\in$ DSPACE(2^{cn}) (for some c)	$\notin$ NTIME($2^{c'n}$) for some c' > 0 (Fischer & Rabin) [FiR74]
$\langle I,\cdot\rangle$	$\in$ DSPACE($2^{2^{2^{cn}}}$) (for some c)	$\notin$ NTIME($2^{2^{2^{c'n}}}$) for some c' > 0 (Fischer & Rabin) [FiR74]
Finite Abelian Groups	$\in$ DSPACE($2^{2^{2^{cn}}}$) (for some c)	$\notin$ NTIME($2^{2^{2^{c'n}}}$) for some c' > 0 (Fischer & Rabin) [FiR74]
Any nonempty collection of pairing functions		$\notin$ NTIME(f(cn)) for some c > 0 where $f:N\to N$ is defined $f(i) = 2^{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}$ } height i

TABLE 1 (cont'd)

of $\langle N,< \rangle$ as above; it is a result due to Stockmeyer [Sto 74] that the theory of $\langle N,< \rangle$ with a monadic predicate is non-elementary in the sense of the following definition:

Definition 1.1 An elementary recursive function (on strings or integers) is one which can be computed by some Turing machine within time bounded above by a fixed composition of exponential functions of the length of the input. (This is shown by Cobham [Cob64] and Ritchie [Rit63] to be equivalent to Kalmar's definition [cf. Pet67].)

In fact, Stockmeyer shows that the first-order theory of any infinite linear order with a monadic predicate is non-elementary.

Let N^* be the set of functions from N to N of finite support, i.e., $N^* = \{f:N \to N \mid f(i) = 0$ for all but finitely many $i \in N\}$. The structure $\langle I,\cdot \rangle$ is isomorphic to the structure $\langle N^*,+ \rangle$ (the weak direct power of $\langle N,+ \rangle$) where addition is defined component-wise. Our results about the theories of weak direct powers enables us to obtain a new procedure, and the upper bound noted in the table, for deciding whether sentences are true over $\langle N^*,+ \rangle$, and thus over $\langle I,\cdot \rangle$. As a corollary, we obtain the same upper bound on decision procedures for the first order theory of finite abelian groups.

A pairing function is a one-one map $\rho: N \times N \to N$, and the associated structure is $\langle N,\rho \rangle$. Our main result on pairing functions is that no nonempty collection of pairing functions has an elementary recursive theory.

Also note that we have not listed an upper bound for the theory of any nonempty collection of pairing functions. In fact, the theory of the set of all pairing functions is undecidable and the theories of some individual pairing functions are undecidable. Tenney [Ten74] has shown

that many commonly used pairing functions have decidable theories. All of these are non-elementary recursive.

We should also mention the following results. The first such lower bound results for a logical theory were obtained by Meyer [Mey75] who showed the weak monadic second order theory of successor is not elementary recursive. Other lower bound results (all exponential or higher) were obtained by Fischer and Rabin [FiR74] on the theories of real addition, finite abelian groups, Presburger arithmetic and integer multiplication. Meyer and Fischer have shown the theory of a unary function is non-elementary; Meyer and Stockmeyer show the theory of linear orders is also non-elementary [Sto74]. Stockmeyer [SM73] obtains a lower bound result for the theory of equality. Ed Robertson [Rob74] obtains results on weak monadic second-order theories of natural numbers.

In terms of upper bound results, Derek Oppen [Opp73], Charles Rackoff [Rack76] and Ferrante and Rackoff [FR75] "settle" the complexity of the theories of real addition, Presburger arithmetic and integer multiplication, by obtaining deterministic upper bounds which closely match the corresponding nondeterministic time lower bounds. Rackoff and Meyer [Rack75'] determine the complexity of satisfiability in the monadic predicate calculus. George Collins [Col75] and Leonard Monk [Monk75] independently obtained good upper bounds on the theory of real closed fields. Richard Ladner [Lad77] determines the computational complexity of various modal propositional theories. Ladner also uses model-theoretic games [Lad77'] to obtain upper bound results for some monadic second-order theories of linear order. Ferrante and Geiser [FeGe77] obtain an upper bound for the theory of rational order not included here. Rackoff's work on the complexity of the theory of products of structures also appears in [Rack76].

Fleischmann, Mahr, and Siefkes [FMS76] introduce a more uniform way to prove complexity lower bounds than that presented here. [FMS76'] contains a general introduction to the type of complexity results which follow.

Recently, Leonard Berman [Ber77] has made some interesting observations about the gaps between the upper and lower bound results which suggest the gaps will not be bridged (solely in terms of an individual time or space class).

A preliminary version of the present work appeared in the authors' Ph.D. theses [Fer74], [Rack75].

The next four sections of this chapter contain the definitions and theorems necessary to understand the precise statements of the results to follow. In the last section, we discuss the relevance of these upper and lower bound results for the practising logician.

Section 2: Basic Notation and Definitions

The purpose of this section is to present the notations and basic definitions used throughout the volume.

We use N to denote the set of nonnegative integers, I (or N^+) the set of positive integers, and Z the entire set of integers. We use Q to denote the set of rational numbers, and R to denote the set of real numbers. For $k \in Z$,

$$N_k = \{n \in N | n \leq k\} = \{n \in N | 0 \leq n \leq k\}, \text{ and}$$

$$I_k = \{n \in I | n \leq k\} = \{n \in N | 1 \leq n \leq k\}.$$

Note that if $k < 0$, both N_k and I_k are empty.

If k is any integer, $|k|$ denotes the absolute value of k. For $k \in N$, $j \in I$, we use the notation $k(\text{mod } j)$ to denote the remainder when k is divided by j. $\approx \text{mod } j$ denotes equivalence mod j, i.e. $k \approx \ell \text{ mod } j$ iff $k(\text{mod } j) = \ell(\text{mod } j)$.

For any positive real number r, log(r) denotes the logarithm to the base 2 of r. $\lfloor r \rfloor$ denotes the largest integer $\leq r$, and $\lceil r \rceil$ denotes the least integer $\geq r$, for any real number r.

ϕ denotes the empty set. If A is any set, P(A) denotes the set of all subsets of A, and $|A|$ denotes the cardinality of A. If $n \in I$ and A_i is any set for $i \in I_n$, then $\underset{k=1}{\overset{n}{X}} A_k$ denotes the usual Cartesian product. If $A_i = A$ for all $i \in I_n$, we use the notation A^n for $\underset{k=1}{\overset{n}{X}} A_k$. Note that A^0 is the set containing the unique 0-tuple.

We use the notation $\bar{a}_n$ for $(a_1,\ldots,a_n) \in \underset{k=1}{\overset{n}{X}} A_k$. For $\bar{a}_n \in \underset{k=1}{\overset{n}{X}} A_k$ and $j \in I_n$, $\pi_j(a_n) = a_j$. For $B \subseteq \underset{k=1}{\overset{n}{X}} A_k$, $\pi_j(B) = \{\pi_j(\bar{a}_n) | \bar{a}_n \in B\}$.

We use the usual definitions of equivalence relation, refinement of an equivalence relation, and equivalence classes. (See e.g. [Men64]). The index of an equivalence relation is the cardinality of the set of its equivalence classes.

We assume the reader is familiar with the basic definitions and concepts of formal language theory, as found in [Hop-Ull69]. We list these here.

By a finite alphabet, we mean any nonempty finite set whose elements are called symbols. A word w over an alphabet Σ is a function $N_\ell \to \Sigma$ for some $\ell \in Z$, $\ell \geq -1$. $\ell+1$ is called the length of the word w, and is denoted by $\ell n(w)$. λ denotes the unique word of length 0.

If Σ is a finite alphabet, Σ^* denotes the set of all words over Σ, including the empty word λ; $\Sigma^+ = \Sigma^* - \{\lambda\}$. If $A \subseteq \Sigma^*$, $\overline{A} = \Sigma^* - A$.

We let

$$\Sigma^k = \{w \in \Sigma^* \mid \ell n(w) = k\} \text{ and}$$

$$\Sigma^{\leq k} = \{w \in \Sigma^* \mid \ell n(w) \leq k\},$$

for $k \in N$. If σ is a symbol, σ^k denotes the word $\sigma\,\sigma \cdot \ldots \cdot \sigma$ of length k.

Juxtaposition, or sometimes "$\cdot$", denotes concatenation of words. We use the usual definitions of prefix and suffix of a word, subword of a word, and occurrence of a subword in a word.

Since our results involve Turing machines, and these operate on words over a finite alphabet, we find it useful to define mappings from numbers to words.

Definition 2.1 Let $n \in N$, $k \in I$, $k \geq 2$. Then we can represent n uniquely as

$$n = k^{\ell}a_{\ell} + k^{\ell-1}a_{\ell-1} + \ldots + a_0,$$

where $a_i \in N_{k-1}$ for $i \in N_{\ell}$, and $a_{\ell} \neq 0$. We define $k\text{-rep}(n) = a_{\ell}a_{\ell-1}\cdot\ldots\cdot a_0 \in N_k^+$. We define $b\text{-rep}(n) = 2\text{-rep}(n)$ for $n \in N$.

Let $k,m,n \in N$, $k \geq 2$. If $\ell n(k\text{-rep}(n)) \leq m$, we define

$$k\text{-rep}(m,n) = 0^{m-\ell n(k\text{-rep}(n))}\cdot k\text{-rep}(n).$$

Otherwise $k\text{-rep}(m,n)$ is undefined.

Definition 2.2 Let $k,i \in I$, $w \in N_{k-1}^*$. The i-th digit of w is $w(\ell n(w)-i)$, i.e. the i-th symbol of w, counting from right to left.

Section 3: Automata Theory Background

In future chapters, we will be describing decision procedures for theories informally, and obtaining upper bounds on such procedures. To give this precise meaning, we interpret such upper bounds in terms of the time or space required to implement the procedure on a Turing machine. For a rigorous definition of these machines, the reader can consult [Sto74, Section 2.2]. In fact, the exact conventions we choose for our Turing machine model do not affect our upper bound results, and so we present a brief verbal description of our chosen model of computation, the input/output Turing machine (IOTM). IOTM's are multitape Turing machines with a single read-only input tape, a single write-only output tape, and one or more work tapes; they may be deterministic or nondeterministic. Because the tape squares used on input and output tapes are not counted in space requirements of a computation it will be possible to consider a set being accepted within space $s(n)$ where $s(n)$ grows more slowly than linearly in n; similarly, we may also consider a function being computed within space $s(n)$ where $\ell n(f(w))$ is larger than $s(\ell n(w))$. Indeed, this is our reason for choosing this particular model.

The length of a computation of IOTM M on input w is the number of steps in the computation; the space used by a computation of IOTM M on input w is the number of work tape squares visited by heads of M during a computation. The output produced by a computation is the word written on the nonblank portion of the output tape at the end of the computation. (We assume the reader can make the notion of a computation, accepting computation, length, space, etc. precise. See, for example, [Sto74].)

If M is an IOTM with input alphabet Σ,

$L(M) = \{w \in \Sigma^* | M \text{ accepts } w\}$.

If $w \in \Sigma^*$,

$Time_M(w) = \min\{\ell | \text{there is an accepting computation of } M \text{ on input } w \text{ of length } \ell\}$,

and

$Space_M(w) = \min\{m | \text{there is an accepting computation of } M \text{ on input } w \text{ which uses space } m\}$.

We note that $Time_M(w)$ and $Space_M(w)$ are undefined if M does not accept w.

Definition 3.1 Let M be an IOTM, $A \subseteq \Sigma^*$, $S,T:N \to \{q \in Q | q \geq 0\}$. M accepts A within time T(n) (within space S(n)) if

1. M accepts $w \iff w \in A$ for all $w \in \Sigma^*$, and
2. for all but finitely many $w \in A$,
   $Time_M(w) \leq T(n(w))$
   $(Space_M(w) \leq S(\ell n(w)))$.

If M is deterministic, and $f:\Sigma^* \to \Delta^*$, M computes f within time T(n) (within space S(n)) iff for all $w \in \Sigma^*$,

1. M accepts w, and the unique computation of M on input w has output f(w), and
2. $Time_M(w) \leq T(\ell n(w))$
   $(Space_M(w) \leq S(\ell n(w)))$.

Definition 3.2

$NTIME(T(n))$ $(DTIME(T(n)))$ $= \{A | \text{there is a nondeterministic (deterministic) IOTM which accepts A within time } T(n)\}$.

NSPACE(S(n)) (DSPACE(S(n))) = {A|there is a nondeterministic (deterministic) IOTM which accepts A within space S(n)}.

For the sets we consider, a lower bound on the set will be a statement that the set does not belong to NTIME(T(n)) or NSPACE(S(n)) (for particular T(n) or S(n), or classes of such functions). This will mean that for any nondeterministic IOTM M which accepts the set,

$Time_M(w) > T(\ell n(w))$

($Space_M(w) > S(\ell n(w))$, respectively)

for infinitely many w in the set. An upper bound on the set will be a statement that the set is in DSPACE(S(n)), or sometimes simultaneously in NSPACE(S(n)) and NTIME(T(n)).

We also remark that if the set happens to be the set of true sentences in a first-order language determined by a particular model, and there is an efficiently recognizable (e.g. within polynomial time) set of axioms for the set of true sentences, then an upper bound on the lengths of proofs of sentences in the set implies an upper bound on the space required to decide membership in this set of true sentences. Thus, a lower bound on the set of sentences implies a lower bound on the lengths of proofs. Although we do not state such results in what follows, we remark here that they follow as a consequence of our results. See [Sto74] or [FiR74] for a further discussion of such results.

In order to compare the upper and lower bounds for the computationa complexity of the theories we shall consider, it is necessary to understand certain relationships known to hold between time and space for deterministic and nondeterministic computations. (These matters are discussed more fully in [Sto 74].)

Fact 3.3 Let $f:N \to N$.

A. Nondeterministic versus determinstic time

a) $DTIME(f(n)) \subseteq NTIME(f(n))$

b) $NTIME(f(n)) \subseteq \bigcup_{c \in N} DTIME(c^{f(n)})$.

B. Nondeterministic versus deterministic space

a) $DSPACE(f(n)) \subseteq NSPACE(f(n))$

b) $NSPACE(f(n)) \subseteq DSPACE((f(n))^2)$

C. Time versus space

a) $DTIME(f(n)) \subseteq DSPACE(f(n))$

$NTIME(f(n)) \subseteq NSPACE(f(n))$

b) $NSPACE(f(n)) \subseteq \bigcup_{c \in N} DTIME(c^{f(n)})$

All of Fact 2.1 is relatively straightforward to prove, with the exception of B.b. B.b is proved by Savitch [Sav70]. By (B), if we are discussing a lower or upper bound of the form "space 2^{cn} for some constant c" it is unnecessary to specify if we are talking about deterministic or nondeterministic space. Similarly, we can talk about a bound of the form "$2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\}$height cn for some constant c" without specifying if we are talking about time or space, either deterministically or nondeterministically.

Each of the gaps between a) and b) in A, B, C above represent important open questions of automata theory.

We also need to extend our notion of IOTM's computing functions to functions of several variables:

Definition 3.4 An IOTM M computes $f:(\Sigma^*)^n \to \Delta^*$ of n variables if M computes $f':(\Sigma \cup \{\#\})^* \to \Delta^*$ where $\# \notin \Sigma$ and

$f'(w_1\#w_2\#\cdot\ldots\cdot\#w_n) = f(w_1,\ldots,w_n)$ for all $w_1,\ldots,w_n \in \Sigma^*$.

The following notation is useful for comparing the growth rates of functions.

Definition 3.5 Let $k \in I$, $f,g:N \to N$. $f(n) = O(g(n))$ iff there is $c \in I$ such that $f(n) \leq c\cdot g(n)$ for sufficiently large n; $f(n) = o(g(n))$ iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.

The next two theorems show, essentially, that a given Turing machine can be "sped up" in time or space by an arbitrary constant factor (cf[HU69]). This implies that the time or space used by a Turing machine is not interesting to within a constant factor, and so (for instance) an upper bound of "$DSPACE(cn^2)$ for some c" implies an upper bound of $DSPACE(n^2)$.

Theorem 3.6 $DSPACE(f(n)) \subseteq DSPACE(c\cdot f(n))$ for any $c > 0$.

Theorem 3.7 $DTIME(f(n)) \subseteq DTIME(c\cdot f(n))$ for any $c > 0$, provided $\lim_{n\to\infty} (f(n)/n) = \infty$.

Similar theorems hold for nondeterministic machines.

Section 4: Using Reducibilities to Prove Upper and Lower Bounds

We now present two notions of efficient transformation of a set A into a set B, (for short, $A \leq B$). Informally, $A \leq B$ will mean that questions about membership of elements in A can efficiently be transformed into questions about membership of elements in B.

Definition 4.1 Let logspace (polylin) denote the class of functions

$\{f \mid f:\Sigma^* \to \Delta^*$ for some finite alphabets Σ and Δ, and there is a deterministic IOTM which computes f within space log (n) (within time p(n) and space (n) for some polynomial p(n))$\}$.

Definition 4.2 Let $f:\Sigma^* \to \Delta^*$. f is linear bounded if there is $c \in I$ such that $\ell n(f(w)) \leq c\, \ell n(w)$ for all $w \in \Sigma^+$.

Definition 4.3 Let $A \subseteq \Sigma^+$, $B \subseteq \Delta^+$ for some finite alphabets Σ and Δ. $A \leq_{\text{log-lin}} B$ ($A \leq_{p\ell} B$) via f iff f is a function, $f:\Sigma^+ \to \Delta^+$, such that

1. $w \in A \iff f(w) \in B$ for all $w \in \Sigma^+$, and
2. $(\leq_{\text{log-lin}})$: f $\in$ logspace and f is linear bounded.
   $((\leq_{p\ell})$: f $\in$ polylin and f is linear bounded).

Let G be a class of sets, and $\leq \in \{\leq_{\text{log-lin}}, \leq_{p\ell}\}$. $G \leq B$ iff for all $A \in G$, $A \leq B$.

We now state Lemma 4.4, which is a very powerful way of proving lower and upper bounds. For a proof (which is really very simple) of this fact and for a very thorough discussion of reducibilities, see [Sto74].

Lemma 4.4 Say that $L_1 \leq_{p\ell} L_2$ ($L_1 \leq_{\text{log-lin}} L_2$). Let $f:N \to N$. (Let f be monotone increasing). If

$$L_2 \in \begin{cases} \text{DTIME}(f(n)) \\ \text{DSPACE}(f(n)) \\ \text{NTIME}(f(n)) \\ \text{NSPACE}(f(n)) \end{cases}, \text{ then } L_1 \in \begin{cases} \text{DTIME}(f(cn)+p(n)) \\ \text{DSPACE}(f(cn)+n)\ (\text{DSPACE}(f(cn)+\log(n)) \\ \text{NTIME}(f(cn)+p(n)) \\ \text{NSPACE}(f(cn)+n)\ (\text{NSPACE}(f(cn)+\log(n)) \end{cases}$$

for some constant $c > 0$ and polynomial $p(n)$.

Contrapositively, if

$$L_1 \notin \begin{cases} \text{DTIME}(f(n)+p(n)) \\ \text{DSPACE}(f(n)+n)\ (\text{DSPACE}(f(n)+\log(n))) \\ \text{NTIME}(f(n)+p(n)) \\ \text{NSPACE}(f(n)+n)\ (\text{NSPACE}(f(n)+\log(n))) \end{cases}, \text{ then } L_2 \notin \begin{cases} \text{DTIME}(f(cn)) \\ \text{DSPACE}(f(cn)) \\ \text{NTIME}(f(cn)) \\ \text{NSPACE}(f(cn)) \end{cases}$$

for some constant $c > 0$ and some polynomial p.

An example of the way we use Lemma 3.2 is the following: say that we have languages L_1 and L_2 such that we know that $L_2 \in \text{SPACE}(2^{2^{cn}})$ for some constant c. If $L_1 \leq_{p\ell} L_2$ then we can conclude that $L_1 \in \text{SPACE}(2^{2^{cn}})$ for some constant c. If we know that $L_1 \notin \text{NTIME}(2^{2^{c'n}})$ for some constant $c' > 0$, and if $L_1 \leq_{p\ell} L_2$, then we can conclude that $L_2 \notin \text{NTIME}(2^{2^{c'n}})$ for some constant $c' > 0$[†]. This latter idea is often used in conjunction with Lemma 4.5.

Lemma 4.5 (see [Co73, SFM73, SFM77].) Let $f:N \to N$ be one of the functions

$$\left.\begin{matrix} 2^n,\ 2^{2^n},\ 2^{2^{2^n}},\ \text{or } 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \end{matrix}\right\} \text{ height } n \ .$$

[†] It is easy to see that if $L \notin \text{NTIME}(f(n))$, then any nondeterministic Turing machine which recognizes L takes time at least f(n) on some $\gamma \in L$ of length n, for infinitely many n.

Then there exists a language L such that $L \in NTIME(f(n))$ and $L \notin NTIME(f(n/2))$.

Theorem 4.6 Let $f:N \to N$ be one of the functions $2^n, 2^{2^n}, 2^{2^{2^n}}$ or $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\}$ height and let $L_0 \subseteq \Sigma^*$ (for some Σ^*) be such that $NTIME(f(n)) \leq_{p\ell} L_0$. Then for some constant $c > 0$, $L_0 \notin NTIME(f(cn))$.

Proof Say that $NTIME(f(n)) \leq_{p\ell} L_0$. By Lemma 4.5, let L be such that $L \notin NTIME(f(n/2))$ and $L \in NTIME(f(n))$. So $L \leq_{p\ell} L_0$. By Lemma 4.4, $L_0 \notin NTIME(f(cn))$ for some constant $c > 0$. □

A typical way Theorem 4.6 is used is the following. Fischer and Rabin [FiR 74] show that if TH is the theory of integer addition, then $NTIME(2^{2^n}) \leq_{p\ell} TH$, concluding that $TH \notin NTIME(2^{2^{cn}})$ for constant c. In Chapter 3 we show that $TH \in SPACE(2^{2^{c'n}})$ for some constant c', and hence that $TH \in DTIME(2^{2^{2^{c'n}}})$ for some constant c'.

A natural question is whether or not we can get a DTIME upper bound for TH and an NTIME lower bound for TH which are closer to each other than are $2^{2^{2^{c'n}}}$ and $2^{2^{cn}}$. If we could, this would settle an important open question of automata theory. For instance, say that we could show that $TH \in DTIME(2^{2^{2^{c'\sqrt{n}}}})$ for some constant c'. Since $NTIME(2^{2^n}) \leq_{p\ell} TH$, Lemma 4.4 would imply that $NTIME(2^{2^n}) \subseteq \bigcup_{c \in N} DTIME(2^{2^{2^{c\sqrt{n}}}})$, narrowing the gap in Fact 3.3.A. This would also contradict the popular conjecture that (for most functions f that are encountered) there is a language in NTIME(f(n)) which <u>requires</u> $DTIME(c^{f(n)})$ for some constant c. The reason therefore that we have not been able to narrow the gap between

our DTIME upper bound and NTIME lower bound for TH, is not because we do not understand the expressive power and other properties of TH, but rather because we don't understand many basic properties of the very notions of deterministic and nondeterministic computation.

In proving $A \leq_{\text{log-lin}} B$ for particular sets A,B in what follows, we generally sketch a proof that $f \in$ logspace based on the properties of log space given by the Lind-Meyer characterization of logspace [Lind74]. In particular, we use the facts that logspace is closed under explicit transformation, composition, concatenation, and two sided recursion of concatenation [Lind74]. where a function $f:(\Sigma^*)^{n+1} \to \Delta^*$ of n+1 variables is defined from functions $g:(\Sigma^*)^n \to \Delta^*$ and $h_1,h_2:(\Sigma^*)^{n+2} \to \Delta^*$ by two sided recursion of concatenation if f satisfies

$$f(\bar{w}_n,\lambda) = g(\bar{w}_n)$$
$$f(\bar{w}_n,w\sigma) = h_1(\bar{w}_n,w,\sigma)\cdot f(\bar{w}_n,\bar{w})\cdot h_2(\bar{w}_n,w,\sigma)$$
$$\text{for all } \bar{w}_n \in (\Sigma^*)^n,\ w \in \Sigma^*,\ \sigma \in \Sigma.$$

Formal proofs that the functions we define are indeed in logspace follow by means of this characterization.

Section 5: Logic Background and Notation

Definition 5.1 A first order language L is specified as follows:

1. L has formal variables $x_0, x_1, x_{10}, \ldots$ (i.e. the subscripts are written in binary notation);
2. L has logical symbols $\sim$, $\wedge$, $\vee$, $\rightarrow$, $\forall$, $\exists$, (,);
3. L has finitely many relation symbols $\underline{R}_1, \underline{R}_2, \ldots, \underline{R}_\ell$, with $\underline{R}_i$ a t_i-ary relation symbol for $1 \leq i \leq \ell$;
4. L has a (possible) constant symbol $\underline{e}$.

For technical convenience, we do not allow function symbols in L. We will often omit the constant symbol $\underline{e}$ from the definition of L. For expository convenience, we use t, u, v, x, y, z, with or without numerical subscripts or primes, to denote formal variables.

The atomic formulas of L are of the form $\underline{R}_i(v_1, v_2, \ldots, v_{t_i})$ where $v_1, v_2, \ldots, v_{t_i}$ represent (not necessarily distinct) formal variables; if L has a constant symbol $\underline{e}$ then each v_j, $1 \leq j \leq t_i$, can represent a formal variable or $\underline{e}$. We define the formulas of L recursively as follows: atomic formulas are formulas; if F_1 and F_2 are formulas and v is a formal variable, then

$$(F_1 \vee F_2)$$
$$(F_1 \wedge F_2)$$
$$(F_1 \rightarrow F_2)$$
$$\sim F_1$$
$$\forall v\, F_1$$
$$\exists v\, F_1$$

are formulas. We use the usual notions of a quantifier-free formula, and of an occurrence of a variable in a formula being bound or free, (c.f.

[Men64] or [Sho67].) A sentence of L is a formula in which there are no free occurrences of variables.

When defining formulas, we often omit the complete parenthetization required by the definition of formula; we also sometimes add superfluous parentheses to improve readability. In all such cases, it will be clear how to rewrite the expression in question to obtain a formula in the language L.

Let L be a first order language and let $k \in \mathbb{N}$. If $k \geq 1$, $F(\bar{y}_k)$ denotes a formula in L with at most the free variables $y_1,\ldots,y_k$; if $k = 0$, $F(\bar{y}_k)$ denotes a formula with no free variables. Also note the notation $F(\bar{y}_k)$ implies a specific ordering of the free variables of F. We use this ordering as follows. Given a formula $F(\bar{y}_k)$, and variables $z_1,\ldots,z_k$ with no bound occurrence in $F(\bar{y}_k)$, $F(\bar{z}_k)$ denotes that formula obtained from $F(\bar{y}_k)$ by substituting z_i for each free occurrence of y_i, for $1 \leq i \leq k$.† We use $\exists \bar{y}_k F(\bar{y}_k)$ as an abbreviation for the formula

$$\exists y_1 \ldots \exists y_k F(\bar{y}_k)$$

for $k \geq 1$.

A structure for L is a tuple $A = \langle A, R_1,\ldots,R_\ell\rangle$ where A is a non-empty set and $R_i \subseteq A^{t_i}$ for $1 \leq i \leq \ell$; if L has a constant symbol $\underline{e}$, then a structure for L is $\langle A, R_1,\ldots,R_\ell, e\rangle$ where $e \in A$. We call A the domain of A. We use the notation $\mathcal{A},\mathcal{B},\mathcal{C},\ldots$ for structures, and $A,B,C,\ldots$ for the domains of $\mathcal{A},\mathcal{B},\mathcal{C},\ldots$, respectively. If $\mathcal{A}$ is a structure, and $B \subseteq A^n$, we let $\langle \mathcal{A}, B\rangle$ denote the structure obtained from $\mathcal{A}$ by adding the n-ary relation B.

We use $\mathbb{N}$ to stand for the structure $\langle \mathbb{N},+,\leq,0\rangle$, $\mathbb{Z}$ for the structure $\langle \mathbb{Z},+,\leq,0\rangle$, and $\mathbb{R}$ for the structure $\langle \mathbb{R},+,\leq,0\rangle$.

† If some of $z_1,\ldots,z_k$ occur bound in F, we obtain $F(\bar{z}_k)$ by first changing the names of the conflicting bound variables, and then substituting each z_i for y_i

If F is a sentence of L we use the usual notion F true in A or A satisfies F or F holds in A, and we write $A \models F$.

Let A be a structure, $F(\bar{y}_k)$ a formula. If $k = 0$, A satisfies $F(\bar{a}_k)$ iff A satisfies F. If $k \geq 1$, let $a_1,\ldots,a_k \in A$. Then A satisfies $F(\bar{a}_k)$ iff A satisfies $F(\bar{y}_k)$ when y_i is interpreted as a_i, for $1 \leq i \leq k$. In both cases we write $A \models F(\bar{a}_k)$.

TH(A) = the theory of A = $\{F \mid F$ is a sentence and $A \models F\}$.

If C is a nonempty collection of structures, TH(C) = theory of $C = \bigcap_{A \in C} TH(A)$.

Let C be a collection of structures. A sentence F is satisfiable in C if $A \models F$ for some $A \in C$. We let

SAT(C) = $\{F \mid F$ is a sentence and F is satisfiable in C$\}$.

We say $F(\bar{x}_k)$ and $G(\bar{x}_k)$ are (logically) equivalent if for all structures A, and all $\bar{a}_k \in A^k$,

$A \models F(\bar{a}_k)$ iff

$A \models G(\bar{a}_k)$.

The logical symbol $\leftrightarrow$ has been omitted from L for technical convenience. This makes L no less powerful a language, since for any formulas F,G, $F \leftrightarrow G$ is equivalent to $((F \rightarrow G) \wedge (G \rightarrow F))$. Also, our language would have been just as powerful had we left out much of our logical notation. For instance, $x \vee y$ is equivalent to $(\sim x) \rightarrow y$ and $\forall x F$ is equivalent to $\sim \exists x \sim F$. It is only for convenience that we have made L as large as we have.

We say a formula F is a Boolean combination of subformulas $F_1, F_2, \ldots, F_k$ if F is obtained by combining $F_1, F_2, \ldots, F_k$ using at most $\wedge$, $\vee$, $\rightarrow$, $\sim$, and no quantifiers. Clearly every formula is equivalent to a Boolean combination of formulas, each of which begins with an existential quantifier.

Since we shall be interested in Turing machines whose input strings are sentences of L, we have to have a precise notion of the alphabet used to write formulas and a precise notion of the length of formulas. Our alphabet consists of $\Sigma = \{(,), \sim, \wedge, \vee, \rightarrow, \forall, \exists, \underline{R}, x, 0, 1\}$ (where 0 and 1 are used to write subscripts of variables and relation symbols in binary); if $\underline{e}$ is a symbol of L, then $\underline{e} \in \Sigma$ also. If F is a formula, then by the length of F, written length (F), we simply mean the length of F as a member of Σ^*.

For $i \in I$, Q_i always represents $\exists$ or $\forall$.

Definition 5.2 A formula F is in prenex normal form if it is of the form $Q_1 v_1 \, Q_2 v_2 \, \ldots \, Q_k v_k \, F'$, where F' is quantifier-free and $v_1, \ldots, v_k$ represent formal variables.

Theorem 5.3 Every formula F is equivalent to a formula G in prenex normal form such that G has at most length (F) quantifiers and is of length at most length (F)·log(length(F)). Furthermore, there is a procedure (i.e. a Turing machine) which given F computes G within time polynomial in length(F).

Proof There is a standard procedure for converting a formula to one in prenex normal form [Men64]. The procedure basically just "pulls out" the quantifiers to the front, except that first the names of certain variables have to be changed in order for the procedure to produce a formula equivalent to the initial one. The procedure does not change the number of quantifiers,

so G has at most length(F) quantifiers. F has at most length (F) occurrences of variables, so if these are given all different names (in the worst case) and the binary subscripts are chosen to be as short as possible, then F grows by a factor of at most log (length(F)) when put prenex normal form. This procedure can be checked to operate within polynomial time. □

Thus, to show that a theory can be decided within space f(cn) for some constant c, where f grows faster than polynomially, it is sufficient to give a procedure which decides the truth of prenex normal form sentences of length at most n log n with at most n quantifiers, within space f(cn) for some constant c.

Definition 5.4 If F is a formula, we write q-depth(F) to mean the quantifier depth of F. Formally, if F is an atomic formula then q-depth(F) = 0; if F_1 and F_2 are formulas then q-depth($F_1 \vee F_2$) = q-depth($F_1 \wedge F_2$) = q-depth($F_1 \rightarrow F_2$) = Max{q-depth(F_1),q-depth(F_2)}, q-depth($\sim F_1$) = q-depth(F_1), and q-depth($\exists v F_1$) = 1+q-depth(F_1).

Section 6: Practical Relevance of the Theoretical Results

It seems wise at this point to discuss the relationship between the theoretical complexity bounds (both lower and upper) that we shall present, and the practical issues involved with real computing.

Consider for example a lower bound result of Chapter 8. This says that if $\mathcal{C}$ is a nonempty collection of pairing functions, then there is a constant $c > 0$ such that no Turing machine can accept $\mathcal{C}$ within time $f(cn)$ where $f(n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\}$ height n. Although this certainly seems to imply that $TH(\mathcal{C})$ is in some sense "difficult", there are a number of possible objections to the relevance of this theorem. One objection is that the value of c is not specified. Another problem is that the theorem only guarantees that any Turing machine accepting $TH(\mathcal{C})$ will take longer than $f(c|x|)$ steps for infinitely many inputs x; in reality one is only interested in a finite number of inputs. This latter point is related to the well known theorem that given any Turing machine M and any finite set of inputs F, one can construct an equivalent machine M' which operates almost as fast as M on all inputs and which operates on members of F very quickly. The construction increases the number of states of the machine, and is not in any sense valid for real computers. Theorems 3.6 and 3.7 of this chapter are two other results not valid for real computation. These constructions speed up the time or space used by a given machine by a constant factor, but only at the expense of increasing the number of states and tape symbols.

While these objections to the practicality of the lower bound results (as stated) are completely valid, it turns out that the proofs contain more information and have a more direct connection with the real

world. This is partly because Turing machines can simulate real computers, although with a loss of efficiency. If one puts together a careful version of this simulation result with the information in the proof that TH($\mathbb{C}$) isn't elementary - recursive, one establishes the following theorem:

Theorem Consider a real computer which, when given a sentence in the language of $\mathbb{C}$ with at most 1000 connectives and quantifiers, eventually decides if the sentence is in TH($\mathbb{C}$) or not. Then the machine will take at least 1,000,000,000 years on some input.

[Sto74] contains some additional theorems along these lines. While the other lower bound results may not imply such a strong theorem as above, similar theorems do follow from the proofs.

Another kind of objection to the lower bound results is that even if one is interested in deciding if long formulas are in TH($\mathbb{C}$), one may not be interested in arbitrary formulas but rather in some particular class of formulas determined by a particular application. Whether the proofs apply in this case depends upon the restricted class one is interested in. What the proofs do make clear, however, is the absolute necessity of studying closely what is special about the restricted class, in the hope of coming up with an efficient algorithm. Although it is very easy to say that one is only interested in a restricted version of the general problem, in practice it is usually enormously difficult to describe what is really special about the sentences arising from a particular application in question.

Turing to upper bounds, what is the practical value of the results stating that various theories can be decided in time one, two, or more exponentials? Firstly, the proofs give algorithms which can be implemented

on real machines, as well as exact upper bounds on their running times. These algorithms, or variants on them, may really be useful when working with sufficiently short formulas.

More importantly, these results have practical significance for the working theoretical logician. When upper and lower bounds match closely, then it is fair to say that one understands reasonably well the power of expressability of a particular theory. All these theorems and proofs help us better understand what formulas in a particular language can say about a class of structures of interest, and hence better understand the nature of the theory in question.

CHAPTER 2

EHRENFEUCHT GAMES AND DECISION PROCEDURES

Section 1: Ehrenfeucht Games

In this chapter we present a development of the Ehrenfeucht game approach to deciding logical theories. This approach was originally described in [Ehr61] and in particular the reader may wish to consult this source to learn about the relationship to game theory. Richard Tenney [Ten74, Ten74'] also discusses game theory, and applies techniques much like those here to decide the theories of certain pairing functions, and to decide the second order theory of an equivalence relation. Neither Ehrenfeucht nor Tenney describe their techniques in great generality. What we shall do here is to provide a very general framework for discussing Ehrenfeucht games; we then show how this framework can be used to provide decision procedures which are not only nearly optimal in efficiency, but which can be very elegantly described as well. Part of our development also involves a characterization of Ehrenfeucht games in terms of the quantifier depth of formulas.

Let L be a first order language with only relational symbols (as described earlier) and let $\mathcal{C}$ be a class of structures. The motivation for our approach to deciding membership in $TH(\mathcal{C})$ is as follows:

We wish to obtain an intuitive upper bound on the power of expressibility of formulas of L with respect to $\mathcal{C}$. For instance, for two models $\mathcal{A}$ and $\mathcal{B}$, we can ask if there is a sentence of L which distinguishes $\mathcal{A}$ from $\mathcal{B}$ -- that is, is true

in $\mathcal{A}$ but not $\mathcal{B}$, or vice-versa. We can also ask if there are any "short" sentences which distinguish $\mathcal{A}$ from $\mathcal{B}$; if not, it suffices to check the truth of a "short" sentence in $\mathcal{A}$, and forget about $\mathcal{B}$. Similarly, given a structure $\mathcal{A}$ and two k-tuples $\bar{a}_k$ and $\bar{b}_k$ from $\mathcal{A}$, one can ask if there is any formula $F(\bar{x}_k)$ which is satisfied by $\bar{a}_k$ and not $\bar{b}_k$, or if there is any "short" formula $F(\bar{x}_k)$ which has this property. Answers to these types of questions are often obtained through application of the (Ehrenfeucht game) theorems to be presented in this chapter. If one is able to obtain <u>suitable</u> answers, then the decision procedure can take on a very simple form. To test if a sentence is true in every member of $\mathcal{C}$, it will suffice to check validity in a particular <u>finite</u> subset of $\mathcal{C}$; to determine truth in each structure in this subset, it will suffice to determine truth when each quantifier is limited to range over a particular finite subset of the domain.

Since in each case our decision procedure will involve testing truth in each of a finite, but possibly large, set of structures, it will be necessary to have finite descriptions of (some) structures which one can actually write down in the course of the procedure. Hence, we associate with each structure in $\mathcal{C}$ a <u>norm</u>, by which we mean a member of Σ^* where Σ is a finite alphabet. (A different Σ may be chosen for different sets $\mathcal{C}$.) Since the decision procedure will also involve letting the quantifiers of a sentence range over members of structures, we also associate with each member of the domain of each structure

in $\mathcal{C}$, a norm which will also be a member of Σ^*. For a structure A we denote the norm of A by $\|A\|$; for $a \in A$, we denote the norm of a by $\|a\|$. In addition, we will have a partial ordering, $\preceq$, on Σ^* which will be useful in our decision procedures. Although we could insist that Σ be always $\{0,1\}$, it will sometimes aid the clarity of our procedures to allow Σ to be a larger, finite set.

For convenience, we will also allow a structure or member of a domain to have the norm ∞; we take every other norm to be $\preceq \infty$. If $m \in \Sigma^* \cup \{\infty\}$, we will write $A \preceq m$ or $a \preceq m$ instead of $\|A\| \preceq m$ or $\|a\| \preceq m$.

Definition Let $H: N \times N \times (\Sigma^* \cup \{\infty\}) \to \Sigma^* \cup \{\infty\}$. Let A be a structure Say that for every $n,k \in N$ and $m \in \Sigma^* \cup \{\infty\}$, and $\bar{a}_k \in A^k$ such that $a_i \preceq m$ for $1 \leq i \leq k$, and formula $F(\bar{x}_{k+1})$ of q-depth $\leq n$ the following is true: if $(A,\bar{a}_k)$ satisfies $\exists x_{k+1} F(\bar{x}_{k+1})$, then for some $a_{k+1} \preceq H(n,k,m)$, $(A,\bar{a}_{k+1})$ satisfies $F(\bar{x}_{k+1})$ [in this case we write that $(A,\bar{a}_k)$ satisfies $(\exists x_{k+1} \preceq H(n,k,m))F(\bar{x}_{k+1})$]; Then we say that A is H-bounded.

If we know a structure to be H-bounded for appropriate H, then often Theorem 1 immediately yields an efficient and easily described decision procedure for the theory of that structure. Many examples of this appear in the chapters which follow.

Theorem 1 Say that A is H-bounded. Let $n,k \in N$ and let $Q_1 x_1 Q_2 x_2 \ldots Q_k x_k F(\bar{x}_k)$ be a sentence with q-depth $\leq n+k$, i.e.,

q-depth(F) $\leq$ n. Let $m_0, m_1, m_2, \ldots, m_k$ be a sequence of members of $\Sigma^* \cup \{\infty\}$ such that $m_0 \preceq m_1 \preceq \ldots \preceq m_k$ and $H(n+k-i, i-1, m_{i-1}) \preceq m_i$ for $1 \leq i \leq k$.

Then $Q_1x_1Q_2x_2\ldots Q_kx_kF(\bar{x}_k)$ is true in A <=> $(Q_1x_1 \preceq m_1)(Q_2x_2 \preceq m_2)\ldots(Q_kx_k \preceq m_k)F(\bar{x}_k)$ is true in A.

Proof We will prove by induction on i, $0<i \leq k+1$, that

$A \models Q_1x_1Q_2x_2\ldots Q_kx_kF(\bar{x}_k)$ <=>

$A \models (Q_1x_1 \preceq m_1)(Q_2x_2 \preceq m_2)\ldots(Q_{i-1}x_{i-1} \preceq m_{i-1})Q_ix_i\ldots Q_kx_kF(\bar{x}_k)$.

The base case, i = 1, is just

$A \models Q_1x_1Q_2x_2\ldots Q_kx_kF(\bar{x}_k)$ <=> $A \models Q_1x_1Q_2x_2\ldots Q_kx_kF(\bar{x}_k)$.

Assume the induction hypothesis for some i, $0 < i \leq k$; we shall prove it for i+1. Consider any $\bar{a}_{i-1} \in A^{i-1}$, such that $a_j \preceq m_j$ for $1 \leq j \leq i-1$. Then since A is H-bounded,

$A \models Q_ix_i[Q_{i+1}x_{i+1}\ldots Q_kx_kF(\bar{a}_{i-1}, x_i, x_{i+1}, \ldots, x_k)]$ <=>

$A \models (Q_ix_i \preceq H(n+k-i, i-1, m_{i-1}))(Q_{i+1}x_{i+1}\ldots Q_kx_kF(\bar{a}_{i-1}, x_i, x_{i+1}, \ldots, x_k))$

(Note: This follows directly from the definition of H-boundedness only if Q_i is $\exists$, but it is easy to see that this implies it must be true if Q_i is $\forall$.) Since $H(n+k-i, i-1, m_{i-1}) \preceq m_i$, we have that for all $\bar{a}_{i-1} \in A^{i-1}$ such that $a_j \preceq m_j$ for $1 \leq j \leq i-1$,

$A \models Q_ix_iQ_{i+1}x_{i+1}\ldots Q_kx_kF(\bar{a}_{i-1}, x_i, x_{i+1}, \ldots, x_k)$ <=>

$A \models (Q_ix_i \preceq m_i)Q_{i+1}x_{i+1}\ldots Q_kx_kF(\bar{a}_{i-1}, x_i, x_{i+1}, \ldots, x_k)$. So

$A \models Q_1x_1Q_2x_2\ldots Q_kx_kF(\bar{x}_k)$ <=>

$A \models (Q_1x_1 \preceq m_1)(Q_2x_2 \preceq m_2)\ldots(Q_{i-1}x_{i-1} \preceq m_{i-1})Q_ix_i\ldots Q_kx_kF(\bar{x}_k)$ <=>

$A \models (Q_1x_1 \preceq m_1)(Q_2x_2 \preceq m_2)\ldots(Q_ix_i \preceq m_i)Q_{i+1}x_{i+1}\ldots Q_kx_kF(\bar{x}_k)$.

So the induction hypothesis is true for i+1.

If we know that a structure A is H-bounded, it is not hard to see how Theorem 1 may possibly lead to a decision procedure for TH(A). Say that we are given a sentence $Q_1x_1 \ldots Q_kx_kF(\bar{x}_k)$ where F is quantifier free. Let $m_0 \preceq m_1 \preceq \ldots \preceq m_k$ be members of Σ^* such that $H(k-i,i-1,m_{i-1}) \preceq m_i$ for $1 \leq i \leq k$. Then essentially we have to decide if $(Q_1x_1 \preceq m_1)\ldots(Q_kx_k \preceq m_k)F(\bar{x}_k)$ is true in A; that is instead of x_i being quantified to range over <u>all</u> elements of A, we can assume x_i only ranges over those elements of A of norm $\preceq m_i$. Say that for each i, there are only a finite number of elements of A of norm $\preceq m_i$, and that we can cycle through representations of all such elements; say also that we can determine if a given k-tuple of such representations satisfies F; then we can determine whether or not $Q_1x_1 \ldots Q_kx_kF(\bar{x}_k)$ is true in A.

In practice in what follows, the space needed to cycle through the representations will be no bigger than the size of the longest such representation. In addition, the space needed to see if a particular k-tuple of such representations satisfies F will be no bigger than the space on which F and the k-tuple is written (in fact, this operation will in general be quite simple). The total space required, therefore, will be no worse than the space needed to write down F and the largest k-tuple of representations.

<u>Definition</u> Let $h:N \to \Sigma^* \cup \{\infty\}$. We say that the class C is (h,H) bounded if for every $n \in N$:

1) If F is a sentence of q-depth $\leq n$, then $F \in SAT(\mathcal{C}) \iff$ F is true in some structure in $\mathcal{C}$ of norm $\preceq h(n)$,

and 2) For every $A \in \mathcal{C}$, A is H-bounded.

In what follows, when we wish to decide membership in $SAT(\mathcal{C})$ we will first show $\mathcal{C}$ to be (h,H) bounded for some appropriate h and H. Given a sentence F of q-depth n, we will then proceed by cycling through representations of all structures of $\mathcal{C}$ of norm $\leq h(n)$, using the technique described above, in order to decide for each such H-bounded structure if it makes F true. The space used by this procedure will be the worst case of space needed to write down the longest representation together with the space needed (as described above) to determine if F is true in the represented structure.

The rest of this chapter will be devoted to developing Ehrenfeucht game methods for showing a structure to be H-bounded, or a class of structures to be (h,H) bounded. We first define the Ehrenfeucht equivalence relations $\equiv_{n,k}$.

Definition Let $n,k \in N$, let $A,\mathcal{B} \in \mathcal{C}$, let $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$. Then we write $(A,\bar{a}_k) \equiv_{n,k} (\mathcal{B},\bar{b}_k)$ if for all formulas $F(\bar{x}_k)$ of q-depth $\leq n$, $(A,\bar{a}_k)$ satisfies $F(\bar{x}_k) \iff (\mathcal{B},\bar{b}_k)$ satisfies $F(\bar{x}_k)$. For convenience, we shall write $(A,\bar{a}_k) \equiv_n (\mathcal{B},\bar{b}_k)$ instead of $(A,\bar{a}_k) \equiv_{n,k} (\mathcal{B},\bar{b}_k)$, and when A and $\mathcal{B}$ are understood we shall merely write $\bar{a}_k \equiv_n \bar{b}_k$.

It is easy to see that each $\equiv_{n,k}$ is an equivalence relation on tuples of the form $(A,\bar{a}_k)$ where $A \in \mathcal{C}$ and $\bar{a}_k \in A^k$. In the

future, we will identify $(A,\bar{a}_0)$ with A; in this we can view $\equiv_{n,0}$ as being an equivalence relation on the class $\mathcal{C}$.

Theorem 2 gives an inductive characterization of $\equiv_{n,k}$. In fact, Ehrenfeucht originally used this as his definition of $\equiv_{n,k}$, for $n > 0$. Theorem 2 will be proved later in this chapter.

Theorem 2 Let $n,k \in N$, let $A,\mathcal{B} \in \mathcal{C}$, let $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$. Then $(A,\bar{a}_k) \equiv_{n+1} (\mathcal{B},\bar{b}_k) \iff$

1) For every $a_{k+1} \in A$, there exists some $b_{k+1} \in B$ such that $(A,\bar{a}_{k+1}) \equiv_n (\mathcal{B},\bar{b}_{k+1})$ and

2) For every $b_{k+1} \in B$, there exists some $a_{k+1} \in A$ such that $(A,\bar{a}_{k+1}) \equiv_n (\mathcal{B},\bar{b}_{k+1})$.

An Ehrenfeucht game decision procedure consists (for our purposes) of defining a set of equivalence relations $E_{n,k}$, which will turn out to be refinements of the relations $\equiv_{n,k}$; one then uses these relations to show H-boundedness of a structure (or (h,H) boundedness of a class), as in Theorem 3 (or 4).

For $n,k \in N$, let $E_{n,k}$ be an equivalence relation on things of the form $(A,\bar{a}_k)$ where $A \in \mathcal{C}$ and $\bar{a}_k \in A^k$. Let $H:N\times N\times(\Sigma^* \cup \{\infty\}) \to \Sigma^* \cup \{\infty\}$ be a function.

Theorem 3 Say that the relations $\{E_{n,k}\}$ satisfy the following properties for every $A,\mathcal{B} \in \mathcal{C}$, $n,k \in N$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$, and $m \in \Sigma^* \cup \{\infty\}$.

1) $(A,\bar{a}_k)\ E_{0,k}\ (\mathcal{B},\bar{b}_k) \Rightarrow (A,\bar{a}_k) \equiv_0 (\mathcal{B},\bar{b}_k)$.

2) If $(A,\bar{a}_k)\ E_{n+1,k}\ (B,\bar{b}_k)$ and $b_i \preceq m$ for all i, $1 \leq i \leq k$, then for all $a_{k+1} \in A$ there exists a $b_{k+1} \in B$ such that $b_{k+1} \succeq H(n,k,m)$ and $(A,\bar{a}_{k+1})\ E_{n,k+1}\ (B,\bar{b}_{k+1})$.

THEN

I) $(A,\bar{a}_k)\ E_{n,k}\ (B,\bar{b}_k) \Rightarrow (A,\bar{a}_k) \equiv_n (B,\bar{b}_k)$ for every $n,k \in N$, $A,B \in C$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$.

II) Every structure in C is H-bounded.

Proof We shall first prove I by induction on n. I is clearly true (for all k) if n = 0.

Assume now that I is true for some particular value of n (and all k); we shall prove it for n+1. So say that $(A,\bar{a}_k)\ E_{n+1,k}\ (B,\bar{b}_k)$, and let $F(\bar{x}_k)$ be a formula of q-depth n+1; we will show that $A \models F(\bar{a}_k) \iff B \models F(\bar{b}_k)$. Since every formula of q-depth n+1 is equivalent to a boolean combination of formulas beginning with an existential quantifier, it is sufficient to show that $A \models \exists x_{k+1} G(\bar{a}_k, x_{k+1}) \iff B \models \exists x_{k+1} G(\bar{b}_k, x_{k+1})$ for G a formula of q-depth n. By symmetry, it is sufficient to show $\Rightarrow$. Assuming $A \models \exists x_{k+1} G(\bar{a}_k, x_{k+1})$, let $a_{k+1} \in A$ be such that $A \models G(\bar{a}_{k+1})$. Since $(A,\bar{a}_k)\ E_{n+1,k}\ (B,\bar{b}_k)$, we have that for some $b_{k+1} \in B$, $(A,\bar{a}_{k+1})\ E_{n,k+1}\ (B,\bar{b}_{k+1})$. By the induction hypothesis, $(A,\bar{a}_{k+1}) \equiv_{n,k+1} (B,\bar{b}_{k+1})$, and since $A \models G(\bar{a}_{k+1})$, we have $B \models G(\bar{b}_{k+1})$. So $B \models \exists x_{k+1} G(\bar{b}_k, x_{k+1})$.

We now prove II. Let k and n be arbitrary, $A \in \mathcal{C}$. Let $F(\bar{x}_{k+1})$ have q-depth $\leq n$, and say that $A \models \exists x_{k+1} F(\bar{a}_k, x_{k+1})$. Then for some a_{k+1}, $A \models F(\bar{a}_{k+1})$. Since $(A,\bar{a}_k)\ E_{n+1,k}\ (A,\bar{a}_k)$, we have by 2) that for some $a'_{k+1} \preceq H(n,k,m)$, $(A,\bar{a}_{k+1})\ E_{n,k+1}\ (A,\bar{a}_k,a'_{k+1})$. By I, $(A,\bar{a}_{k+1})\ \equiv_{n,k+1}\ (A,\bar{a}_k,a'_{k+1})$. Since $A \models F(\bar{a}_{k+1})$ we have $A \models F(\bar{a}_k,a'_{k+1})$. So A is H-bounded.

Theorem 4 Say that $\{E_{n,k}\}$ and H satisfy the hypotheses of Theorem 3. Let $h:N \to \Sigma^* \cup \{\infty\}$ be a function such that for every $n \in N^+$ and every $A \in \mathcal{C}$, there is some structure $A' \in \mathcal{C}$ such that $A'\ E_{n,0}\ A$ and $\|A'\| \preceq h(n)$.

THEN $\mathcal{C}$ is (h,H) bounded.

Proof Since every $A \in \mathcal{C}$ is H-bounded, it remains to show for every sentence F of q-depth $\leq n$, $F \in SAT(\mathcal{C}) \iff$ F is true in some structure in $\mathcal{C}$ of norm $\preceq h(n)$. $\Leftarrow$ is obviously true. So say that $F \in SAT(\mathcal{C})$, q-depth(F) $\leq n$. Then for some $A \in \mathcal{C}$, $A \models F$. By hypothesis, there is some $A' \in \mathcal{C}$ such that $A'\ E_{n,0}\ A$ and $\|A'\| \preceq h(n)$. Since $A'\ E_{n,0}\ A \Rightarrow A'\ \equiv_{n,0}\ A$, and $A \models F$, we have $A' \models F$. □

To show that condition 1 of Lemma 3 holds, we shall us usually appeal to the very simple Lemma 5.

Lemma 5 In order to show $(A,\bar{a}_k) \equiv_0 (B,\bar{b}_k)$, it is sufficient to show that $A \models F(\bar{a}_k) \iff B \models F(\bar{b}_k)$, for every atomic formula $F(\bar{x}_k)$.

Proof The proof that $A \models F(\bar{a}_k) \iff B \models F(\bar{b}_k)$ for every quantifier-free formula F, proceeds by induction on the number of connectives in F. The details are left to the reader. □

We now prove Lemma 6, since the proof is very similar to the proof of part I of Theorem 3; Lemma 6 is one half of Theorem 2.

Lemma 6 Let $n,k \in N$, and let $(A,\bar{a}_k)$ and $(B,\bar{b}_k)$ be such that

1) For each $a_{k+1} \in A$ there exists some $b_{k+1} \in B$ such that $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$

and

2) For each $b_{k+1} \in B$ there exists some $a_{k+1} \in A$ such that $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$.

Then $(A,\bar{a}_k) {}_n\!\equiv_{1} (B,\bar{b}_k)$.

Proof Say that 1) and 2) hold. As in the proof of Theorem 3, it suffices to show for each formula $G(\bar{x}_{k+1})$ of q-depth n that $A \models \exists x_{k+1}\, G(\bar{a}_k,x_{k+1}) \iff B \models \exists x_{k+1}\, G(\bar{b}_k,x_{k+1})$. Because of symmetry, we only show $\Rightarrow$. Assume $A \models \exists x_{k+1}\, G(\bar{a}_k,x_{k+1})$. Let $a_{k+1} \in A$ be such that $A \models G(\bar{a}_{k+1})$. Let $b_{k+1} \in B$ be such that $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$. Then $B \models G(\bar{b}_{k+1})$, so $B \models \exists x_{k+1}\, G(\bar{b}_k,x_{k+1})$

Although in general our decision procedures will proceed by using Theorem 3 or 4 to show H or (h,H) boundedness, it is important to note that it is sometimes the case that there is no nice notion of norm or H that one can think of, and in this case one can still often make use of a version of Theorem 3 or 4 which makes no notion of norm or boundedness. For instance, we obtain Theorem 3' by setting the norm and the function H to be identically 0.

Theorem 3' Say that the equivalence relations $\{E_{n,k}\}$ satisfy the following.

1) $(A,\bar{a}_k)\ E_{0,k}\ (B,\bar{b}_k) \Rightarrow (A,\bar{a}_k) \equiv_0 (B,\bar{b}_k)$

2) $(A,\bar{a}_k)\ E_{n+1,k}\ (B,\bar{b}_k) \Rightarrow$ for every $a_{k+1} \in A$ there is a $b_{k+1} \in B$ such that $(A,\bar{a}_{k+1})\ E_{n,k}\ (B,\bar{b}_{k+1})$.

Then

$$(A,\bar{a}_k)\ E_{n,k}\ (B,\bar{b}_k) \Rightarrow (A,\bar{a}_k) \equiv_n (B,\bar{b}_k).$$

In the next section, we present a more general discussion of Ehrenfeucht games, with relevance to Theorem 3'. We also discuss there the complexity of this type of decision procedure, and a crucial role is played there by the number of $\equiv_{n,k}$ equivalence classes.

Definition 7 Assume C is fixed. Then for each $n,k \in N$, let $M(n,k)$ be the number of $\equiv_{n,k}$ equivalence classes.

Lemma 8 Let $n,k \in N$. Then $M(n,k)$ is finite and for each $A \in C$, $\bar{a}_k \in A^k$, there is a formula $F(\bar{x}_k)$ of q-depth n such that for all $B \in C$, $\bar{b}_k \in B^k$: $B \models F(\bar{b}_k) \Leftrightarrow (B,\bar{b}_k) \equiv_n (A,\bar{a}_k)$. That is, F

defines the $\equiv_n$ equivalence class of $(A,\bar{a}_k)$.

Proof (by induction on n)

If n = 0, then we can clearly take $F(\bar{x}_k)$ to be a conjunction of atomic formulas and negations of atomic formulas. Say that there are ℓ formal relations, of arity $t_1,t_2,\ldots,t_\ell$. Since an argument place of an atomic formula can be occupied by either a formal variable or the constant symbol $\underline{e}$, the number of atomic formulas in which at most $\underline{e}, x_1,x_2,\ldots,x_k$ occur is $\sum_{i=1}^{\ell} (k+1)^{t_i}$. So

$$M(0,k) \leq 2^{\sum_{i=1}^{\ell} (k+1)^{t_i}}.$$

Now assume the lemma true for n (and all k). We shall prove it for n+1 (and k). Let $F_1(\bar{x}_{k+1}),F_2(\bar{x}_{k+1})\ldots,F_{M(n,k+1)}(\bar{x}_{k+1})$ be a sequence of formulas of q=depth n such that for each $(A,\bar{a}_{k+1})$ there exists an i, $1 \leq i \leq M(n,k+1)$, such that F_i defines the $\equiv_n$ equivalence class of $(A,\bar{a}_{k+1})$.

For each $(A,\bar{a}_k)$ define $W(A,\bar{a}_k) = \{i \mid 1 \leq i \leq M(n,k+1)$ and $A \models \exists x_{k+1} F_i(\bar{a}_k,x_{k+1})\}$. We shall show that for all $(A,\bar{a}_k)$, $(\mathcal{B},\bar{b}_k)$, $(A,\bar{a}_k) \equiv_{n+1} (\mathcal{B},\bar{b}_k) \iff W(A,\bar{a}_k) = W(\mathcal{B},\bar{b}_k)$. Thus the formula

$$F(\bar{x}_k) = \left(\bigwedge_{i \in W(A,\bar{a}_k)} \exists x_{k+1} F_i(\bar{x}_{k+1}) \right) \wedge \left(\bigwedge_{\substack{i \notin W(A,\bar{a}_k) \\ 1 \leq i \leq M(n,k+1)}} \sim \exists x_{k+1} F_i(\bar{x}_{k+1}) \right)$$

defines the $\equiv_{n+1}$ equivalence class of $(A,\bar{a}_k)$.

Clearly if $(A,\bar{a}_k) \equiv_{n+1} (B,\bar{b}_k)$, then $W(A,\bar{a}_k) = W(B,\bar{b}_k)$ since each formula $\exists x_{k+1} F_i(\bar{x}_{k+1})$ is of q-depth n+1. To prove the converse, we first prove the following claim.

Claim If $W(A,\bar{a}_k) = W(B,\bar{b}_k)$, then for each $a_{k+1} \in A$ there exists some $b_{k+1} \in B$ such that $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$ (and by symmetry, for each b_{k+1} there is some a_{k+1} such that $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$).

Proof of Claim Say that $W(A,\bar{a}_{k+1}) = W(B,\bar{b}_{k+1})$ and $a_{k+1} \in A$. Let i, $1 \le i \le M(n,k+1)$, be such that $F_i(\bar{x}_{k+1})$ defines the $\equiv_n$ equivalence class of $\bar{a}_{k+1}$. $A \models F_i(\bar{a}_{k+1})$, so $A \models \exists x_{k+1} F_i(\bar{a}_k, x_{k+1})$, so $i \in W(A,\bar{a}_k)$. So $i \in W(B,\bar{b}_k)$. This implies $B \models \exists x_{k+1} F_i(\bar{b}_k, x_{k+1})$, and therefore we can find $b_{k+1} \in B$ such that $B \models F_i(\bar{b}_{k+1})$. Since F_i defines the $\equiv_n$ equivalence class of $(A,\bar{a}_{k+1})$, we must have $(A,\bar{a}_{k+1}) \equiv_n (B,\bar{b}_{k+1})$, and the claim is proved.

By the claim and Lemma 6, we can conclude $W(A,\bar{a}_k) = W(B,\bar{b}_k) \iff (A,\bar{a}_k) \equiv_{n+1} (B,\bar{b}_k)$. Note that the $\equiv_{n+1}$ equivalence class of $(A,\bar{a}_k)$ is determined by $W(A,\bar{a}_k) \subseteq \{1,2,\ldots,M(n,k+1)\}$. So $M(n+1,k) \le 2^{M(n,k+1)}$. This and the bound on M(0,k) imply that

$$M(n,k) \le 2^{2^{\cdot^{\cdot^{\cdot^{2^{(n+k)^c}}}}}} \Big\} \text{height } n+1 \quad \text{for some constant } c. \quad \square$$

There are in fact single structures for which

$$M(n,k) \ge 2^{2^{\cdot^{\cdot^{\cdot^{2^{n+k}}}}}} \Big\} \text{height } \varepsilon n \quad \text{for some constant } \varepsilon > 0, \text{ so } M(n,k)$$

is not in general bounded above by an elementary recursive

function. For many structures and classes of structures, however, M(n,k) grows considerably more slowly.

Lemma 9 Say that $\mathcal{C}$ is H-bounded. Let $n,k \in N$ and say that $(A,\bar{a}_k) \underset{n+1}{\equiv} (\mathcal{B},\bar{b}_k)$, and $a_i \preceq m$ for $1 \leq i \leq k$. Then for each $a_{k+1} \in A$ there exists some $b_{k+1} \in B$ such that $(A,\bar{a}_{k+1}) \underset{n}{\equiv} (\mathcal{B},\bar{b}_{k+1})$ and $\|b_{k+1}\| \preceq H(n,k,m)$.

Proof Say that $(A,\bar{a}_k) \underset{n+1}{\equiv} (\mathcal{B},\bar{b}_k)$ and $a_{k+1} \in A$. By Lemma 8 there is a formula $F(\bar{x}_{k+1})$ of q-depth n which defines the $\underset{n}{\equiv}$ equivalence class of $(A,\bar{a}_{k+1})$. Since $A \models \exists x_{k+1} F(\bar{a}_k,x_{k+1})$ and $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$, $\mathcal{B} \models \exists x_{k+1} F(\bar{b}_k,x_{k+1})$. Since $\mathcal{B}$ is H-bounded, we can choose $b_{k+1} \in B$ such that $\mathcal{B} \models F(\bar{b}_{k+1})$ and $\|b_{k+1}\| \preceq H(n,k,m)$. But $\mathcal{B} \models F(\bar{b}_{k+1})$ implies $(\mathcal{B},\bar{b}_{k+1}) \underset{n}{\equiv} (A,\bar{a}_{k+1})$. □

We can now prove Theorem 2, our alternative characterization of the relations $\underset{n+1,k}{\equiv}$.

Proof of Theorem 2

<= is just the statement of Lemma 6.

=> follows from Lemma 9, if we take our norm function and H to be identically 0. □

The next lemma will only be used in Chapter 5. It says, essentially, that every $\underset{n,k}{\equiv}$ equivalence class with respect to an H-bounded structure has members with "small" norms.

Lemma 10 Say that $A \in \mathcal{C}$ is H-bounded; let $n,k \in N$ and let $m_0 \preceq m_1 \preceq \ldots \preceq m_k$ be a sequence of members of $\Sigma^* \cup \{\infty\}$ such that $H(n+k-i,i-1,m_{i-1}) \preceq m_i$ for $1 \leq i \leq k$. Then for every $\bar{a}_k \in A^k$

there is some $\bar{a}'_k \in A^k$ such that $(A,\bar{a}_k) \equiv_n (A,\bar{a}'_k)$ and $\|a'_i\| \preceq$ for $1 \leq i \leq k$.

Proof Let $\bar{m}_k$ be as stated, and let $\bar{a}_k \in A^k$. By Lemma 8 there is a formula $F(\bar{x}_k)$ of q-depth n which defines the $\equiv_n$ equivalence class of $(A,\bar{a}_k)$. Since $A \models F(\bar{a}_k)$, $A \models \exists x_1 \exists x_2 \ldots \exists x_k F(\bar{x}_k)$. By Theorem 1, $A \models (\exists x_1 \preceq m_1)\ldots(\exists x_k \preceq m_k)F(\bar{x}_k)$. So for some $\bar{a}'_k \in A^k$, $A \models F(\bar{a}'_k)$ (and hence $(A,\bar{a}_k) \equiv_n (A,\bar{a}'_k)$) and $\|a'_i\| \preceq m_i$ for $1 \leq i \leq k$. □

Section 2: Complexity of Ehrenfeucht Games

We now wish to discuss in some generality the Ehrenfeucht game technique for deciding the theory of, say, a single structure A. Sometimes we can define a norm for A and relations $E_{n,k}$, and then use Theorem 3 to conclude H-boundedness, and then use H-boundedness to decide TH(A). More generally, however, no nice norm and function H present themselves, and one can only define equivalence relations $E_{n,k}$ and prove that the hypotheses of Theorem 3' hold. In this case, instead of restricting quantifiers to range over elements of bounded norm, we can restrict quantifiers to range over (representations of) the $E_{n,k}$ classes, and in this way possibly decide the theory.

More precisely, to decide $Q_1x_1Q_2x_2\ldots Q_nx_nF(\bar{x}_n)$ where F is quantifier free, we restrict quantifier Q_i to range over (representatives) of the $E_{n-i,i}$ equivalence classes. For instance, $\exists x_1\forall x_2\exists x_3F(x_1,x_2,x_3)$ is equivalent to "for some $E_{2,1}$ class x_1, it is the case that for every $E_{1,2}$ class x_2 which is an extension of x_1, there exists some $E_{0,3}$ class which is an extension of x_2 and satisfies F. (We use extension in the sense that for every $\bar{a}_{k+1} \in A^{k+1}$, the $E_{n,k+1}$ class containing $\bar{a}_{k+1}$ is an extension of the $E_{n+1,k}$ class containing $\bar{a}_k$.) We do not wish to belabor this formalism too much, since it is not hard to see that every decision procedure for A can be translated into it.

We cannot, therefore, give an interesting, precise, and totally general definition of an Ehrenfeucht game procedure. However, every primitive-recursive decision procedure that we

know of can be transformed, in a way which seems natural, into the above formalism; in addition, this transformation can be accomplished in a straightforward way and doesn't even cause a significant increase in the complexity of the procedure. In particular, this is true of all quantifier-elimination procedures that we know of.

Although the word "natural" is vague, one can say some precise things if one discusses the <u>complexity</u> of decision procedures. In fact, the above formalism implies a lower bound on the complexity of deciding the formula $Q_1x_1 \ldots Q_nx_nF(\bar{x}_n)$ of $\sum_{i=0}^{n} M(n-i,i)$, since by Theorem 3, each $E_{n-i,i}$ is a refinement of $\equiv_{n-i,i}$. If M is not an elementary recursive function, then the above formalism cannot yield an elementary recursive procedure And it seems reasonable to say that any transformation which transforms a procedure which is elementary recursive into one that isn't, is not "natural".

We would like to conjecture that <u>every</u> decision procedure can be transformed naturally into an Ehrenfeucht game procedure. We cannot state this formally, but we can state the following precise conjecture.

<u>Conjecture</u> If TH(A) has an elementary recursive decision procedure, then M(n,k) is bounded above by an elementary recursive function.

Although this conjecture is open, its converse is definitely false.

Counter-example to the Converse of the Conjecture

For the purposes of this counter-example, let L be the language of the first order predicate calculus with the formal predicates $v_1 = v_2$ and $v_1 \sim v_2$ (v_1 is equivalent to v_2), and the constant symbol 0.

For every nonempty set S of positive integers, let $\tilde{S}$ be an equivalence relation on N such that for every positive integer i

1) If $i \in S$, then there is exactly one $\tilde{S}$ equivalent class of size i

and 2) If $i \notin S$, then there are no $\tilde{S}$ equivalence class of size i.

Define the structure $A_S = \langle N, =, \tilde{S}, 0 \rangle$.

For any $i \in N^+$, there is a sentence F_i which can be obtained in time polynomial in i, which says that there is an equivalence class of size exactly i. Therefore, if $TH(A_S)$ can be decided within time g(n), then we can decide if $n \in S$ in time $p(n) + g(p(n))$ for some fixed polynomial p. Since we can make S arbitrarily hard to decide or arbitrarily nonrecursive, we can make $TH(A_S)$ arbitrarily hard to decide or arbitrarily nonrecursive.

Now let S be fixed, and consider M(n,k) for A_S; we will show that M(n,k) is bounded above by a fixed elementary recursive function (independent of S), contradicting the converse of the conjecture.

For each $\bar{a}_k, \bar{b}_k \in N^k$, define $\bar{a}_k \; E_n \; \bar{b}_k$ iff for all i,j such that $1 \leq i,j \leq k$:

I) $a_i \tilde{<}_S 0 \iff b_i \tilde{<}_S 0$, and $a_i = 0 \iff b_i = 0$

II) $a_i \tilde{<}_S a_j \iff b_i \tilde{<}_S b_j$, and $a_i = a_j \iff b_i = b_j$

III) the sets $\{a \in N | a \tilde{<}_S a_i\}$ and $\{b \in N | b \tilde{<}_S b_i\}$ are either of the same cardinality, or both of size bigger than n+k.

One can now use Theorem 3 to prove that $\bar{a}_k \; E_n \; \bar{b}_k \Rightarrow \bar{a}_k \equiv_n \bar{b}_k$.

Since the number of $E_{n,k}$ equivalence classes is bounded above by $2^{2^{c(n+k)}}$ for some constant c, $M(n,k)$ for A_S is bounded above by the same elementary recursive function.

CHAPTER 3

INTEGER ADDITION - AN EXAMPLE OF AN EHRENFEUCHT GAME DECISION PROCEDURE

Section 1: An Upper Bound for Integer Addition

We now present some applications of Chapter 2. For the rest of this section let L_1 be the language of the first order predicate calculus with the formal predicates $v_1+v_2 = v_3$ and $v_1 \leq v_2$, and the constant symbol 0.

We let $\mathbb{Z}$ be the structure $\langle Z,+,\leq,0\rangle$ where Z is the set of integers and + and $\leq$ are the usual integer addition and order. For $a \in Z$, we define $\|a\|$, the norm of a, to be $|a|$, the absolute value of a considered as a string in $\{0,1\}^*$, and we say $|a| \lesssim |b|$ iff $|a| \leq |b|$. $TH(\mathbb{Z})$ was first shown decidable by Presburger [Pre29].

We will obtain a theoretically efficient procedure for $TH(\mathbb{Z})$ in the manner outlined in Chapter 2. Although we will be using an Ehrenfeucht game approach, many of the ideas we shall use come from a quantifier elimination decision procedure for $TH(\mathbb{Z})$ obtained by Cooper [Coo72] and analysed from a complexity viewpoint by Oppen [Opp73]. We choose this example because it illustrates the formalism of Chapter 2, as well as our thesis that all known quantifier elimination procedures can be converted to Ehrenfeucht game decision procedures without significant loss of time and sometimes with a saving of space. Some of our results about $TH(\mathbb{Z})$ appeared in preliminary form in [FR75].

Although our procedure for $TH(\mathbb{Z})$ has about the same time complexity as Cooper's, it only requires the logarithm of the space used by Cooper's procedure.

We will begin by defining equivalence relations $E_{n,k}$ on objects of the form $\langle Z,\bar{a}_k\rangle$, $\bar{a}_k \in Z^k$; since we are dealing with only one structure, we will abbreviate $\langle Z,\bar{a}_k\rangle$ by $\bar{a}_k$. Intuitively, $\bar{a}_k E_{n,k} \bar{b}_k$ (abbreviated $\bar{a}_k E_n \bar{b}_k$) will mean that $\bar{a}_k$ and $\bar{b}_k$ satisfy the same (subject to certain restrictions) linear inequalities, and have certain divisibility properties in common.

Definition 1.1 If $A \subseteq Z$, then by $\ell cm\ A$ we mean the least positive integer which every nonzero member of A divides. Define the sequence of sets $V_0, V_0', V_1, V_1', \ldots$ by:

$$V_0 = \{-2,-1,0,1,2\};$$

$$V_i' = \{\tfrac{\delta}{v}\cdot v' \mid \delta = \ell cm\ V_i; v,v' \in V_i; v \neq 0\} \text{ for } i \geq 0;$$

$$V_{i+1} = V_i \cup \{a+b \mid a,b \in V_i'\} \text{ for } i \geq 0.$$

Definition 1.2 Let $n,k \in N$. Define the equivalence relation E_n on Z^k as follows: Let $\bar{a}_k, \bar{b}_k \in Z^k$, let $\delta = \ell cm\ V_n$; then $\bar{a}_k E_n \bar{b}_k$ iff for every $v_1, v_2, \ldots, v_k \in V_n$ and every v, $|v| \leq \delta^2$

1) $v + \sum_{i=1}^{k} v_i a_i \leq 0 \iff v + \sum_{i=1}^{k} v_i b_i \leq 0$

and 2) $a_i \approx b_i \bmod \delta^2$ for $1 \leq i \leq k$.

Lemma 1.3 For all $k \in N^+$, $\bar{a}_k E_0 \bar{b}_k \Rightarrow \bar{a}_k \equiv_{\bar{0}} \bar{b}_k$.

Proof Say that $\bar{a}_k E_0 \bar{b}_k$. To show $\bar{a}_k \equiv_{\bar{0}} \bar{b}_k$, it is sufficient to show that $\bar{a}_k$ and $\bar{b}_k$ satisfy the same atomic formulas, that is, that

$$a_i + a_j = a_\ell \iff b_i + b_j = b_\ell \text{ and}$$
$$a_i \le a_j \iff b_i \le b_j \text{ for all } 1 \le i,j,\ell \le k$$

This follows from the fact that

$$\sum_{i=1}^{k} v_i a_i \le 0 \iff \sum_{i=1}^{k} v_i b_i \le 0 \text{ for } |v_i| \le 2,\ 1 \le i \le k,$$

which in turn follows from $\bar{a}_k E_0 \bar{b}_k$. □

We now wish to show that if $\bar{a}_k E_{n+1} \bar{b}_k$, then for each a_{k+1} there is a "small" b_{k+1} such that $\bar{a}_{k+1} E_n \bar{b}_{k+1}$, thereby establishing the hypotheses of Theorem 3 of Chapter 2. We will do this in two steps, Lemma 1.5 and Lemma 1.6. We first prove

Lemma 1.4 For some constant c, $|V_n| \le 2^{2^{cn}}$ and $V_n = \{-a \mid a \in V_n\}$ and Max $V_n \le 2^{2^{2^{cn}}}$ for all $n \in N$ (Max V_n is the largest element of V_n).

Proof $|V_0| = 5$. In general, $|V_i'| \le |V_i|^2$ and $|V_{i+1}| \le |V_i| + |V_i'|^2 \le |V_i|^5$. So $|V_n| \le 5^{5^n}$.

It is trivial to show that $V_n = \{-a \mid a \in V_n\}$. Max $V_0 = 2$. In general, $\ell cm\ V_i \le (\max V_i)^{|V_i|}$. So

Max $V_{i+1} \le$ Max(Max V_i, 2·Max V_i') $\le 2 \cdot \ell cm\ V_i \cdot$ Max V_i
$\le 2 \cdot (\text{Max } V_i)^{5^{5^i}} \cdot \text{Max } V_i \le (\text{Max } V_i)^{6^{6^i}}$. So Max $V_n \le 2^{(6^{6^n})^n}$.
So for some constant c, $V_n \le 2^{2^{cn}}$ and Max $V_n \le 2^{2^{2^{cn}}}$.

Lemma 1.5 There is a constant c such that the following is true.

Let $n,k \in N$ and say that $\bar{a}_k E_{n+1} \bar{b}_k$ and $|b_1|,|b_2|,\ldots,|b_k| \leq m \in N$. Let $a_{k+1} \in Z$. Then there is some $b_{k+1} \in Z$ such that

$$|b_{k+1}| \leq (m+1)2^{2^{2^{c(n+k)}}} \quad \text{and}$$

1) for all $v_1,v_2,\ldots,v_k \in V_n'$ and $v \in Z$ such that $|v| \leq (\ell cm\ V_n)^3$, $v+\sum_{i=1}^{k} v_i a_i \leq a_{k+1} \iff v+\sum_{i=1}^{k} v_i b_i \leq b_{k+1}$

2) for all $v_1,v_2,\ldots,v_k \in V_n'$ and $v \in Z$ such that $|v| \leq (\ell cm\ V_n)^3$, $v+\sum_{i=1}^{k} v_i a_i \geq a_{k+1} \iff v+\sum_{i=1}^{k} v_i b_i \geq b_{k+1}$

and 3) $a_{k+1} \equiv b_{k+1} \bmod (\ell cm\ V_n)^3$.

Proof Let $k,n,m,\bar{a}_{k+1},\bar{b}_k$ be as given. Let $\delta = \ell cm\ V_n$. Consider the set T of formal terms $v+\sum_{i=1}^{k} v_i x_i$ where $v_1,\ldots,v_k \in V_n'$ and $|v| \leq \delta^3$. For any such term t, let $t(\bar{a}_k) \in Z$ be defined as $v+\sum_{i=1}^{k} v_i a_i$, and let $t(\bar{b}_k)$ be defined similarly.

Lemma 1.5.1 Let t and t' be the terms $v+\sum_{i=1}^{k} v_i x_i$ and $v'+\sum_{i=1}^{k} v_i' x_i$ in T; let $0 \leq J \leq \delta^3$. Then $t'(\bar{a}_k)-t(\bar{a}_k) \leq J \iff t'(\bar{b}_k)-t(\bar{b}_k) \leq J$.

Proof of Lemma 1.5.1 In effect, we wish to show that $(v'-v-J)+\sum_{i=1}^{k}(v_i'-v_i)a_i \leq 0 \iff (v'-v-J)+\sum_{i=1}^{k}(v_i'-v_i)b_i \leq 0$. $|v'-v-J| \leq 3\delta^3$, and it is not hard to show that $3\delta^3 \leq (\ell cm\ V_{n+1})^2$. By the definition of V_{n+1}, $v_i'-v_i \in V_{n+1}$ (since $-v_i \in V_n'$ if $v_i \in V_n'$) So from $\bar{a}_k E_{n+1} \bar{b}_k$, we can conclude Lemma 1.5.1.

Now number the members of T $t_1, t_2, \ldots, t_\ell$ such that $t_1(\bar{a}_k) \leq t_2(\bar{a}_k) \leq \ldots \leq t_\ell(\bar{a}_k)$. Lemma 1.5.1 implies that $t_1(\bar{b}_k) \leq t_2(\bar{b}_k) \leq \ldots \leq t_\ell(\bar{b}_k)$. Furthermore, for $1 \leq i < \ell$, either $t_{i+1}(\bar{a}_k) - t_i(\bar{a}_k) = t_{i+1}(\bar{b}_k) - t_i(\bar{b}_k)$, or $t_{i+1}(\bar{a}_k) - t_i(\bar{a}_k) > \delta^3$ and $t_{i+1}(\bar{b}_k) - t_i(\bar{b}_k) > \delta^3$.

In addition, since $\bar{a}_k E_{n+1} \bar{b}_k$ and δ^3 divides $(\ell cm\ V_{n+1})^2$, $t_i(\bar{a}_k) \approx t_i(\bar{b}_k) \bmod \delta^3$.

Say that $t_i(\bar{a}_k) \leq a_{k+1} \leq t_{i+1}(\bar{a}_k)$; the cases where $a_{k+1} < t_1(\bar{a}_k)$ or $a_{k+1} > t_\ell(\bar{a}_k)$ are similar and in fact easier. If $a_{k+1} = t_i(\bar{a}_k)$ or $a_{k+1} = t_{i+1}(\bar{a}_k)$, then let $b_{k+1} = t_i(\bar{b}_k)$ or $t_{i+1}(\bar{b}_k)$ respectively, and 1) and 2) above are satisfied. So assume $t_i(\bar{a}_k) < a_{k+1} < t_{i+1}(\bar{a}_k)$. Let b_{k+1} be such that $a_{k+1} \approx b_{k+1} \bmod \delta^3$, and $t_i(\bar{b}_k) < b_{k+1} \leq t_i(\bar{b}_k) + \delta^3$. For 1) and 2) to be true, we must have $b_{k+1} < t_{i+1}(\bar{b}_k)$. The only way this can fail to be true is if $t_{i+1}(\bar{b}_k) - t_i(\bar{b}_k) \leq \delta^3$, in which case $t_{i+1}(\bar{a}_k) - t_i(\bar{a}_k) = t_{i+1}(\bar{b}_k) - t_i(\bar{b}_k) \leq \delta^3$, contradicting the facts that $t_i(\bar{a}_k) \approx t_i(\bar{b}_k) \bmod \delta^3$ and $a_{k+1} \approx b_{k+1} \bmod \delta^3$ and $a_{k+1} < t_{i+1}(\bar{a}_k)$.

It remains to get an upper bound on the absolute value of b_{k+1}. We have b_{k+1} is within δ^3 of $t_j(\bar{b}_k) = v + \sum_{i=1}^{k} v_i b_i$ for some j. So $|b_{k+1}| \leq \delta^3 + (km)\text{Max } V_i' \leq (\text{Max } V_{n+1})^{3|V_{n+1}|} + (km)\text{Max } V_n$ $\leq$ (by Lemma 1.4) $(m+1)2^{2^{2^{c(n+k)}}}$ for some constant c. □

Lemma 1.6 For some constant c, $TH(\mathbb{Z})$ is H bounded where

$$H(n,k,m) = (m+1)2^{2^{2^{c(n+k)}}}.$$

Proof Let $n,k \in N$, $\bar{a}_k E_{n+1} \bar{b}_k$; say that $|b_1|,|b_2|,\ldots,|b_k| \leq m \in N$ Let $a_{k+1} \in Z$. We will show that there is some $b_{k+1} \in Z$ such that $\bar{a}_{k+1} E_n \bar{b}_{k+1}$ and $|b_{k+1}| \leq (m+1)2^{2^{2^{c(n+k)}}}$ for some constant c. This fact, together with Lemma 1.3 and Theorem 3 of Chapter 2, implies that $TH(\mathbb{Z})$ is $(m+1)2^{2^{2^{c(n+k)}}}$ bounded.

We will find an appropriate b_{k+1} by using Lemma 1.5. Let $\delta = \ell cm\ V_n$, and let $b'_{k+1} \in Z$ be such that

$|b'_{k+1}| \leq (m+1)2^{2^{2^{c(n+k)}}}$ and

1) for all $v_1,v_2,\ldots,v_k \in V'_n$ and $v \in Z$ such that $|v| \leq \delta^3$,
$$v + \sum_{i=1}^{k} v_i a_i \leq \delta a_{k+1} \iff v + \sum_{i=1}^{k} v_i b_i \leq b'_{k+1},$$

2) for all $v_1,v_2,\ldots,v_k \in V'_n$ and $v \in Z$ such that $|v| \leq \delta^3$,
$$v + \sum_{i=1}^{k} v_i a_i \geq \delta a_{k+1} \iff v + \sum_{i=1}^{k} v_i b_i \geq b'_{k+1},$$

and 3) $\delta a_{k+1} \approx b'_{k+1} \bmod \delta^3$.

Note that 3) implies that δ divides b'_{k+1}. Let $b_{k+1} = (b'_{k+1})/\delta$. 3) now implies that $a_{k+1} \approx b_{k+1} \bmod \delta^2$. Since δ^2 divides $(\ell cm\ V_{n+1})^2$, $\bar{a}_k E_{n+1} \bar{b}_k$ implies that $a_i \approx b_i \bmod \delta^2$ for $1 \leq i \leq k$. It remains to show the following: Say that $u_1,u_2,\ldots,u_{k+1} \in V_n$ and $|u| \leq \delta^2$; then we want $u + \sum_{i=1}^{k+1} u_i a_i \leq 0 \iff u + \sum_{i=1}^{k+1} u_i b_i \leq 0$. Since $\bar{a}_k E_{n+1} \bar{b}_k$, this holds

immediately if $u_{k+1} = 0$, so assume $u_{k+1} \neq 0$. Letting $v = u\delta/|u_{k+1}|$ and $v_i = u_i\delta/|u_{k+1}|$ for $1 \leq i \leq k$, we wish to show that $v + \sum_{i=1}^{k} v_i a_i + j\delta a_{k+1} \leq 0 \iff v + \sum_{i=1}^{k} v_i a_i + j\delta b_{k+1} \leq 0$ where $j \in \{1,-1\}$. Note that $|v| \leq \delta^3$ and $v_i \in V_n'$ for $1 \leq i \leq k$. So this last "<=>" follows from either 1) or 2) depending on the sign of j. □

Theorem 1.7 For some constant c, $TH(\langle Z,+,\leq,0\rangle)$ can be decided within space $2^{2^{cn}}$.

Proof Let c be as in the previous lemma, so that $TH(\mathbb{Z})$ is H-bounded where

$$H(n,k,m) = (m+1)2^{2^{2^{c(n+k)}}}.$$

Now let F be a sentence of length n. In time polynomial in n we can convert F to an equivalent sentence $Q_1x_1Q_2x_2\ldots Q_nx_nG(\bar{x}_n)$. Define $m_i = 2^{2^{2^{cn+i}}}$ for $0 \leq i \leq n$. Note that $m_i \geq H(n-i,i-1,m_{i-1})$ for $1 \leq i \leq n$. So by Theorem 1 of Chapter 2, F is equivalent to

$$(Q_1x_1 \preceq m_1)(Q_2x_2 \preceq m_2)\ldots(Q_nx_n) \preceq m_n)G(\bar{x}_n).$$

F can be decided in Z by setting aside for quantifier Q_i, $1 \leq i \leq n$, $2^{2^{cn+i}}+2$ tape squares; every integer $\leq 2^{2^{2^{cn+i}}}$ in absolute values can be written in this space in binary. Then decide F by cycling through each quantifier space appropriately, all the time testing the truth of G on different n-tuples of

integers. We let the reader convince himself that a Turing machine implementing this outlined procedure need use only $2^{2^{cn}}$ tape squares for some constant c. □

Theorem 1.8 For some constant c', any nondeterministic Turing machine which recognizes TH(Z,+,≤,0) requires time $2^{2^{c'n}}$ on some sentence of length n, for infinitely many $n \in \mathbb{N}$.

See Fischer and Rabin [FiR74] for a proof of this theorem. Their proof uses the method (for lower bounds) described in Chapter 1, and hence, for the reasons described in Chapter 1, the upper bound of Theorem 1.7 matches the lower bound of Theorem 1.8 reasonably well.

Let $\mathbb{R}$ be the structure $\langle R,+,\leq,0\rangle$ where R is the set of real numbers. As above, the upper bound for TH($\mathbb{R}$) in Theorem 1.9 is close to the lower bound in Theorem 1.10.

Theorem 1.9 For some constant c, TH($\mathbb{R}$) can be decided in space 2^{cn}.

Theorem 1.10 For some constant c', any nondeterministic Turing machine which recognizes TH($\mathbb{R}$) requires time $2^{c'n}$ on some sentence of length n, for infinitely many n.

Theorem 1.9 is proved in Ferrante and Rackoff [FR75]. Although part of their proof uses quantifier elimination, it could be rewritten to follow the above Ehrenfeucht game format without loss of efficiency. Theorem 1.10 is proved in [FiR74].

CHAPTER 4

SOME ADDITIONAL UPPER BOUNDS

Section 1: Introduction

In this section, we state the definitions and results needed in the next four sections. In particular, we state the definition of "equal up to size n", $\equiv_n$, to be used in sections 2 through 5. A basic technique of these sections is showing formulas with a fixed number of quantifiers can only distinguish up to a fixed number of structures,and no more, where $\equiv_n$ is our measure of "large".

We find the introduction of the symbol ∞ useful in defining $\equiv_n$.

Definition 1.1 For any integer ℓ, we define $\ell+\infty = \ell-\infty = \infty+\ell = \infty-\ell = \infty$, and $\ell < \infty$. We also define $\infty+\infty = \infty$, and $|\infty| = \infty$.

Definition 1.2 Let n be an integer, i an integer or the symbol ∞. We define

$$[i]_n = \begin{cases} i & \text{if } |i| \leq n \\ n+1 & \text{otherwise.} \end{cases}$$

Definition 1.3 Let n be any integer, i,j integers or the symbol ∞. We define

$$i \equiv_n j \quad \text{iff} \quad [i]_n = [j]_n.$$

We remark that if n is negative, then for i an integer or ∞, $[i]_n = n+1$, and so for all i,j integers or ∞, we have $i \equiv_n j$

We also remark for any integer n, $\equiv_n$ is an equivalence relation, and if m is any integer $\leq n$, $i \equiv_n j$ implies $i \equiv_m j$.

We now prove a series of technical lemmas concerning $\underset{n}{=}$ which will be used throughout sections 2 through 5.

Lemma 1.4 Let i,j be integers or the symbol ∞, k,n any integers. Suppose $i \underset{n}{=} j$. Then $i+k \underset{n-|k|}{=} j+k$.

Proof The proof is by cases.

Case 1. $|i| \leq n$.

We have in this case, by definition, i = j, and so i+k = j+k. Then by definition $i+k \underset{n-|k|}{=} j+k$.

Case 2. $|i| > n$.

We must have in this case, by definition, $|i| \geq n+1$ and $|j| \geq n+1$. Then $|i+k| \geq |i| - |k| \geq n+1 - |k|$, and likewise $|j+k| \geq n+1 - |k|$, and so by definition $i+k \underset{n-|k|}{=} j+k$. □

Lemma 1.5 Let i,j,k and ℓ be elements of N or the symbol ∞, and let $n \in N$. If $i \underset{n}{=} j$ and $k \underset{n}{=} \ell$, then $i+k \underset{n}{=} j+\ell$.

The proof is an easy verification by cases, which we omit.

Lemma 1.6 Let $i,j,\ell,n \in N$. Suppose $j \leq i$, and $i \underset{2^{n+1}-1}{=} \ell$. Then there is $j' \in N_\ell$ such that

1. $j \underset{2^n-1}{=} j'$, and
2. $i-j \underset{2^n-1}{=} \ell-j'$.

Proof The proof is by cases.

Case 1. $j \leq 2^n-1$.

We let $j' = j$. Then certainly requirement 1 of the lemma is satisfied, and requirement 2 follows immediately from Lemma 1.4 (with $k=-j$).

Case 2. $i-j \leq 2^n-1$.

Choose j' so that $\ell-j' = i-j$. Then clearly requirement 2 of the lemma is satisfied, and requirement 1 follows immediately from Lemma 1.4 (with $k=-(i-j)$).

Case 3. Both $j \geq 2^n$ and $i-j \geq 2^n$.

We must have in this case $i \geq 2^{n+1}$. Since $i \underset{2^{n+1}-1}{=} \ell$, by definition we have $\ell \geq 2^{n+1}$. We let $j' = 2^n$. Then $\ell-j' \geq 2^n$, and so clearly requirements 1 and 2 of the lemma are satisfied. □

We also need a coarser definition of cardinality which we define as follows:

Definition 1.7 Let A be any set.

$$|A|_\infty = \begin{cases} |A| & \text{if } |A| \text{ is finite} \\ \infty & \text{otherwise.} \end{cases}$$

In the next four sections, we present four main results and their various corollaries. Our mode of presentation will be to present the proof of the first result in great detail, the second and third in some detail, and to present an outlined proof, with important definitions, of the last result, leaving it to the reader to fill in the majority of proof details.

We feel that although the details of proof differ, the technique is uniform, and the interested reader should be able to supply the details omitted.

Our method of proceeding through the proofs can be outlined as follows.

We wish to use Theorems 3 and 4 of Chapter 2 to establish our results. To show Theorem 3 is applicable to a class of structures $\mathcal{C}$ we define an equivalence relation $E_{n,k}$ for all $n,k \in N$ and show

(1) $E_{0,k}$ is a refinement of $\equiv_{0,k}$. As an intermediate step, we then show

(2) For all $A,B \in \mathcal{C}$, $n,k \in N$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$, if $\bar{a}_k E_{n+1} \bar{b}_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $\bar{a}_{k+1} E_n \bar{b}_{k+1}$.

Careful examination of the proof will allow us to define norms over Σ and $H: N \times N \times (\Sigma^* \cup \{\infty\}) \to \Sigma^* \cup \{\infty\}$ in each case so that

(3) For all $A,B \in \mathcal{C}$, $n,k \in N$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$ and $m \in \Sigma^* \cup \{\infty\}$, if $\bar{a}_k E_{n+1} \bar{b}_k$, and $b_i \preceq m$ for all i, $1 \leq i \leq k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$, such that $b_{k+1} \preceq H(n,k,m)$ and $\bar{a}_{k+1} E_n \bar{b}_{k+1}$.

We will thereby have established the premises of Theorem 3.

We then define $h: N \to \Sigma^* \cup \{\infty\}$ and show (in most cases quite easily)

(4) For every $n \in I$ and $A \in C$, there is some structure $A' \in C$ such that $A' E_{n,o} A$ and $A' \leq h(n)$.

The premises of Theorem 4 will then be satisfied, and so we ca conclude C is (h,H) bounded.

Section 2: Upper Bounds for the Theory of a 1-1 Unary Function

We first apply our technique to the theory of a 1-1 unary function because of the easily analyzed structure of its models.

Let 11F denote the class of structures $\{\langle A,f_A\rangle \mid f_A:A \to A$ is a 1-1 function$\}$. To apply the results of Chapter 2 we must consider a language with only relation symbols, so let L be a first order language with a single binary relation symbol in it. We use the notation $f(x) = y$ to stand for the assertion in the language L that the binary relation holds for the pair of variables x,y. Using this notation, then, for any $\langle A,f_A\rangle$ in 11F, and any $a,b \in A$, we interpret $f(a) = b$ as $f_A(a) = b$.

We first wish to present an analysis of 11F. We need a few preliminary definitions. Henceforth we drop the subscript A on f_A.

Definition 2.1 Let $\langle A,f\rangle \in$ 11F, $a,b \in A$. We define $f^0(a) = b$ iff $a = b$. If $\ell \in I$, we define $f^{\ell}(a) = b$ iff there are $\bar{a}_{\ell+1} \in A^{\ell+1}$ such that $a_1 = a$, $a_{\ell+1} = b$, and $f(a_i) = a_{i+1}$ for all $i \in I_{\ell}$. Note that $f^1(a) = b \iff f(a) = b$. If ℓ is any integer, $\ell < 0$, $f^{\ell}(a) = b \iff f^{-\ell}(b) = a$.

For any function $f:A \to A$, range $(f) = \{b \in A \mid$ there is $a \in A$ such that $f(a) = b\}$.

Definition 2.2 Let $\langle A,f\rangle \in$ 11F. For $a \in A$, we define the component of A, C_a, to be $\{b \in A \mid$ for some integer ℓ, $f^{\ell}(a) = b\}$

For any $a \in A$, and $n \in I$, C_a is a loop of size n if $|C_a| = n$; C_a is a one-sided chain if there is $b \in C_a$ such that $b \notin$ range (f); C_a is a two-sided chain if for every n it is not a loop of size n, and it is not a one-sided chain. Figure 2.3 provides pictorial representations of components as graphs whose nodes correspond to elements of A, with a directed edge from a to b iff f(a) = b.

loop of size 2 one-sided chain two-sided chain

Figure 2.3 Components of a model

If C is a one-sided chain, the element $b \in C$ such that $b \notin$ range (f) is obviously unique, and will be called the origin of C. We let o_a denote the origin of C_a when C_a is a one-sided chain. For any $j \in I$, we let LOOPS $(A,j) = \{a \in A | C_a$ is a loop of size $j\}$; we let 1-CHAINS $(A) = \{a \in A | C_a$ is a one-sided chain$\}$, 2-CHAINS $(A) = \{a \in A | C_a$ is a two-sided chain$\}$, and ORIGINS $(A) = \{o_a | a \in$ 1-CHAINS $(A)\}$.

It should be clear for any $A \in$ 11F, A is a disjoint union of the components of its elements. In fact, it is easy to see any $A \in$ 11F is determined up to isomorphism by the cardinalities of each type of component in it. That is, given any $\mathcal{B} \in$ 11F such that for all $j \in I$, $\mathcal{B}$ has the same cardinality of

loops of size j as A, the same cardinality of 1-sided chains as A, and the same cardinality of two-sided chains as A, then $\mathcal{B}$ is isomorphic to A.

The elementary equivalence classes of models are also easily described. By the Skolem-Lowenheim theorem [Smu 68], as well as by our analysis of $\equiv_{n,k}$ which appears below, it is clear that each $A \in \mathrm{IIF}$ is elementarily equivalent to a countable model. For countable $A,\mathcal{B} \in \mathrm{IIF}$, it will turn out that A is not elementarily equivalent to $\mathcal{B}$ (i.e., there is a sentence $F \in L$ such that A satisfies F and $\mathcal{B}$ satisfies $\sim F$) iff either

1. For some $j \in I$, $|\mathrm{LOOPS}\ (A,j)| \neq |\mathrm{LOOPS}\ (\mathcal{B},j)|$, or
2. $|\mathrm{ORIGINS}\ (A)| \neq |\mathrm{ORIGINS}\ (\mathcal{B})|$, or
3. Conditions 1 and 2 do not hold, A does not have loops of arbitrarily large size, nor a one-sided chain, and A and $\mathcal{B}$ differ with respect to the property of having a two-sided chain.

Our analysis of the relation $\equiv_{n,k}$, or rather an appropriate refinement of it, consists of characterizing the information that quantifier-depth n formulas can provide about k elements. We shall show that quantifier-depth n formulas with k free variables $\bar{z}_k$ can at most determine, up to certain bounds depending on n and k, the number of elements in distinct loops of different sizes, the number of origins, and the total number of elements in "large" loops and two-sided chains. Also such formulas can define the distances (i.e., the number of applications of f) up

to certain bounds between z_i and z_j and o_{z_i} and z_i. We make this precise in our definition of $E_{n,k}$ below.

Definition 2.4 Let $\langle A,f\rangle \in \mathbb{IF}$, $a,b \in A$. We define the distance from a to b, $d(a,b)$ as follows:

1. If $b \notin C_a$, then $d(a,b) = \infty$.
2. If $b \in C_a$, and C_a is not a loop, then $d(a,b)$ is the necessarily unique integer such that $f^{d(a,b)}(a) = b$.
3. If $b \in C_a$ and C_a is a loop, consider $\{|n| \mid n \text{ is an integer and } f^n(a) = b\}$. It must have a unique minimal element $m \in N$. If both $f^m(a) = b$ and $f^{-m}(a) = b$, (viz. if C_a is a loop of size $2\cdot m$) we let $d(a,b) = m$. Otherwise, there is a unique integer ℓ such that $f^{\ell}(a) = b$ and $|\ell| = m$. We let $d(a,b) = \ell$.

If C_a is not a one-sided chain, we define $d(o_a,b) = \infty$.

We collect here some facts concerning d, whose easy verification is left to the reader:

Lemma 2.5 Let $A,B \in \mathbb{IF}$, $a,a_1,a_2 \in A$, $b \in B$, $n \in I$.

1. Either $d(a_1,a_2) = -d(a_2,a_1)$, or else both $d(a_1,a_2) = d(a_2,a_1)$ and $|C_{a_1}|_\infty = 2\cdot d(a_1,a_2)$.
2. If $d(a_1,a_2) \neq \infty$, then $d(o_{a_2},a_2) = d(o_{a_1},a_1) + d(a_1,a_2)$.
3. $d(o_a,a) \overset{=}{n} d(o_b,b) \iff$ for every integer $\ell \geq -n$, [there is $a' \in A$ such that $f^{\ell}(a) = a' \iff$ there is $b' \in B$ such that $f^{\ell}(b) = b'$].

Definition 2.6 Let $A,\mathcal{B} \in \mathrm{IIF}$, $a_1,a_2 \in A$, $b_1,b_2 \in B$, $n \in I$. The pair (a_1,a_2) is distance-n equivalent to the pair (b_1,b_2) iff [for every integer ℓ, $|\ell| \leq n$, $f^{\ell}(a_1) = a_2 \iff f^{\ell}(b_1) = b_2$].

We collect here some facts about distance-n equivalence, whose easy verification is left to the reader.

Lemma 2.7 Let $A,\mathcal{B} \in \mathrm{IIF}$, $a,a_1,a_2 \in A$, $b,b_1,b_2 \in B$, $n \in I$.

1. (a_1,a_2) distance-n equivalent to $(b_1,b_2) \iff (a_2,a_1)$ distance-n equivalent to (b_2,b_1).
2. $|d(a_1,a_2)| > n$ and $|d(b_1,b_2)| > n \Rightarrow (a_1,a_2)$ distance-n equivalent to (b_1,b_2).
3. $|C_a|_\infty \underset{n}{=} |C_b|_\infty \iff (a,a)$ distance-n equivalent to (b,b).

Definition 2.8 Let $n,k \in N$, $A,\mathcal{B} \in \mathrm{IIF}$. A is n;k equivalent to $\mathcal{B}$ if

1. for any $j \in I_{2^n}$,
$$|\text{LOOPS } (A,j)|_\infty \underset{(n+k)(2^{n+k}+1)}{=} |\text{LOOPS } (\mathcal{B},j)|_\infty, \text{ and}$$
2. $$|\text{ORIGINS } (A)|_\infty \underset{n+k}{=} |\text{ORIGINS } (\mathcal{B})|_\infty, \text{ and}$$
3. $$|\text{2-CHAINS } (A)|_\infty + \sum_{j>2^n} |\text{LOOPS } (A,j)|_\infty \underset{(n+k)\cdot(2^{n+k}+1)}{=} |\text{2-CHAINS } (\mathcal{B})|_\infty + \sum_{j>2^n} |\text{LOOPS } (\mathcal{B},j)|_\infty.$$

Fact 2.9 For all $n,k \in N$, $A,\mathcal{B} \in \mathrm{IIF}$, A n+1; k equivalent to $\mathcal{B}$ implies A n; k+1 equivalent to $\mathcal{B}$.

We note that for any $A,\mathcal{B} \in \mathrm{IIF}$, if $|\text{ORIGINS } (A)|_\infty \underset{0}{=} |\text{ORIGINS } (\mathcal{B})|_\infty$, then trivially $|\text{1-CHAINS } (A)|_\infty = |\text{1-CHAINS } (\mathcal{B})|_\infty$.

Definition 2.10 Let $n \in N$, $A,B \in \mathrm{IIF}$. $A\ E_{n,0}\ B$ iff A is n; 0 equivalent to B.

Definition 2.11 Let $n \in N$, $k \in I$. Let $A,B \in \mathrm{IIF}$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$. We define $\bar{a}_k\ E_n\ \bar{b}_k$ iff

1. A is n; k equivalent to B, and
2. for all $i,j \in I_k$, (a_i,a_j) is distance-2^n equivalent to (b_i,b_j), and
3. for all $i \in I_k$, $d(o_{a_i},a_i) \doteq_{2^n} d(o_{b_i},b_i)$.

We remark here that we have actually chosen a somewhat finer refinement of $\equiv_{n,k}$ than is necessary. For instance, in conditions 1 and 3 of the definition of n; k equivalence, $\doteq_{(n+k)(2^{n+k}+1)}$ could be replaced by $\doteq_{k(2^n+1)}$, at some sacrifice of simplicity in the proofs to follow. However, since our definition of $E_{n,k}$ yields an upper bound of as good an order as can be expected, it seems reasonable to use our definition as given.

Clearly, $E_{n,k}$ is an equivalence relation of finite index.

We next wish to define our concept of norm in IIF. It turns out that to decide if a sentence F is satisfiable in IIF it will suffice to look at the satisfiability of F in a finite number of structures of the following restricted nature:

Notation 2.12 For $m \in N$, $N(m)$ denotes

$$\left(N_{m(1+2^m)+1}\right)^{2^m} \times N_{m+1} \times \left(N_{m(1+2^m)+1}{}^{-I}{}_{2^m}\right).$$

For $\bar{k}_{2+2^m} \in N(m)$ we now define $A(\bar{k}_{2+2^m})$ to consist of, for each $j \in I_{2^m}$, k_j loops of size j, k_{1+2^m} one-sided chains, and a "large" loop of size k_{2+2^m} if $k_{2+2^m} > 0$, no "large" loop if $k_{2+2^m} = 0$.

Definition 2.13 Let $m \in N$, $\bar{k}_{2+2^m} \in N(m)$. We define $A(\bar{k}_{2+2^m}) \in \mathrm{IIF}$ as follows:

$$A = \{(i,j,k) \mid i,j \in I,\ k \in N,\ j \leq 2^m,\ i \leq k_j,\ k < j\} \cup$$
$$\{(p,q) \mid p \in I,\ q \in N,\ p \leq k_{1+2^m}\} \cup \{(1,k_{2+2^m},k) \mid k \in N,\ k < k_{2+2^m}$$

f is defined so that $(i,j,k) \in A$ means (i,j,k) is the $k+1^{st}$ element on the i^{th} loop of size j, and $(p,q) \in A$ means (p,q) is the $q+1^{st}$ element on the p^{th} one-sided chain. That is $f((i,j,k)) = (i,j,k+1 \pmod j)$, and $f((p,q)) = (p,q+1)$. We refer to any $A(\bar{k}_{2+2^m})$ as an Ehrenfeucht structure for IIF.

We consequently define our notion of norm over the alphabet $\{0,1,\#\}$ as follows:

Definition 2.14 For any structure $A(\bar{k}_{2+2^m})$ as defined above, we define

$$\|A(\bar{k}_{2+2^m})\| = \mathrm{brep}(k_1)\#\mathrm{brep}(k_2)\#\ldots\#\mathrm{brep}(k_{2+2^m}).$$

For any other structure A in IIF, we define $\|A\| = \infty$.

We define a partial order on norms as follows:

$$\mathrm{brep}(k_1)\#\mathrm{brep}(k_2)\#\ldots\#\mathrm{brep}(k_{2+2^m}) \preceq$$
$$\mathrm{brep}(\ell_1)\#\mathrm{brep}(\ell_2)\#\ldots\#\mathrm{brep}(\ell_{2+2^m})$$

iff $k_i \leq \ell_i$ for all $i \in I_{2+2^m}$ and k_{2+2^m} is either equal to 0 or greater than 2^m. We also define $w \preceq \infty$ for any norm w. $\preceq$ is undefined for all other pairs of words in $\{0,1,\#\}^*$.

Our notion of norm for elements in structures is as follows:

Definition 2.15 For any structure $A(\bar{k}_{2+2^m})$, and $(i,j,k) \in A$(respectively,$(i,j) \in A$)

$$\|(i,j,k)\| = brep(i)\#brep(j)\#brep(k)$$

(respectively,$\|(i,j)\| = brep(i)\#brep(j)$).

For all other structures A in IIF and all $a \in A$ we define $\|a\| =$ We define a partial order on norms of elements as follows:

$$brep(i)\#brep(j) \preceq brep(\hat{i})\#brep(\hat{j})\#brep(\hat{k})$$

iff $i \leq \hat{i}$ and $j \leq \hat{j}$.

$$brep(i)\#brep(j)\#brep(k) \preceq brep(\hat{i})\#brep(\hat{j})\#brep(\hat{k})$$

iff $i \leq \hat{i}$, $j \leq \hat{j}$, and either $k \leq \hat{k}$ or $k \geq j\text{-}\hat{k}$.
We also define $w \preceq \infty$ for any norm w. $\preceq$ is undefined for all other pairs of words in $\{0,1,\#\}^*$.

Following our outline, we first show for all $k \in N$, $E_{0,k}$ is a refinement of $\equiv_{0,k}$.

For all $A,B \in \mathrm{IIF}$, $\bar{a}_k \in A^k$, and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_0 \; \bar{b}_k$, then for all $i,j \in I_k$ (a_i,a_j) is distance-1 equivalent to (b_i,b_j). Thus by definition, for $i,j \in I_k$, $f(a_i) = a_j \iff f(b_i) = b_j$. Using Lemma 2 of Chapter 2, we can conclude $\bar{a}_k \equiv_0 \bar{b}_k$.

Again, following our outline, we establish

Lemma 2.16 For all $n,k \in N$, all $A,B \in \mathbb{IF}$, $\bar{a}_k \in A^k$, and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

Proof As noted in Fact 2.9, clause 1 of the definition of $E_{n,k+1}$ is satisfied for any choice of $a_{k+1} \in A$, $b_{k+1} \in B$.

Let $a_{k+1} \in A$. It will be sufficient to show that there is $b_{k+1} \in B$ such that clauses 2 and 3 of the definition of $E_{n,k+1}$ are satisfied.

Our choice of b_{k+1} depends on the properties of a_{k+1}, and so we proceed by cases.

Case 1. For some $j \in I_k$, $|d(a_j,a_{k+1})| \leq 2^n$.

We want to choose $b_{k+1} \in B$ such that $d(b_j,b_{k+1}) = d(a_j,a_{k+1})$, i.e., such that $f^{d(a_j,a_{k+1})}(b_j) = b_{k+1}$.

By hypothesis, $d(o_{a_j},a_j) \; {}_{2^{n+1}}\!\equiv \; d(o_{b_j},b_j)$. By Lemma 2.5.3, this is equivalent to the assertion that for every integer $\ell \geq -2^{n+1}$, [there is $a \in A$ such that $f^\ell(a_j) = a$ <=> there is $b \in B$ such that $f^\ell(b_j) = b$]. Since for $\ell = d(a_j,a_{k+1}) \geq -2^n$ and $a = a_{k+}$ we have $f^\ell(a_j) = a$, we conclude that such a choice of b_{k+1} is possible in this case.

We now verify that this choice of b_{k+1} satisfies clauses 2 and 3 of the definition of $E_{n,k+1}$.

We first verify clause 2. By hypothesis, for all $i,\ell \in I_k$, (a_i,a_ℓ) is distance -2^{n+1} equivalent to (b_i,b_ℓ), and thus (a_i,a_ℓ) is *a fortiori* distance -2^n equivalent to (b_i,b_ℓ).

Using Lemma 2.7.1, it will be sufficient to verify for all $i \in I_{k+1}$, that (a_{k+1},a_i) is distance -2^n equivalent to (b_{k+1},b_i).

First let $i \in I_k$. The fact that $f^{d(a_j,a_{k+1})}(a_j) = a_{k+1}$ implies that for any integer ℓ, $[f^{\ell}(a_{k+1}) = a_i \iff f^{\ell+d(a_j,a_{k+1})}(a_j) = a_i]$. Let ℓ be any integer such that $|\ell| \leq 2^n$. Then $|\ell+d(a_j,a_{k+1})| \leq 2^{n+1}$, and since (a_j,a_i) is distance -2^{n+1} equivalent to (b_j,b_i), we conclude that $f^{\ell+d(a_j,a_{k+1})}(a_j) = a_i \iff f^{\ell+d(a_j,a_{k+1})}(b_j) = b_i$. By choice of b_{k+1}, $f^{d(a_j,a_{k+1})}(b_j) = b_{k+1}$, and so $f^{\ell+d(a_j,a_{k+1})}(b_j) = b_i \iff f^{\ell}(b_{k+1}) = b_i$. Altogether, we have that $f^{\ell}(a_{k+1}) = a_i \iff f^{\ell}(b_{k+1}) = b_i$ for $|\ell| \leq 2^n$. In other words, (a_{k+1},a_i) is distance -2^n equivalent to (b_{k+1},b_i), as required.

To finish clause 2, it remains only to show that (a_{k+1},a_{k+1}) is distance -2^n equivalent to (b_{k+1},b_{k+1}). By Lemma 2.7.3, it is sufficient to show that $|C_{a_{k+1}}|_\infty \underset{2^n}{=} |C_{b_{k+1}}|_\infty$. But $C_{a_{k+1}} = C_{a_j}$ since $|d(a_j,a_{k+1})| \leq 2^n$ by assumption, and $|C_{a_j}|_\infty \underset{2^n}{=} |C_{b_j}|_\infty$ by the fact that (a_j,a_j) is distance -2^n equivalent to (b_j,b_j) and Lemma 2.7.3, and $C_{b_j} = C_{b_{k+1}}$ by choice of b_{k+1}.

We now verify clause 3. By hypothesis, for all $i \in I_k$, $d(o_{a_i},a_i) \underset{2^{n+1}}{=} d(o_{b_i},b_i)$, and so <u>a fortiori</u> $d(o_{a_i},a_i) \underset{2^n}{=} d(o_{b_i},b_i)$. We therefore must verify that $d(o_{a_{k+1}},a_{k+1}) \underset{2^n}{=} d(o_{b_{k+1}},b_{k+1})$. Using Lemma 2.5.2,

$d(o_{a_{k+1}},a_{k+1}) = d(o_{a_j},a_j) + d(a_j,a_{k+1})$, and $d(o_{b_{k+1}},b_{k+1}) = d(o_{b_j},b_j) + d(b_j,b_{k+1})$. Since $d(o_{a_j},a_j) \underset{2^{n\dot{-}1}}{=} d(o_{b_j},b_j)$ by hypothesis, and $d(a_j,a_{k+1}) = d(b_j,b_{k+1})$ by choice of b_{k+1} in this case, and $|d(a_j,a_{k+1})| \leq 2^n$ by assumption, Lemma 1.4 immediately implies that $d(o_{a_{k+1}},a_{k+1}) \underset{2^n}{=} d(o_{b_{k+1}},b_{k+1})$.

Clause 3 is thus verified.

Before proceeding to Case 2 of the proof of Lemma 2.16, we need the following lemma:

Lemma 2.17 Let $n,k \in N$, $A,\mathcal{B} \in 1\mathbb{F}$, and suppose $\bar{a}_k \; E_{n+1} \; \bar{b}_k$. Then

1. For any $j \in N_{2^n}$

$|\{a \in 1\text{-CHAINS}(A) | d(o_a,a) = j$ and for all $i \in I_k$, $|d(a_i,a)| > 2^n\}|_\infty \underset{n\dot{-}1}{=} |\{b \in 1\text{-CHAINS}(\mathcal{B}) | d(o_b,b) = j$ and for all $i \in I_k$, $|d(b_i,b)| > 2^n\}|_\infty$.

2. For all $j \in I_{2^n}$

$|\{a \in \text{LOOPS}(A,j)|$ for all $i \in I_k$, $d(a_i,a) = \infty\}| \underset{n\dot{-}1}{=} |\{b \in \text{LOOPS}(\mathcal{B},j)|$ for all $i \in I_k$, $d(b_i,b) = \infty\}|$

3. $|\{a| \; |C_a|_\infty > 2^n$, $d(o_a,a) > 2^n$, and for all $i \in I_k$, $|d(a_i,a)| > 2^n\}|_\infty \underset{n\dot{-}1}{=} |\{b| \; |C_b|_\infty > 2^n$, $d(o_b,b) > 2^n$, and for all $i \in I_k$, $|d(b_i,b)| > 2^n\}|$.

Actually, it would be sufficient for our purposes to prove Lemma 2.17 with $\underset{0}{=}$ in place of $\underset{n\dot{-}1}{=}$. However, our choice of $E_{n,k}$ allows us to prove the stronger result.

Proof 1. Let $j \in N_{2^n}$. By clause 1 of the definition of $E_{n+1,k}$,

$$|\text{ORIGINS } (A)|_{\infty} \overset{=}{n+1+k} |\text{ORIGINS } (B)|_{\infty}.$$

Since for any $C \in \mathbb{F}$ and $o \in \text{ORIGINS } (C)$, there is a unique $c \in 1\text{-CHAINS } (C)$ such that $d(o,c) = j$, we conclude

$$|\{a \in 1\text{-CHAINS } (A) | d(o_a,a) = j\}|_{\infty} \overset{=}{n+1+k}$$
$$|\{b \in 1\text{-CHAINS } (B) | d(o_b,b) = j\}|_{\infty}.$$

We claim for any $i \in I_k$ that [there is $a \in A$ such that $d(o_a,a) = j$ and $|d(a_i,a)| \leq 2^n$ iff there is $b \in B$ such that $d(o_b,b) = j$ and $|d(b_i,b)| \leq 2^n$]. Obviously if such $a \in A$, or $b \in B$, exists, it is unique.

To prove the claim, if $d(o_{a_i},a_i) > 2^{n+1}$, then there is no $a \in A$ such that $d(o_a,a) = j$ and $|d(a_i,a)| \leq 2^n$. For if such $a \in A$ existed, by Lemma 2.5.2, $d(o_a,a) = d(o_{a_i},a_i) + d(a_i,a)$; since $d(o_{a_i},a_i) > 2^{n+1}$ and $|d(a_i,a)| \leq 2^n$, we conclude $d(o_a,a) > 2^n$. But by assumption, $d(o_a,a) = j \leq 2^n$, a contradiction. Similarly if $d(o_{b_i},b_i) > 2^{n+1}$, then there is no $b \in B$ such that $d(o_b,b) = j$ and $|d(b_i,b)| \leq 2^n$. Since $d(o_{a_i},a_i) \overset{=}{2^{n+1}} d(o_{b_i},b_i)$ by hypothesis, either both $d(o_{a_i},a_i)$ and $d(o_{b_i},b_i) > 2^{n+1}$, or $d(o_{a_i},a_i) = d(o_{b_i},b_i) \leq 2^{n+1}$. If the former holds, the above argument verifies the claim. If the latter holds, then o_{a_i} and o_{b_i} exist, so there are unique $a \in A$, $b \in B$ such that $d(o_{a_i},a) = d(o_{b_i},b) = j$. For this choice of $a \in A$ and $b \in B$, Lemma 2.5.2 implies $d(a_i,a) = d(b_i,b)$, and thus $|d(a_i,a)| \leq 2^n \iff |d(b_i,b)| \leq 2^n$. The claim then follows easily. We conclude

$$|\{a \in 1\text{-CHAINS } (A) | d(o_a,a) = j \text{ and for some } i \in I_k,$$
$$|d(a_i,a)| \leq 2^n\}| = |\{b \in 1\text{-CHAINS } (B) | d(o_b,b) = j \text{ and for some}$$

$i \in I_k$, $|d(b_i,b)| \leq 2^n\}| \leq k$. Since for any $C \in \mathbb{1}\mathbb{F}$ and any $\bar{c}_k \in C^k$, $\{c \in 1\text{-CHAINS } (C) | d(o_c,c) = j$ and for all $i \in I_k$, $|d(c_i,c)| > 2^n\} = \{c \in 1\text{-CHAINS } (C) | d(o_c,c) = j\} -$ $\{c \in 1\text{-CHAINS } (C) | d(o_{\ },c) = j$ and for some $i \in I_k$, $|d(c_i,c)| \leq 2^n\}$, the result then follows easily, using Lemma 1.4. This completes the proof of Lemma 2.17.1.

The arguments proving 2.17.2 and 2.17.3 are similar to each other; we therefore present only the proof of 3.

3. By assumption,

$$|2\text{-CHAINS } (A)|_\infty + \sum_{j>2^n} |\text{LOOPS } (A,j)|_\infty \underset{(n+k+1)(1+2^{n+k+1})}{=}$$

$$|2\text{-CHAINS } (\mathcal{B})|_\infty + \sum_{j>2^n} |\text{LOOPS } (\mathcal{B},j)|_\infty.$$

Thus

$$\Big| \bigcup_{j>2^n} \text{LOOPS } (A,j) \cup 2\text{-CHAINS } (A) \Big|_\infty \underset{(n+k+1)(1+2^{n+k+1})}{=}$$

$$\Big| \bigcup_{j>2^n} \text{LOOPS } (\mathcal{B},j) \cup 2\text{-CHAINS } (\mathcal{B}) \Big|_\infty.$$

Now, by assumption $|\text{ORIGINS } (A)|_\infty \underset{n+\bar{k}+1}{=} |\text{ORIGINS } (\mathcal{B})|_\infty$; therefore by definitions, for any $m \in N$, $|\{a | a \in 1\text{-CHAINS } (A)$ and $d(o_a,a) > 2^n\}|_\infty \underset{m}{=} |\{b | b \in 1\text{-CHAINS } (\mathcal{B})$ and $d(o_b,b) > 2^n\}|_\infty$. Thus we conclude, by the definition of d and Lemma 1.5, that $|\{a|\ |C_a|_\infty > 2^n$ and $d(o_a,a) > 2^n\}|_\infty = |\{a | a \in \bigcup_{j>2^n} \text{LOOPS } (A,j) \cup$ 1-CHAINS $(A) \cup 2$-CHAINS (A) and $d(o_a,a) > 2^n\}|_\infty \underset{(n+k+1)(2^{n+k+1}+1)}{=}$ $|\{b | b \in \bigcup_{j>2^n} \text{LOOPS } (\mathcal{B},j) \cup 1\text{-CHAINS } (\mathcal{B}) \cup 2\text{-CHAINS } (\mathcal{B})$ and $d(o_b,b) > 2^n\}|_\infty = |\{b|\ |C_b|_\infty > 2^n$ and $d(o_b,b) > 2^n\}|_\infty$. For any

$C \in \mathcal{IF}$, we denote $\{c \in C \mid |C_c|_\infty > 2^n$ and $d(o_c,c) > 2^n\}$ by LSTRINGS (C). Then we have shown above that $|\text{LSTRINGS } (A)|_\infty \underset{(n+k+1)(2^{n+k+1}+1)}{=} |\text{LSTRINGS } (\mathcal{B})|_\infty$, and we wish to prove $|\{a \in \text{LSTRINGS } (A) \mid \text{for all } i \in I_k, |d(a_i,a)| > 2^n\}|_\infty \underset{n+1}{=} |\{b \in \text{LSTRINGS } (\mathcal{B}) \mid \text{for all } i \in I_k, |d(b_i,b)| > 2^n\}|_\infty$. We claim $|\{a \in \text{LSTRINGS } (A) \mid \text{for some } i \in I_k, |d(a_i,a)| \leq 2^n\}| = |\{b \in \text{LSTRINGS } (\mathcal{B}) \mid \text{for some } i \in I_k, |d(b_i,b)| \leq 2^n\}| \leq k\cdot(2^{n+1}+1)$. Noting that $n+1 \leq (n+k+1)(2^{n+k+1}+1) - k(2^{n+1}+1)$, the proof of 3 follows as in 1 from Lemma 1.4.

It remains only to prove the claim; we note that the proof of Lemma 2.16, Case 1 shows that for each $a \in A$ such that $|d(a_i,a)| \leq 2^n$ for some $i \in I_k$, there is a corresponding $b \in B$ such that both $d(b_j,b) \underset{2^n}{=} d(a_j,a)$ for all $j \in I_k$, and $d(o_a,a) \underset{2^n}{=} d(o_b,b)$. Conversely, for each $b \in B$ such that $|d((b_j,b)| \leq 2^n$ for some $j \in I_k$, there is a corresponding $a \in A$ Moreover, this correspondence is a bijection.

Now, by assumption, (a_i,a_i) is distance-2^n equivalent to (b_i,b_i) for all $i \in I_k$, so by Lemma 2.7.3, $|C_{a_i}|_\infty \underset{2^n}{=} |C_{b_i}|_\infty$. In particular, if $|d(a,a_i)| \leq 2^n$ and $a \in$ LSTRINGS (A), then since $C_{a_i} = C_a$ and $|C_a|_\infty > 2^n$, we conclude that $|C_b|_\infty > 2^n$ for the b corresponding to a, so $b \in$ LSTRINGS $(\mathcal{B})$. Hence $|\{a \in \text{LSTRINGS } (A)|$ for some $i \in I_k$, $|d(a_i,a)| \leq 2^n\}| = |\{b \in \text{LSTRINGS } (\mathcal{B})|$ for some $i \in I_k$, $|d(b_i,b)| \leq 2^n\}|$. Moreover, since for each $i \in I_k$, there are at most $2^{n+1}+1$

elements within distance 2^n (in absolute value) of a_i, the cardinality of the sets above is at most $k\cdot(2^{n+1}+1)$, which completes the proof of the claim. □

Proof of Lemma 2.16 Case 2. For all $i \in I_k$, $|d(a_i,a_{k+1})| > 2^n$.

As in Case 1, to establish Case 2, it will be sufficient to choose $b_{k+1} \in B$ such that first, for all $i \in I_k$, $|d(b_i,b_{k+1})| > 2^n$, second, (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b_{k+1},b_{k+1}), and third, $d(o_{a_{k+1}},a_{k+1}) =_{2^n} d(o_{b_{k+1}},b_{k+1})$.

First suppose a_{k+1} is such that $d(o_{a_{k+1}},a_{k+1}) \leq 2^n$. Then $C_{a_{k+1}}$ is a one-sided chain such that for all $i \in I_k$, $|d(a_i,a_{k+1})| > 2^n$. Using Lemma 2.17.1 with $j = d(o_{a_{k+1}},a_{k+1})$, there is $b \in B$ such that C_b is a one-sided chain, $d(o_b,b) = d(o_{a_{k+1}},a_{k+1})$, and for all $i \in I_k$, $|d(b_i,b)| > 2^n$. It follows directly from Lemma 2.7.3 that (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b,b). If we let $b_{k+1} = b$, then the desired result follows immediately.

Next suppose $|C_{a_{k+1}}| \leq 2^n$. Then for all $i \in I_k$, $d(a_i,a_{k+1}) = \infty$. Using Lemma 2.17.2 with $j = |C_{a_{k+1}}|$, there is a $b \in B$ such that C_b is a loop of size $|C_{a_{k+1}}|$, and for all $i \in I_k$, $d(b_i,b) = \infty$. It follows directly from Lemma 2.7.3 that (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b,b). By definition, we have $d(o_{a_{k+1}},a_{k+1}) = d(o_b,b) = \infty$. If we let $b_{k+1} = b$, the desired result follows immediately.

Finally, suppose both $d(o_{a_{k+1}},a_{k+1}) > 2^n$ and $|C_{a_{k+1}}| > 2^n$, that is $a_{k+1} \in$ LSTRINGS (A). Using Lemma 2.17.3,

there is $b \in \text{LSTRINGS}(B)$ such that for all $i \in I_k$, $|d(b_i,b)| > 2^n$. Hence, choose $b_{k+1} = b$. □

Corollary 2.18 For all $n,k \in N$, $A,B \in \text{IIF}$, $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$, and such that

1. $|d(b_i,b_{k+1})| \leq 2^n+1$ for some $i \in I_k$, or
2. $|d(o_{b_{k+1}},b_{k+1})| \leq 2^n$, or
3. $d(b_i,b_{k+1}) = \infty$ for all $i \in I_k$.

Proof Examination of the proof of Lemma 2.16 shows we always obtain such a b_{k+1}, except possibly in the case that both $d(o_{a_{k+1}},a_{k+1}) > 2^n$ and $|C_{a_{k+1}}|_\infty > 2^n$. Let b denote the choice of b_{k+1} in this case in Lemma 2.16. If there is $i \in I_k$ such that $d(b_i,b) \neq \infty$, let b_j, for $j \in I_k$, be such that $|d(b_j,b)| \leq |d(b_i,b)|$ for all $i \in I_k$. If $d(b_j,b) > 0$, we let $\ell = 1+2^n$; if $d(b_j,b) < 0$, we let $\ell = -1-2^n$. We then choose $b_{k+1} = f^\ell(b_j)$. It remains to show, for this choice of b_{k+1}, firstly, for all $i \in I_k$, $|d(b_i,b_{k+1})| > 2^n$, secondly (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b_{k+1},b_{k+1}), and thirdly, $d(o_{b_{k+1}},b_{k+1}) > 2^n$.

We first establish, for all $i \in I_k$, $|d(b_i,b_{k+1})| > 2^n$. If $b_i \in C_{b_j}$ for $i \in I_k$, then either $d(b_i,b_{k+1}) = d(b_i,b_j) + d(b_j,b_{k+1})$, and all three distances are similarly signed, or $d(b_i,b_{k+1}) = d(b_i,b) + d(b,b_{k+1})$ and all three distances are similarly signed, by choice of b_{k+1} and the fact $|d(b_j,b)|$ is

minimal. Now, since $|d(b_j,b_{k+1})| > 2^n$ by choice of b_{k+1}, and $|d(b_i,b)| > 2^n$ by assumption, the result then follows.

The fact that (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b_{k+1},b_{k+1}), and $d(o_{b_{k+1}},b_{k+1}) > 2^n$, follow routinely and are left to the reader. □

Corollary 2.18 motivates our definition of H in this case; we therefore define

Definition 2.19 For $n,k,k_1,k_2,k_3 \in N$, we define

$$H(n,k,brep(k_1)\#brep(k_2)\#brep(k_3)) =$$
$$brep(max(k_1,k+1))\#brep((n+k+1)(1+2^{n+k+1})+1)\#brep(k_3+2^n+1).$$

For all other $m \in \{0,1,\#\}^* \cup \{\infty\}$, $H(n,k,m) = \infty$.

We now establish:

Theorem 2.20 For all $n,k \in N$, $A,\mathcal{B} \in \mathbb{IF}$, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$, and $m \in \{0,1,\#\}^* \cup \{\infty\}$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, and $b_i \leq m$ for all $i \in I_k$, then for all $a_{k+1} \in A$ there is a $b_{k+1} \in B$ such that $b_{k+1} \preceq H(n,k,m)$ and $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

Proof Clearly it suffices to consider only the case $H(n,k,m) \neq \infty$, since otherwise the result follows from Lemma 2.16. Thus we can assume by definition of H that $m = brep(k_1)\#brep(k_2)\#brep(k_3)$ for some $k_1,k_2 \in I$, $k_3 \in N$, and by definition of $\| \; \|$ that $\mathcal{B}$ is an Ehrenfeucht structure for $\mathbb{IF}$, as given by definition 2.13.

Now, using Corollary 2.18, there is a $b \in B$ such that either

1. $|d(b_i,b)| \leq 2^n+1$ for some $i \in I_k$,

2. $|d(o_b,b)| \leq 2^n$, or

3. $|d(b_i,b)| = \infty$ for all $i \in I_k$.

In the first case, consider b_i. Either $b_i = (j_1,j_2,j_3)$ or $b_i = (j_1,j_2)$ for $j_1,j_2 \in I$, $j_3 \in N$. Since $|d(b_i,b)| \leq 2^n+1$, we must have $b = (j_1,j_2,j_3+d)$ or (j_1,j_2+d) for $d = d(b_i,b)$. It then follows from $b_i \preceq m$ and the definition of H that $b \preceq H(n,k,m)$. We therefore choose $b_{k+1} = b$.

In case $|d(o_b,b)| \leq 2^n$, we must have $b = (i,d)$, where $d = d(o_b,b)$. Now if $i \leq k+1$, we take $b_{k+1} = b$; it then follows from the definition of H that $b_{k+1} \preceq H(n,k,m)$. If $i > k+1$, then since there are only k b_i's, and at least k+1 one-sided chains, we can choose $b_{k+1} = (i',d)$ with $i' \leq k+1$ and $d(b_i,b_{k+1}) = \infty$ for all $i \in I_k$. Then $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$, and $b_{k+1} \preceq H(n,k,m)$ follows from the definition of H.

In case $|d(b_i,b)| = \infty$ for all $i \in I_k$ and $b = (i,j)$, then by an argument similar to the above we can choose $b' = (i',j)$, $i' \leq k+1$, with $d(b_i,b') = \infty$ for all i. We let $b_{k+1} = (i',2^n+1)$. It follows easily that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$ and $b_{k+1} \preceq H(n,k,m)$. If $b = (i,j,k)$ the proof is similar to the above and is left to the reader. □

Lastly, following our outline,

Definition 2.21 We define $h:N \to \{0,1,\#\}^* \cup \{\infty\}$ as follows:

$$h(m) = \underbrace{brep(m(1+2^m)+1)\#brep(m(1+2^m)+1)\#\ldots\#brep(m(1+2^m)+1)}_{2+2^m}.$$

We note that for any A in 11F, $\sum_{j>2^n} |\text{LOOPS}(A,j)|_\infty$ + $|2\text{-CHAINS}(A)|_\infty$ is either 0 or larger than 2^n. It is then easy to verify that for any A in 11F, and every $n \in I$, there is some (Ehrenfeucht) structure A' such that A' $E_{n,o}$ A and A' $\preceq$ h(n). We have thus verified all the steps necessary to conclude 11F is (h,H) bounded by using Theorem 4 of Chapter 2.

Theorem 2.22 For H as in Definition 2.19, and h as in Definition 2.21, the class of structures 11F is (h,H) bounded.

Theorem 2.23 Let A be an Ehrenfeucht structure, and let $Q_1x_1 \ldots Q_nx_n\ F(\bar{x}_n)$ be a sentence such that $F(\bar{x}_n)$ is quantifier-free. Let m_0 = 0#0#0, and let

m_{i+1} = brep(i+1)#brep(n(1+2^n)+1)#brep($\sum_{j=1}^{i}(2^{n-j}+1)$) for $i \in N_{n-1}$.

Then $A \models Q_1x_1 \ldots Q_nx_n\ F(\bar{x}_n)$ <=>

$A \models Q_1x_1 \preceq m_1 \ldots Q_nx_n \preceq m_n\ F(\bar{x}_n)$.

Proof Since A is H-bounded by Theorem 2.22, using theorem 1 of Chapter 2, we need only verify that $m_0 \preceq m_1 \preceq \ldots \preceq m_n$ and $H(n-i,i-1,m_{i-1}) \preceq m_i$ for $1 \leq i \leq n$, which follows directly by definition. □

Theorem 2.24 Let m_i, $i \in N_n$, be as in the previous theorem, and let $Q_1x_1 \ldots Q_nx_n\ F(\bar{x}_n)$ be a sentence such that $F(\bar{x}_n)$ is quantifier free. Then $Q_1x_1 \ldots Q_nx_n\ F(\bar{x}_n)$ is satisfiable in 11F <=> there is an A in 11F, A $\preceq$ h(n), such that

$A \models Q_1x_1 \preceq m_1 \ldots Q_nx_n \preceq m_n\ F(\bar{x}_n)$.

Proof Since 11F is (h,H) bounded, it follows by definition [$Q_1x_1 \ldots Q_nx_n\ F(\bar{x}_n)$ satisfiable in 11F <=> there is A in 11F, $A \preceq h(n)$, such that $A \models Q_1x_1 \ldots Q_nx_n\ F(x_n)$].

The theorem then follows immediately from the previous theorem. □

Corollary 2.25 For some constant $c > 0$, whether a sentence F of length n is in SAT(11F) can be decided within space 2^{cn}.

Proof Using Theorem 5.3 of Chapter 1, convert F to an equivalent sentence $Q_1x_1 \ldots Q_nx_n\ G(\bar{x}_n)$ where G is quantifier-free and of length at most n log n. By the previous theorem, F is in SAT(11F) iff for some A in 11F such that $A \preceq h(n)$ we have $A \models Q_1x_1 \preceq m_1 \ldots Q_nx_n \preceq m_n\ G(\bar{x}_n)$ (where $m_0 = 0\#0\#0$, and $m_{i+1} = \text{brep}(i+1)\#\text{brep}(n(1+2^n)+1)\#\text{brep}(\sum_{j=1}^{i}(2^{n-j}+1))$ for $i \in N_{n-1}$). Therefore, whether F is in SAT(11F) can be decided by first cycling through all $\|A\|$, A in 11F, such that $A \preceq h(n)$, and then deciding if $A \models F$.

Now, there is a straightforward procedure which given n, prints out, one at a time (by orderly cycling through binary representations of integers) all $\|A\| \preceq h(n)$. Furthermore, this procedure uses space at most that required to write the longest word in $\{\|A\| \mid A \preceq h(n)\}$, which is of length at most $(2+2^n)(\log(n(2^n+1)+1)) \leq (2+2^n)\log((n+1)2^{n+1}) = (2+2^n)(n+1+\log(n+1)) \leq 2^{2n+2}$.

Given $\|A\|$, to decide if $A \models F$, cycle through, at each quantifier Q_i, all $\|A\| \preceq m_i$, and test the truth of G on the n-tuple of norms so generated. There is a straightforward

effective procedure which given $\|A\|$, i, and n, prints out all $\|A\| \preceq m_i$. Furthermore, this procedure uses space at most that required to write the largest word in $\{\|A\| \mid A \in A, A \preceq m_i\}$, which is at most $3(\log(n(2^n+1)))+2 \leq 3(\log(n)+n+1)+2 \leq 3(2n+1)+2 = 6n+5$.

We let the reader convince themselves that a Turing machine implementing this outlined procedure uses space at most 2^{cn} for some constant c. □

Corollary 2.26 For some constants c,d, whether a sentence F of length n is in SAT(11F) can be decided nondeterministically within space 2^{cn} and, simultaneously, time d^{n^2}.

Proof The procedure nondeterministically guesses $\|A\| \preceq h(n)$ and then decides if $A \models F$, as in the proof of Corollary 2.25.

Now, this procedure takes space 2^{cn}, as previously noted; however, it can also be shown to take time d^{n^2} for some constant $d > 1$. For to decide if $A \models F$, we have to first cycle through at most n different sets of norms of cardinality at most k^n for some $k \in I$, which takes k^{n^2} time, and then on each cycle decide a quantifier-free sentence, which takes at most time n. Since nondeterministically guessing $\|A\|$ takes only the time required to write the longest word in $\{\|A\| \mid A \preceq h(n)\}$ which is of length at most 2^{2n+2}, we conclude this nondeterministic procedure takes time d^{n^2} for some constant d, and, simultaneously, space 2^{cn}. □

Section 3: Upper Bounds for the Theory of a 1-1 Unary Function, with a Monadic Predicate

We next apply our technique to the theory of a 1-1 unary function with a monadic predicate. Let 11FM denote the class of models $\{\langle A,C\rangle \mid A \in 11F,\ C \subseteq A\}$. Since for $\langle A,C\rangle \in 11FM$, we have $A \in 11F$, we use the definitions and analysis of Section 2 freely in what follows.

Again we consider a language with only relation symbols in it; let L(M) be a first-order language with a single binary relation symbol and a single unary relation symbol in it. We use the notation $f(x) = y$, and interpret $f(a) = b$ for $\langle A,C\rangle \in 11FM$, $a,b \in A$, as in Section 2, for $A \in 11F$. For $\langle A,C\rangle \in 11FM$, $a \in A$, we interpret the assertion that the unary relation holds for a as $a \in C$.

Quantifier-depth n formulas in L(M) with k free variables obviously provide at least as much information as such formulas in the language L of a 1-1 unary function. In addition, quantifier-depth n formulas in L(M) with k free variables $\bar{z}_k$ can determine, for all elements whose distance from z_i is less than or equal to a certain bound depending on n, the membership or nonmembership of that element in the distinguished subset of a model. Also, such formulas can provide, for each word in $\{0,1\}^*$ of length up to a certain bound depending on n, the number, up to a certain bound depending on n and k, of distinct elements whose "nearby" elements' membership or nonmembership in the distinguished subset, taken as 1 or 0,

respectively, and concatenated in order, yield that word. We make this precise in our definition of $E_{n,k}$ below.

Definition 3.1 Let $n \in N$. An n-word is any word in $\{0,1\}^{\leq 1+2^{n+1}} \cup \emptyset \cdot \{0,1\}^{\leq 1+2^{n+1}}$.

Definition 3.2 Let $\langle A,C \rangle \in$ 1IFM, $a \in A$, $n \in N$. The n-word of a, $W(n,a)$ is an n-word defined as follows.

1. If $|C_a| < 2^{n+1}$, $W(n,a) = w_1 \ldots w_{|C_a|}$, with $w_i \in \{0,1\}$ for $i \in I_{|C_a|}$, such that for all $i \in I_{|C_a|}$,
$w_i = 1 \iff f^{i-1}(a) \in C$.
2. If $d(o_a,a) \leq 2^n$, $W(n,a) = \emptyset \cdot w_1 \cdot \ldots \cdot w_{1+d(o_a,a)+2^n}$, with $w_i \in \{0,1\}$ for all $i \in I_{1+d(o_a,a)+2^n}$, such that for all $i \in I_{d(o_a,a)+2^n+1}$,
$w_i = 1 \iff f^{i-1}(o_a) \in C$.
3. Otherwise, $W(n,a) = w_1 \cdot \ldots \cdot w_{1+2^{n+1}}$, $w_i \in \{0,1\}$ for $i \in I_{1+2^{n+1}}$, such that for all $i \in I_{1+2^{n+1}}$,
$w_i = 1 \iff f^{i-2^n-1}(a) \in C$.
(By Lemma 2.5.3, since $d(o_a,a) > 2^n$, for $i \in I_{1+2^{n+1}}, f^{i-2^n-1}(a) \in A$).

We state without proof the following; the proof is routine and is left to the reader:

Lemma 3.3 Let $A,B \in$ 1IFM, $a,a' \in A$, $b,b' \in B$, and $n \in N$.

1. $W(n,a) = W(n,b) \Rightarrow |C_a|_\infty \underset{2^{n+1}}{=} |C_b|_\infty$, and thus (a,a) is distance-2^{n+1} equivalent to (b,b).

2. $W(n,a) = W(n,b) \Rightarrow d(o_a,a) =_{2^n} d(o_b,b)$.

3. If $W(n+1,a) = W(n+1,b)$, and if $d(a,a') = d(b,b')$, and $|d(a,a')| \leq 2^n$, then $W(n,a') = W(n,b')$.

Definition 3.4 Let $A \in$ llFM, $n \in N$, w an n-word.
$WORDS\ (A,w) = \{a \in A | W(n,a) = w\}$.

Definition 3.5 Let $n,k \in N$, $A,B \in$ llFM. A is n,k equivalent to B if for any n-word w, $|WORDS\ (A,w)|_\infty =_{(n+k)(2^{n+k}+1)} |WORDS\ (B,w)|_\infty$

It follows easily from Lemma 1.5 for all $n,k \in N$, $A,B \in$ llFM, that A $n+1$, k equivalent to B implies A n, $k+1$ equivalent to B.

Definition 3.6 Let $n \in N$, $A,B \in$ llFM. $A\ E_{n,0}\ B$ if A is $n,0$ equivalent to B.

Definition 3.7 Let $n \in N$, $k \in I$. Let $A,B \in$ llFM, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$. We define $\bar{a}_k\ E_n\ \bar{b}_k$ iff

1. A is n,k equivalent to B, and
2. for all $i,j \in I_k$, (a_i,a_j) is distance-2^n equivalent to (b_i,b_j), and
3. for all $i \in I_k$, $W(n,a_i) = W(n,b_i)$.

Again, we have chosen a somewhat finer refinement $E_{n,k}$ than is necessary for the proofs to follow, for the sake of technical simplicity.

It should be clear $E_{n,k}$ is an equivalence relation of finite index.

Following our outline, we first show for all $k \in N$, $E_{0,k}$ is a refinement of $\equiv_{0,k}$.

For all $A,B \in$ llFM, $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_0 \; \bar{b}_k$, then for all $i,j \in I_k$, (a_i,a_j) is distance-1 equivalent to (b_i,b_j). Thus, by definition, for all $i,j \in I_k$, $f(a_i) = a_j \iff f(b_i) = b_j$. Again, since $\bar{a}_k \; E_0 \; \bar{b}_k$, for all $i \in I_k$, $W(0,a_i) = W(0,b_i)$. Thus for all $i \in I_k$, a_i is in the distinguished subset of $A \iff b_i$ is in the distinguished subset of B. Using Lemma 5 of Chapter 2, we conclude $\bar{a}_k \; E_0 \; \bar{b}_k$.

Again, following our outline, we establish

Lemma 3.8 For all $n,k \in N$, all $A,B \in$ llFM, $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

Proof We first note that A n+1, k equivalent to B implies A n,k+1 equivalent to B, so clause 1 of the definition of $E_{n,k+1}$ is satisfied for any choice of $a_{k+1} \in A$, $b_{k+1} \in B$.

Let $a_{k+1} \in A$. It will be sufficient to show that there is $b_{k+1} \in B$ such that clauses 2 and 3 of the definition of $E_{n,k+1}$ are satisfied.

Our choice of b_{k+1} depends on the properties of a_{k+1}, and so we proceed by cases.

Case 1. For some $j \in I_k$, $|d(a_j,a_{k+1})| \leq 2^n$.

The proof proceeds as in Case 1 of Lemma 2.16. That is, we choose $b_{k+1} \in B$ such that $f^{d(a_j,a_{k+1})}(b_j) = b_{k+1}$. Since $W(n+1,a_j) = W(n+1,b_j)$ by assumption, from Lemma 3.3.2 we can

conclude $d(o_{a_j},a_j) \ _{2^{n+1}}\!\!= d(o_{b_j},b_j)$. Then, as in Case 1 of Lemma 2.16, we infer such a choice of b_{k+1} is possible.

We now verify that this choice of b_{k+1} satisfies clauses 2 and 3 of the definition of $E_{n,k+1}$.

We first verify clause 2. Using Lemma 2.7.1, it will be sufficient to verify for all $i \in I_{k+1}$, that (a_{k+1},a_i) is distance-2^n equivalent to (b_{k+1},b_i).

If $i \in I_k$, the proof proceeds exactly as in Case 1 of Lemma 2.5.3. Thus to finish clause 2, it remains only to show that (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b_{k+1},b_{k+1}). To do this, it is convenient to verify first that clause 3 holds.

To verify clause 3, it will be sufficient to show $W(n,a_{k+1}) = W(n,b_{k+1})$. Now, by assumption, $W(n+1,a_j) = W(n+1,b_j)$. $d(b_j,b_{k+1}) = d(a_j,a_{k+1})$ holds by choice of b_{k+1}; $|d(a_j,a_{k+1})| \leq 2^n$ by assumption of this case. Thus by Lemma 3.3.3, $W(n,a_{k+1}) = W(n,b_{k+1})$. Clause 3 is thus verified.

To finish clause 2, since $W(n,a_{k+1}) = W(n,b_{k+1})$, it follows directly from Lemma 3.3.1 that (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b_{k+1},b_{k+1}). This completes the proof of Case 1.

Before we proceed to Case 2, we need the following:

Lemma 3.9 Let $n,k \in N$, $A,B \in$ llFM, and suppose $\bar{a}_k \ E_{n+1} \ \bar{b}_k$. Then for any n-word w

$|\{a \in A | W(n,a) = w \text{ and for all } i \in I_k, |d(a_i,a)| > 2^n\}|_\infty \ _{n+1}\!\!= |\{b \in B | W(n,b) = w \text{ and for all } i \in I_k, |d(b_i,b)| > 2^n\}|_\infty$.

Proof Let w be an n-word. By assumption,

$|\text{WORDS}(A,w)|_{\infty} = |\text{WORDS}(\mathcal{B},w)|$.

We claim $|\{a \in \text{WORDS}(A,w)|$ for some $i \in I_k$, $|d(a_i,a)| \leq 2^n\}| = |\{b \in \text{WORDS}(\mathcal{B},w)|$ for some $i \in I_k$, $|d(b_i,b)| \leq 2^n\}| \leq k \cdot (2^{n+1}+1)$. Noting that $n+1 \leq (n+k+1)(2^{n+k+1}+1) - k(2^{n+1}+1)$, the proof then follows from Lemma 1.4.

The proof of the claim follows, as in Lemma 2.17, directly from the proof of Lemma 3.8, Case 1. □

Proof of Lemma 3.8 Case 2. For all $i \in I_k$, $|d(a_i,a_{k+1})| > 2^n$.

Let $W(n,a_{k+1}) = w$. By Lemma 3.9, there must be $b \in B$ such that $W(n,b) = w$, and for all $i \in I_k$, $|d(b_i,b)| > 2^n$. Since $W(n,a_{k+1}) = W(n,b)$, Lemma 3.3.1 implies (a_{k+1},a_{k+1}) is distance-2^n equivalent to (b,b). If we let $b_{k+1} = b$, then clauses 2 and 3 of the definition of $E_{n,k+1}$ are immediately verified. □

We show that we can restrict our decision procedure to structures of the following kind.

Notation 3.10 For any structure A in llFM with distinguished subset $C \subseteq A$ and any one-sided chain O, seq(A,O) denotes the infinite sequence determined by membership in C on O, i.e. for $n \in N$, with the origin of O denoted by org,

$\text{seq}(A,O)(n) = 1 \iff f^n(\text{org}) \in C$.

Definition 3.11 For $m \in N$, $V(m)$ denotes

$$\left(N_{2(1+2^{m+1})_{[m(2^m+1)+1]}}\right)^{2+2^{m+1}}.$$

For $\bar{k}_{2+2^{m+1}} \in V(m)$, we now define $A(\bar{k}_{2+2^{m+1}}) \in$ llF to contain, for each $1 \leq j \leq 2^{m+1}$, k_j loops of size j, $k_{1+2^{m+1}}$ one-sided chains, and $k_{2+2^{m+1}}$ loops of size $1+2^{m+1}$.

Definition 3.12 Let $m \in N$, $\bar{k}_{2+2^{m+1}} \in V(m)$. We define $A\left(\bar{k}_{2+2^{m+1}}\right)$ as follows:

$$A = \{(i,j,k) \mid i,j \in I,\ k \in N,\ j \leq 2^{m+1},\ i \leq k_j,\ k<j\} \cup$$
$$\{(p,q) \mid p \in I,\ q \in N,\ p \leq k_{1+2^{m+1}}\} \cup$$
$$\left\{(i,1+2^{m+1},k) \mid i \in I,\ k \in N,\ i \leq k_{2+2^{m+1}},\ k<1+2^{m+1}\right\}.$$

We define $f((i,j,k)) = (i,j,k+1 (\mathrm{mod}\ j))$, and $f((p,q)) = (p,q+1)$.

Definition 3.13 Let $m \in N$, $(\bar{k}_{2+2^{m+1}}) \in$ llF. We define Monad $\left(A\left(\bar{k}_{2+2^{m+1}}\right)\right)$ to be $\Big\{<A(\bar{k}_{2+2^{m+1}}),C> \mid C \subseteq A$, and for each one-sided chain O in A, $\mathrm{seq}\left(<A\left(\bar{k}_{2+2^{m+1}}\right),C>,O\right) = w_1 \cdot w_2^{\omega}$, for some $w_1,w_2 \in \{0,1\}^*$ such that $\ell n(w_1) \leq 1+2^{m+1}+2^{2^{m+5}}$, $\ell n(w_2) \leq 2^{2^{m+4}}\Big\}$. An Ehrenfeucht structure for llFM is any structure in

$$\bigcup_{m \in N}\left(\bar{k}_{2+2^{m+1}} \in V(m) \bigcup \text{Monad}\left(A\left(\bar{k}_{2+2^{m+1}}\right)\right)\right).$$

We define our concept of norm as follows:

Definition 3.14 Let $\langle A(\bar{k}_{2+2^{m+1}}),C\rangle$ be an Ehrenfeucht structure for llFM. We define

$$\left\|\langle A(\bar{k}_{2+2^{m+1}}),C\rangle\right\| =$$

$$w_1^1\#w_2^1\cdot\ldots\cdot w_{k_1}^1\#w_1^2\#\cdot\ldots\cdot\#w_{k_2}^2\#\cdot\ldots$$

$$\#w_1^{2^{m+1}}\#\cdot\ldots\cdot\#w_{k_{2^{m+1}}}^{2^{m+1}}\#\cdot\#\alpha_1\#\beta_1\#\alpha_2\#\beta_2\#\cdot\ldots\cdot\#$$

$$\alpha_{k_{1+2^{m+1}}}\#\beta_{k_{1+2^{m+1}}}\#\cdot\#\gamma_1\#\cdot\ldots\cdot\#\gamma_{k_{2+2^{m+1}}},$$

where

(a) for $j \in I_{2^{m+1}}$, and $i \in I$, $i \leq k_j$, $w_i^j \in \{0,1\}^j$, and $w_i^j(k) = 1 \iff (i,j,k) \in C$,

(b) for $i \in I_{k_{1+2^{m+1}}}$, the i^{th} one-sided chain 0 is such that $\mathrm{seq}\left(\langle A(\bar{k}_{2+2^{m+1}}),C\rangle,0\right) = \alpha_i\cdot\beta_i^{\omega}$, with

$\ell n(\alpha_i) \leq 1+2^{m+1}+2^{2^{m+5}}$, and $\ell n(\beta_i) \leq 2^{2^{m+4}}$, and

(c) for $i \in I_{k_{2+2^{m+1}}}$, $\gamma_i \in \{0,1\}^{1+2^{m+1}}$, and

$\gamma_i(k) = 1 \iff (i,1+2^{m+1},k) \in C$.

For any other structure A in llFM, $\|A\| = \infty$. We define a partial order on norms as follows. For $m_i,n_i \in I$ for $1 \leq i \leq 3$, and $j_i,\hat{j}_i \in I$ for all I, we define

$$\text{brep}(j_1)\#\text{brep}(j_2)\#\ldots\#\text{brep}(j_{m_1})\#\#$$

$$\text{brep}(k_1)\#\ldots\#\text{brep}(k_{m_2})\#\#$$

$$\text{brep}(\ell_1)\#\ldots\#\text{brep}(\ell_{m_3}) \preceq$$

$$\text{brep}(\hat{j}_1)\#\ldots\#\text{brep}(\hat{j}_{n_1})\#\#\text{brep}(\hat{k}_1)\#\ldots$$

$$\text{brep}(\hat{k}_{n_2})\#\#\text{brep}(\hat{\ell}_1)\#\ldots\#\text{brep}(\hat{\ell}_{n_3})$$

iff $m_i \leq n_i$ for $i = 1,2,3$ and

$j_i \leq \hat{j}_i$ for $i \leq m_1$,

$k_i \leq \hat{k}_i$ for $i \leq m_2$, and

$\ell_i \leq \hat{\ell}_i$ for $i \leq m_3$.

We also define $w \preceq \infty$ for any norm w. $\preceq$ is undefined for all other pairs of words in $\{0,1,\#\}^*$.

Definition 3.15 Let $\langle A(k_{2+2^{m+1}}), C\rangle$ be an Ehrenfeucht structure for llFM. For $a = (i_1,i_2,i_3) \in A$ ($a = (i_1,i_2) \in A$) we define

$$\|(i_1,i_2,i_3)\| = \text{brep}(i_1)\#\text{brep}(i_2)\#\text{brep}(i_3)\#e$$

$$(\|(i_1,i_2)\| = \text{brep}(i_1)\#\text{brep}(i_2)\#e)$$

where $e = 1$ if $a \in C$, and 0 otherwise. For all other structures A in llFM and all $a \in A$, we define

$$\|a\| = \infty.$$

We define a partial order on norms of elements as follows. Let $i,j,\hat{i},\hat{j} \in I$, $\hat{k} \in N$, $e,\hat{e} \in \{0,1\}$. We define $\text{brep}(i)\#\text{brep}(j)\#e \preceq \text{brep}(\hat{i})\#\text{brep}(\hat{j})\#\text{brep}(\hat{k})\#\hat{e}$ iff $i \leq \hat{i}$, $j \leq \hat{k}$ and $e \leq \hat{e}$.

brep(i)#brep(j)#brep(k)#e $\preceq$

brep($\hat{i}$)#brep($\hat{j}$)#brep($\hat{k}$)#$\hat{e}$ iff

either $i \leq \hat{i}$, $j \leq \hat{j}$, $k \leq \hat{k}$, and $e \leq \hat{e}$, or

$i \leq \hat{i}$, $j \leq \hat{j}$, $e \leq \hat{e}$, and $k \geq j\text{-}\hat{k}$.

We also define $w \preceq \infty$ for any norm w. $\preceq$ is undefined for all other pairs of words in $\{0,1,\#\}^*$.

For $m \in N$, and $A \in$ llFM, we remark there is a $\mathcal{B} \in$ llFM with

1. for each $j \in I_{2^{m+1}}$, at most $2^j \cdot [(m)\cdot(2^m+1)+1]$ loops of size j, and
2. at most $2^{(1+2^{m+1})}[(m)(2^m+1)+1]$ one-sided chains, and
3. at most $2^{(1+2^{m+1})}[(m)(2^{m+1})+1]$ loops of size $1+2^{m+1}$,

such that A is m,0 equivalent to $\mathcal{B}$, and thus $A \; E_{m,0} \; \mathcal{B}$. However, this does not determine a <u>finite</u> set of structures in llFM as it would in llF; for there are infinitely many ways of defining membership in the distinguished subset of a structure on a single one-sided chain.

We now proceed to analyze infinite sequences in terms of the n-words which occur in them. Since it is only those n-words in $\{0,1\}^{1+2^{n+1}}$ which can be n-words of infinitely many elements of a one-sided chain, we restrict our attention to these.

Definition 3.16 Let α be an infinite sequence, $m,n \in N$, and w an n-word. We say w occurs in α at m iff $w \in \{0,1\}^{1+2^{n+1}}$, $m \geq 2^n$, and $w = \alpha(m-2^n)\cdot\ldots\cdot\alpha(m+2^n)$.

Definition 3.17 Let α be an infinite sequence, $n \in N$, and $w \in \{0,1\}^{1+2^n}$. We define

WORDS $(\alpha,w) = \{m \in N \mid w$ occurs in α at $m\}$.

Lemma 3.18 Let α be an infinite sequence, and $m \in N$. Then there is an infinite sequence $w_1\cdot w_2^\omega$, for $w_1,w_2 \in \{0,1\}^*$, such that $\alpha \restriction N_{2^m} = w_1\cdot w_2^\omega \restriction N_{2^m}$, and such that for any n-word w

$$|\text{WORDS } (\alpha,w)| = |\text{WORDS } (w_1\cdot w_2^\omega,w)|.$$

Proof Consider $\{w \mid w$ is an m-word and $|\text{WORDS } (\alpha,w)|$ is finite It is a finite set, since the set of all m-words is finite. Since each word in this finite set occurs in α at only finitely many elements of N, there must exist $n \in N$ such that for all $j > n$, no m-word in the above finite set occurs in α at j. We let $w_1 = \alpha(0)\cdot\ldots\cdot\alpha(n+2^m)$. It is then obvious that $\alpha \restriction N_{2^m} = w_1\cdot w_2^\omega \restriction N_{2^m}$, for any $w_2 \in \{0,1\}^*$.

Next consider $\{w \mid w$ is an m-word and $|\text{WORDS } (\alpha,w)|$ is infinite$\}$, which is finite in cardinality by the same argument as above. Since each m-word in this finite set occurs in α at infinitely many elements of N, there must exist $\ell \in N$, $\ell > n+1+2^{m+1}$, such that, firstly $\{w \mid w$ is an m-word and w occurs in α at some i, $n < i < \ell-2^m\}$ is equal to the set of infinitely occurring m-words, and secondly, the m-word which

occurs in α at n+1 is equal to the m-word which occurs at ℓ. We let $w_2 = \alpha(n+1)\cdot\ldots\cdot\alpha(\ell-1)$. Now, the set of m-words which occur in α is obviously a subset of the set of m-words which occur in $w_1\cdot w_2^{\omega}$, by construction; since the m-word which occurs in α at n+1 is equal to the m-word which occurs in α at ℓ, this inclusion is actually equality. It should be clear, then, for any m-word w,

$$|\text{WORDS}\ (\alpha,w)| = |\text{WORDS}\ (w_1\cdot w_2^{\omega},w)|. \qquad \square$$

Lemma 3.18 suggests analyzing <u>words</u> in $\{0,1\}^*$, instead of infinite sequences, in terms of the m-words which occur in them. We therefore extend our definitions to cover the case of words in $\{0,1\}^*$, and prove a series of technical lemmas about words in $\{0,1\}^*$ that will yield the desired results.

<u>Definition 3.19</u> Let $m,n \in N$, $\alpha \in \{0,1\}^*$. We define

$$W(m,n,\alpha) = \begin{cases} \alpha(n-2^m)\cdot\ldots\cdot\alpha(n+2^m) & \text{if } 2^m \le n \le \ell n(\alpha)-2^m-1 \\ \lambda & \text{otherwise.} \end{cases}$$

If $W(m,n,\alpha) \neq \lambda$, we say $W(m,n,\alpha)$ <u>occurs in α</u> at n. (Note this is the obvious extension of our definition for infinite sequences). We say w occurs in α if w occurs in α at some $n \in N$ (We note this definition is consistent with our definition of an occurrence of a subword.) We define, for $w \in \{0,1\}^{1+2^{m+1}}$,

$$\text{WORDS}\ (\alpha,w) = \{n \in N \mid W(m,n,\alpha) = w\}.$$

(We note this is also the obvious extension of our definition for infinite sequences.)

The next definition provides a condition that will allow us to concatenate words in $\{0,1\}^*$ together to form an infinite sequence, without creating any new m-words at the point of concatenation.

Definition 3.20 Let $\alpha,\beta \in \{0,1\}^*$, $m \in N$. α is m-end equivalent to β if $\alpha = \beta$ or there are $\gamma_1,\gamma_2 \in \{0,1\}^{1+2^m}$, and $\alpha',\beta' \in \{0,1\}^*$, such that $\alpha = \gamma_1 \cdot \alpha' \cdot \gamma_2$ and $\beta = \gamma_1 \cdot \beta' \cdot \gamma_2$.

We remark that if α is m-end equivalent to β, and γ is m-end equivalent to δ, then $\alpha \cdot \gamma$ is m-end equivalent to $\beta \cdot \delta$.

Lemma 3.21 Let $m \in N$, $\alpha \in \{0,1\}^*$. There is $\beta \in \{0,1\}^*$ such that $\ell n(\beta) \leq \left(2^{m+1}+1\right)\left(1+2^{\left(1+2^{m+1}\right)}\right)+2^{m+1}+1$, and such that

1. α is m-end equivalent to β, and
2. For any $w \in \{0,1\}^{1+2^{m+1}}$,

$$|\text{WORDS}(\beta,w)| \leq |\text{WORDS}(\alpha,w)|.$$

Proof Suppose $\ell n(\alpha) \geq (2^{m+1}+1)\left(1+2^{\left(1+2^{m+1}\right)}\right)+2+2^{m+1}$. Then $\alpha = \gamma_1 \cdot \alpha_1 \cdot \alpha_2 \cdot \ldots \cdot \alpha_{1+2^{\left(1+2^{m+1}\right)}} \cdot \alpha' \cdot \gamma_2$, where $\gamma_1,\gamma_2 \in \{0,1\}^{1+2^m}$, $\alpha_i \in \{0,1\}^{1+2^{m+1}}$ for $i \in I_{1+2^{\left(1+2^{m+1}\right)}}$, and $\alpha' \in \{0,1\}^*$. Since $|\{0,1\}^{1+2^{m+1}}| = 2^{\left(1+2^{m+1}\right)}$, and there are $\left(1+2^{\left(1+2^{m+1}\right)}\right)$ α_i's, there must be $i,j \in I_{1+2^{\left(1+2^{m+1}\right)}}$, $i < j$ such that $\alpha_i = \alpha_j$. We let $\eta = \gamma_1\alpha_1 \cdot \ldots \cdot \alpha_i \cdot \alpha_{j+1} \cdot \ldots \cdot \alpha_{1+2^{\left(1+2^{m+1}\right)}} \alpha' \gamma_2$. Then

$\ell n(\eta) < \ell n(\alpha)$, α is m-end equivalent to η, and since $\alpha_i = \alpha_j$, we have by construction for any $w \in \{0,1\}^{1+2^{m+1}}$, $|WORDS\ (\beta,w)| \leq |WORDS\ (\alpha,w)|$. Now, if $\ell n(\eta) < (2^{m+1}+1)\left(1+2^{(1+2^{m+1})}\right) + 2 + 2^{m+1}$, the lemma holds with $\beta = \eta$. If not, it is clear by repetition of the above argument sufficiently many times, we do obtain such a β. □

We now make clear the kind of word in $\{0,1\}^*$ we will apply Lemma 3.21 to.

Definition 3.22 Let $k,m \in N$, $\alpha \in \{0,1\}^*$. $i \in N$ is a k-m garbage point of α if $W(m,i,\alpha) \neq \lambda$, and $|\{j < i \mid W(m,j,\alpha) = W(m,i,\alpha)\}| > k$. An occurrence α' of a subword of α, $\alpha' = \alpha(n)\cdot\ldots\cdot\alpha(n+\ell n(\alpha')-1)$, for some $n \in N$, consists solely of k-m garbage points of α if for all $i \in N$, $n \leq i \leq n+\ell n(\alpha')-1$, i is a k-m garbage point of α.

We remark for any $\alpha \in \{0,1\}^*$ there are at most $(k+1)\cdot 2^{(1+2^{m+1})} + 2^{m+1}$ elements of N less than $\ell n(\alpha)$ which are not k-m garbage points of α.

Lemma 3.23 Let $k,m \in N$, $\gamma \in \{0,1\}^*$. There is $\delta \in \{0,1\}^*$, $\ell n(\delta) \leq \left((k+1)2^{(1+2^{m+1})}+2^{m+1}\right)(1+2^{m+1})\left(3+2^{(1+2^{m+1})}\right)$, such that

1. γ is m-end equivalent to δ, and
2. for any $w \in \{0,1\}^{1+2^{m+1}}$ $|WORDS\ (\gamma,w)| \overset{\leq}{k} |WORDS\ (\delta,w)|$.

Proof We construct δ from γ as follows. For every occurrence γ' of a maximal length subword of $\gamma, \gamma' = \gamma(n) \cdot \ldots \cdot \gamma(n+\ell n(\gamma')-1)$ for some $n \in N$, such that

1. $\ell n(\gamma') \geq (2^{m+1}+1)\left(1+2^{(1+2^{m+1})}\right)+2^{m+1}+2$, and
2. γ' consists solely of k-m garbage points of γ,

we apply Lemma 3.21 with $\alpha = \gamma'$, obtaining $\delta' \in \{0,1\}^*$ such that

a. $\ell n(\delta') \leq (2^{m+1}+1)\left(1+2^{(1+2^{m+1})}\right)+2^{m+1}+1$,

b. γ' is m-end equivalent to δ', and

c. for any $w \in \{0,1\}^{1+2^{m+1}}$,
$|\text{WORDS}(\delta', w)| \leq |\text{WORDS}(\gamma', w)|$.

Let δ denote the result of replacing each such occurrence γ' in γ by δ'. Obviously, γ is m-end equivalent to δ, since each δ' is m-end equivalent to each γ'.

Let $w \in \{0,1\}^{1+2^{m+1}}$. Now the first k+1 occurrences in γ of w are at elements of N which are <u>not</u> k-m garbage points of γ. Thus these elements are not part of any occurrence γ' of γ which consists solely of k-m garbage points. Since any γ' which consists solely of k-m garbage points is replaced by δ which is m-end equivalent to it, these first occurrences of w are not deleted in transforming γ to δ. Thus each occurrence δ consists solely of k-m garbage points of δ. In addition, since each γ' is an occurrence of a maximal length subword consisting solely of k-m garbage points of γ, it is easy to see δ' is an occurrence of a maximal length subword consisting

solely of k-m garbage points of δ, using the fact γ' is m-end equivalent to δ'.

We can now verify the claimed bound on $\ell n(\delta)$. Now, there are at most $(k+1)\left(2^{(1+2^{m+1})}\right) + 2^{m+1}$ elements in N less than $\ell n(\delta)$ which are not k-m garbage points of δ. In addition, by the above, all occurrences of maximal length subwords of δ which consist solely of k-m garbage points of δ are of length at most $(2^{m+1}+1)\left(1+2^{(1+2^{m+1})}\right)+2^{m+1}+1$; thus $\ell n(\delta) \leq \left((k+1)\left(2^{(1+2^{m+1})}\right)+2^{m+1}\right)$ $[(1+2^{m+1})+(1+2^{m+1})\left(1+2^{(1+2^{m+1})}\right)+(1+2^{m+1})] =$ $(k+1)\left(2^{(1+2^{m+1})}+2^{m+1}\right)(1+2^{m+1})\left(3+2^{(1+2^{m+1})}\right)$.

It remains to show for any $w \in \{0,1\}^{1+2^{m+1}}$, $|\text{WORDS}(\gamma,w)| \underset{k}{=} |\text{WORDS}(\delta,w)|$.

Firstly, $|\text{WORDS}(\gamma,w)| \geq |\text{WORDS}(\delta,w)|$, by the fact that for any occurrence γ' replaced by δ' to obtain δ, $|\text{WORDS}(\delta',w)| \leq |\text{WORDS}(\gamma',w)|$, and γ' is m-end equivalent to δ'. Since by a previous argument, the first k+1 occurrences of w are <u>not</u> deleted in transforming γ to δ, we must have $|\text{WORDS}(\gamma,w)| \underset{k}{=} |\text{WORDS}(\delta,w)|$. □

We now apply these results on words in $\{0,1\}^*$ to infinite sequences.

<u>Lemma 3.24</u> Let α be an infinite sequence, $m \in N$. There is an infinite sequence $\hat{w}_1 \cdot \hat{w}_2^{\omega}$, $\hat{w}_1, \hat{w}_2 \in \{0,1\}^*$, with $\ell n(\hat{w}_1) \leq 2^{2^{m+5}}$, $\ell n(\hat{w}_2) \leq 2^{2^{m+4}}$, such that for every $w \in \{0,1\}^{1+2^{m+1}}$,

$|\text{WORDS } (\alpha,w)|_\infty \underset{m(1+2^m)}{=} |\text{WORDS } (\hat{w}_1 \cdot \hat{w}_2^\omega, w)|_\infty$ and such that

$\alpha \restriction N_{2^m} = w_1 \cdot w_2^\omega \restriction N_{2^m}$.

Proof Using Lemma 3.18, we may assume that $\alpha = w_1 \cdot w_2^\omega$, for $w_1, w_2 \in \{0,1\}^*$. We apply Lemma 3.23 first with $\gamma = w_1$ and $k = m(2^m+1)$, obtaining $\hat{w}_1$ such that

$\ell n(\hat{w}_1) \leq (1+m(2^m+1))\left(2^{(1+2^{m+1})}+2^{m+1}\right)(1+2^{m+1})\left(3+2^{(1+2^{m+1})}\right) \leq 2^{2^{m+5}}$

We then apply Lemma 3.23 with $\gamma = w_2$ and $k = 0$, obtaining $\hat{w}_2$ such that $\ell n(\hat{w}_2) \leq \left(2^{(1+2^{m+1})}+2^{m+1}\right)(1+2^{m+1})\left(3+2^{(1+2^{m+1})}\right) \leq 2^{2^{m+4}}$.

Now let $w \in \{0,1\}^{1+2^{m+1}}$. By Lemma 3.23, $|\text{WORDS } (w_1,w)| \underset{m(1+2^m)}{=} |\text{WORDS } (\hat{w}_1,w)|$, and $|\text{WORDS } (w_2,w)| \underset{0}{\equiv} |\text{WORDS } (\hat{w}_2,w)|$. Since w_1 is m-end equivalent to $\hat{w}_1$, and w_2 is m-end equivalent to $\hat{w}_2$, it is an easy matter to verify $|\text{WORDS } (w_1 \cdot w_2^\omega, w)|_\infty \underset{m(1+2^m)}{=} |\text{WORDS } (\hat{w}_1 \cdot \hat{w}_2^\omega, w)|_\infty$. Since w_1 is m-end equivalent to $\hat{w}_2$, it is also easy to verify $\alpha \restriction N_{2^m} = \hat{w}_1 \cdot \hat{w}_2^\omega \restriction N_{2^m}$. □

We now wish to apply these results in the theory of a 1-1 unary function with a monadic predicate.

For $m \in N$, and $A \in$ 11FM, we claim there is $\mathcal{B} \in$ 11FM with

1. for each $j \in I_{2^{m+1}}$, at most $2^j[m(2^m+1)+1]$ loops of size j, and

2. at most $2^{(1+2^{m+1})}[m(2^m+1)+1]$ one-sided chains, such that for each one-sided chain O, the infinite sequence seq($\mathcal{B}$,O) obtained by restricting the distinguished subset C of $\mathcal{B}$ to O, as previously defined, actually equals $w_1 \cdot w_2^{\omega}$, for some $w_1, w_2 \in \{0,1\}^*$, with $\ell n(w_1) \leq 1+2^{m+1}+2^{2^{m+5}}$, $\ell n(w_2) \leq 2^{2^{m+4}}$, and

3. at most $2^{(1+2^{m+1})}[(m)(2^m+1)+1]$ loops of size $1+2^{m+1}$,

such that A is m,o equivalent to $\mathcal{B}$.

To prove the claim, by our previous remarks, we may assume there is $\mathcal{B} \in$ 11FM such that A is m,o equivalent to $\mathcal{B}'$, and the components of $\mathcal{B}'$ are numerically as described above.

For each of the one-sided chains which are components of $\mathcal{B}'$, we consider the distinguished subset C' of $\mathcal{B}'$ restricted to that one-sided chain. Now, membership in C' restricted to the elements on this one-sided chain O whose distance from its origin org is greater than 2^{m+1} yield an infinite sequence $\mathcal{B}(0)$ in the obvious manner. That is, for $n \in N$, we let $\mathcal{B}(0)(n) = 1 \iff f^{1+2^{m+1}+n}(org) \in C'$. By Lemma 3.24, there is an infinite sequence $\mathcal{B}'(0) = \hat{w}_1 \cdot \hat{w}_2^{\omega}$, with $\hat{w}_1, \hat{w}_2 \in \{0,1\}^*$ such $\ell n(\hat{w}_1) \leq 2^{2^{m+5}}$, $\ell n(\hat{w}_2) \leq 2^{2^{m+4}}$, $\beta(0) \upharpoonright N_{2^m} = \beta'(0) \upharpoonright N_{2^m}$, and such that

$$|\text{WORDS}(\beta(0),w)|_{\infty} \underset{m(1+2^m)}{=} |\text{WORDS}(\beta'(0),w)|_{\infty}.$$

We now define $\mathcal{B}$ as follows:

$\mathcal{B} = \langle B', f, C \rangle$, where for $a \in B'$, $a \in C$ iff

1. C_a is not a one-sided chain, and $a \in C'$, or
2. C_a is a one-sided chain, $d(o_a,a) < 1+2^{m+1}$, and $a \in C'$, or
3. C_a is a one-sided chain, $\beta'(C_a)$ is the ω-sequence obtained from $\beta(C_a)$, as above, $d(o_a,a) \geq 1+2^{m+1}$, and $\beta'(C_a)(d(o_a,a)-1-2^{m+1}) = 1$.

It is then an easy matter to verify for any m-word w,

$$|\text{WORDS}(\mathcal{B}',w)|_\infty \underset{m(1+2^m)}{=} |\text{WORDS}(\mathcal{B},w)|_\infty.$$

Since we have $|\text{WORDS}(A,w)|_\infty \underset{m(1+2^m)}{=} |\text{WORDS}(\mathcal{B}',w)|_\infty$, the claim is established.

Definition 3.25 For $m \in N$, let $\text{limit}(m) = 2^{(1+2^{m+1})[m(2^m+1)+1]}$

We define $h: N \to \{0,1,\#\}^* \cup \{\infty\}$ as follows:

$$h(m) = (1\#)^{\text{limit}(m)}(1^2\#)^{\text{limit}(m)}(1^3\#)^{\text{limit}(m)}\ldots \left(1^{2^{m+1}}\#\right)^{\text{limit}(m)}\#\left(1^{(1+2^{m+1}+2^{2^{m+5}})}\#1^{2^{2^{m+4}}}\#\right)^{\text{limit}(m)} \left(\#1^{(1+2^{m+1})}\right)^{\text{limit}(m)}$$

where for any $k \in I$, 1^k denotes $\underbrace{1\ 1\ldots 1}_{k} \in \{1\}^*$.

It should be clear from the claim above and our definition of norm that for any $m \in N$ and $A \in$ 11FM there is $\mathcal{B} \in$ 11FM such that $A\ E_{m,0}\ \mathcal{B}$ and $\mathcal{B} \preceq h(m)$.

Definition 3.26 For $n,k,k_1,k_2,k_3 \in N$, $e \in \{0,1\}$

$H(n,k,brep(k_1)\#brep(k_2)\#brep(k_3)\#e) =$

$$brep(max(k_1,2^{1+2^{n+k+1}}(n+k+1(2^{n+k+1}+1))\#brep(1+2^{n+k+1})\#$$

$$brep(k_3+1+2^{n+k+1}+2^{2^{n+k+5}})\#1.$$

For all other $m \in \{0,1,\#\}^* \cup \{\infty\}$, $H(n,k,m) = \infty$.

It then suffices to show

Theorem 3.27 For all $n,k \in N$, $A,B \in$ llFM, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$ and $m \in \{0,1,\#\}^* \cup \{\infty\}$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$ and $b_i \preceq m$ for all $i \in I_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $b_{k+1} \preceq H(n,k,m)$ and $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

The proof is left to the reader.

We thus have verified all the steps necessary to conclude by Theorem 4 of Chapter 2 llFM is (h,H) bounded.

Theorem 3.28 For H as in Definition 3.26, and h as in Definition 3.25, the class of structures in llFM is (h,H) bounded

By a series of steps similar to that in the case of llF, we obtain

Theorem 3.29 For some constant $c > 0$, whether a sentence F of length n is in SAT(llFM) can be decided within space $2^{2^{cn}}$.

As in llF, we also obtain as corollary

Corollary 3.30 For some constants c,d > 0, whether a sentence F of length n is in SAT(11FM) can be decided nondeterministically within space $2^{2^{cn}}$ and, simultaneously, time $2^{n \cdot d^n}$ (and hence time $2^{2^{dn}}$ for some constant d).

We claim our method also applies to the theory of a 1-1 unary function with k monadic predicates, for $k \in I$; we let the reader convince himself that this is indeed the case.

We now consider the theory of one successor with a monadic predicate, that is, the theory of the class MONN of models $\{<N,',M>|M \subseteq N\}$, where

$': N \to N$ is such that for all $n \in N$, $n' = n+1$. This is exactly the same as the theory of a single one-sided chain with a monadic predicate.

Now, the results of this section show that for all $n,k \in N$, $E_{n,k}$, as given in Definitions 3.6 and 3.7, and restricted to the models in 11FM consisting of a single one-sided chain, is a refinement of the Ehrenfeucht equivalence relation $\equiv_{n,k}$ for the theory of a single one-sided chain with a monadic predicate. We can thus apply the results of this section to this theory, thereby obtaining

Theorem 3.31 For some constant c > 0, whether a sentence F of lengt n is in SAT(MONN) can be decided in nondeterministic time $2^{2^{cn}}$.

Since by Lemma 2.2 of Chapter 9, for any $k \in I$, the theory T_k of $<N,'>$ with k monadic predicates is such that $T_k \leq_{p\ell} T_1$, we obtain a decision procedure, and an upper bound of the same order on the procedure, for T_k as for T_1.

Section 4: Upper Bounds for the Theory of Two Successors and Equal Length

We next apply our technique to the theory of two successors and equal length. We consider the model $2SEL = \langle \{0,1\}^*, r_0, r_1, E\ell n \rangle$, where for $a,b \in \{0,1\}^*$,

$$r_0(a) = a \cdot 0$$
$$r_1(a) = a \cdot 1, \text{ and}$$
$$E\ell n(a,b) \iff \ell n(a) = \ell n(b).$$

Again, to apply the results of Chapter 2 we must consider a language with only relation symbols in it, so let L be a first-order language with three binary relation symbols in it. For any $a,b \in \{0,1\}^*$, we interpret the assertion in the language L that the first binary relation holds for the pair a,b as $r_0(a) = b$, that the second binary relation holds for the pair a,b as $r_1(a) = b$, and that the third binary relation holds for the pair a,b as $E\ell n(a,b)$.

Since an analysis of structures is obviously unnecessary in this case, we move towards a characterization of the information that quantifier-depth n formulas can provide about k elements. We will show that quantifier-depth n formulas with k free variables $\bar{z}_k$ can give the symbols in z_i, if z_i is of bounded length depending on n and k, and can give the suffixes, of length up to a certain bound depending on n and k, such that z_i minus a suffix equals z_j minus a suffix; such formulas can also give the difference, up to a certain bound depending on n and k, between $\ell n(z_i)$ and $\ell n(z_j)$. We make this precise in our definition of $E_{n,k}$ below.

Definition 4.1 Let $a,w_1 \in \{0,1\}^*$. a/w_1 denotes that $w \in \{0,1\}^*$, if it exists, such that $a = w \cdot w_1$.

Definition 4.2 We use the notation $a_1/w_1 \stackrel{=}{ex} a_2/w_2$ to mean

a_1/w_1 exists $\iff$ a_2/w_2 exists,

and if a_1/w_1 exists, a_1/w_1 equals a_2/w_2.

We use the notation $a_1/w_1 = a_2/w_2$ to mean

a_1/w_1 and a_2/w_2 exist, and

a_1/w_1 equals a_2/w_2.

Thus $a_1/w_1 = a_1/w_1$ iff a_1/w_1 exists. We collect here some useful facts concerning /, whose easy verification is left to the reader.

Lemma 4.3 Let $a_1,a_2,w_1,w_2 \in \{0,1\}^*$.

1. For all $n \in N$ such that $n \leq \ell n(a)$, there is a unique $w \in \{0,1\}^n$ such that a/w exists.
2. $a/w_1 \cdot w_2 \stackrel{=}{ex} (a/w_2)/w_1$.
3. $a_1/w_1 = a_2/w_2 \Rightarrow$ for any $a,w_3,w_4 \in \{0,1\}^*$, with $\ell n(w_4) \geq \ell n(w_2)$, $a/w_3 = a_1/(w_4/w_2) \cdot w_1 \iff a/w_3 = a_2/w_4$.

Definition 4.4 Let $n \in N$, $k \in I$. Let $\bar{a}_k,\bar{b}_k \in (\{0,1\}^*)^k$. We define $\bar{a}_k \; E_n \; \bar{b}_k$ iff for all $i,j \in I_k$, for all $w_1,w_2 \in \{0,1\}^{\leq 2^{(3\cdot n)+k}}$,

1. $a_i/w_1 = a_j/w_2 \iff b_i/w_1 = b_j/w_2$, and,

2. $a_i = w_1 \iff b_i = w_1$, and

3. for all integers ℓ, $|\ell| \leq 2^{(3\cdot n)+k}$,
$\ell n(a_i) = \ell n(a_j)+\ell \iff \ell n(b_i) = \ell n(b_j)+\ell$.

Clearly, $E_{n,k}$ is an equivalence relation of finite index. We wish to show for all $n \in N$, and $k \in I$, that $E_{n,k}$ is a refinement of $\underset{n}{\equiv}_{,k}$.

To accomplish this, we first show for all $k \in I$, $E_{0,k}$ is a refinement of $\underset{0}{\equiv}_{,k}$.

For all $\bar{a}_k, \bar{b}_k \in (\{0,1\}^*)^k$, if $\bar{a}_k \; E_0 \; \bar{b}_k$, then for all $i,j \in I_k$, $a_i/0 = a_j \iff b_i/0 = b_j$ and $a_i/1 = a_j \iff b_i/1 = b_j$. Thus $a_i = r_0(a_j) \iff b_i = r_0(b_j)$, and $a_i = r_1(a_j) \iff b_i = r_1(b_j)$. Again, by $\bar{a}_k \; E_0 \; \bar{b}_k$, for all $i,j \in I_k$, $\ell n(a_i) = \ell n(a_j) \iff \ell n(b_i) = \ell n(b_j)$, and thus $E\ell n(a_i,a_j) \iff E\ell n(b_i,b_j)$. Using Lemma 5 of Chapter 2, we conclude $\bar{a}_k \underset{0}{\equiv} \bar{b}_k$.

Lemma 4.5 For all $k,n \in N$, all $\bar{a}_k, \bar{b}_k \in (\{0,1\}^*)^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all a_{k+1} there is b_{k+1} such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

Proof Let $a_{k+1} \in \{0,1\}^*$. Our choice of b_{k+1} depends on the properties of a_{k+1} and so we proceed by cases.

Case 1. For some $j \in I_k$, there are $w_1, w_2 \in \{0,1\}^{\leq 2^{(3\cdot n)+k+1}}$ such that $a_j/w_1 = a_{k+1}/w_2$.

Since by the first clause of the definition of $E_{n+1,k}$, $[a_j/w_1 = a_j/w_1 \iff b_j/w_1 = b_j/w_1]$, we infer b_j/w_1 exists. We let $b_{k+1} = (b_j/w_1)\cdot w_2$. Note then $b_j/w_1 = b_{k+1}/w_2$. We now

verify this choice of b_{k+1} satisfies clauses 1, 2 and 3 of the definition of $E_{n,k+1}$.

We first verify clause 1. By hypothesis, for all $i,\ell \in I_k$, and all $w_3,w_4 \in \{0,1\}^{\leq 2^{(3\cdot(n+1))+k}}$, and thus for all $w_3,w_4 \in \{0,1\}^{2^{(3\cdot n)+(k+1)}}$, $a_i/w_3 = a_\ell/w_4 \iff b_i/w_3 = b_\ell/w_4$. It will thus be sufficient to verify, for all $i \in I_{k+1}$, and all $w_3,w_4 \in \{0,1\}^{\leq 2^{(3\cdot n)+k+1}}$, $a_i/w_3 = a_{k+1}/w_4 \iff b_i/w_3 = b_{k+1}/w_4$.

We first suppose $i \in I_k$. Suppose $\ell n(w_2) \leq \ell n(w_4)$. Then by Lemma 4.3.3, $a_i/w_3 = a_{k+1}/w_4 \iff a_i/w_3 = a_j/(w_4/w_2)\cdot w_1$. Figure 4.6 illustrates this case by presenting a pictorial representation of a subtree of $\{0,1\}^*$ containing a_i, a_j, a_{k+1}, and a_i/w_3.

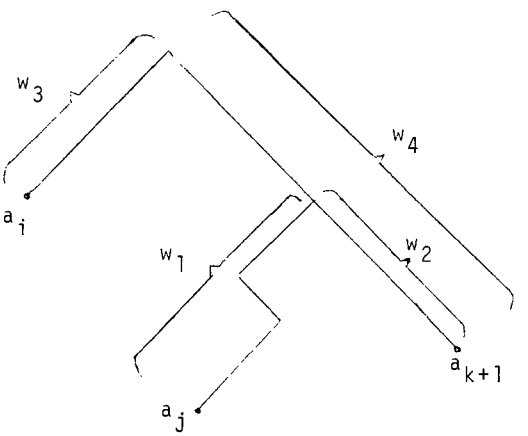

Figure 4.6

Now if $(w_4/w_2)\cdot w_1$ exists, $\ell n((w_4/w_2)\cdot w_1) \leq 2\cdot 2^{(3n+k+1)} \leq 2^{3(n+1)+k}$ Thus, by the first clause of the definition of $E_{n+1,k}$, by considering the cases when $(w_4/w_2)\cdot w_1$ exists and doesn't exist, we have

$$a_i/w_3 = a_j/(w_4/w_2)\cdot w_1 \iff b_i/w_3 = b_j/(w_4/w_2)\cdot w_1.$$

Since $b_j/w_1 = b_{k+1}/w_2$, by Lemma 4.3.3 we have

$$b_i/w_3 = b_j/(w_4/w_2)\cdot w_1 \iff b_i/w_3 = b_{k+1}/w_4.$$

Altogether, we have that if $\ell n(w_2) \leq \ell n(w_4)$,

$a_i/w_3 = a_{k+1}/w_4 \iff b_i/w_3 = b_{k+1}/w_4$.

If $\ell n(w_4) \leq \ell n(w_2)$, the argument is similar and is left to the reader.

To complete the verification of clause 1, it is sufficient to show for $w_3 \in \{0,1\}^{2^{(3n+k+1)}}$,

$a_{k+1}/w_3 = a_{k+1}/w_3 \iff b_{k+1}/w_3 = b_{k+1}/w_3$. Since $a_j/w_1 = a_{k+1}/w_2$, and $\ell n(w_2) \leq \ell n(w_3)$, by Lemma 4.3.3, $a_{k+1}/w_3 = a_j/(w_3/w_2)\cdot w_1 \iff a_{k+1}/w_3 = a_{k+1}/w_3$. Since $b_j/w_1 = b_{k+1}/w_2$, and $\ell n(w_2) \leq \ell n(w_3)$ by Lemma 4.3.3 again, we have $b_{k+1}/w_3 = b_j/(w_3/w_2)\cdot w_1 \iff b_{k+1}/w_3 = b_{k+1}/w_3$. But since if $(w_3/w_2)\cdot w_1$ exists, $\ell n((w_3/w_2)\cdot w_1) \leq 2^{(3(n+1))+k}$, by considering cases and using the hypothesis,

$$a_j/(w_3/w_2)\cdot w_1 \text{ exists} \iff b_j/(w_3/w_2)\cdot w_1 \text{ exists}.$$

It is then easy to see

$$a_{k+1}/w_3 = a_{k+1}/w_3 \iff b_{k+1}/w_3 = b_{k+1}/w_3.$$

Clause 1 is thus verified.

We next verify clause 2. It will be sufficient to verify, for all $w \in \{0,1\}^{\leq 2^{(3n)+k+1}}$, $a_{k+1} = w \iff b_{k+1} = w$. Now, $a_{k+1} = w \iff a_j = (w/w_2)\cdot w_1$, since $a_j/w_1 = a_{k+1}/w_2$. But $\ell n((w/w_2)\cdot w_1) \leq 2^{3(n+1)+k}$, so by hypothesis, $a_j = (w/w_2)\cdot w_1 \iff b_j = (w/w_2)\cdot w_1$. Since $b_j/w_1 = b_{k+1}/w_2$, $b_j = (w/w_2)\cdot w_1 \iff b_{k+1} = w$.
Clause 2 is thus verified.

To verify clause 3, it is sufficient to verify for all $i \in I_k$, and any integer ℓ, $|\ell| \leq 2^{(3n)+k+1}$, $\ell n(a_i) = \ell n(a_{k+1})+\ell \iff \ell n(b_i) = \ell n(b_{k+1})+\ell$. Clearly, for some integer m, $|m| \leq 2^{(3n)+k+1}$, $\ell n(a_{k+1}) = \ell n(a_j)+m$, and $\ell n(b_{k+1}) = \ell n(b_j)+m$. Thus for any integer ℓ, $\ell n(a_i) = \ell n(a_{k+1})+\ell \iff \ell n(a_i) = \ell n(a_j)+m+\ell$. Now, if $|\ell| \leq 2^{(3n)+k+1}$, so $|\ell+m| \leq 2^{(3(n+1))+k}$, by hypothesis, $\ell n(a_i) = \ell n(a_j)+m+\ell \iff \ell n(b_i) = \ell n(b_j)+m+\ell$. Since $\ell n(b_{k+1}) = \ell n(b_j)+m$, $\ell n(b_i) = \ell n(b_j)+m+\ell \iff \ell n(b_i) = \ell n(b_{k+1})+\ell$
We thus have for all $|\ell| \leq 2^{(3n)+k+1}$, $\ell n(a_i) = \ell n(a_{k+1})+\ell \iff \ell n(b_i) = \ell n(b_{k+1})+\ell$.
Clause 3 is thus verified.

Case 2. $\ell n(a_{k+1}) \leq 2^{(3n)+k+1}$.
We choose $b_{k+1} = a_{k+1}$. The proof that clauses 1, 2 and 3 of the definition of $E_{n,k+1}$ hold for this choice of b_{k+1} is routine and is left to the reader.

Case 3. $\ell n(a_{k+1}) > 2^{(3n)+k+1}$, and for all $i \in I_k$, there are no $w_1, w_2 \in \{0,1\}^{\leq 2^{(3n)+k+1}}$ such that $a_i/w_1 = a_{k+1}/w_2$.

Subcase a. For all $i \in I_k$, there is no integer ℓ, $|\ell| \leq 2^{(3n)+k+1}$, such that $\ell n(a_{k+1}) = \ell n(a_i)+\ell$.

If $k \geq 1$, let b_j, for $j \in I_k$, be such that for all $i \in I_k$, $\ell n(b_i) \leq \ell n(b_j)$. If $k = 0$, let $b_j = \lambda$. By Lemma 4.3.1, since $\ell n(a_{k+1}) > 2^{(3n)+k+1}$, there is a unique $w \in \{0,1\}^{2^{(3n)+k+1}}$ such that a_{k+1}/w exists. We let $b_{k+1} = b_j 0 w$. It is then an easy matter to verify clauses 1, 2 and 3 of the definition of $E_{n,k+1}$ hold.

Subcase b. There is $j \in I_k$, and an integer ℓ, $|\ell| \leq 2^{(3n)+k+1}$, such that $\ell n(a_{k+1}) = \ell n(a_j)+\ell$. Using Lemma 4.3.1, let $w \in \{0,1\}^{2^{(3n)+k+1}}$ be that unique word of length $2^{(3n)+k+1}$ such that a_{k+1}/w exists. The proof now proceeds by considering the size of $\ell n(a_{k+1}/w)$.

If $\ell n(a_{k+1}/w) < 2^{n+k}$, then $\ell n(a_{k+1}) < 2^{n+k}+2^{(3n)+k+1} \leq 2^{(3n)+k+2}$. We let $b_{k+1} = a_{k+1}$. The proof then follows from the fact that by hypothesis, for all $i \in I_k$, if $\ell n(a_i)$, $\ell n(b_i) \leq 2^{3(n+1)+k}$, then $a_i = b_i$.

If $\ell n(a_{k+1}/w) \geq 2^{n+k}$, let $Lev = \{b \in \{0,1\}^* \mid \ell n(b) = \ell n(b_j)+\ell-2^{(3n)+k+1}\}$. Note that $\ell n(a_{k+1}/w) = \ell n(a_j)+\ell-2^{(3n)+k+1} \geq 2^{n+k}$, and thus

$\ell n(a_j) \geq -\ell+2^{(3n)+k+1}+2^{n+k}$. Since $-\ell+2^{(3n)+k+1}+2^{n+k} \leq 2^{3(n+1)+k}$, by clause 2 of the definition of $E_{n+1,k}$ we must have $\ell n(b_j) \geq -\ell+2^{(3n)+k+1}+2^{n+k}$. Thus $\ell n(b_j)+\ell-2^{(3n)+k} \geq 2^{n+k}$. Thus Lev is of cardinality at least $2^{2^{n+k}}$. Let $Lev(w) = \{b \cdot w | b \in Lev\}$. Then Lev(w) is of cardinality at least $2^{2^{n+k}}$, since Lev is. We claim there is $b_{k+1} \in Lev(w)$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

For any $c \in Lev(w)$, we denote by Nbhd(c) the set $\{a \in \{0,1\}^* | \text{there are } w_1, w_2 \in \{0,1\}^{\leq 2^{(3n)+k+1}} \text{ such that } c/w_1 = a/w_2$

We note that for $c,c' \in Lev(w)$, if $c \neq c'$, then $Nbhd(c) \cap Nbhd(c') = \phi$. This follows from the fact that $\ell n(c/w) = \ell n(c'/w)$ but $c/w \neq c'/w$. Thus if for some $i \in I_k$, and for some $c \in Lev(w)$, $b_i \in Nbhd(c)$, then $b_i \notin Nbhd(c')$ for all $c' \in Lev(w)$ with $c' \neq c$. Thus $|\{c \in Lev(w) | \text{for some } i \in I_k, b_i \in Nbhd(c)\}| \leq k$.

Now $k < 2^{2^{n+k}}$, and since $|Lev(w)| \geq 2^{2^{n+k}}$, $|\{c \in Lev(w) | \text{for all } i \in I_k, b_i \notin Nbhd(c)\}| > 0$. We let b_{k+1} be any element of $\{c \in Lev(w) | \text{for all } i \in I_k, b_i \notin Nbhd(c)\}$. It is then a trivial exercise to verify $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$. □

Corollary 4.7 For all $k,n \in N$, all $\bar{a}_k, \bar{b}_k \in (\{0,1\}^*)^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all a_{k+1} there is b_{k+1} such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$, and either

1. $\ell n(b_{k+1}) \leq 2^{(3n)+k+1}$, or
2. for some integer ℓ, $|\ell| \leq 2^{(3n)+k+1}+1$ and some $i \in I_k$, $\ell n(b_{k+1}) = \ell n(b_i)+\ell$.

Proof Examination of our choice of b_{k+1} in each case shows b_{k+1} does satisfy the above. □

Corollary 4.7 motivates our definition of norm and H:

Definition 4.8 For $w \in \{0,1\}^*$, we define $\|w\| = \ell n(w)$. We take as our partial order on norms of elements in $\{0,1\}^*$ $\leq$, the usual less-than-or-equal-to order relation on integers.

Definition 4.9 For $n,k,m \in N$, we define $H(n,k,m) = m+2^{3n+k+1}+1$. It now follows easily from Corollary 4.7 that

Theorem 4.10 For all $n,k \in N$, $\bar{a}_k, \bar{b}_k \in (\{0,1\}^*)^k$, and $m \in N$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$ and $b_i \preceq m$ for all $i \in I_k$, then for all a_{k+1} there is b_{k+1} such that $b_{k+1} \preceq H(n,k,m)$ and $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

We have verified all steps necessary to conclude by Theorem 3 of Chapter 2 that

Theorem 4.11 For H as in definition 4.9, 2SEL is H-bounded.

Theorem 4.12 Let $Q_1x_1 \ldots Q_nx_n \; F(\bar{x}_n)$ be a sentence such that $F(\bar{x}_n)$ is quantifier-free. Let $m_0 = 0$, and let $m_{i+1} = m_i+2^{3(n-i)+i}+1$ for $i \in N_{n-1}$. Then

$$2SEL \models Q_1x_1 \ldots Q_nx_n \; F(\bar{x}_n) \iff$$
$$2SEL \models Q_1x_1 \preceq m_1 \ldots Q_nx_n \preceq m_n \; F(\bar{x}_n).$$

Proof Since 2SEL is H-bounded, we need only verify that $m_0 \preceq m_1 \preceq \ldots \preceq m_n$ and $H(n-i,i-1,m_{i-1}) \preceq m_i$, which follows easily. □

Corollary 4.13 For some constant $c > 0$, whether a sentence F of length n is in TH(2SEL) can be decided within space 2^{cn}.

Proof Using Theorem 5.3 of Chapter 1, convert F to an equivalent sentence $Q_1x_1 \ldots Q_nx_n\ G(\bar{x}_n)$ where G is quantifier-free and of length at most $n \log(n)$. By the previous theorem, F is in TH(2SEL) iff $2SEL \models Q_1x_1 \leq m_1 \ldots Q_nx_n \leq m_n\ G(\bar{x}_n)$, (where $m_0 = 0$, $m_{i+1} = m_i + 2^{3(n-i)+i+1}$ for $i \in N_{n-1}$). We let the reader convince themselves that the straightforward implementation of the procedure suggested by the above uses space at most 2^{cn} for some constant c. □

Corollary 4.14 For some constant $d > 0$, whether a sentence of length n is in $TH(\langle\{0,1\}^*, r_0, r_1\rangle)$ can be decided within space 2^{dn}.

Section 5: Upper Bounds for the Theories of <N,<>, Lexicographical Order, and Well-order

We now wish to apply our technique to the theories of <N,<>, lexicographical order, and well-order.

Let L be a first-order language with a single binary relation symbol. We mean by the theory of <N,<> the theory of the model <N,<> in the language L, where < is the usual binary relation of (strict) order on elements in N. By the theory of lexicographical order we mean the theory of the model $<\{0,1\}^*,\prec>$ in the language L, where $\prec$ is the lexicographical or dictionary order relation on words in $\{0,1\}^*$. By the theory of well-order, we mean the theory of the class of models $\{<\alpha,<> | \alpha$ is an ordinal number, < the usual order relation on ordinal numbers$\}$ in the language L. The goal of this section is to obtain decision procedures, and upper bounds on such decision procedures, for all three of these theories.

The way we accomplish this goal is as follows. For any ordinals α,β,γ, we define JOIN(α,β,γ) to be the model consisting of the set $J(\alpha,\beta,\gamma)$ of ordered pairs equal to

$\{(0,a) | a$ an ordinal, $a < \alpha\} \cup$

$\{(q,b) | q \in Q, 0 < q < 1, b$ an ordinal, $b < \beta\} \cup$

$\{(1,c) | c$ an ordinal, $c < \gamma\}$,

and the order relation on this set defined as follows:

for $(q,a), (r,b) \in J(\alpha,\beta,\gamma)$,

$(q,a) < (r,b) \iff q < r$, or $q = r$

and $a < b$.

Clearly JOIN(α,β,γ) is isomorphic to the ordered sum of the three orders α, the ordered product of β with η, and γ.

Our motivation for considering such models is the following: for each of the models $\langle N,<\rangle$, $\langle\{0,1\}^*,\frown\rangle$, and $\langle\alpha,<\rangle$ α an ordinal, there are ordinals $\gamma_1,\gamma_2,\gamma_3$ such that that model is isomorphic to JOIN($\gamma_1,\gamma_2,\gamma_3$). For the models $\langle N,<\rangle$, and $\langle\alpha,<\rangle$, this should be apparent: the proof that $\langle\{0,1\}^*,\frown\rangle$ is isomorphic to JOIN($\gamma_1,\gamma_2,\gamma_3$), for some ordinals $\gamma_1,\gamma_2,\gamma_3$, will be given below.

Let JOINS = {JOIN(α,β,γ)$|\alpha,\beta,\gamma$ ordinals}. We will in this case give the basic definitions and lemmas needed to prove our upper bound for JOINS, and omit many of the proofs. The interested reader is referred to [Fer74] for a fuller treatment. Then, using the fact that $\langle N,<\rangle$, $\langle\{0,1\}^*,\frown\rangle$, and $\langle\alpha,<\rangle$ are all isomorphic to some model in JOINS, we obtain decision procedures for each of these theories. With this motivation in mind, we proceed to outline a decision procedure for JOINS.

Our analysis of $\equiv_{n,k}$, or rather an appropriate refinement of it, will be as follows. We will show that quantifier-depth n formulas with k free variables $\bar{z}_k$ can determine, for each ordinal number which defines the model, the coefficients, up to certain bounds depending on n, of a unique polynomial in ω (see below) of degree n which is the "end segment" of that ordina

Also such formulas can determine the order relation between z_i and z_j, and for each ordinal number which defines the model in JOINS isomorphic to the segment between z_i and z_j, can determine the coefficients up to certain bounds depending on n, of a unique polynomial in ω of degree n which is the "end segment" of the ordinal. We make this precise in our definition of $E_{n,k}$ below.

We first present some facts and definitions concerning orderings and ordinal numbers.

Consider the equivalence relation on linearly ordered sets defined by the notion of order-isomorphism. (That is, if A and B are ordered sets, A is equivalent to B iff A is order-isomorphic to B). We choose a representative from each equivalence class of this equivalence relation, and call it the order-type of any set in the equivalence class. (See [Kam 50] for a full development of the notion of order-type.) The order-types of well-ordered sets are called ordinal numbers. The ordinal numbers themselves form a well-ordered class determined by

$\alpha < \beta$ <=> there is a 1-1, order-preserving function from α strictly into β.

We assume the reader is familiar with the basic facts and definitions of ordinal arithmetic: that is, the definitions of ordinal addition, multiplication and subtraction, and the basic facts concerning these, e.g. the distributive law when the second factor is a sum. For a full review, see [Kam 50]. ω denotes

the order-type of the set N with the usual (less than) order.

We now define what we mean by a segment of a linear order. For this purpose, it is useful to introduce the symbols $+\infty$ and $-\infty$.

Definition 5.1 Let $<L,<_L>$ be any linear order. For any $a \in L$, we define $-\infty <_L a$, and $a <_L +\infty$. Let $b \in L \cup \{-\infty\}$, $c \in L \cup \{+\infty\}$. We define $[b,c)_L$, the segment from b to c in L, to be the order-type of the set $\{d \in L | b \leq_L d <_L c\}$ with the order relation $<_L$ restricted to this set.

Lemma 5.2 For all ordinals γ,β, with $\beta > 0$, there are unique ordinals ε,ε' such that

$$\gamma = \beta \cdot \varepsilon' + \varepsilon$$

with $\varepsilon < \beta$.

Lemma 5.3 Let $n \in N$, α an ordinal. α can be expressed uniquely as $\omega^n \cdot \alpha_n + \omega^{n-1} \cdot \alpha_{n-1} + \ldots + \alpha_0$, where for $i \in N_{n-1}$, α_i is a finite ordinal.

Lemma 5.3 is proved by repeated application of Lemma 5.2 for a proof of Lemma 5.2, see [Kam 50].

Fact 5.4 If $\alpha = \omega^n \alpha_n + \ldots + \alpha_0$, and $\beta = \omega^n \beta_n + \ldots + \beta_0$, such that for $i \in N_{n-1}$, α_i and β_i are finite ordinals, and $\beta_n < \alpha_n$, then $\alpha-\beta$ exists and equals $\omega^n(\alpha_n-\beta_n) + \omega^{n-1}\alpha_{n-1} + \ldots + \alpha_0$.

Fact 5.5 If $\alpha = \omega^n \alpha_n + \ldots + \alpha_0$, $\beta = \omega^n \beta_n + \ldots + \beta_0$, such that for $i \in N_{n-1}$, α_i and β_i are finite ordinals, then [$\alpha > \beta$ <=> for some $j \in N_n$, $\alpha_j > \beta_j$, and for all i, $j < i \leq n$, $\alpha_i = \beta_i$].

We now extend our definition of $\equiv_n$ to include ordinal numbers.

Definition 5.6 Let n be an integer, α any ordinal number. We define

$$[\alpha]_n = \begin{cases} \alpha & \text{if } \alpha \leq n \\ n+1 & \text{otherwise.} \end{cases}$$

For n an integer, α,β ordinals, we define

$$\alpha \equiv_n \beta \text{ iff } [\alpha]_n = [\beta]_n.$$

We remark that this definition is consistent with our Definition 1.3 of $\equiv_n$, and that $\equiv_n$ restricted to the class of ordinals is an equivalence relation. For later use, we state the following; the proof is an easy argument by cases.

Lemma 5.7 Let $n \in N$, $\alpha,\beta,\gamma,\alpha'$ ordinal numbers. Suppose $\alpha = \beta + \gamma$, $\gamma \neq 0$ and $\alpha \equiv_{2^{n+1}-1} \alpha'$. Then $[\beta]_{2^n-1} + [\gamma]_{2^n} \equiv_{2^n} \alpha'$.

We also remark that the analogue of Lemma 1.4, which we state below, holds for $\equiv_n$ restricted to the ordinals; the proof is similar to that of Lemma 1.4 and is left to the reader.

Lemma 5.8 Let γ,δ be ordinals, $n \in N$, k any integer with $|k| \leq \gamma$, $|k| \not\leq \delta$. Suppose $\gamma \equiv_n \delta$. Then $\gamma+k \equiv_{n-|k|} \delta+k$.

Definition 5.9 Let $n \in N$, α,β ordinal numbers. Suppose $\alpha = \omega^n\alpha_n + \ldots + \alpha_0$, and $\beta = \omega^n\beta_n + \ldots + \beta_0$, such that for all $i \in N_{n-1}$, α_i and β_i are finite ordinals. (Note then, by Lemma 5.3, these representations are unique). We define $\alpha \cong_n \beta$ iff

1. $\alpha_n \overset{=}{0} \beta_n$, and

2. for all $i \in N_{n-1}$, $\alpha_{i\ 2^{n-1}} \overset{=}{} \beta_i$.

We collect here some facts about $\overset{\simeq}{n}$, whose verification is left to the reader.

Lemma 5.10 Let $n \in N$, $\alpha,\beta,\gamma,\delta$ ordinals.

1. $\overset{\simeq}{n}$ is an equivalence relation on the class of ordinal numbers.

2. $\alpha\ _{n+1}\tilde{} \beta \Rightarrow \alpha \overset{\simeq}{n} \beta$.

3. $\alpha \overset{\simeq}{n} \beta$ and $\gamma \overset{\simeq}{n} \delta \Rightarrow \alpha+\gamma \overset{\simeq}{n} \beta+\delta$.

Definition 5.11 Let $n,j \in N$. We define $\omega\text{-POLY}(n,j) = \{\alpha \mid \alpha = \omega^n a_n + \ldots + a_0$, such that for all $i \in N_n$, $a_i \in N_j\}$.

We remark that $\omega\text{-POLY}(n,2^n)$ is a set of representatives for the equivalence classes of the equivalence relation $\overset{\simeq}{n}$.

We now apply these results to JOINS.

Definition 5.12 For ordinals α,β,γ, $\alpha + \beta\eta + \gamma$ denotes the order-type of $JOIN(\alpha,\beta,\gamma)$.

We now extend $\overset{\simeq}{n}$ to the class of order-types of models in JOINS.

Definition 5.13 Let $n \in N$, α_i,β_i ordinals for $i = 1,2,3$. If $\alpha_2 = 0$ and $\beta_2 = 0$ we define $\alpha_1 + \alpha_2 \cdot \eta + \alpha_3 \overset{\simeq}{n} \beta_1 + \beta_2\eta + \beta_3$ iff $(\alpha_1+\alpha_3) \overset{\simeq}{n} (\beta_1+\beta_3)$. Otherwise, we define $\alpha_1 + \alpha_2\eta + \alpha_3 \simeq \beta_1 + \beta_2\eta + \beta_3$ iff $\alpha_i \overset{\simeq}{n} \beta_i$ for $i = 1,2,3$.

We note that if α_i, β_i are ordinals for i = 1,2,3, such that $\alpha_1 + \alpha_2\eta + \alpha_3 = \beta_1 + \beta_2\eta + \beta_3$, then either $\alpha_2 = \beta_2 = 0$ and $\alpha_1 + \alpha_3 = \beta_1 + \beta_3$, or $\alpha_2 \neq 0$ and $\alpha_i = \beta_i$ for i = 1,2,3. Thus $\tilde{n}$ is a well-defined equivalence relation on $\{\alpha + \beta\eta + \gamma | \alpha,\beta,\gamma \text{ ordinals}\}$.

We remark that any segment of an element of JOINS is the order-type of some element in JOINS. In fact for later use, we prove the following result which explicitly specified these order types:

Lemma 5.14 Let $A \in$ JOINS, and let $c \in A \cup \{-\infty\}$, $d \in A \cup \{+\infty\}$. Suppose $[c,d)_A = \alpha_1 + \alpha_2\eta + \alpha_3$ for some ordinals $\alpha_1,\alpha_2,\alpha_3$. Let $b \in A$ be such that $c \leq b < d$. Then one of the following must hold.

1. $\alpha_2 = 0$, $[c,b)_A$ is an ordinal $< \alpha_1 + \alpha_3$, and $[b,d)_A = (\alpha_1 + \alpha_3) - [c,b)_A$.
2. $\alpha_2 \neq 0$, $[c,b)_A$ is an ordinal $< \alpha_1$, and $[b,d)_A = (\alpha_1 - [c,b)_A) + \alpha_n\eta + \alpha_3$.
3. $\alpha_2 \neq 0$, $[c,b)_A = \alpha_1 + \alpha_2\eta + \delta$ for some ordinal $\delta < \alpha_2$, and $[b,d)_A = (\alpha_2 - \delta) + \alpha_2\eta + \alpha_3$.
4. $\alpha_2 \neq 0$, $[c,b)_A = \alpha_1 + \alpha_2\eta + \delta$, for some ordinal $\delta < \alpha_3$, and $[b,d)_A = \alpha_3 - \delta$.

Proof When $\alpha_2 \neq 0$, Case 2 occurs when either

1. $c = -\infty$, $b = (0,b')$ for some $b' < \alpha_1$, and $d = +\infty$ or $\pi_1(d) > 0$*, or

---

* π_i is the i-th projection function.

2. $c \in A$, $b = (\pi_1(c),b')$ for some $b' < \alpha_1$, and
$d = +\infty$ or $\pi_1(d) > \pi_1(c)$.

Case 3 occurs when $b = (q,b')$ for some $0 < q < 1$ and $b' < \alpha_2$, such that if $c \in A$, $\pi_1(c) < q$, and if $d \in A$, then $\pi_1(d) > q$.
Case 4 occurs when $b = (q,b')$ for some $b' < \alpha_3$ and either

1. $q = 1$, $d = +\infty$, and if $c \in A$, $\pi_1(c) < q$, or
2. $0 < q < 1$, $\pi_1(d) = q$, and if $c \in A$, $\pi_1(c) < q$.

The rest of the verification is left to the reader. □

Definition 5.15 Let $n \in N$, $JOIN(\alpha_1,\alpha_2,\alpha_3)$, and $JOIN(\beta_1,\beta_2,\beta_3) \in JOINS$. We define $JOIN(\alpha_1,\alpha_2,\alpha_3)\ E_{n,0}\ JOIN(\beta_1,\beta_2,\beta_3)$ iff $\alpha_1 + \alpha_2\eta + \alpha_3 \tilde{\tilde{n}} \beta_1 + \beta_2\eta + \beta_3$.

Definition 5.16 Let $n \in N$, $k \in I$. Let $JOIN(\alpha_1,\alpha_2,\alpha_3)$, $JOIN(\beta_1,\beta_2,\beta_3) \in JOINS$, $\bar{a}_k \in (J(\alpha_1,\alpha_2,\alpha_3))^k$, $\bar{b}_k \in (J(\beta_1,\beta_2,\beta_3))^k$. We define $\bar{a}_k\ E_n\ \bar{b}_k$ iff

1. for all $i,j \in I_k$, $a_i < a_j \iff b_i < b_j$.
2. Letting c_j, for $j \in I_k$ (respectively, d_j, for $j \in I_k$) denote the jth smallest of the a_i, for $i \in I_k$ (respectively, b_i, for $i \in I_k$), and letting $c_0 = d_0 = -\infty$ and $c_{k+1} = d_{k+1} = +\infty$,

then for any $i \in N_k$,

$$[c_i,c_{i+j})_{J(\alpha_1,\alpha_2,\alpha_3)} \tilde{\tilde{n}} [d_i,d_{i+1})_{J(\beta_1,\beta_2,\beta_3)}.$$

Since, as previously remarked, every segment of a model in JOINS is equal to $\alpha' + \beta'\eta + \gamma'$, for some ordinals α',β',γ', it should be clear that $E_{n,k}$ is a well-defined equivalence relation of finite index.

We outline a proof that we can restrict ourselves to structures $JOIN(\alpha_1,\alpha_2,\alpha_3)$ where α_i, for $1 \leq i \leq 3$, is in ω-POLY(m,ℓ) for some numbers m,ℓ. We therefore define our notion of norm as follows:

Definition 5.17 An Ehrenfeucht Structure for JOINS is any structure $JOIN(\alpha_1,\alpha_2,\alpha_3)$ such that $\alpha_i \in \omega$-POLY(m,ℓ) for some m,$\ell \in I$. We define, for any Ehrenfeucht structure $JOIN(\alpha_1,\alpha_2,\alpha_3)$,

$$\|JOIN(\alpha_1,\alpha_2,\alpha_3)\| =$$
$$brep(a_m)\#brep(a_{m-1})\#\ldots\#brep(a_0)\#$$
$$brep(b_m)\#brep(b_{m-1})\#\ldots\#brep(b_0)\#$$
$$brep(c_m)\#brep(c_{m-1})\#\ldots\#brep(c_0)\ ,$$

where $\alpha_1 = \omega^m a_m + \ldots + a_0$, $\alpha_2 = \omega^m b_m + \ldots + b_0$, and $\alpha_3 = \omega^m c_m + \ldots + c_0$. For all other structure A in JOINS we let $\|A\| = \infty$. We also define a partial order on norms as follows: $brep(c_k)\#\ldots brep(c_0) \preceq brep(d_k)\#\ldots\#brep(d_0)$ iff $c_i \leq d_i$ for all $i \in I_k$. We define $w \preceq \infty$ for any norm w, and leave $\preceq$ undefined for all other pairs of words in $\{0,1,\#\}^*$.

Our notion of norm for elements in structures is as follows:

Definition 5.18 For any $JOIN(\alpha_1,\alpha_2,\alpha_3)$ and any $b \in J(\alpha_1,\alpha_2,\alpha_3)$ such that $b = (j/2^k,a)$ with $a \in \omega$-POLY(m,ℓ) for ℓ,m $\in I$, we define

$$\|b\| = \text{brep}(j)\#\text{brep}(2^k)\#\text{brep}(a_m)\#\ldots\#\text{brep}(a_0)$$

where $a = \omega^m a_m + \omega^{m-1} a_{m-1} + \ldots + \omega a_1 + a_0$. For all other $b \in J(\alpha_1,\alpha_2,\alpha_3)$, we define $\|b\| = \infty$.

We define a partial order on norms of elements as follows:

$$\text{brep}(k_1)\#\text{brep}(k_2)\#\text{brep}(a_m)\#\ldots\#\text{brep}(a_0) \lesssim$$
$$\text{brep}(\ell_1)\#\text{brep}(\ell_2)\#\text{brep}(b_m)\#\ldots\#\text{brep}(b_0)$$

iff $k_1/k_2 < \ell_1/\ell_2$, or $k_1/k_2 = \ell_1/\ell_2$ and $a_i \leq b_i$ for $i \in I_m$. We also define $w \lesssim \infty$ for any norm w, and leave $\lesssim$ undefined for all other pairs of words in $\{0,1,\#\}^*$.

We wish to show for all $n,k \in N$, $E_{n,k}$ is a refinement of $\overset{\equiv}{}_{n,k}$. We first show for all $k \in N$, $E_{0,k}$ is a refinement of $\overset{\equiv}{}_{0,k}$.

For all $A,B \in$ JOINS, $\bar{a}_k \in A^k$, and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_0 \; \bar{b}_k$, then for all $i,j \in I_k$, $a_i < a_j \iff b_i < b_j$. Using Lemma 5 of Chapter 2, we can conclude $\bar{a}_k \overset{\equiv}{0} \bar{b}_k$.

Again, following our outline, to complete the proof that for all $n,k \in N$, $E_{n,k}$ is a refinement of $\overset{\equiv}{}_{n,k}$, we would establish

Lemma 5.19 For all $n,k \in N$, $A,B \in$ JOINS, $\bar{a}_k \in A^k$ and $\bar{b}_k \in B^k$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$, then for all $a_{k+1} \in A$ there is $b_{k+1} \in B$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

To prove Lemma 5.19 we need two further lemmas.

Lemma 5.20 Let $n \in N$, α,β ordinals. Suppose $\alpha \overset{\sim}{\underset{n+1}{}} \beta$. Then for all $a < \alpha$ there is a $b < \beta$ such that

1. $a \overset{\sim}{\underset{n}{}} b$, and
2. $\alpha - a \overset{\sim}{\underset{n}{}} \beta - b$.

Lemma 5.21 Let $A \in$ JOINS, and let $c \in A \cup \{-\infty\}$, $d \in A \cup \{+\infty\}$. Suppose $[c,d)_A = \alpha_1 + \alpha_2\eta + \alpha_3$, such that $\alpha_2 \neq 0$. Then for any $\gamma < \alpha_2$ there is $b \in A$ such that $[c,b)_A = \alpha_1 + \alpha_2\eta + \gamma$, and $[b,d)_A = (\alpha_2 - \gamma) + \alpha_2\eta + \alpha_3$.

We leave the proofs of Lemmas 5.19, 5.20 and 5.21 to the reader.

It remains to give our definitions of H and h in this case, and to conclude that JOINS is (h,H)-bounded.

Definition 5.22 We define $h:N \to \{0,1,\#\}^* \cup \{\infty\}$ as follows:

$$h(m) = \text{brep}(1)(\#\text{brep}(2^m))^m\#\text{brep}(1)(\#\text{brep}(2^m))^m \#\text{brep}(1)(\#\text{brep}(2^m))^m.$$

It follows directly from the definition of $\overset{\sim}{\underset{n}{}}$ and ω-POLY(n,k) that for any $\text{JOIN}(\beta_1,\beta_2,\beta_3) \in$ JOINS, and $m \in N$, there is a $\text{JOIN}(\alpha_1,\alpha_2,\alpha_3) \trianglelefteq h(m)$ such that $\alpha_1 + \alpha_2\eta + \alpha_3 \overset{\sim}{\underset{m}{}} \beta_1 + \beta_2\eta + \beta_3$, and thus $\text{JOIN}(\alpha_1,\alpha_2,\alpha_3)\ E_{m,0}\ \text{JOIN}(\beta_1,\beta_2,\beta_3)$.

Our definition of H in this case is as follows:

Definition 5.23 For $n,k \in N$, and
norm $m = brep(j)\#brep(2^{\ell})\#brep(a_n)\#...\#brep(a_0)$, we define

$$H(n,k,m) = brep(2^{\ell+1})\#brep(2^{\ell+1})(\#brep(2^{n+2k}))^{n+k}.$$

For all other $m \in \{0,1,\#\}^* \cup \{\infty\}$, we define $H(n,k,m) = \infty$.

By an examination of the proof of Lemma 5.19, one can then establish

Theorem 5.24 For all $A,B \in$ JOINS, $\bar{a}_k \in A^k$, $\bar{b}_k \in B^k$, and $m \in \{0,1,\#\}^* \cup \{\infty\}$, if $\bar{a}_k \; E_{n+1} \; \bar{b}_k$ and $b_i \leq m$ for all $i \in I_k$, then for all $a_{k+1} \in A$ there is a $b_{k+1} \in B$ such that $b_{k+1} \leq H(n,k,m)$ and $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$.

We could then conclude by Theorem 4 of Chapter 2.

Theorem 5.25 For H as in Definition 5.23, and h as in Definition 5.22, the class of structures JOINS is (h,H) bounded

As in the previous cases, we would obtain

Theorem 5.26 For some constant $c > 0$, whether a sentence F of length n is in SAT(JOINS) can be decided within space $c \cdot n^3$.

We now state our results for the theories of well-order, discrete order, and lexicographical order.

We first consider the theory of well-order. Now if α is an ordinal, $< \alpha,< >$ is clearly isomorphic to JOIN(α,0,0). We henceforth identify α and JOIN(α,0,0).

Now, we can obtain a decision procedure for the theory of well-order by restricting our decision procedure for SAT(JOINS) to ordinals. That is, given any sentence F of length n, we check the satisfiability of F only in ordinals $\preceq$ h(n). In this manner, we obtain:

Theorem 5.27 For some constant c > 0, whether a sentence F of length n is satisfiable in any well-order can be decided within space $c \cdot n^3$.

We next consider $<N,< >$. Clearly $< N,< >$ is isomorphic to JOIN(ω,0,0). We henceforth identify $< N,< >$ with this structure. By restricting our decision procedure for SAT(JOINS) to $< N,< >$, and examining the procedure carefully, we obtain,

Theorem 5.28 For some constant c > 0, whether a sentence F of length n is in TH($< N,< >$) can be decided within space $c \cdot n^2$.

We lastly consider the theory of lexicographical order. We first prove the following:

Lemma 5.29 $<\{0,1\}^*,\prec>$ is isomorphic to JOIN($\omega,\omega,0$).

Proof We first show $<\{0,1\}^*\cdot 1,\prec>$ is a dense linear order without end points. Clearly it is a linear order. To show it has no end points, let $w1 \in \{0,1\}^*\cdot 1$. Then $0^{\ell n(w)+1}\cdot 1 \prec w\cdot 1$, and $w\cdot 1 \prec 1^{\ell n(w)+2}$. Since $0^{\ell n(w)+1}\cdot 1$ and $1^{\ell n(w)+2} \in \{0,1\}^*\cdot 1$, we have $<\{0,1\}^*\cdot 1,\prec>$ a linear order without end points.

We next show this order is dense. For this purpose let $w_1,w_2 \in \{0,1\}^*\cdot 1$, with $w_1 \prec w_2$. Then either there are $w,w_3,w_4 \in \{0,1\}^*$ such that $w_1 = w0w_3$ and $w_2 = w1w_4$, or there is $w \in \{0,1\}^*\cdot 1$ such that $w_2 = w_1\cdot w$.

If the first holds, let $w_5 = w01^{\ell n(w_3)+1}$. We claim in this case $w_1 = w0w_3 \prec w_5 \prec w1w_4$.

If the second holds, let $w_5 = w_1 0^{\ell n(w)}1$. We claim in this case $w_1 \prec w_5 \prec w_1 w = w_2$. We thus conclude $<\{0,1\}^*\cdot 1,\prec>$ is a dense linear order without end points.

Since by Cantor's Theorem [Kam 50], any two countable dense linear orders without end points are isomorphic, we must have $<\{0,1\}^*\cdot 1,\prec>$ isomorphic to $<\{q \in Q \mid 0<q<1\},<>$. We let is : $\{0,1\}^*\cdot 1 \to \{q \in Q \mid 0<q<1\}$ denote such an isomorphism.

We now define a function is* from words in $\{0,1\}^*$ to elements of $J(\omega,\omega,0)$, as follows. Let $w \in \{0,1\}^*$. If $w = 0^n$ for some $n \in N$, we define

$$is^*(w) = (0,n);$$

if $w = w_1 0^n$ for some $n \in N$, $w_1 \in \{0,1\}^*\cdot 1$, we define $is^*(w) = (is(w_1),n)$. It is an easy matter to verify is* is an isomorphism. □

We henceforth identify $<\{0,1\}^*, \prec >$ with JOIN($\omega,\omega,0$).

Again, by restricting our decision procedure for SAT(JOINS) to $<\{0,1\}^*, \prec >$, and examining the procedure carefully, we obtain

Theorem 5.30 For some constant $c > 0$, whether a sentence F of length n is in TH($<\{0,1\}^*, \prec >$) can be decided within space $c \cdot n^2$.

We note some further results that follow easily. Since $<Q, <>$ is isomorphic to JOIN(0,1,0), we obtain

Theorem 5.31 For some constant $c > 0$, whether a sentence F of length n is in TH($<Q, <>$) can be decided within space $c \cdot n^2$.

However, a better space bound can be obtained for this theory by a more careful analysis; see [FeGe 77].

We also consider $<N, '>$, where $': N \to N$ is such that for all $n \in N$, $n' = n+1$. There is a function f from {sentences of L} $\to$ {sentences of L} such that for any sentence F F is in TH($<N, '>$) $\Leftrightarrow$ f(F) is in TH($<N, <>$). This function is defined inductively, noting that for all $m,n \in N$,

$m' = n \Leftrightarrow m < n$, and for no $b \in N$ is it true $m < b < n$

We leave the details to the reader, as well as the verification that

length(f(F)) = O(length(F)).

We then have:

Theorem 5.32 For some constant $c > 0$, whether a sentence F of length n is in TH($<N, '>$) can be decided within space $c \cdot n^2$.

Section 6: Historical Remarks

All of the complexity results presented in this chapter are for theories already known to be decidable. The decidability of the theory of monadic functions and monadic predicates, without equality, was proved by Eichholz [Eich 57]; Ehrenfeucht proved the decidability of the theory of one monadic function and monadic predicates with equality [Ehr 59]. Hence the theory of a 1-1 unary function, with or without monadic predicates, is decidable, since the property of 1-1-ness is expressible in the language. Elgot and Rabin [ER 66] obtain decidability results for the first and second order theories of one and two successors. The decidability of $\langle N, < \rangle$ follows from a result of Presburger [Pre 29]. The theory of well-order was proved decidable by Mostowski and Tarski [MT 49].

One of the most powerful (and difficult) decidability results is Rabin's theorem that the second order theory of two successors is decidable [Rab 69]; many earlier theorems, as well as many new ones, follow quite easily from it. In particular, it leads to a very elegant proof of the decidability of the theory of a monadic function.

CHAPTER 5

DIRECT PRODUCTS OF THEORIES

Section 1: Weak Direct Powers and Ehrenfeucht Games

Let L be a language of the first order predicate calculus with a finite number of predicate symbols $\underline{R}_1,\underline{R}_2,\ldots,\underline{R}_\ell$ such that $\underline{R}_i$ is a t_i place formal predicate for $1 \leq i \leq \ell$, and with a constant symbol $\underline{e}$.

Definition 1.1 Let $A = \langle A,R_1,R_2,\ldots,R_\ell,e\rangle$ be a structure for L. For all $a \in A$, let $\|a\|$ be the norm of a. For convenience, we will assume $\|a\| \in N$, and the ordering $\preceq$ is just $\leq$. The weak direct power of A is $A^* = \langle A^*,R_1^*,R_2^*,\ldots,R_\ell^*,e^*\rangle$ where $A^* = \{f:N \to A \mid f(i) \neq e$ for only finitely many $i \in N\}$; for $1 \leq j \leq \ell$, if $\vec{f}_{t_j} \in (A^*)^{t_j}$, then $\vec{f}_{t_j} \in R_j^*$ iff $\vec{f}_{t_j}(i) \in R_j$ for all $i \in N$ (where $\vec{f}_{t_j}(i)$ abbreviates $(f_1(i),f_2(i),\ldots,f_{t_j}(i))$); $e^*(i) = e$ for all $i \in N$.

For a norm on A^* we define, for $f \in A^*$, $\|f\| = \mathrm{Max}(\{i \in N \mid f(i) \neq e\} \cup \{\|f(i)\| \mid i \in N\})$. By $f \preceq m$ we will mean $\|f\| \leq m$.

Mostowski [Mos52] and Feferman and Vaught [FV59] both show that TH(A) decidable $\Rightarrow$ TH(A^*) decidable. However, their proofs are such that in every case, the decision procedure for TH(A^*) obtained is not elementary recursive. In this section we will present some general theorems which will allow us to derive significantly more efficient decision procedures for

$TH(A^*)$ in many cases, and in particular to obtain a procedure for $TH(\mathbb{Z}^*)$ (where $\mathbb{Z}$ is the structure of integer addition defined in Chapter 3) which closely matches the known lower bound. In Section 3 we discuss even more general theorems which give a condition under which we can conclude $TH(A^*)$ elementary recursive if $TH(A)$ is elementary recursive.

Now let $H: N^3 \to N$ be such that A is H-bounded. Let $M(n,k)$ be the function as defined for A in Chapter 2, Section 1.

Definition 1.2 Define the function $\mu:N^2 \to N$ by setting $\mu(0,k) = 1$, and $\mu(n+1,k) = M(n,k+1)\cdot\mu(n,k+1)$. Hence $\mu(n,k) = \prod_{i=1}^{n} M(n-i,k+i)$.

Definition 1.3 Define $H^*:N^3 \to N$ by $H^*(n,k,m) = \text{Max}\{H(n,k,m), m + \mu(n+1,k), \|e\|\}$.

The major theorem of this section will show that A^* is H^*-bounded.

We now prove a combinatorial lemma. $\underset{n}{=}$ is defined in Chapter 4 to be, essentially, the relation whereby two things are $\underset{n}{=}$ if they are "the same size up to n". It is convenient now to define $\underset{n}{=}$ slightly differently than in Chapter 4.

Definition 1.4 Let $x,y,n \in N$. Then $x \underset{n}{=} y$ if either $x = y$ or $x \geq n$ and $y \geq n$. If A and B are two sets, $A \underset{n}{=} B$ if $|A| \underset{n}{=} |B|$.

Lemma 1.5 Let N_1 and N_2 be sets and let $n,m \in N^+$ such that $N_1 \underset{n\cdot m}{=} N_2$. Let $A_1,A_2,\ldots,A_n$ be a sequence of (possibly empty) pairwise disjoint subsets of N_1 such that $\bigcup_{i=1}^{n} A_i = N_1$.

Then there exists a sequence $B_1,B_2,\ldots,B_n$ of pairwise disjoint subsets of N_2 such that $\bigcup_{i=1}^{n} B_i = N_2$ and such that $A_i \underset{m}{=} B_i$ for $1 \leq i \leq n$.

Proof If $|N_1| = |N_2|$ then the Lemma is obvious. Assume $|N_1| \geq n\cdot m$ and $|N_2| \geq n\cdot m$. For some i, $1 \leq i \leq n$, we must have $|A_i| \geq m$, so assume without loss of generality that $|A_1| \geq m$.

Define numbers $p_2,p_3,\ldots,p_n \in N$ by

$$p_i = \begin{cases} |A_i| & \text{if } |A_i| < m \\ m & \text{if } |A_i| \geq m \end{cases} \quad \text{for } 2 \leq i \leq n.$$

Clearly $\sum_{i=2}^{n} p_i \leq (n-1)m = n\cdot m-m$. Since $|N_2| \geq n\cdot m$, there exists a sequence of pairwise disjoint subsets of N_2, namely $B_2,B_3,\ldots,B_n$, such that $|B_i| = p_i$ for $2 \leq i \leq n$. So $A_i \underset{m}{=} B_i$ for $2 \leq i \leq n$. Let $B_1 = N_2 - \bigcup_{i=2}^{n} B_i$. $|N_2| \geq n\cdot m$ and $\bigcup_{i=2}^{n} B_i \leq n\cdot m-m$, so $|B_1| \geq m$. Since $|A_1| \geq m$, $A_1 \underset{m}{=} B_1$. □

For every $n,k \in N$, define the Ehrenfeucht relation $\underset{n}{\equiv}$ on both A^k and $(A^*)^k$ as in Chapter 2, Section 1.

Definition 1.6 Let $n,k \in N$ and $\bar{f}_k,\bar{g}_k \in (A^*)^k$. Then we say $\bar{f}_k \; E_n \; \bar{g}_k$ iff for all $\bar{a}_k \in A^k$,

$$\{i \in N \mid \bar{f}_k(i) \underset{n}{\equiv} \bar{a}_k\}_{\mu(n,k)}\{i \in N \mid \bar{g}_k(i) \underset{n}{\equiv} \bar{a}_k\}.$$

Lemma 1.7 For all $k \in N$, $\bar{f}_k,\bar{g}_k \in (A^*)^k$, if $\bar{f}_k \; E_0 \; \bar{g}_k$ then $\bar{f}_k \underset{0}{\equiv} \bar{g}_k$.

Proof Say that $\bar{f}_k \; E_0 \; \bar{g}_k$. We wish to show that for every quantifier free formula $F(\bar{x}_k)$, $A^* \models F(\bar{f}_k) \iff A^* \models F(\bar{g}_k)$. It is clearly sufficient to prove this for the case where F is atomic. By symmetry, it is sufficient to show that $F(\bar{f}_k)$ false in $A^* \Rightarrow F(\bar{g}_k)$ false in A^*.

Thus assume that $F(\bar{f}_k)$ is false in A^*. By definition of the relations of A^* we can choose $i_0 \in N$ such that $F(\bar{f}_k(i_0))$ is false in A. Since $\bar{f}_k \; E_0 \; \bar{g}_k$, we have that $\{i \in N | \bar{f}_k(i) \equiv_{\bar{0}} \bar{f}_k(i_0)\} =_{\mu(0,k)} \{i \in N | \bar{g}_k(i) \equiv_{\bar{0}} \bar{f}_k(i_0)\}$. Since $\mu(0,k) = 1$, we have $|\{i \in N | \bar{g}_k(i) \equiv_{\bar{0}} \bar{f}_k(i_0)\}| \geq 1$. So let $i_1 \in N$ be such that $\bar{g}_k(i_1) \equiv_{\bar{0}} \bar{f}_k(i_0)$. By definition of $\equiv_{\bar{0}}$, $F(\bar{f}_k(i_0))$ false in $A \Rightarrow F(\bar{g}_k(i_1))$ false in A. So $F(\bar{g}_k)$ is false in A^*. □

Lemma 1.8 Let $n,k \in N$ and $\bar{f}_k, \bar{g}_k \in (A^*)^k$ such that $\bar{f}_k \; E_{n+1} \; \bar{g}_k$. Then for each $f_{k+1} \in A^*$ there exists some $g_{k+1} \in A^*$ such that

1) $\bar{f}_{k+1} \; E_n \; \bar{g}_{k+1}$

and

2) $\|g_{k+1}\| \leq H^*(n,k, \operatorname{Max}_{1 \leq i \leq k} \{\|g_i\|\})$.

Proof Let $\bar{f}_k, \bar{g}_k \in (A^*)^k$ be such that $\bar{f}_k \; E_{n+1} \; \bar{g}_k$. Let $m = \operatorname{Max}_{1 \leq i \leq k} \{\|g_i\|\}$ and let $f_{k+1} \in A^*$. Let $\bar{b}^1_{k+1}, \bar{b}^2_{k+1}, \ldots, \bar{b}^{M(n,k+1)}_{k+1}$ be a sequence of representatives of all the $\equiv_n$ equivalence classes on A^{k+1}. Our goal is to find $g_{k+1} \in A^*$ such that if $1 \leq j \leq M(n,k+1)$, then $\{i \in N | \bar{f}_{k+1}(i) \equiv_n \bar{b}^j_{k+1}\} =_{\mu(n,k+1)} \{i \in N | \bar{g}_{k+1}(i) \equiv_n \bar{b}^j_{k+1}\}$; we also want $\|g_{k+1}\| \leq H^*(n,k,m)$. Instead of defining g_{k+1} simultaneously on all of N, we will define it separately on various pieces of N.

For each $\bar{a}_k \in A^k$ define $N_1(\bar{a}_k) = \{i \in N | \bar{f}_k(i) \underset{n+1}{\equiv} \bar{a}_k\}$ and $N_2(\bar{a}_k) = \{i \in N | \bar{g}_k(i) \underset{n+1}{\equiv} \bar{a}_k\}$. We claim it is sufficient to define g_{k+1} on each $N_2(\bar{a}_k)$ such that

I) $\{i \in N_1(\bar{a}_k) | \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} \{i \in N_2(\bar{a}_k) | \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$

for all j, $1 \le j \le M(n,k+1)$.

II) If $i \in N_2(\bar{a}_k)$ and $i > m+\mu(n+1,k)$, then $g_{k+1}(i) = e$.

and

III) If $i \in N_2(\bar{a}_k)$ and $i \le m+\mu(n+1,k)$, then $\|g_{k+1}(i)\| \le H(n,k,m)$.

An examination of the definitions of H^* and the norm on A^* will show that II) and III) together imply $\|g_{k+1}\| \le H^*(n,k,m)$. Since $\{N_1(\bar{a}_k) | \bar{a}_k \in A^k\}$ and $\{N_2(\bar{a}_k) | \bar{a}_k \in A^k\}$ are each a collection of disjoint sets, it is easy to see from I) and the definition of $\underset{\mu(n,k+1)}{=}$ that if $1 \le j \le M(n,k+1)$ then

$$\left(\bigcup_{\bar{a}_k \in A^k} \{i \in N_1(\bar{a}_k) | \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}\right) \underset{\mu(n,k+1)}{=} \left(\bigcup_{\bar{a}_k \in A^k} \{i \in N_2(\bar{a}_k) | \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}\right),$$

i.e., $\{i \in N | \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} \{i \in N | \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$.

So now let $\bar{a}_k \in A^k$ be fixed for the rest of this proof. Abbreviate $N_1(\bar{a}_k)$ by N_1 and $N_2(\bar{a}_k)$ by N_2. Begin by defining $g_{k+1}(i) = e$ if $i \in N_2$ and $i > m+\mu(n+1,k)$; this guarantees II) above. It remains to define g_{k+1} on $N_3 = \{i \in N_2 | i \le m+\mu(n+1,k)\}$.

The definition of E_{n+1} implies that $N_1 \underset{\mu(n+1,k)}{=} N_2$. We now demonstrate that $N_1 \underset{\mu(n+1,k)}{=} N_3$; if $\bar{a}_k \underset{n+1}{\equiv} e^k$ then N_1 is an infinite set, and $|N_3| \ge \mu(n+1,k)$ since $\bar{g}_k(i) = e^k$ for

$m < i \leq m+\mu(n+1,k)$; if $\bar{a}_k \not\underset{n}{=}_1 e^k$ then $N_3 = N_2$ (since $i > m+\mu(n+1,k) \Rightarrow \bar{g}_k(i) = e^k \Rightarrow i \notin N_2$). So $N_1 \underset{\mu(n+1,k)}{=} N_3$

Define, for $1 \leq j \leq m(n,k+1)$, $A_j = \{i \in N_1 \mid \bar{f}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\}$. $A_1, A_2, \ldots, A_{M(n,k+1)}$ form a sequence of pairwise disjoint sets whose union is N_1. Since $N_1 \underset{\mu(n+1,k)}{=} N_3$ and $\mu(n+1,k) = M(n,k+1)\cdot\mu(n,k+1)$, Lemma 1.5 tells us there exists a sequence $B_1, B_2, \ldots, B_{M(n,k+1)}$ of pairwise disjoint subsets of N_3 whose union is N_3 such that $A_j \underset{\mu(n,k+1)}{=} B_j$ if $1 \leq j \leq M(n,k+1)$.

Now let $i \in N_3$; we want to define g_{k+1} on i. Let j be such that $i \in B_j$. Since $B_j \neq \phi$, we also have $A_j \neq \phi$. So let $i_0 \in A_j$. Since $i_0 \in N_1$ and $i \in N_2$, we have $\bar{f}_k(i_0) \underset{n}{\equiv}_1 \bar{a}_k \underset{n}{\equiv}_1 \bar{g}_k(i)$. By Lemma 9 of Chapter 2 we can define $g_{k+1}(i)$ such that $\bar{f}_{k+1}(i_0) \underset{n}{=} \bar{g}_{k+1}(i)$ and $\|g_{k+1}(i)\| \leq H(n,k,\text{Max}\{\|g_1(i)\|, \|g_2(i)\|, \ldots, \|g_k(i)\|\}) \leq H(n,k,m)$. Clearly III) above holds. Since $i_0 \in A_j$, $\bar{f}_{k+1}(i_0) \underset{n}{=} \bar{b}^j_{k+1}$. So $\bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}$. Thus, we have defined $g_{k+1} \in A^*$ so that for $1 \leq j \leq M(n,k+1)$, $\{i \in N_3 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\} = B_j \underset{\mu(n,k+1)}{=} A_j = \{i \in N_1 \mid \bar{f}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\}$.

To complete the proof of Lemma 1.8, we must show I), i.e. $\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} A_j$ when $1 \leq j \leq M(n,k+1)$.

So fix j, $1 \leq j \leq M(n,k+1)$. If $\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\} = \{i \in N_3 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\}$ we are done, so assume $\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\} \neq \{i \in N_3 \mid \bar{g}_{k+1}(i) \underset{n}{=} \bar{b}^j_{k+1}\}$.

Since $N_3 = \{i \in N_2 | i \leq m+\mu(n+1,k)\}$, there must exist some $i > m+\mu(n+1,k)$ such that $i \in N_2$ (hence $\bar{g}_k(i) \underset{n+1}{\equiv} \bar{a}$) and $\bar{g}_{k+1}(i) \underset{n}{\not\equiv} \bar{b}^j_{k+1}$. But since $i > m+\mu(n+1,k)$ implies $\bar{g}_{k+1}(i) = e^{k+1}$, this means that $\bar{a}_k \underset{n+1}{\not\equiv} e^k$ and $\bar{b}^j_{k+1} \underset{n}{\not\equiv} e^{k+1}$. Hence, both A_j and $\{i \in N_2 | \bar{g}_{k+1}(i) \underset{n}{\not\equiv} \bar{b}^j_{k+1}\}$ are infinite, so $\{i \in N_2 | \bar{g}_{k+1}(i) \underset{n}{\not\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{} A_j$. □

Theorem 1.9 A^* is H^*-bounded. Also, for every $n,k \in N$ and $\bar{f}_k, \bar{g}_k \in (A^*)^k$, $\bar{f}_k \; E_n \; \bar{g}_k \Rightarrow \bar{f}_k \underset{n}{\equiv} \bar{g}_k$.

Proof This follows immediately from Lemmas 1.7 and 1.8, and Theorem 3 of Chapter 2. □

Section 2: Upper Bounds for The Theories of Integer Multiplication and Abelian Groups

We now present some applications of the material in Section 1. Let L_1 be the language of Chapter 3.

Let $\mathbb{Z} = \langle Z,+,\leq,0\rangle$ be the structure of Chapter 3 and let $\mathbb{Z}^* = \langle Z^*,+,\leq,0^*\rangle$ be the weak direct power of $\mathbb{Z}$. As before, for $a \in Z$ let $\|a\| = |a|$ and, following Definition 1.1, for $f \in Z^*$ let $\|f\| = \text{Max}(\{i \in N | f(i) \neq 0\} \cup \{|f(i)| | i \in N\})$.

Lemma 2.1 There exists a constant e such that $\mathbb{Z}^*$ is $(1+m)\cdot 2^{2^{2^{e(n+k)}}}$-bounded.

Proof By Lemma 1.6 of Chapter 3, $\mathbb{Z}$ is H-bounded where $H(n,k,m) = (1+m)\cdot 2^{2^{2^{d(n+k)}}}$ for some constant d. We now calculate bounds for the function M(n,k) for $\mathbb{Z}$. Letting $m_i = 2^{2^{2^{d(n+k)+i}}}$ for $0 \leq i \leq k$, we see that $m_i \geq H(n+k-i,i-1,m_{i-1})$ for $1 \leq i \leq k$. So by Lemma 10 of Chapter 2, for each $\bar{a}_k \in Z^k$ there is some $\bar{b}_k \in Z^k$ such that $\bar{a}_k \equiv_{\bar{n}} \bar{b}_k$ and $|b_i| \leq m_i$ for $1 \leq i \leq k$. Hence, since $m_i \leq m_k$, we certainly have $M(n,k) \leq (2\cdot 2^{2^{2^{d(n+k)+k}}}+1)^k$. So $\mu(n,k) = \prod_{i=1}^{n} M(n-i,k+i) \leq 2^{2^{2^{d'(n+k)}}}$ for some constant d'.

So for some constant e, $H^*(n,k,m) = \text{Max}\{H(n,k,m),m+\mu(n+1,k),0\} \leq (1+m)\cdot 2^{2^{2^{e(n+k)}}}$

By Theorem 1.9, Z^* is $(1+m)\cdot 2^{2^{2^{e(n+k)}}}$-bounded. □

Lemma 2.2 Let F be the sentence of L_1, $Q_1x_1Q_2x_2\ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free. Then for some constant e independent of n, F is equivalent in $\mathbb{Z}^*$ to $(Q_1x_1 \leq 2^{2^{2^{en+1}}})(Q_2x_2 \leq 2^{2^{2^{en+2}}})\ldots (Q_nx_n \leq 2^{2^{2^{en+n}}})G(\bar{x}_n)$.

Proof By Lemma 2.1, Z^* is H-bounded where $H(n,k,m) = (1+m)2^{2^{2^{e(n+k)}}}$. Define $m_i = 2^{2^{2^{en+i}}}$ for $0 \leq i \leq n$. Since $m_i \geq H(n-i,i-1,m_{i-1})$ for $1 \leq i \leq n$, Lemma 2.2 follows from Theorem 1 of Chapter 2.

□

Theorem 2.3 For some constant c, $TH(\langle Z,+,\leq,0\rangle^*)$ can be decided within space $2^{2^{2^{cn}}}$.

Proof It is sufficient to consider the sentence F of L_1 which in prenex normal form is $Q_1x_1Q_2x_2\ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free and of length at most n log n.

By Lemma 2.2, F is equivalent to $(Q_1x_1 \leq 2^{2^{2^{en+1}}})(Q_2x_2 \leq 2^{2^{2^{en+2}}})\ldots(Q_nx_n \leq 2^{2^{2^{en+n}}})G(\bar{x}_n)$ for some constant e.

Now if $f \in Z^*$ and $f \leq 2^{2^{2^{en+i}}}$, then $f(j) = 0$ for $j > 2^{2^{2^{en+i}}}$ and $|f(j)| \leq 2^{2^{2^{en+i}}}$ for all $j \in N$, so the first $2^{2^{2^{en+i}}}$ successive values of f can be represented on a tape with roughly $(2^{2^{en+i}}+2)\cdot 2^{2^{2^{en+i}}}$ tape squares. So a procedure like the one outlined in Theorem 1.7 of Chapter 3 would decide $TH(\mathbb{Z}^*)$ in space $2^{2^{2^{cn}}}$ for some constant c. □

Remark 2.5 The structure $\langle \mathbb{N}^*, + \rangle$ is isomorphic to the structure $\langle \mathbb{N}^+, \cdot \rangle$ (i.e., the positive integers under multiplication). So an upper bound on the complexity of $TH(\mathbb{N}^*)$ is an upper bound on $TH(\langle \mathbb{N}^+, \cdot \rangle)$. The decidability of the theory of $\langle \mathbb{N}^+, \cdot \rangle$ was proved by Skolem [Sko31].

Corollary 2.6 $TH(\mathbb{N}^*)$ can be decided in space $2^{2^{2^{cn}}}$ for some constant c.

Proof Since $x \geq 0$ is a formula of L_1, it is easy to see that $TH(\mathbb{N}^*) \leq_{p\ell} TH(\mathbb{Z}^*)$. So Corollary 2.6 follows from Lemma 4.4 of Chapter 1. □

The upper bound of Corollary 2.3 and Corollary 2.6 matches the lower bound of Theorem 2.7 reasonably well.

Theorem 2.7 (Fischer and Rabin [FiR74]) For some constant $c' > 0$, any nondeterministic Turing machine which recognizes $TH(\mathbb{Z}^*)$ (or $TH(\mathbb{N}^*)$) requires time $2^{2^{2^{c'n}}}$ on some sentence of length n, for infinitely many n.

Our next goal is to present a decision procedure for the first order theory of finite abelian groups;† this theory was originally shown to be decidable (see [Szm55],††[ELTT65]) by a less efficient procedure than ours. Our approach will be to show that this theory is $\leq_{p\ell} TH(\mathbb{N}^*)$ and conclude

Theorem 2.8 The first order theory of finite abelian groups can be decided within space $2^{2^{2^{cn}}}$ for some constant c.

---

† This topic is also discussed in Section 4 from a slightly different viewpoint.

†† In fact, Szmielew actually shows the theory of abelian groups to be decidable.

There is still a significant gap between the upper bound of Theorem 2.8 and the known lower bound of Theorem 2.9.

Theorem 2.9 (Fischer and Rabin [FiR74]) For some constant $c' > 0$, any nondeterministic Turing machine which recognizes the theory of finite abelian groups requires time $2^{2^{c'n}}$ on some sentence of length n, for infinitely many n.

The language of groups, L_2, merely contains the formal predicate $v_1 + v_2 = v_3$. We are interested in deciding which sentences of L_2 are true of every finite abelian group. Recall that every finite abelian group (henceforth abbreviated FAG) is isomorphic to a finite direct product of finite cyclic groups [MB68]. For i a positive integer, let Z_i denote the cyclic group $\{0,1,\ldots,i-1\}$ where addition is performed mod i. The basic idea of the embedding (due to Michael J. Fischer) is to think of every nonzero $f \in N^*$ as representing an FAG, G_f. This is made precise in the following definition.

Definition 2.10 Let $f \in N^*$, $f \neq 0^*$. Define $\ell_f = |\{i \in N \mid f(i) \neq 0\}|$. Define m_f: $\{1,2,\ldots,\ell_f\} \to N$ by

$m_f(j)$ = the j-th smallest member of $\{i \in N \mid f(i) \neq 0\}$ for $1 \leq j \leq \ell_f$.

Define the FAG $G_f = G_1 \times G_2 \times \ldots \times G_{\ell_f}$ where $G_j = Z_{f(m_f(j))}$ for $1 \leq j \leq \ell_f$.

Clearly every FAG is isomorphic to G_f for some $f \in N^*$, $f \neq 0^*$.

Definition 2.11 Let $f,g \in N^*$, $f \neq 0^*$, be such that for all $i \in N$

a) $f(i) = 0 \Rightarrow g(i) = 0$

and b) $f(i) > 0 \Rightarrow 0 \leq g(i) < f(i)$.

Then we say that g <u>represents</u> $<g(m_f(1)),g(m_f(2)),\ldots,g(m_f(\ell_f))> \in G_f$. Clearly for each $f \neq 0^*$, every member of G_f is represented by a unique $g \in N^*$.

We now describe some properties definable in L_1 by formulas interpreted over N^*. (<u>Properties</u> are discussed in Chapter 7.)

1) ONE(x). For $f \in N^*$, ONE(f) will hold iff for some $i \in N$, $f(i) = 1$ and for every $j \neq i$, $f(j) = 0$. ONE(x) is equivalent to $x \neq 0^* \wedge \forall x'((0^* \leq x' \wedge x' \leq x) \rightarrow (x'=0^* \vee x'=x))$.

2) $ZERO(x_1,x_2)$. For $f_1,f_2 \in N^*$, $ZERO(f_1,f_2)$ will hold iff $ONE(f_1)$ and $f_1(i) = 0 \Rightarrow f_2(i) = 0$. $ZERO(x_1,x_2)$ is equivalent to $ONE(x_1) \wedge \forall x'((ONE(x') \wedge x' \neq x_1) \rightarrow \sim(x' \leq x_2))$.

3) $PICK(x_1,x_2,x_3)$. For $f_1,f_2,f_3 \in N^*$, $PICK(f_1,f_2,f_3)$ will hold iff $ONE(F_1)$ and

$$(f_1(i) = 0 \Rightarrow f_2(i) = 0) \wedge (f_1(i) = 1 \Rightarrow f_2(i) = f_3(i)).$$

$PICK(x_1,x_2,x_3)$ is equivalent to

$$ZERO(x_1,x_2) \wedge x_2 \leq x_3 \wedge \sim(x_1+x_2 \leq x_3).$$

4) $MEM(x_1,x_2)$. For $f_1,f_2 \in N^*$, $MEM(f_1,f_2)$ will hold iff $f_1 \neq 0^*$ and f_2 represents a member of G_{f_1}. $MEM(x_1,x_2)$ is equivalent to $x_1 \neq 0^* \wedge x_2 \leq x_1 \wedge \forall x\forall x_1'\forall x_2'([PICK(x,x_1',x_1) \wedge PICK(x,x_2',x_2)] \rightarrow (x_1' \neq 0^* \rightarrow x_2' \neq x_1'))$.

5) $PLUS(x_1,x_2,x_3,x_4)$. For $f_1,f_2,f_3,f_4 \in N^*$, $PLUS(f_1,f_2,f_3,f_4)$ will hold iff $f_1 \neq 0^*$ and f_2,f_3,f_4 represent members of G_{f_1} and the member represented by f_4 is the sum in G_{f_1} of the members represented by f_2 and f_3. $PLUS(x_1,x_2,x_3,x_4)$ is equivalent to

$$MEM(x_1,x_2) \wedge MEM(x_1,x_3) \wedge MEM(x_1,x_4) \wedge \forall x\forall x_1'\forall x_2'\forall x_3'\forall x_4'$$
$$[(PICK(x,x_1',x_1) \wedge PICK(x,x_2',x_2) \wedge PICK(x,x_3',x_3) \wedge PICK(x,x_4',x_4)) \rightarrow$$
$$(x_2' + x_3' = x_4' \vee x_2' + x_3' = x_4' + x_1')].$$

Proof of Theorem 2.8 Using formulas defining MEM and PLUS and the fact that $f \in N^*$ represents a FAG if and only if $f \neq 0^*$, we obtain a procedure which operates in polynomial time and linear space which takes a sentence F of L_2 to a sentence F' of L_1, such that F is true of every FAG $\iff$ F' $\in$ TH(N^*). So TH(FAG) $\leq_{p\ell}$ TH(N^*). Theorem 2.8 therefore follows from Corollary 2.6, and Lemma 4.4 of Chapter 1. □

Section 3: The Complexity of Theories of Weak Direct Powers

Let L, A, A^* be as in Section 1, and let $M(n,k)$ be defined for A as in Definition 7 of Chapter 2.

Theorem 3.1 If $TH(A)$ is elementary recursive and if $M(n,k)$ is bounded above by an elementary recursive function, then $TH(A^*)$ is elementary recursive.

Theorem 3.1 can be proven by modifying either Mostowski's or Feferman and Vaught's decision procedure for $TH(A^*)$ [Mos52,FV59], but we now outline a slightly different proof based on the general description of Ehrenfeucht games given in Section 2 of Chapter 2, as well as the results of Section 1 of this chapter.

Let $\equiv_{n,k}$ be the Ehrenfeucht equivalence relations for A as defined in Chapter 2, and let $E_{n,k}$ be the equivalence relations for A^* as defined in Section 1 of this chapter. $M(n,k)$ is just the number of $\equiv_{n,k}$ equivalence classes.

Say now that we wish to decide truth in A^* of the formula $Q_1x_1Q_2x_2\ldots Q_nx_nG(\bar{x}_n)$. We begin, for each i, $1 \leq i \leq n$, by writing down formulas $F(\bar{x}_i)$ defining all the $\equiv_{n-i,i}$ equivalence classes. This is done following the proof of Lemma 8 in Chapter 2. Every formula defining a $\equiv_{0,n}$ class will be a conjunction of atomic formulas and negations of atomic formulas; which of these conjunctions are satisfiable, and hence define a (nonempty) $\equiv_{0,n}$ class, can be determined using the given procedure for $TH(A)$.

For $i < n$, every formula defining a $\equiv_{n-i,i}$ class will be a conjunction of formulas, and negations of formulas, of the form $\exists x_{i+1} F(\bar{x}_{i+1})$ where F defines a $\equiv_{n-(i+1),i+1}$ class; which of these conjunctions are satisfiable can be determined using the given procedure for TH(A).

Following the outline of Chapter 2, we now wish to define suitable representations for the $E_{n-i,i}$ equivalence classes for $1 \leq i \leq n$ and the E equivalence relations as in Definition 1.6. Let u be as in Definition 1.2. Then we can represent an $E_{n-i,i}$ equivalence class by a sequence of formulas $F_1, F_2, \ldots, F_k$ each of which defines an $\equiv_{n-i,i}$ class, so that no formula occurs more than u(n-i,i) times in the sequence and such that at least one formula occurs u(n-i,i) times in the sequence. A member $\bar{f}_i \in (A^*)^i$ is in the $E_{n-i,i}$ class represented by $F_1, F_2, \ldots, F_k$ iff for every formula F defining a $\equiv_{n-i,i}$ class, the number of times F appears in $F_1, F_2, \ldots, F_k$ is $u(n-i,i)$ to $\{j \mid \bar{f}_i(j)$ satisfies $F\}$.

It is not too difficult now to follow the outline in Chapter 2 to obtain a decision procedure for A* which is elementary recursive if TH(A) is and if M(n,k) is. If this is done carefully, one can obtain Theorem 3.2 a more quantitative version of Theorem 3.1. Corollary 3.3 follows immediately from Theorem 3.2.

Theorem 3.2 Say that $T_1: N \to N$ is an increasing function such that TH(A) can be decided by some algorithm within time $T_1(n)$, and such that $T_1(n) \geq 2^n$ for all $n \in N$. Say that $T_2: N \to N$ is an increasing function such that $T_2(k+k') \geq M(k,k')$ for all $k,k' \in N$.

Then there is an algorithm for deciding $TH(A^*)$ which operates within time $[T_1((T_2(n))^{cn})]^c$ for some constant c.

Corollary 3.3 Let $s_1, s_2, c \in N$, $s_1 \geq 1$ and $s_2 \geq 2$, such that $TH(A)$ can be decided within time

$$2^{2^{\cdot^{\cdot^{\cdot^{2^{cn}}}}}} \Big\} \text{height } s_1$$

and such that $M(n,k) \leq 2^{2^{\cdot^{\cdot^{\cdot^{2^{c(n+k)}}}}}} \Big\} \text{height } s_2$

for all $n,k \in N$.

Then $TH(A^*)$ can be decided within time $2^{2^{\cdot^{\cdot^{\cdot^{2^{c'n}}}}}} \Big\} \text{height } s_1+s_2$

for some constant c'. □

We lastly remark that the converse of Theorem 3.1 is false. Let S be a set of positive integers > 1, and let A_S be the structure defined at the end of Section 2 of Chapter 1. It is not hard to see that $(A_S)^*$ consists of an infinite number of infinite equivalence classes. Hence, one can show that $TH(A_S)^*$ is elementary-recursive, and its complexity doesn't depend on S. However, S can be chosen so as to make $TH(A_S)$ nonrecursive or arbitrarily hard to decide.

Section 4: Results about Other Kinds of Direct Products

In this section we state some results about other kinds of direct products, thus giving quantitative versions of some additional theorems of Mostowski and Feferman and Vaught [Mos52,FV59]. We will not present proofs here, but our results follow from extensions of the ideas in the preceding parts of this chapter.

Definition 4.1 Let I be a nonempty set, and let $(A^{(i)}|i \in I)$ be a collection of structures for L, indexed by I; say that $A^{(i)} = \langle A^{(i)}, R_1^{(i)}, R_2^{(i)}, \ldots, R_\ell^{(i)}, e^{(i)} \rangle$ for all $i \in I$. Let $D = \{f: I \to \bigcup_{i \in I} A^{(i)} | f(i) \in A^{(i)} \text{ for } i \in I\}$. For each j, $1 \leq j \leq \ell$, define $R_j \subseteq D^{t_j}$ as follows: if $\bar{f}_{t_j} \in D^{t_j}$, then $\bar{f}_{t_j} \in R_j$ iff $\bar{f}_{t_j}(i) \in R_j^{(i)}$ for all $i \in I$. Define $e \in D$ by $e(i) = e^{(i)}$ for all $i \in I$. Define the strong direct product of the system $(A^{(i)}|i \in I)$ by $\text{STRONG}(A^{(i)}|i \in I) = \langle D, R_1, R_2, \ldots, R_\ell, e \rangle$ Let $D' \subseteq D$ be the set $\{f \in D |$ for all but finitely many $i \in I$, $f(i) = e^{(i)}\}$, and let R_j' be the relation R_j restricted to $(D')^{t_j}$ for $1 \leq j \leq \ell$. Define the weak direct product of the system $(A^{(i)}|i \in I)$ by $\text{WEAK}(A^{(i)}|i \in I) = \langle D', R_1', R_2', \ldots, R_\ell', e \rangle$. If I is finite, then $\text{STRONG}(A^{(i)}|i \in I) = \text{WEAK}(A^{(i)}|i \in I)$.

If we take I to be N and $A^{(i)} = A$ for some fixed structure A and all $i \in N$, then we denote $\text{STRONG}(A^{(i)}|i \in N)$ by A^ω and call it the strong direct power of A; $\text{WEAK}(A^{(i)}|i \in N)$ is A^*, the weak direct power of A, which was defined earlier. If C

is a nonempty collection of structures, then STRONG(C) is the class $\{STRONG(A^{(i)}) | i \in I) | I$ is a set and $A^{(i)} \in C$ for $i \in I\}$ and WEAK(C) is the class

$\{WEAK(A^{(i)}) | i \in I) | I$ is a set and $A^{(i)} \in C$ for $i \in I\}$.

Mostowski shows that if TH(A) is decidable, then $TH(A^{\omega})$ is decidable. Feferman and Vaught show that $TH(STRONG(C)) = TH(\{STRONG(A^{(i)}) | i \in I) | I$ is a finite set and $A^{(i)} \in C$ for $i \in I\})$, and if TH(C) is decidable, then TH(STRONG(C)) and TH(WEAK(C)) are decidable. We can prove stronger versions of these theorems.

Theorem 4.1 Let A be a structure and let M(n,k) be defined as in Chapter 2. Say that $T_1: N \to N$ is an increasing function such that TH(A) can be decided by some algorithm within time $T_1(n)$ and such that $T_1(n) \geq 2^n$ for all $n \in N$. Say that $T_2: N \to N$ is an increasing function such that $T_2(k+k') \geq M(k,k')$ and for all $k,k' \in N$.

Then there exists an algorithm for deciding $TH(A^{\omega})$ which operates within time $[T_1((T_2(n))^{cn})]^c$ for some constant c.

Definition 4.2 If C is a collection of structures, let $INFSTRONG(C) = \{STRONG(A^{(i)}) | i \in I) | I$ is an infinite set and $A^{(i)} \in C$ For $i \in I\}$.

Let $INFWEAK(C) = \{WEAK(A^{(i)}) | i \in I) | I$ is an infinite set and $A^{(i)} \in C$ for $i \in I\}$.

Theorem 4.3 Let C be a nonempty collection of structures and for each $A \in C$, let $M_A(n,k)$ be defined for A as in Chapter 2. Say that $T_1:N \to N$ is an increasing function such that $TH(C)$ can be decided by some algorithm within time $T_1(n)$ and such that $T_1(n) \geq 2^n$ for all $n \in N$. Say that $T_2:N \to N$ is an increasing function such that $T_2(k+k') \geq M_A(k,k')$ for all $k,k' \in N$ and all $A \in C$.

Then there exists algorithms for deciding TH(STRONG(C)), TH(INFSTRONG(C)), TH(WEAK(C)), and TH(INFWEAK(C)) which operate within time $[T_1(2^{(T_2(n))^{cn}})]^c$ for some constant c.

It is important to note that in Theorems 3.2, 4.1 and 4.3, the decision procedure that is produced is obtained effectively from the one that is given. For instance, in Theorem 4.3 TH(STRONG(C)) is completely determined by TH(C).

Now let C be the collection of finite cyclic group structures. Since every finite abelian group is isomorphic to a finite direct product of finite cyclic groups, the first order theory of finite abelian groups is the same as TH(STRONG(C)). TH(C) is decidable, and we could have used the technique involved in proving Theorem 4.3 to prove Theorem 2.8. Every finitely generated abelian group is isomorphic to a finite direct product of cyclic groups [MB68]. So if C' is the collection of cyclic group structures then TH(STRONG(C')) is the first order theory

of finitely generated abelian groups. But using results of [Szm55] it can be shown that TH($\mathcal{C}$) = TH($\mathcal{C}'$), and so by Theorem 2.8 we see that TH(STRONG($\mathcal{C}'$)) can also be decided within space $2^{2^{2^{cn}}}$ for some constant c.

CHAPTER 6

LOWER BOUND PRELIMINARIES

Section 1: Simple Turing Machines

We now present the basic model of computation used in our lower bound results, the simple Turing machine or STM. Most of the proofs of such results entail the efficient arithmetization of STM's, and for this reason we give a detailed account of them.

Definition 1.1 A (nondeterministic) STM is a six-tuple $M = (\Sigma,\Gamma,S,\delta,q_0,q_a)$ consisting of a finite set Γ (the tape alphabet), a finite set $\Sigma \subseteq \Gamma$ (the input alphabet), a finite set S (the set of states), a transition function $\delta: S \times \Gamma \to P(S\times\Gamma\times\{-1,0,1\})$, designated states $q_0 \in S$ (the initial state) and $q_a \in S$ (the accept state).

M is deterministic if $|\delta(q,s)| = 1$ for all $q \in S$, $s \in \Gamma$. An instantaneous description (i.d.) of M is any word in $\Gamma^*S\Gamma^*$. Given $\$ \notin \Gamma \cup S$, and an i.d. d of M, d with the end-marker \$ is \$d\$.

Informally, STM's are single-tape single-head Turing machines, whose tape is infinite only to the right. No move shifts the head off the left end of the tape.

I.d.'s give the particular local tape configuration, state of the machine and symbol being scanned, at a particular step. STM's accept their input only in state q_a, and only in the following configuration: the entire tape is blank, and the head is scanning the leftmost tape square. Thus for each $n \in N$, there is a unique accepting i.d. of length n+1; this i.d. which is $q_a\not{b}^n$ where $\not{b}$ denotes the blank tape symbol, is denoted by acc(n). In addition, we will insist that state q_a can only be entered on accepting input, and that STM's never halt.

We next define a function $\text{Next}_M: \Gamma^* S \cdot \Gamma^* \to P(\Gamma^* \cdot S \cdot \Gamma^*)$ which maps i.d.'s of M to the set of i.d.'s which can occur one step after the given i.d. We follow the convention that the i.d.'s in $\text{Next}_M(d)$ are all of the same length as d, for all i.d.'s d.

We first define

Definition 1.2 Let $M = (\Sigma, \Gamma, S, \delta, q_0, q_a)$ be an STM. Let $d = yqsz$, $y, z \in \Gamma^*$, $q \in S$, $s \in \Gamma$. Let $u = (q', s', m') \in S \times \Gamma \times \{-1, 0, 1\}$

$$\text{Next1}_M(d,u) = \begin{cases} \{yq's'z\} & \text{if } m=0 \\ \{ys'q'z\} & \text{if } m=1 \\ \{wq'ts'z\} & \text{if } m=-1 \text{ and} \\ & \quad y=wt \text{ for some } w \in \Gamma^* \text{ and} \\ & \quad t \in \Gamma \\ \phi & \text{if } m=-1 \text{ and } y=\Lambda\ . \end{cases}$$

$$\text{Next}_M(d) = \begin{cases} \bigcup_{u \in \delta(q,s)} \text{Next1}_M(d,u) & \text{if } d=yqsz \text{ as above} \\ \phi & \text{if } d=yq \text{ for some} \\ & y \in \Gamma^*,\ q \in S. \end{cases}$$

We now define $\text{Next}_M(d, \ell)$, the set of i.d.'s occurring ℓ steps after d:

Definition 1.3 Let M be an STM. If d is any i.d. of M, we define inductively

$$\text{Next}_M(d,0) = \{d\}.$$

$$\text{Next}_M(d,\ell+1) = \{d'' \mid d'' \in \text{Next}_M(d') \text{ for some } d' \in \text{Next}_M(d,\ell)\}.$$

If d is an i.d. of M, and $\$ \notin \Gamma \cup S$, we define

$\text{Next}_M(\$d\$) = \{\$e\$ \mid e \in \text{Next}_M(d)\}$; $\text{Next}_M(\$d\$, \ell)$ for $\ell \in N$ is then defined as above.

We next give our criterion for acceptance:

Definition 1.4 Let $M = (\Sigma, \Gamma, S, \delta, q_0, q_a)$ be an STM, $A \subseteq \Sigma^*$, $\not\!b \in \Gamma$ the blank tape symbol. M accepts A within time $T(n)$ (within space $S(n) \geq n$) iff

1. for all $w \in A$, there are $\ell, k \in N$ with $\ell \leq T(\ell n(w))$ and $k \geq \ell n(w)$ (resp., with $\ell n(w) \leq k \leq S(\ell n(w))$) such that
   $q_a \not\!b^k \in \text{Next}_M(q_0 w \not\!b^{k-\ell n(w)}, \ell)$, and
2. for all $w \in \Sigma^* - A$, there do not exist $\ell, k \in N$ and $y, z \in \Gamma^*$ such that $y q_a z \in \text{Next}_M(q_0 w \not\!b^{k-\ell n(w)}, \ell)$, and
3. for all $w \in \Sigma^*$, and all $\ell, k \in N$, $\text{Next}_M(q_0 w \not\!b^k, \ell) \neq \phi$.

We next state without proof a technical lemma which makes precise the idea that $\text{Next}_M(d_1) = d_2$ iff a sequence of "local checks" of three successive digits of d_2 (based on the corresponding three digits of d_1) succeed. This lemma forms the heart of our arithmetizations of Turing machines.

Lemma 1.5 Let $M = (\Sigma, \Gamma, S, \delta, q_0, q_a)$ be an STM. Assume $\$ \notin \Gamma \cup S$. Let $\Delta = \Gamma \cup S \cup \{\$\}$. There is a function $N_M: \Delta^3 \to P(\Delta^3)$ with the following property:
if d is any i.d. of M with the end-marker \$, $d = d_1 \cdot \ldots \cdot d_k$, with $d_i \in \Delta$ for $i \in I_k$, and if $f = f_1 \cdot \ldots \cdot f_k \in \Delta^*$, with $f_i \in \Delta$ for $i \in I_k$, then

$$f \in \text{Next}_M(d) \text{ iff } f_{j-1} f_j f_{j+1} \in N_M(d_{j-1} d_j d_{j+1})$$

for $1 < j < k$.

For a proof of Lemma 1.5, see [Sto74].

We also find the following corollary useful:

Definition 1.6 Let $M = (\Sigma,\Gamma,S,\delta,q_0,q_a)$ be an STM. Assume $\$ \notin \Gamma \cup S$, and let $\Delta = \Gamma \cup S \cup \{\$\}$. $d \in \Delta^*$ is an accepting computation of M on input w if for some $k \in N$ and some $\ell \in I$, and $n = \ell(w)+k$,

$d = d_1\$d_2\$\cdot\ldots\cdot\$d_\ell$, with

$d_i \in (\Gamma \cup S)^{n+1}$ for $i \in I_\ell$, and

1. $d_1 = q_0 w \not\beta^k$,
2. $d_{i+1} \in Next_M(d_i)$ for $i \in I_{\ell-1}$,
3. $d_\ell = acc(n)$.

Corollary 1.7 Let $M = (\Sigma,\Gamma,S,\delta,q_0,q_a)$ be an STM, with $\$ \notin \Gamma \cup S$, and $\Delta = \Gamma \cup S \cup \{\$\}$. There is a function $N_M\colon \Delta^3 \to P(\Delta^3)$ with the following property:

For any $w \in \Sigma^*$, $d \in \Delta^*$ is an accepting computation of M on input w <=> for some $k \in N$ and $\ell \in I$, with $n = \ell n(w)+k$, the following hold:

1. $d = d_1\cdot\ldots\cdot d_\ell$, with $d_i \in (\Gamma \cup S)^{n+2}$ for $i \in I_\ell$,
2. $d_1 = q_0 w \not\beta^k\$$
3. q_a appears in d
4. For all $i \in I_{\ell-1}$ if $d_i = d_{i,1}d_{i,2}\cdot\ldots\cdot d_{i,n+2}$ and $d_{i+1} = d_{i+1,1}d_{i+1,2}\cdots d_{i+1,n+2}$ then for all $j \in I_{n+1}$ $d_{i+1,j-1}d_{i+1,j}d_{i+1,j+1} \in N_M(d_{i,j-1}d_{i,j}d_{i,j+1})$.

Section 2: Regular-like Expressions

Most of our lower bound results are obtained by the efficient arithmetization of Turing machines which run in a bounded amount of time or space; however, one of our results will be obtained by the efficient reducibility of an already established "hard" problem into a theory. We now define this problem.

Definition 2.1 We simultaneously define the $\{0,1\}$-$\{\cup,\cdot,^2\}$ expressions, and the language $L(E)$ defined by each such expression E, as follows.

1. (0) is a $\{0,1\}$-$\{\cup,\cdot,^2\}$ expression, and $L((0)) = \{0\}$.
2. (1) is a $\{0,1\}$-$\{\cup,\cdot,^2\}$ expression, and $L((1)) = \{1\}$.
3. If E_1 and E_2 are $\{0,1\}$-$\{\cup,\cdot,^2\}$ expressions

$$\left\{\begin{array}{c}(E_1 \cup E_2)\\ \hline (E_1 \cdot E_2)\\ \hline (E_1^2)\end{array}\right\} \text{ is a } 0,1\text{-}\{\cup,\cdot,^2\} \text{ expression,}$$

and

$$L(E) = \left\{\begin{array}{l}L(E_1) \cup L(E_2)\\ L(E_1) \cdot L(E_2)\\ L(E_1) \cdot L(E_1)\end{array}\right\}.$$

Definition 2.2 We define

$$INEQ(\{0,1\},\{\cup,\cdot,^2\}) = \{(E_1,E_2) \mid L(E_1) \neq L(E_2)\}.$$

The following theorem is due to Stockmeyer:

Theorem 2.3 [Sto74] There is a rational $d>1$ such that

$$INEQ(\{0,1\},\{\cup,\cdot,^2\}) \notin NTIME(d^n).$$

CHAPTER 7

A TECHNIQUE FOR WRITING SHORT FORMULAS DEFINING COMPLICATED PROPERTIES

The method we shall use for showing theories difficult to decide is to efficiently reduce to them problems already known to be difficult. For instance, if $NTIME(2^n) \leq_{p\ell} TH(C)$, then we know from Chapter 1 that $TH(C)$ requires exponential time to decide. To exhibit this reducibility we will have to create, given a word of length n, special formulas of length proportional to n in time polynomial in n. Often we will wish to define a sequence of formulas $F_0, F_1, \ldots, F_n$ where we define F_{i+1} from F_i, and where we want length (F_i) to be proportional to i.

As an example, consider the language L with atomic formulas of the form $z_1 = z_2+1$ and $z_1 = z_2$, and consider the natural interpretation A with the domain N. Say that we wish to define formulas $F_i(x,y)$ for $i = 0,1,2,\ldots$ such that $A \models F_i(a,b) \iff b = a+2^i$. We first define $F_0(x,y)$ to be $y = x+1$. The most obvious thing to do next is to define $F_{i+1}(x,y)$ as $\exists z(F_i(x,z) \wedge F_i(z,y))$ for $i > 0$. The trouble with this is that length (F_i) will be exponential in i since F_i occurs more than once in the definition of F_{i+1}. We could avoid this problem, however, by defining F_{i+1} as $\exists z \forall w_1 w_2(((w_1=x \wedge w_2=z) \vee (w_1=z \wedge w_2=y)) \rightarrow F_i(w_1,w_2))$, where w_1, w_2 are new variables. In this way, length (F_i) is order i log i. If we were even more careful, then we could define F_i in such a way as to use a <u>fixed</u> number of variables (independent of i), and thus obtain F_i of length order i.

Theorems 2 and 8 are generalizations of the techniques of the preceding paragraph. Theorem 3 is due to Fischer and Meyer [cf. FiR74] working from earlier ideas of Stockmeyer [SM73]. In order to be able to state our theorems, we have to have the notion of a property relative to a class of structures.

Let L be the language of the first order predicate calculus with a finite number of relational symbols $\underline{R}_1,\underline{R}_2,\ldots,\underline{R}_\ell$. Let C be a nonempty class of structures for L. By a k-place property P we mean a set of elements of the form $(A,\bar{a}_k)$ where $A \in C$. Sometimes we will indicate that P is a k-place property by writing $P(\bar{x}_k)$. Sometimes we will write $A \models P(\bar{a}_k)$, or $(A,\bar{a}_k) \models P$, instead of $(A,\bar{a}_k) \in P$. If $F(\bar{x}_k)$ is a formula of L, then by the (k-place) property defined by F we mean $\{(A,\bar{a}_k) \mid A \in C \text{ and } A \models F(\bar{a}_k)\}$.

Theorem 2 below will essentially say the following: given a sequence of properties $G_0,G_1,\ldots$ such that G_0 is defined by a formula of L and such that G_{i+1} can be expressed in a fixed way (independent of i) from G_i using the language L, then for every $i > 0$ there is a formula of L of length propertional to i which defines the property G_i.

We assume that equality is definable in C, and hence for convenience assume that $v_1 = v_2$ is an atomic formula of L. We also assume that every structure in C has a domain of cardinality ≥ 2. These assumptions are convenient but not necessary. Although it appears that the use of $\leftrightarrow$ is crucial in constructing our short formulas, an idea due to R. Solovay

(personal communication) would allow us to prove these theorems even if $\leftrightarrow$ were not in our language.

Now let $k \in N$ be fixed and let L' be the language of the first order predicate calculus which is the same as L except that a k-place formal predicate $\underline{R}$, has been added. Two formulas of L' are equivalent if they are equivalent in any structure obtained by adding to a structure from $\mathcal{C}$ an interpretation for $\underline{R}$.

Definition 1 Let $\underline{F}(\bar{x}_k)$ be a formula of L' and let $G(\bar{x}_k)$ be a property. We define an infinite sequence of properties, $G_0(\bar{x}_k), G_1(\bar{x}_k), \ldots$ as follows: Let $G_0(\bar{x}_k)$ be $G(\bar{x}_k)$. For every $i \in N$ and for every structure $A \in \mathcal{C}$ with domain A and for every $\bar{a}_k \in A^k$, we say that $A \models G_{i+1}(\bar{a}_k)$ iff $A \models \underline{F}(\bar{a}_k)$ when the formal predicate $\underline{R}$ is interpreted in A as G_i (restricted to A).

Theorem 2 Let $\underline{F}(\bar{x}_k)$ be a formula of L' and let $\underline{G}(\bar{x}_k)$ be a formula of L defining the property $G(\bar{x}_k)$. Let $G_0(\bar{x}_k), G_1(\bar{x}_k), \ldots$ be the properties defined in Definition 1. Then there exists a sequence $\underline{G}_0(\bar{x}_k), \underline{G}_1(\bar{x}_k), \ldots$ of formulas of L such that

(I) $\underline{G}_i$ defines the property G_i for each $i \in N$.

(II) There is a procedure which given $i \in I$ computes $\underline{G}_i$ within time a fixed polynomial in i, and space linear in i including the length of the output $\underline{G}_i$.

Lemma 3 is a key part of the proof of Theorem 2.

Lemma 3 Let $\underline{F}$ be a formula of L'. Then there exists a formula $\underline{F}'$ of L' equivalent to $\underline{F}$ such that $\underline{F}'$ has exactly one occurrence

of the predicate letter $\underline{R}$; this occurs in an atomic formula in which all the k formal variables are distinct.

Proof Let $\underline{F}$ be a formula of L' . Since any formula of L' can trivially be extended to an equivalent one with at least one occurrence of $\underline{R}$, assume that $\underline{F}$ contains at least one occurrence of $\underline{R}$. Assume $\underline{F}$ is in prenex normal form so that $\underline{F}$ looks like $Q_1 v_1 Q_2 v_2 \ldots Q_J v_J \underline{A}$ where $\underline{A}$ is a quantifier free formula containing $m \geq 1$ occurrences of the symbol $\underline{R}$ and where $v_1, v_2, \ldots, v_J$ represent formal variables. Let us say that the m atomic formulas of $\underline{A}$ in which $\underline{R}$ occurs, from left to right are $\underline{R}(v_{11}, v_{12}, \ldots, v_{1k}), \underline{R}(v_{21}, v_{22}, \ldots, v_{2k}), \ldots, \underline{R}(v_{m1}, v_{m2}, \ldots, v_{mk})$ where the symbols $v_{i,j}$ for $1 \leq i \leq m$ and $1 \leq j \leq k$ represent formal variables.

Let $y_1, y_1', y_2, y_2', \ldots, y_m, y_m'$ be distinct formal variables not appearing in $\underline{A}$. Let $\underline{A}'$ be the formula obtained from $\underline{A}$ by replacing $\underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})$ by $y_i = y_i'$ for $1 \leq i \leq m$. Since in every structure of $\mathfrak{C}$ there are interpretations of y and y' which cause the formula $y = y'$ to be true, and interpretations which cause $y = y'$ to be false, we see that $\underline{A}$ is equivalent to

$$\exists y_1 \exists y_1' \exists y_2 \exists y_2' \ldots \exists y_m \exists y_m' (\underline{A}' \wedge \bigwedge_{1 \leq i \leq m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]).$$

Now let $y, y', z_1, z_2, \ldots, z_k$ be distinct formal variables not occurring in $\bigwedge_{1 \leq i \leq m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]$.

$\bigwedge_{1 \leq i \leq m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]$ is equivalent to

$$\forall y \forall y' \forall z_1 \ldots \forall z_k [(\bigvee_{1 \leq i \leq m} (y = y_i \wedge y' = y_i' \wedge z_1 = v_{i1} \wedge z_2 = v_{i2} \wedge \ldots \wedge z_k = v_{ik})) \rightarrow ((y = y') \leftrightarrow \underline{R}(z_1, z_2, \ldots, z_k))].$$

So we have shown that $\underline{F}$ is equivalent to a formula with exactly one occurrence of $\underline{R}$, which occurs in the atomic formula $\underline{R}(\bar{z}_k)$. □

Definition 4 Let $\underline{F}(\bar{x}_k)$ be a formula of L and let $z_1, z_2, \ldots, z_k$ be distinct variables all of which are different from $x_1, x_2, \ldots, x_k$. Then let $\underline{F}^{(\bar{z}_k | \bar{x}_k)}(\bar{z}_k)$ be the formula obtained from $\underline{F}$ in the following way: If v is an occurrence (not necessarily free) of a formal variable in $\underline{F}$, then if $v = z_i$ for some i, $1 \leq i \leq k$, replace v by x_i; if $v = x_i$ for some i, $1 \leq i \leq k$, replace v by z_i.

Definition 5 If $\underline{F}$ is a formula of L, define the size of $\underline{F}$, $s(\underline{F})$, to be the length of $\underline{F}$ when each variable subscript is counted to be of length 1 and all other symbols are counted normally.

The following lemma follows immediately from the definitions.

Lemma 6 Let $\underline{F}(\bar{x}_k)$ and $\underline{F}^{(\bar{z}_k | \bar{x}_k)}(\bar{z}_k)$ be as in Definition 4. Then $s(\underline{F}) = s(\underline{F}^{(\bar{z}_k | \bar{x}_k)})$, and $\underline{F}(\bar{x}_k)$ and $\underline{F}^{(\bar{z}_k | \bar{x}_k)}(\bar{z}_k)$ define the same property.

Proof of Theorem 2 Let $\underline{F}(\bar{x}_k)$ be a formula of L' and let $\underline{G}(\bar{x}_k)$ be a formula of L defining the property $G(\bar{x}_k)$. By Lemma 3 assume that $\underline{F}$ contains exactly one occurrence of $\underline{R}$; the proof of Lemma 3 assures us in fact that we can insist that the atomic formula in which $\underline{R}$ occurs is $\underline{R}(\bar{z}_k)$ where $z_1, z_2, \ldots, z_k$ are distinct variables not occurring in $\{x_1, x_2, \ldots, x_k\}$.

Now define a sequence $\underline{G}_0(\bar{x}_k), \underline{G}_1(\bar{x}_k), \ldots$ of formulas of L as follows. Let $\underline{G}_0$ be $\underline{G}$. For all $i \in N$, let $\underline{G}_{i+1}$ be the formula obtained by substituting $\underline{G}_i^{(\bar{z}_k \mid \bar{x}_k)}$ for $\underline{R}(\bar{z}_k)$ in $\underline{F}$. It is easy to see by induction (using Lemma 6) that $\underline{G}_i(\bar{x}_k)$ defines $G_i(\bar{x}_k)$ for each $i \in N$.

For $c_0 = \text{length}(\underline{F})$ we have $s(\underline{G}_{i+1}) \leq c_0 + s(\underline{G}_i^{(\bar{z}_k \mid \bar{x}_k)}) = c_0 + s(\underline{G}_i)$ for $i \in N$, so $s(\underline{G}_i) \leq s(\underline{G}) + i \cdot c_0$. Every variable occurring in each $\underline{G}_i$ is either from the set $\{x_1, x_2, \ldots, x_k\}$ or occurs in $\underline{F}$ or occurs in $\underline{G}$. If c_1 is the maximum length of any such variable subscript, then length $(\underline{G}_i) \leq c_1 \cdot s(\underline{G}_i) \leq c_1 \cdot (s(\underline{G}) + i \cdot c_0) \leq c \cdot i$ for $i \in I$ and some constant c independent of i. It can also be checked that one can compute $\underline{G}_i$ within time polynomial in i and space linear in i. □

Theorem 2 can be generalized in a number of ways. We will only present the particular generalization which we need in the text.

To begin with, let L" be the language of the first order predicate calculus which is the same as L except that we have added two new formal k-place predicates: $\underline{R}$ and $\underline{R}'$ for some fixed $k \in N$.

Definition 7 Let $\underline{F}_0(\bar{x}_k), \underline{F}_1(\bar{x}_k), \underline{F}_0'(\bar{y}_k), \underline{F}_1'(\bar{y}_k)$ be formulas of L" . Let $G(\bar{x}_k)$ and $G'(\bar{y}_k)$ be properties. For every $\gamma \in \{0,1\}^*$ we let $G_\gamma(\bar{x}_k)$ and $G'_\gamma(\bar{y}_k)$ be properties as follows: If λ is the empty string, let G_λ be G and let G'_λ be G'. For every $\delta \in \{0,1\}^*$

and every $A \in \mathbb{C}$ with domain A and every $\bar{a}_k \in A^k$ we say $A \models G_{\delta i}(\bar{a}_k)$ (where $i \in \{0,1\}$) iff $A \models \underline{F}_i(\bar{a}_k)$ when $\underline{R}$ is interpreted as G_δ (restricted to A) and $\underline{R}'$ is interpreted as G'_δ; we say $A \models G'_{\delta i}(\bar{a}_k)$ iff $A \models \underline{F}'_i(\bar{a}_k)$ when $\underline{R}$ is interpreted as G_δ and $\underline{R}'$ is interpreted as G'_δ.

Theorem 8 Let $\underline{F}_0, \underline{F}_1, \underline{F}'_0, \underline{F}'_1$ be formulas of L" and let $\underline{G}(\bar{x}_k), \underline{G}'(\bar{y}_k)$ be formulas of L defining, respectively, the properties $G(\bar{x}_k)$ and $G'(\bar{y}_k)$. For each $\gamma \in \{0,1\}^*$, let $G_\gamma(\bar{x}_k)$ and $G'_\gamma(\bar{y}_k)$ be as in Definition 7. Assume that for any $A \in \mathbb{C}$, the relations obtained by restricting G_γ and G'_γ to A are both nonempty. Then for each $\gamma \in \{0,1\}^*$ there exist formulas $\underline{G}_\gamma(\bar{x}_k), \underline{G}'_\gamma(\bar{y}_k)$ such that

(I) $\underline{G}_\gamma$ defines G_γ and $\underline{G}'_\gamma$ defines G'_γ.

(II) There is a procedure which given $\gamma \in \{0,1\}^+$ computes $\underline{G}_\gamma$ and $\underline{G}'_\gamma$ within time a fixed polynomial in $\ell n(\gamma)$, and space linear in $\ell n(\gamma)$ including the lengths of $\underline{G}_\gamma$ and $\underline{G}'_\gamma$.

Proof The basic idea of this proof, discovered independently in [FiR74], is what we call "simultaneous definition"; for every $\gamma \in \{0,1\}^*$ we will write down a formula which defines both G_γ and G'_γ, as described below.

For each γ, let $H_\gamma(\bar{x}_k, \bar{y}_k)$ be a 2k-place property which we define informally to be "$G_\gamma(\bar{x}_k) \wedge G'_\gamma(\bar{y}_k)$"; more formally, if $A \in \mathbb{C}$ with domain A and $\bar{a}_k, \bar{b}_k \in A^k$, then we say $A \models H_\gamma(\bar{a}_k, \bar{b}_k) \iff A \models G_\gamma(\bar{a}_k)$ and $A \models G'_\gamma(\bar{b}_k)$. The formula $\underline{H}_\lambda(\bar{x}_k, \bar{y}_k) = \underline{G}(\bar{x}_k) \wedge \underline{G}'(\bar{y}_k)$ defines $H_\lambda(\bar{x}_k, \bar{y}_k)$.

Let $\delta \in \{0,1\}^*$ and let $i \in \{0,1\}$. We now show informally (this will be made precise below) how $H_{\delta i}$ can be expressed from H_δ: It is sufficient to show that $G_{\delta i}$ and $G'_{\delta i}$ can be expressed from H_δ. Using $\underline{F}_i$ and $\underline{F}'_i$ we can express $G_{\delta i}$ and $G'_{\delta i}$ by using G_δ and G'_δ. Since for any $A \in \mathcal{C}$ with domain A and any $a_k \in A^k$,

$$A \models G_\delta(\bar{a}_k) \iff \text{for some } \bar{b}_k \in A^k,\ A \models H_\delta(\bar{a}_k,\bar{b}_k), \text{ and}$$
$$A \models G'_\delta(\bar{a}_k) \iff \text{for some } \bar{b}_k \in A^k,\ A \models H_\delta(\bar{b}_k,\bar{a}_k),^\dagger$$

we see that G_δ and G'_δ can be expressed from H_δ.

Proceeding more formally, let L''_0 be the language of the first order predicate calculus obtained from L by adding a 2k-place formal predicate $\underline{U}$. Let $w_1,w_2,\ldots,w_k$ be distinct variables not occurring in $\underline{F}_0,\underline{F}_1,\underline{F}'_0,\underline{F}'_1$. For $i \in \{0,1\}$, let $\underline{F}_i(\bar{x}_k)$ be the formula of L''_0 obtained from $\underline{F}_i$ by substituting $\exists w_1 \exists w_2,\ldots,\exists w_k \underline{U}(\bar{v}_k,\bar{w}_k)$ for $\underline{R}(\bar{v}_k)$ every time $\underline{R}$ appears (where $v_1,v_2,\ldots v_k$ represent formal variables), and substituting $\exists w_1 \exists w_2 \ldots \exists w_k \underline{U}(\bar{w}_k,\bar{v}_k)$ for $\underline{R}'(\bar{v}_k)$ every time $\underline{R}'$ appears; obtain $\underline{F}'_i(\bar{y}_k)$ from $\underline{F}'_i$ in the same manner.

For $i \in \{0,1\}$, define the formula $\underline{T}_i(\bar{x}_k,\bar{y}_k)$ of L''_0 as $\underline{F}_i(\bar{x}_k) \wedge \underline{F}'_i(\bar{y}_k)$. One can now see that for $\delta \in \{0,1\}^*$, $i \in \{0,1\}$, $A \in \mathcal{C}$ with domain A, and $\bar{a}_k,\bar{b}_k \in A^k$, we have $A \models G_{\delta i}(\bar{a}_k) \iff A \models \underline{F}_i(\bar{a}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to A, $A \models G'_{\delta i}(\bar{b}_k) \iff A \models \underline{F}'_i(\bar{b}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to A, and therefore $A \models H_{\delta i}(\bar{a}_k,\bar{b}_k) \iff A \models \underline{T}_i(\bar{a}_k,\bar{b}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to A.

---

† Since the relations obtained by restricting G_δ and G'_δ to A are nonempty.

Now let $\{z_1,z_2,\ldots,z_{2k}\}$ be a set of 2k distinct variables not intersecting $\{x_1,x_2,\ldots,x_k,y_1,y_2,\ldots,y_k\}$ or the set of variables in $\underline{T}_0$ and $\underline{T}_1$. Let $\underline{\Gamma}_0(\bar{x}_k,\bar{y}_k)$ and $\underline{\Gamma}_1(\bar{x}_k,\bar{y}_k)$ be formulas of L_0'' such that each contains exactly one occurrence of $\underline{U}$, namely in the atomic formula $\underline{U}(\bar{z}_{2k})$, and such that $\underline{\Gamma}_0$ is equivalent to $\underline{T}_0$ and $\underline{\Gamma}_1$ is equivalent to $\underline{T}_1$. For every $\gamma \in \{0,1\}^*$ define the formula $\underline{H}_\gamma(\bar{x}_k,\bar{y}_k)$ of L as follows. Let $\underline{H}_\lambda(\bar{x}_k,\bar{y}_k)$ be, as before, the formula $\underline{G}(\bar{x}_k) \wedge \underline{G}'(\bar{y}_k)$; for $\delta \in \{0,1\}^*$ and $i \in \{0,1\}$, let $\underline{H}_{\delta i}$ be the formula obtained by substituting, for $\underline{U}(\bar{z}_{2k})$ in T_i, the formula $\underline{H}_\delta^{(\bar{z}_{2k}|(\bar{x}_k,\bar{y}_k))}$. It is now easy to see that $\underline{H}_\gamma(\bar{x}_k,\bar{y}_k)$ defines $H_\gamma(\bar{x}_k,\bar{y}_k)$ for $\gamma \in \{0,1\}^*$. As in the proof of Theorem 2, we can check that length $(\underline{H}_\gamma) \leq c\cdot \ell n(\gamma)$ for $\ell n(\gamma) > 0$. Lastly, for $\gamma \in \{0,1\}^*$, let $\underline{G}_\gamma(\bar{x}_k)$ be $\exists y_1 \exists y_2 \ldots \exists y_k \underline{H}_\gamma(\bar{x}_k,\bar{y}_k)$ and let $\underline{G}'_\gamma$ be $\exists x_1 \exists x_2 \ldots \exists x_k \underline{H}_\gamma(\bar{x}_k,\bar{y}_k)$. It is clear that conditions (I) and (II) of Theorem 8 hold.

CHAPTER 8

A LOWER BOUND ON THE THEORIES OF PAIRING FUNCTIONS

Section 1: Introduction

A pairing function is defined to be a one-one map $\rho: N \times N \to N$. The language L we shall use to talk about pairing functions in this chapter is the usual language of the first order predicate calculus with the formal relation $\rho(v_1,v_2) = v_3$. If $\rho: N \times N \to N$ is a particular pairing function, then we can interpret formulas and sentences of L in the structure $\langle N,\rho\rangle$ in the obvious way; by a P-structure we shall mean a pair $\langle N,\rho\rangle$ where ρ is a pairing function. Let P be the collection of all P-structures. Note that although equality is not a formal predicate of L, we can define equality in P by writing $\forall x(\rho(v_1,v_1) = x \leftrightarrow \rho(v_2,v_2) = x)$, which we will henceforth abbreviate as $v_1 = v_2$ (where v_1 and v_2 represent formal variables) In [Ten74] Richard Tenney refers to some unpublished results of Hanf and Morley which show that $TH(P)$ is undecidable. We will present our own proof of this in Section 2. Tenney also proves that the theories of a large class of pairing functions, including the most common examples, are in fact decidable; however, none of the decision procedures for P-structures that he arrives at are elementary recursive. The major result of this chapter will be that this is an intrinsic difficulty of pairing functions. We shall show that no nonempty collection of P-structures (and hence no single P-structure) has an elementary recursive theory.

Definition 1.1 Define $f: N \to N$ by $f(i) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$ } height i.
that is, $f(0) = 1$ and $f(i+1) = 2^{f(i)}$ for $i \geq 0$.

Theorem 1.2 Let $\mathcal{C}$ be a nonempty collection of P-structures. Then $NTIME(f(n)) \leq_{p\ell} TH(\mathcal{C})$.

Theorem 1.2 will be proved in Sections 3 and 4. Using the methods described in Chapter 1 for proving lower bounds, Theorem 1.2 yields the following corollary.

Corollary 1.3 For some constant $c > 0$, the following is true: Let $\mathcal{C}$ be a nonempty collection of P-structures and let M be a nondeterministic Turing machine which recognizes $TH(\mathcal{C})$. Then for infinitely many n, there is a sentence in $TH(\mathcal{C})$ which M takes at least $f(cn)$ steps to accept.

We have remarked that Tenney shows that many pairing functions have decidable theories; in fact, some of the decision procedures that he presents run within time $f(c'n)$ for some constant c'. So the lower bound of Corollary 1.3 is achievable (except for the value of c).

We conclude this section with some simple generalizations of Corollary 1.3.

Definition 1.4 Let n be an integer > 2. Then an n-ling function is a one-one map $\rho: N^n \to N$. L_n, the language for n-ling functions, is the language of the first order predicate calculus with the formal predicate $\rho(v_1, v_2, \ldots, v_n) = v_{n+1}$. An n-structure is a pair $\langle N, \rho \rangle$ where ρ is an n-ling function.

Corollary 1.5 Let $n > 2$ and let $\mathcal{C}$ be a nonempty collection of n-structures. Then $TH(\mathcal{C})$ has no elementary recursive decision procedure.

Proof Assume for convenience that $n = 3$; the other cases are handled similarly. If ρ is a 3-ling functions and $a \in N$, define the pairing function ρ_a by $\rho_a(a_1,a_2) = \rho(a,a_1,a_2)$. If F is a sentence of $\mathcal{L}$ (the language of pairing functions) and x is a variable not occurring in F, define F'(x) to be the formula of $\mathcal{L}_3$ obtained by replacing every atomic formula of F of the form $\rho(v_1,v_2) = v_3$ by $\rho(x,v_1,v_2) = v_3$. It is easy to see that for any 3-structure $\langle N,\rho\rangle$ and any $a \in N$, $\langle N,\rho\rangle \models F'(a) \iff \langle N,\rho_a\rangle \models F$.

Now let $\mathcal{C}'$ be a nonempty collection of 3-structures and define $\mathcal{C} = \{\langle N,\rho_a\rangle \mid \langle N,\rho\rangle \in \mathcal{C}' \text{ and } a \in N\}$; $\mathcal{C}$ is a nonempty collection of P-structures. Let F be a sentence of $\mathcal{L}$. Then $\mathcal{C} \models F \iff$ for every $\langle N,\rho\rangle \in \mathcal{C}'$ and $a \in N$, $\langle N,\rho_a\rangle \models F \iff$ for every $\langle N,\rho\rangle \in \mathcal{C}'$ and $a \in N$, $\langle N,\rho\rangle \models F'(a) \iff \mathcal{C}' \models \forall x F'(x)$. An elementary recursive decision procedure for $TH(\mathcal{C}')$ would therefore yield an elementary recursive procedure for $TH(\mathcal{C})$, contradicting Corollary 1.3. □

Section 2: Some Undecidability Results

Our goal in this section is to prove that the set of sentences true of all P-structures is not recursive, and that some individual P-structures also have undecidable theories. These proofs are due to the authors and Robert Hossley.

Definition 2.1 Let $F_{REL}(x_1,x_2)$ be the formula $\exists x_3 \exists x_4(\rho(x_1,x_2) = x_3 \wedge \rho(x_3,x_4) = x_4)$.

If $A = \langle N,\rho\rangle$ is a P-structure, define $REL(A) = \{(a_1,a_2) \in N^2 \mid A \models F_{REL}(a_1,a_2)\}$.

Let $N_e \subseteq N$ be the set of even, nonnegative integers.

Lemma 2.2 Let $R \subseteq N_e \times N_e$. Then for some pairing function ρ, $REL(\langle N,\rho\rangle) = R$; furthermore, we can choose ρ to be onto as well as one-one.

Proof Let $(a_1,b_1),(a_2,b_2),\ldots$ be an enumeration of N^2 such that each pair occurs exactly once and such that $b_i \neq 2i$ for each $i \in N^+$. (For instance, we can choose an enumeration $(0,0),(0,1),(1,0),(0,2),(1,1),\ldots$ where the numbers grow sufficiently slowly to ensure that $b_i \neq 2i$.) We will now define the sequence $\rho(a_1,b_1),\rho(a_2,b_2),\ldots$.

Let $n \in N^+$ and assume that $\rho(a_i,b_i)$ has been defined for $0 < i < n$; we now define $\rho(a_n,b_n)$.

Case 1: $(a_n,b_n) \in R$, Define $\rho(a_n,b_n) = 2n$.

Case 2: $b_n = 2i+1$ and $a_n = 2i$ and $(a_i,b_i) \in R$. Define $\rho(a_n,b_n) = b_n$.

Case 3: Otherwise. Let m be the least member of N such that

a) m is not equal to either 2i or 2i+1 for any i such that $(a_i,b_i) \in R$.

b) $m \notin \{\rho(a_i,b_i) \mid i < n\}$

and c) $m \neq b_n$.

Then define $\rho(a_n,b_n) = m$.

We first show that ρ is one-one. Say that $\rho(a_j,b_j) = \rho(a_k,b_k) = J$. If $J = 2i$ where $(a_i,b_i) \in R$, then both $\rho(a_k,b_k)$ and $\rho(a_j,b_j)$ must have been defined via Case 1, so j=k=i. If $J = 2i+1$ where $(a_i,b_i) \in R$, then both $\rho(a_j,b_j)$ and $\rho(a_k,b_k)$ must have been defined via Case 2, so $b_j = b_k = 2i+1$ and $a_j = a_k = 2i$. If we do not have either $J = 2i$ or $J = 2i+1$ where $(a_i,b_i) \in R$, then both $\rho(a_j,b_j)$ and $\rho(a_k,b_k)$ must have been defined via Case 3; by Case 3b), we must have j = k. So ρ is one-one.

We will now show that ρ is onto. Let $m \in N$. Assume that ρ is not defined to take on the value m via either Case 1 or Case 2. Then we do not have $m = 2i$ or $m = 2i+1$ where $(a_i,b_i) \in R$. Let $A = \{(a,b) \in N^2 \mid b \in N_e \text{ and } a \notin N_e \text{ and } b \neq m\}$. ρ cannot have been defined on any member of A via Case 1 or Case 2, so ρ must have been defined on every member of A via Case 3. Since A is infinite, $\{(a,b) \mid \rho \text{ is defined on } (a,b) \text{ via Case 3 and } b \neq m\}$ is infinite. So ρ eventually takes on the value m via Case 3, and hence ρ is onto.

It remains to show that $REL(\langle N,\rho\rangle) = R$. Say that $(a_i,b_i) \in R$. By Case 1, $\rho(a_i,b_i) = 2i$, and by Case 2(since Case 1 doesn't apply to (2i,2i+1)), $\rho(2i,2i+1) = 2i+1$ and hence

$(a_i,b_i) \in REL(<N,\rho>)$. Say that $(a_i,b_i) \in REL(<N,\rho>)$. Then for some $c \in N$ and some $j \in N^+$ we have $(\rho(a_i,b_i),c) = (a_j,c_j)$ and $\rho(a_j,c_j) = c_j$. Since we can't have $c_j = 2j$, ρ cannot have been defined on (a_j,c_j) via Case 1, and looking at Case 3c), we see that ρ cannot have been defined on (a_j,c_j) via Case 3. So ρ was defined on (a_j,c_j) via Case 2. This means that $c_j = a_j+1$ and $a_j = 2k$ where $(a_k,b_k) \in R$; that is, $\rho(a_i,b_i) = 2k$ and $(a_k,b_k) \in R$. $\rho(a_i,b_i)$ cannot therefore have been defined via Cases 2 or 3, and therefore we have that $i = k$ and $(a_i,b_i) \in R$.

□

Definition 2.3 Let L_1 be the language of the first order predicate calculus with only a 2-place formal predicate REL. Define the class of structures for L_1, $\mathcal{C} = \{<D,R> | R \subseteq D^2$ and D = domain R} (where domain R for a 2-place relation R means {a| for some b, (a,b) $\in$ R or (b,a) $\in$ R}).

Lemma 2.4 (Kalmar [cf. Ch56]). TH($\mathcal{C}$) is undecidable.†

Theorem 2.5 a) TH(P) is undecidable.

b) There exist particular P-structures with undecidable theories.

---

†Actually, the theorem as stated by Church is TH($\{<D,R> | R \subseteq D^2\}$) is undecidable, but Lemma 2.4 follows immediately from the proof.

Proof If F is a sentence of L_1, let F' be the sentence of L obtained in the following way:

1) For every quantification Qv in F, change it into a quantification over the values of v which satisfy $\exists x_1 \exists x_2 (F_{REL}(x_1,x_2) \wedge (v=x_1 \vee v=x_2))$, and

2) Replace each atomic formula of F of the form $REL(v_1,v_2)$ by $\exists x_1 \exists x_2 (F_{REL}(x_1,x_2) \wedge x_1=v_1 \wedge x_2=v_2)$. (We are assuming that neither x_1 nor x_2 occur in F.) It is easy to see that for any $A \in \mathcal{P}$ and sentence F of L_1, $\langle domain(REL(A)),REL(A)\rangle \models F \iff A \models F$

Proof of (a) We will show that $\mathcal{C} \models F \iff \mathcal{P} \models F'$.

$\mathcal{C} \models F \Rightarrow$ for all $\langle D,R\rangle \in \mathcal{C}$, $\langle D,R\rangle \models F \Rightarrow$

for all $A \in \mathcal{P}$, $\langle domain(REL(A)),REL(A)\rangle \models F \Rightarrow$

for all $A \in \mathcal{P}$, $A \models F' \Rightarrow \mathcal{P} \models F'$.

Conversely, $\mathcal{P} \models F' \Rightarrow$ for all $A \in \mathcal{P}$ $A \models F' \Rightarrow$

for all $A \in \mathcal{P}$, $\langle domain(REL(A)),REL(A)\rangle \models F \Rightarrow$ (by Lemma 2.2)

for all $\langle D,R\rangle \in \mathcal{C}$ such that $D \subseteq N_e$, $\langle D,R\rangle \models F$.

By the Skolem-Löwenheim theorem [cf. Men64], this implies that for every $\langle D,R\rangle \in \mathcal{C}$, $\langle D,R\rangle \models F$, implying $\mathcal{C} \models F$. So $\mathcal{C} \models F \iff \mathcal{P} \models F'$.

Hence, a decision procedure for TH($\mathcal{P}$) would yield one for TH($\mathcal{C}$), contradicting Lemma 2.4.

Proof of (b) It is easy to see that there exists some $R \subseteq N_e \times N_e$ such that N_e = domain R and $TH(\langle N_e,R\rangle)$ (in L_1) is undecidable.

(We can, for example, choose R to be an equivalence relation so as to make $TH(\langle N_e, R\rangle)$ undecidable, as described in Section 4 of Chapter 2.) By Lemma 2.2 we can find $A = \langle N, \rho\rangle$ such that $REL(A) = R$. Then for any sentence F of L_1 we have $\langle N_e, R\rangle \models F \Longleftrightarrow A \models F'$. So TH(A) is undecidable. □

Remark 2.6 Let $P' = \{\langle N, \rho\rangle \in P \mid \rho \text{ is onto}\}$. The proof of Theore 2.5 shows that (a) TH(P') is undecidable and (b) TH(A) is undecidable for some $A \in P'$.

Section 3: Construction of Formulas which Talk About Large Sets

Our goal in these next two sections is to prove Theorem 1.2, i.e. that $NTIME(f(n)) \leq_{p\ell} TH(\mathcal{C})$ for any nonempty collection $\mathcal{C}$ of P-structures. We shall do this as follows: Let M be a simple (nondeterministic) Turing machine over the alphabet Σ as in Chapter 6. Then for every $w \in \Sigma^+$ we will produce a sentence F_w of L, such that for any P-structure A $A \models F_w \iff$ M accepts w within time $f(\ell n(w))$; furthermore, the time it takes to produce F_w will be polynomial in $\ell n(w)$, and the space needed (including the output) will be linear in $\ell n(w)$. If M operates within time f(n) and $\mathcal{C}$ is a nonempty collection of P-structures, then we have $\mathcal{C} \models F_w \iff$ M accepts w within time $f(\ell n(w)) \iff$ M accepts w, and hence $NTIME(f(n)) \leq_{p\ell} TH(\mathcal{C})$.

The way F_w will "say" that M accepts w within time $f(\ell n(w))$ is as follows: We regard the instantaneous description of a computation of M on w at any time as a string of length about $f(\ell n(w))$, and hence the concatenation of the first $(f(\ell n(w)+1)/f(\ell n(w))$ (which is $\geq f(\ell n(w))$)successive instantaneous configurations is a string of length $f(\ell n(w)+1)$. Using Corollary 1.7 of Chapter 6, F_w will "say" roughly that there exists such a string of length $f(\ell n(w)+1)$ which represents an accepting computation. In order to write such sentences as F_w, we will first have to be able to write down formulas of L of length proportional to n which allow us to describe the basic set-theoretic relations on the subsets of an ordered set of size f(n+1).

The above is an intuitive outline of our approach. The ideas for this outline first appeared in Meyer's proof that WSIS is not elementary recursive [Mey75], and also occur in [FiR74], [Fer74], [MS72], [SM73], [Rob73], [Sto74]. In the rest of this section we shall show how to write formulas of length proportional to n which "talk about" sets of size f(n+1); these theorems do not appeal to any of these previous papers since the development in this section is necessarily intimately connected with the nature of P-structures. In Section 4 we shall present a development along the lines of Meyer, etc., which shows how to use the formulas derived in Section 3 to prove Theorem 1.2.

Let $\langle N,\rho\rangle$ be a P-structure. We first define partial functions $\ell:N \to N$ and $r: N \to N$ as follows: for $a \in N$, $\ell(a) = b$ if for some $c \in N$, $\rho(b,c) = a$; $r(a) = b$ if for some $c \in N$, $\rho(c,b) = a$. Since ρ is one-one, r and ℓ are indeed partial functions. Clearly r and ℓ depend on ρ, but it will always be clear from the context what pairing function a particular r and ℓ come from. Let $\sigma \in \{r,\ell\}^*$ be a string; we define the partial function $f_\sigma:N \to N$ in the obvious way, namely, if λ is the empty string then $f_\lambda(a) = b$ iff $a = b$, and if σ is $\ell\sigma'(r\sigma')$ then $f_\sigma = \ell \circ f_{\sigma'}(= r \circ f_{\sigma'})$. Henceforth we will use σ ambiguously to designate both the string in $\{r,\ell\}^*$ and the function f_σ. We shall sometimes use $|\sigma|$ instead of $\ell n(\sigma)$.

Let $F_\ell(x_1,x_2)$ be the formula $\exists x_3(\rho(x_2,x_3) = x_1)$ and let $F_r(x_1,x_2)$ be the formula $\exists x_3(\rho(x_3,x_2) = x_1)$. Then for any $A \in P$ and any $a,b \in N$, $A \models F_\ell(a,b)$ iff $\ell(a) = b$ and

$A \models F_r(a,b)$ iff $r(a) = b$. Since we will be expressing properties using the partial functions r and ℓ, and since we will be interested in writing down formulas that define these properties, it is important to realize that we will be implicitly using the formulas F_ℓ and F_r.

Definition 3.1 Let $\preceq$ be the reverse lexicographical ordering on $\{r,\ell\}^*$. That is, $\sigma_1 \preceq \sigma_2$ if either $\sigma_2 = \sigma_3\sigma_1$ for some $\sigma_3 \in \{r,\ell\}^*$, or if $\sigma_1 = \sigma_1'\ell\sigma$ and $\sigma_2 = \sigma_2'r\sigma$ for some $\sigma_1',\sigma_2',\sigma \in \{r,\ell\}^*$. $\sigma_1 \prec \sigma_2$ means $\sigma_1 \preceq \sigma_2$ and $\sigma_1 \neq \sigma_2$.

All the properties mentioned in this chapter will be with respect to P.

Definition 3.2 For each $n \in N$, we define the property $ORD_n(x,y_1,y_2)$ as follows: let $\langle N,\rho\rangle \in P$, let $a,b_1,b_2 \in N$. Then $\langle N,\rho\rangle \models ORD_n(a,b_1,b_2)$ iff there exists $\sigma_1,\sigma_2 \in \{r,\ell\}^*$ such that

(I) $|\sigma_1| = |\sigma_2| = f(n)$

(II) $\sigma_1 \preceq \sigma_2$

(III) $\sigma_1 a = b_1$ and $\sigma_2 a = b_2$

Remark 3.3 $\langle N,\rho\rangle \models ORD_n(a,b,b)$ iff for some $\sigma \in \{r,\ell\}^*$, $|\sigma| = f(n)$ and $\sigma a = b$. Clearly $|\{b \mid \langle N,\rho\rangle \models ORD_n(a,b,b)\}| \leq 2^{f(n)} = f(n+1)$.

Definition 3.4 For $n \in N$ we define the property $FULL_n(x)$ as follows: Let $\langle N,\rho\rangle \in P$, let $a \in N$. Then $\langle N,\rho\rangle \models FULL_n(a)$ iff $|\{b \mid \langle N,\rho\rangle \models ORD_n(a,b,b)\}| = f(n+1)$.

Lemma 3.5 Let $\langle N,\rho\rangle$ be a structure and let $n \in N$. Let $\sigma_1,\sigma_2,\ldots,\sigma_{2^n}$ be the increasing (with respect to $\preceq$) sequence of those members of $\{r,\ell\}^*$ of length n. Let $b_1,b_2,\ldots b_{2^n}$ be a sequence of (not necessarily distinct) members of N. Then there exists $a \in N$ such that $\sigma_i a = b_i$ for $1 \leq i \leq 2^n$.

Proof (by induction on n)

Let $\langle N,\rho\rangle$ be a P-structure. Lemma 3.5 is true if n = 0, since we can choose $a = b_1$. So assume the Lemma for n; we will prove it for n+1.

Let $b_1,b_1',b_2,b_2',\ldots,b_{2^n},b'_{2^n}$ be a sequence of members of N of length 2^{n+1}. Define the sequence $c_1,c_2,\ldots,c_{2^n}$ by $c_i = \rho(b_i,b_i')$ for $1 \leq i \leq 2^n$. Let $\sigma_1,\sigma_2,\ldots,\sigma_{2^n}$ be the increasing sequence of those members of $(r,\ell)^*$ of length n. By the induction hypothesis, we can choose $a \in N$ such that $\sigma_i a = c_i$ for $1 \leq i \leq 2^n$. By definition of $\preceq$, $\ell\sigma_1,r\sigma_1,\ell\sigma_2,r\sigma_2,\ldots,\ell\sigma_{2^n},r\sigma_{2^n}$ is the increasing sequence of members of $(\ell,r)^*$ of length n+1. Since $\ell\sigma_i a = \ell c_i = b_i$ and $r\sigma_i a = rc_i = b_i'$, a is the element we were looking for. Hence we are done. □

Lemma 3.6 Let $\langle N,\rho\rangle \in P$ and let $a,n \in N$. Then the following two statements are equivalent.

(I) $\langle N,\rho\rangle \models \mathrm{FULL}_n(a)$

(II) For every $a' \in N$, if $[(\mathrm{ORD}_n(a,b,b) \Rightarrow \mathrm{ORD}_n(a',b,b))$ for all $b \in N]$ then $[(\mathrm{ORD}_n(a',b,b) \Rightarrow \mathrm{ORD}_n(a,b,b))$ for all $b \in N]$

Proof

(I => II): Say that $FULL_n(a)$ holds in $\langle N,\rho\rangle$ and that $a' \in N$ has the property that for all $b \in N$, $\langle N,\rho\rangle \models ORD_n(a,b,b) \Rightarrow \langle N,\rho\rangle \models ORD_n(a',b,b)$. We have $f(n+1) = |\{b|\langle N,\rho\rangle \models ORD_n(a,b,b)\}| \leq |\{b|\langle N,\rho\rangle \models ORD_n(a',b,b)\}| \leq f(n+1)$. Hence $\langle N,\rho\rangle \models ORD_n(a',b,b) \Rightarrow \langle N,\rho\rangle \models ORD_n(a,b,b)$.

(II => I): Say that II is true. Let $A \subseteq N$ be a set of cardinality $f(n+1)$ such that $\{b|\langle N,\rho\rangle \models ORD_n(a,b,b)\} \subseteq A$. By Lemma 3.5 we can choose $a' \in N$ such that $\{b|\langle N,\rho\rangle \models ORD_n(a',b,b)\} = A$, so $\{b|\langle N,\rho\rangle \models ORD_n(a,b,b)\} \subseteq \{b|\langle N,\rho\rangle \models ORD_n(a',b,b)\}$. So by II, $\{b|\langle N,\rho\rangle \mid ORD_n(a,b,b)\} = \{b|\langle N,\rho\rangle \models ORD_n(a',b,b)\} = A$. Hence, $|\{b|\langle N,\rho\rangle \models ORD_n(a,b,b)\}| = |A| = f(n+1)$ and so $\langle N,\rho\rangle \models FULL_n(a)$. □

Remark 3.7 If $\langle N,\rho\rangle \models FULL_n(a)$, then clearly σa is defined for every σ of length $f(n)$; furthermore, if $|\sigma_1| = |\sigma_2| = f(n)$ and $\sigma_1 \neq \sigma_2$, then $\sigma_1 a \neq \sigma_2 a$. Hence $\{(b_1,b_2)|\langle N,\rho\rangle \models ORD_n(a,b_1,b_2)\}$ is a linear ordering on the set $\{b|\langle N,\rho\rangle \models ORD_n(a,b,b)\}$ of cardinality $f(n+1)$.

Lemma 3.6 showed how $FULL_n$ can be expressed from the property ORD_n; the purpose of Lemma 3.8 is to show how ORD_{n+1} can be expressed from ORD_n and $FULL_n$. Let $\langle N,\rho\rangle \in \mathcal{P}$ and let $a,b_1,b_2 \in N$. Lemma 3.8 says that $\langle N,\rho\rangle \models ORD_{n+1}(a,b_1,b_2)$ if and only if there exists some $c \in N$ which "codes" strings $\sigma_1,\sigma_2 \in \{r,\ell\}^*$ of length $f(n+1)$ such that $\sigma_1 a = b_1$, $\sigma_2 a = b_2$, and $\sigma_1 \preceq \sigma_2$. To see how this coding is done, examine Figure 3.1.

Every node in the tree in Figure 3.1 represents a (not necessarily distinct) member of N. The value at a node is ρ of the values of the two sons (if they exist); for instance, $\rho(g,h) = c$. In order for c to code the strings $\sigma_1 = \gamma_{f(n+1)}\cdots\gamma_2\gamma_1$ and $\sigma_2 = \delta_{f(n+1)}\cdots\delta_2\delta_1$ it is necessary that $d_i = \gamma_i\cdots\gamma_2\gamma_1 a$ and $e_i = \delta_i\cdots\delta_2\delta_1 a$ for $1 \leq i \leq f(n+1)$; note that c may code numerous pairs of strings. In order to say that c codes strings σ_1,σ_2 such that $\sigma_1(a) = b_1$ and $\sigma_2(a) = b_2$ and $\sigma_1 \preceq \sigma_2$, one has to be able to talk about the nodes labelled by $d_1,e_1,d_2,e_2,\ldots,e_{f(n+1)}$ and their ordering from left to right, and for this reason we insist that $c_1,c_2,\ldots,c_{f(n+1)}$ all be distinct so that we can talk about their ordering using ORD_n.

Lemma 3.8 Let $\langle N,\rho\rangle \in \mathcal{P}$, let $n \in N$, let $a,b_1,b_2 \in N$. Then $\langle N,\rho\rangle \models ORD_{n+1}(a,b_1,b_2)$ if and only if there exists $c \in N$ such that the following four facts hold.

1) $\langle N,\rho\rangle \models FULL_n(c)$.

Let $\textcircled{\leq}$ be the linear order imposed on the set $\{b \mid \langle N,\rho\rangle \models ORD_n(c,b,b)\}$ by ORD_n. Let $c_1,c_2,\ldots,c_{f(n+1)}$ be the elements ordered by $\textcircled{\leq}$ listed in increasing order (with respect to $\textcircled{\leq}$).

2) $\ell\ell c_i$ is defined for $1 \leq i \leq f(n+1)$.

Define the sequence $d_0,d_1,\ldots,d_{f(n+1)}$ by $d_0 = a$ and $d_i = \ell\ell c_i$ for $0 < i \leq f(n+1)$. Define the sequence $e_0,e_1,\ldots,e_{f(n+1)}$ by $e_0 = a$ and $e_i = r\ell c_i$ for $0 < i \leq f(n+1)$ ($r\ell c_i$ is defined since $\ell\ell c_i$ is defined).

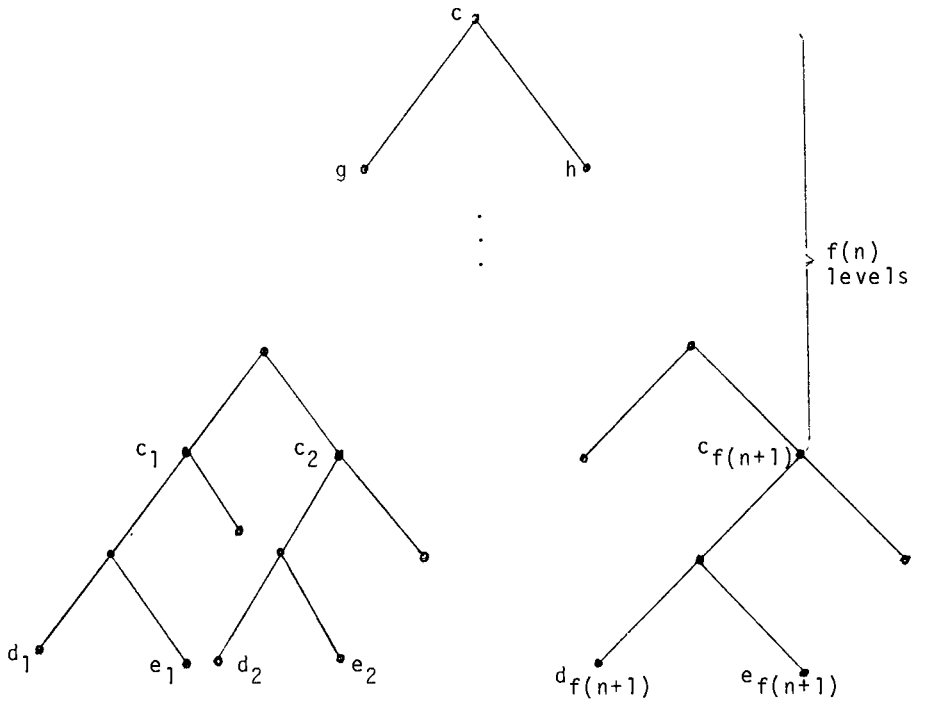

Figure 3.1 Illustrating Lemma 3.8.

3) For $0 < i \leq f(n+1)$, either $d_i = rd_{i-1}$ or $d_i = \ell d_{i-1}$, and either $e_i = re_{i-1}$ or $e_i = \ell e_{i-1}$. Also, $d_{f(n+1)} = b_1$ and $e_{f(n+1)} = b_2$.

4) Either $d_i = e_i$ for all i, $0 \leq i \leq f(n+1)$, or there exists some i, $0 < i \leq f(n+1)$ such that 4.1) $d_j = e_j$ for $0 \leq j < i$ and 4.2) $d_i = \ell d_{i-1}$ and $e_i = re_{i-1}$.

Proof Fix $\langle N,\rho\rangle$, n,a,b_1,b_2.

(If): Say that for some $c \in N$, 1) through 4) hold. If $0 < i \leq f(n+1)$, define $\gamma_i = \ell$ if $d_i = \ell d_{i-1}$, and $\gamma_i = r$ if $d_i = rd_{i-1}$ and $d_i \neq \ell d_{i-1}$. If $0 < i \leq f(n+1)$, define $\delta_i = r$ if $e_i = re_{i-1}$, and $\delta_i = \ell$ if $e_i = \ell e_{i-1}$ and $e_i \neq re_{i-1}$. Define $\sigma_1,\sigma_2 \in (r,\ell)^*$ by $\sigma_1 = \gamma_{f(n+1)}\cdots\gamma_2\gamma_1$ and $\sigma_2 = \delta_{f(n+1)}\cdots\delta_2\delta_1$. It is clear from 2) and 3) that $\sigma_1 a = b_1$ and $\sigma_2 a = b_2$. We wish to show $\sigma_1 \preceq \sigma_2$. If $\sigma_1 \neq \sigma_2$, then for some i we have $\gamma_j = \delta_j$ when $0 < j < i$, and $\gamma_i \neq \delta_i$. So $d_j = e_j$ for $0 < j < i$. If $d_i = e_i$, then $\gamma_i d_{i-1} = d_i = e_i = \delta_i e_{i-1} = \delta_i d_{i-1}$, so $\ell d_{i-1} = rd_{i-1} = d_i$. By definition of γ_i, $\gamma_i = \ell$ and so $\sigma_1 \prec \sigma_2$. If $d_i \neq e_i$, then by 4.2) $d_i = \ell d_{i-1}$. So $\gamma_i = \ell$ and $\sigma_1 \prec \sigma_2$.

(Only if): Say that $\langle N,\rho\rangle \models ORD_{n+1}(a,b_1,b_2)$. Let $\sigma_1,\sigma_2 \in (r,\ell)^*$ be such that $\sigma_1 \preceq \sigma_2$ and $|\sigma_1| = |\sigma_2| = f(n+1)$ and $\sigma_1 a = b_1$ and $\sigma_2 a = b_2$. Say that σ_1 is $\gamma_{f(n+1)}\cdots\gamma_2\gamma_1$ and that σ_2 is $\delta_{f(n+1)}\cdots\delta_2\delta_1$ where $\gamma_i \in \{r,\ell\}$ and $\delta_i \in \{r,\ell\}$ for $0 < i \leq f(n+1)$. Define the sequence $d_0,d_1,\ldots,d_{f(n+1)}$ by $d_0 = a$ and $d_i = \gamma_i d_{i-1}$

for $0 < i \leq f(n+1)$. Define the sequence $e_0, e_1, \ldots, e_{f(n+1)}$ by $e_0 = a$ and $e_i = \delta_i e_{i-1}$ for $0 < i \leq f(n+1)$. Clearly $d_{f(n+1)} = b_1$ and $e_{f(n+1)} = b_2$.

Define the sequence $g_1, g_2, \ldots, g_{f(n+1)}$ by $g_i = \rho(d_i, e_i)$ for $1 \leq i \leq f(n+1)$. Define $h_1, h_2, \ldots, h_{f(n+1)} \in N$ as follows: let h_1 be any element of N; for $1 \leq i < f(n+1)$ let h_{i+1} be such that $\rho(g_{i+1}, h_{i+1}) \neq \rho(g_j, h_j)$ for any j, $1 \leq j \leq i$. (h_{i+1} can be chosen in this way since ρ is one-one.) Define the sequence of distinct members of N--$c_1, c_2, \ldots, c_{f(n+1)}$-- by $c_i = \rho(g_i, h_i)$ for $1 \leq i \leq f(n+1)$. Clearly $d_i = \ell\ell c_i$ and $e_i = r\ell c$ for $1 \leq i \leq f(n+1)$. By Lemma 3.5, we can find $c \in N$ such that if $\alpha_1, \alpha_2, \ldots, \alpha_{f(n+1)}$ are those members of $\{r,\ell\}^*$ of length $f(n)$ listed in increasing order, then $c_i = \alpha_i c$ for $1 \leq i \leq f(n+1)$. Clearly c satisfies properties 1), 2) and 3).

If $\sigma_1 = \sigma_2$, then $d_i = e_i$ for $0 \leq i \leq f(n+1)$. Otherwise $\sigma_1 \prec \sigma_2$ implies that there exists i, $0 < i \leq f(n+1)$, such that $\gamma_j = \delta_j$ if $0 < j < i$, and $\gamma_i = \ell$ and $\delta_i = r$. This means that $d_j = e_j$ if $0 \leq j < i$ and $d_i = \ell d_{i-1}$ and $e_i = r e_{i-1}$, so 4) holds also. □

Lemma 3.9 There exists a sequence of formulas of L $\underline{ORD}_0(x,y_1,y_2), \underline{ORD}_1(x,y_1,y_2), \ldots$ such that

(I) $\underline{ORD}_n(x,y_1,y_2)$ defines the property ORD_n for $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{ORD}_n$ within time polynomial in n and space linear in n (including the length of the output formula).

Proof Define $\underline{ORD}_0(x,y_1,y_2)$ to be

$[y_1 = y_2 \wedge \exists z(\rho(z,y_1) = x \vee \rho(y_1,z) = x)] \vee \rho(y_1,y_2) = x$.

If we have $\underline{ORD}_n$ defining ORD_n, then by using Lemma 3.6 we can obtain a formula $\underline{FULL}_n(x)$ which is of length proportional to the length of $\underline{ORD}_n$ and which defines the property $FULL_n$. Lemma 3.8 therefore gives a way to define $\underline{ORD}_{n+1}$ using $\underline{ORD}_n$. (This is completely straightforward if one notes the following fact: in Lemma 3.8 we occasionally quantify over i, $1 \leq i \leq f(n+1)$, but this can be expressed indirectly as quantification over the ordered set $\{b|ORD_n(c,b,b)\}$).

If one used Lemma 3.8 in the simplest way to write down $\underline{ORD}_{n+1}$ using subformulas $\underline{ORD}_n$, then since $\underline{ORD}_n$ would occur more than once in $\underline{ORD}_{n+1}$, the length of $\underline{ORD}_n$ would be at least exponential in n. We can, however, use Theorem 2 of Chapter 7 to obtain (using Lemma 3.8) a formula $\underline{ORD}_n$ of length <u>proportional to n</u> which defines ORD_n for all $n \in N^+$. □

<u>Corollary 3.10</u> There exists a sequence of formulas of L, $\underline{FULL}_0(x), \underline{FULL}_1(x), \ldots$ such that

(I) $\underline{FULL}_n(x)$ defines the property $FULL_n$ for all $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{FULL}_n$ within time polynomial in n and within space linear in n.

Proof Use Lemma 3.6 to express $\underline{FULL}_n$ using $\underline{ORD}_n$ for $n \in N$. □

Lemma 3.11 There exists a sequence of formulas of L, $\underline{DIST}_0(x,y_1,y_2)$, $\underline{DIST}_1(x,y_1,y_2)$,... such that

(I) If $A \in P$ and $n,a,b_1,b_2 \in N$, then $A \models \underline{DIST}_n(a,b_1,b_2)$ $\iff$

1) $A \models FULL_n(a)$

2) $A \models ORD_n(a,b_1,b_2)$

3) The distance from b_1 to b_2 in the ordering determined by ORD_n is exactly $f(n)$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{DIST}_n$ within time polynomial in n and space linear in n including the length of output.

Proof Let $\underline{DIST}_0$ be $\rho(y_1,y_2) = x \wedge y_1 \neq y_2$.

Let $A \in P$, $n \in N^+$, $a,b_1,b_2 \in N$. We wish to say that $A \models FULL_n(a)$ and $|\{c \in N | c \neq b_1$, and $A \models ORD_n(a,b_1,c)$ and $A \models ORD_n(a,c,b_2)\}| = f(n)$. (This implies that $A \models ORD_n(a,b_1,b_2)$.) But by Lemma 3.5, this will be true iff $A \models FULL_n(a)$ and there is some $c' \in N$ such that $A \models FULL_{n-1}(c')$ and such that for all $c \in N$, $(A \models ORD_{n-1}(c',c,c)) \iff (c \neq b_1$ and $A \models ORD_n(a,b_1,c)$ and $A \models ORD_n(a,c,b_2))$. We can therefore write down a formula $\underline{DIST}_n(x,y_1,y_2)$ for $n \in N$ (by using $\underline{FULL}_n$, $\underline{ORD}_n$, $\underline{FULL}_{n-1}$ and $\underline{ORD}_{n-1}$) such that (I) and (II) are satisfied. □

Definition 3.12 For all $n \in N$, let $SET_n(x,y_1,y_2)$ be the property such that for $A \in P$ and $n,a,b_1,b_2 \in N$, $A \models SET_n(a,b_1,b_2)$ $A \models FULL_n(a)$ and $A \models ORD_n(a,b_2,b_2)$ and $A \models ORD_n(b_1,b_2,b_2)$.

Lemma 3.13 Let $A \in P$ and let $n,a \in N$ such that $A \models FULL_n(a)$. Let $A \subseteq \{b \mid A \models ORD_n(a,b,b)\}$. Then for some $b_1 \in N$, $A = \{b_2 \mid A \models SET_n(a,b_1,b_2)\}$.

Proof Say that $A \models FULL_n(a)$ and $A \subseteq \{b \mid A \models ORD_n(a,b,b)\}$. Let $A' \subseteq N$ be such that $0 < |A'| \leq f(n+1)$ and $A = A' \cap \{b \mid A \models ORD_n(a,b,b)\}$. By Lemma 3.5 we can find some $b_1 \in N$ such that $A' = \{b_2 \mid A \models ORD_n(b_1,b_2,b_2)\}$. Hence, $A = \{b_2 \mid A \models SET_n(a,b_1,b_2)\}$. □

Lemma 3.14 There exists a sequence of formulas of L, $\underline{SET}_0(x,y_1,y_2), \underline{SET}_1(x,y_1,y_2), \ldots$ such that

(I) $\underline{SET}_n(x,y_1,y_2)$ defines the property SET_n for $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{SET}_n$ within the time polynomial in n and space linear in n.

Proof One can easily write down $\underline{SET}_n$ using $\underline{FULL}_n$ and $\underline{ORD}_n$. □

Note that by Lemma 3.5, $\underline{FULL}_n(x)$ is satisfiable in any P-structure. Hence, the formulas $\underline{FULL}_n$ and $\underline{ORD}_n$ allow us to write formulas which, no matter which P-structure they are interpreted in, talk about an ordered set of size f(n+1). Using $\underline{DIST}_n$ we can talk about two members of this ordered set being f(n) apart. Using $\underline{SET}_n$ we can talk about all subsets of this ordered set and refer to the basic set-theoretic relations

In what follows we will think of a subset of this ordered set as corresponding to the binary string which is the characteristic sequence of the subset. It will be useful to be able to express the property that such a binary string begins in a particular way.

Definition 3.15 For every $\gamma \in \{0,1\}^+$, let $START_\gamma(x,y,z)$ be the property such that if $n+1 = \ell n(\gamma)$, $A \in P$, $a,b,c \in N$, then $A \mid START_\gamma(a,b,c)$ iff

1) $A \vdash FULL_n(a)$.

Let $\leqslant$ be the ordering determined on $\{b' \mid A \vdash ORD_n(a,b',b')\}$ by ORD_n. Let α be the characteristic sequence (with respect to $\leqslant$) of the set $\{b' \mid A \vdash SET_n(a,b,b')\} = \{b' \mid A \vdash ORD_n(a,b',b')$ and $A \vdash ORD_n(b,b',b')\}$; that is α is the binary string of length $f(n+1)$ determined by a, b and A. That is, α is the binary string of length $f(n+1)$ which has a 1 in position i iff $A \vdash SET_n(a,b,b')$ where b' is the i-th smallest member with respect to $\leqslant$ of $\{b' \mid A \vdash ORD_n(a,b',b')\}$.

2) $\alpha = \gamma\delta$ for some $\delta \in \{0,1\}^*$ of length $f(n+1)-(n+1)$.

3) c is the n+1 smallest member (with respect to $\leqslant$) of $\{b' \mid A \vdash ORD_n(a,b',b')\}$.

Lemma 3.16 Let $\gamma \in \{0,1\}^+$, $\ell n(\gamma) = n+1$, and let $i \in \{0,1\}$. Let $A \in P$ and let $a,b,c \in N$. Then $A \vdash START_{\gamma i}(a,b,c)$ <=> the following six properties hold for some $a',b',c' \in N$.

1) $A \models START_{\gamma}(a',b',c')$

2) $A \models FULL_{n+1}(a)$

Let $\leqslant$ be the ordering determined on $\{c" | A \models ORD_{n+1}(a,c",c")\}$ by ORD_{n+1}. Similarly, let $\leqslant'$ be the ordering determined on $\{c" | A \models ORD_{n}(a',c",c")\}$ by ORD_{n}.

3) c is the immediate successor of c' in the ordering $\leqslant$.

4) For all $c_1, c_2 \in N$, $c_1 \leqslant' c_2 \leqslant' c' \iff c_1 \leqslant c_2 \leqslant c'$.

5) For all $c_1 \in N$, if $c_1 \leqslant c'$, then
$A \models SET_{n+1}(a,b,c_1) \iff A \models SET_{n}(a,b',c_1)$, that is,
$A \models ORD_{n+1}(b,c_1,c_1) \iff A \models ORD_{n}(b',c_1,c_1)$.

6) $A \models SET_{n+1}(a,b,c) \iff i=1$.

Proof $\Rightarrow$): Say that $A \models START_{\gamma i}(a,b,c)$. Let α be the binary string of length f(n+2) determined by a, b and A. By Lemma 3.5 we can choose a' such that $A \models FULL_{n}(a')$ and the set $\{c" | A \models ORD_{n}(a',c",c")\}$ ordered by $\leqslant'$ is an initial segment of the set $\{c" | A \models ORD_{n+1}(a,c",c")\}$ ordered by $\leqslant$.
Let c' be the predecessor of c in the ordering $\leqslant$ (or equivalently $\leqslant'$)
Let b' be such that for all $c_1 \leqslant c'$,
$A \models ORD_{n+1}(b,c_1,c_1) \iff A \models ORD_{n}(b',c_1,c_1)$; b' exists by Lemma 3.5. It is not hard to check that a',b',c' satisfy 1) through 6) above.

$\Leftarrow$): Let a',b',c' be such that 1) through 6) above hold. Clearly $A \models FULL_{n+1}(a)$. Let $\leqslant$ and $\leqslant'$ be as above. By 3), 4) and 1), we see that c is the n+2 element in the $\leqslant$ ordering. Let α be the binary sequence of length f(n+2) determined by a, b, and A;

let α' be the binary sequence of length $f(n+1)$ determined by a', b' and A. 1), 4) and 5) imply that the first $n+1$ elements of α are the same as α', that is γ. 6) tells us that the $n+2$ element of α is i. Hence, $A \vdash \underline{START}_{\gamma i}(a,b,c)$. □

Lemma 3.17 For every $\gamma \in \{0,1\}^+$ there exists a formula $\underline{START}_\gamma(x,y,z)$ such that

(I) $\underline{START}_\gamma(x,y,z)$ defines the property $START_\gamma$ for all $\gamma \in \{0,1\}^+$.

(II) There is a procedure which given $\gamma \in \{0,1\}^+$ computes $\underline{START}_\gamma$ within time polynomial in $\ell n(\gamma)$, and space linear in $\ell n(\gamma)$ including the length of output.

Proof Let $\underline{START}_0(x,y,z)$ be the formula $FULL_0(x) \wedge \exists z'(\rho(z,z') = x) \wedge \sim ORD_0(y,z,z)$; let $\underline{START}_1(x,y,z)$ be $FULL_0(x) \wedge \exists z'(\rho(z,z') = x) \wedge ORD_0(y,z,z)$.

Lemma 3.16 shows that $START_{\gamma i}$ can be expressed in a fixed way (depending on i but independent of γ) using $START_\gamma$, together with $FULL_{n+1}$, $FULL_n$, ORD_{n+1}, ORD_n, SET_{n+1}, SET_n, where $n+1 = \ell n(\gamma)$. All of these latter properties can be expressed in a fixed way from $START_\gamma$ and ORD_n. Since for all $n \in N$, ORD_{n+1} can be expressed in a fixed way from ORD_n, we can appeal to a special case of Theorem 3 of Chapter 7 (in which $\underline{F}'_0 = \underline{F}'_1$) to conclude Lemma 3.17. □

Section 4: Using Formulas to Simulate Turing Machines

In this section we will use the formulas $\underline{FULL}_n$, $\underline{ORD}_n$, $\underline{SET}_n$, $\underline{DIST}_n$, $\underline{START}_\gamma$ to talk about Turing machines which recognize languages in NTIME(f(n)), and hence prove Theorem 1.2.

Theorem 1.2 $NTIME(f(n)) \leq_{p\ell} TH(\mathcal{C})$ for any nonempty collection $\mathcal{C}$ of P-structures.

Proof Let $L \subseteq \Sigma^*$ be a member of NTIME(f(n)). By a well known speed-up result, L is accepted in time f(n)-2 by some STM M, as defined in Chapter 6. Let Δ be as in Corollary 1.7 of Chapter 6; recall that $\Sigma \subseteq \Delta$, and that Δ contains other symbols as well (depending on M), including q_0, q_a and $\not\emptyset$. Let $N_M: \Delta^3 \to P(\Delta^3)$ be as in Corollary 1.7 of Chapter 6. Since we can view an accepting computation of M on an input of length n, as the concatenation of f(n+1)/f(n) i.d.^{s} of length f(n) (including the \$), Corollary 1.7 of Chapter 6 implies the following Fact:

Fact: Let $w \in \Sigma^*$. Then $w \in L$ if and only if there is some string $d = d(1)d(2)\ldots d(f(n+1)) \in \Delta^*$ of length f(n+1) such that

(I) d begins with $q_0 w \not\emptyset^{f(n)-(n+2)}\$$

(II) d contains q_a

(III) for all j, $1 < j < f(n+1)-f(n)$, if neither d(j-1) nor d(j) is equal to \$, then $(d(f(n)+j-1), d(f(n)+j), d(f(n)+j+1)) \in N_M(d(j-1), d(j), d(j+1))$

Now let $w \in \Sigma^+$, $\ell n(w) = n$. We have shown that with formulas of length proportional to n we can talk about an ordered set of size f(n+1). Every subset of this set can be thought of as a string of length f(n+1) in $\{0,1\}^*$. We wish, however, to talk

about strings over Δ. For convenience, say that $|\Delta| = 2^v$ for some integer v, and identify every member of Δ with some distinct v-tuple of elements of $\{0,1\}$. Then there is a natural way we can think of a sequence $\gamma_1,\gamma_2,\ldots,\gamma_v$ of strings in $\{0,1\}^*$, all of the same length k, as <u>representing</u> a string in Δ^* of length k: say that $\gamma_i = \gamma_i(1)\gamma_i(2)\ldots\gamma_i(k)$ for $1\leq i\leq v$; then the string <u>represented</u> by $\gamma_1,\gamma_2,\ldots,\gamma_v$ is the string $\alpha = \alpha(1)\alpha(2)\ldots\alpha(k)$ where $\alpha(j) = (\gamma_1(j),\gamma_2(j),\ldots,\gamma_v(j))$ for $1\leq j\leq k$. Clearly every string in Δ^* is represented by v strings in $\{0,1\}^*$. Say that $\gamma_1,\gamma_2,\ldots,\gamma_v$ represent the string q_0w.

It is now not hard to see that using $\underline{\text{FULL}}_n$, $\underline{\text{ORD}}_n$, $\underline{\text{DIST}}_n$, $\underline{\text{SET}}_n$, $\underline{\text{START}}_{\gamma_1}$, $\underline{\text{START}}_{\gamma_2}$,..., $\underline{\text{START}}_{\gamma_v}$, we can write a sentence F_w of length cn which "says" that there exist strings $\beta_1,\beta_2,\ldots,\beta_v$ of length f(n+1) in $\{0,1\}^*$ such that the string d represented by them satisfies I, II, III. That is, for any $A \in \mathcal{P}$, F_w will be true in A <=> M accepts w. Hence, if $\mathcal{C}$ is a nonempty collection of P-structures, $F_w \in TH(\mathcal{C})$ <=> M accepts w. So $L(M) \leq_{p\ell} TH(\mathcal{C})$. □

CHAPTER 9

SOME ADDITIONAL LOWER BOUNDS

Section 1: Lower Bounds for the Theory of One Successor

We now consider the structure $\langle N,' \rangle$, where $':N \to N$ is such that $n' = n+1$ for all $n \in N$. Let L be a first-order language with a single binary relation symbol in it. As in previous sections, we use the notation $x' = y$ to stand for the assertion in the language L that the binary relation holds for the pair of variables x,y; we give the relation symbol the standard interpretation in $\langle N,' \rangle$ implied by this notation.

The aim of this section is to prove:

Theorem 1.1 (Meyer, Ferrante) For any function $s:N \to N$ such that $s(n) = o(n)$, $TH(\langle N,' \rangle) \notin NSPACE(s(n))$.

The proof will be by efficient arithmetization, in the language L, of simple Turing machines which run in linear space; more specifically, we use the following theorem, whose proof we omit:

Theorem 1.2 [Sto74] Let Σ be a finite alphabet. Let $B \subseteq \Sigma^*$ be such that $\{A \mid A$ is accepted by a (nondeterministic) STM within space $n\} \leq_{log\text{-}lin} B$. Then for any function $s:N \to N$ such that $s(n) = o(n)$, $B \notin NSPACE(s(n))$.

To prove Theorem 1.1, then, using Theorem 1.2, it will be sufficient to prove that given any (nondeterministic) STM M with input alphabet Σ which accepts a set $A \subseteq \Sigma^*$ within space n, there is a function $f: \Sigma^* \to \{$the sentences of L$\}$ such that

1. for all $w \in \Sigma^*$, $w \in A \iff f(w) \in TH(<N,'>)$,

2. f is linear bounded, and

3. f is in logspace.

We note for later use the following:

Fact 1.3 Let M be an STM which accepts A within space n. Then M accepts A within time c^n for some $c \in I$.

The fact holds since M can enter at most c^n different i.d.'s on input of length n, for some $c \in I$. For a proof of this fact, see [Co71].

Given a (nondeterministic) STM $M = (\Sigma,\Gamma,S,\delta,q_0,q_a)$ such that M accepts A within space n and within time c^n for some $c \in I$, we outline our plan for defining a function $f: \Sigma^* \to$ {the sentences of L} satisfying the three properties stated above.

We assume $\$ \notin \Gamma \cup S$, and $\Gamma \cap S = \phi$. We also assume, without loss of generality, that $\{\$\} \cup \Gamma \cup S = N_{k-1}$, for some $k \in I$.

Given any element $w \in \Sigma^*$ of length n, our conventions require that every i.d. of M which appears in a computation with input w is of length exactly n+1, and thus any such i.d. with the end-marker \$ is of length exactly n+3. Given $w \in \Sigma^*$ of length n, our aim is to associate elements of N less than k^{n+4} with i.d.'s of M with the end-marker \$ of length n+3. We associate $m \in N$, $m < k^{n+4}$, with k-rep $(n+3,m) \in N_{k-1}^{n+3}$. Then every word in N_{k-1}^{n+3}, and thus every i.d. of M with the end-marker \$ of length n+3, has a unique $m \in N$, $m < k^{n+4}$, associated with it.

Our goal is to prove the following two lemmas:

Lemma 1.4 There is a logspace function mapping $w \in N^+_{k-1}$ to a formula $H_w(x)$ such that

1. $\ell n(H_w(x)) = O(\ell n(w))$, and
2. for all $a \in N$, $<N,'>$ satisfies $H_w(a) \iff$ k-rep($\ell n(w)$,a) = w.

Lemma 1.5 There is a logspace function mapping $w \in \Sigma^*$ to a formula $S_{\ell n(w)}(x,y)$, such that if $\ell n(w) = n$,

1. $\ell n(S_n(x,y)) = O(n)$, and
2. for all $a,b \in N$ such that $a < k^{n+4}$ and k-rep(n+3,a) is an i.d. of M with the end-marker \$,

$<N,'>$ satisfies $S_n(a,b) \iff b < k^{n+4}$ and k-rep(n+3,b) $\in$ $Next_M$(k-rep(n+3,a),c^n).

Once we accomplish this goal, we can easily prove Theorem 1.1 as follows

If $w \in \Sigma^*$ is of length n, we define f(w) to be the sentence

$$\exists y \exists z\ H_{\$q_0w\$}(y) \wedge H_{\$acc(n)\$}(z) \wedge S_n(y,z).$$

It should be clear that $w \in A \iff f(w) \in TH(<N,'>)$, given Lemmas 1.4 and 1.5. That f is linear bounded follows easily from condition 1 of Lemmas 1.4 and 1.5.

To verify that f $\in$ logspace, note that the functions h_1, h_2, where $h_1(w) = \$q_0w\$$, and $h_2(w) = \$acc(\ell n(w))\$$, for $w \in \Sigma^*$, are obviously in logspace. Hence, for $w \in \Sigma^*$, the functions mapping w to $H_{\$q_0w\$}(y)$, and w to $H_{\$acc(\ell n(w))\$}(z)$, are in logspace. Since f is obtained by concatenation of these two functions, the function mapping $w \in \Sigma^*$ to $S_{\ell n(w)}(y,z)$, and a fixed number of symbols, and since logspace is closed under concatenation, we conclude that f is in logspace.

It remains only to construct the logspace functions required in Lemmas 1.4 and 1.5.

To express the predicate [x=0] in L, we can write $\sim\exists y[x=y']$; similarly, to express [z=1] we can write $\exists x[x=0 \wedge z=x']$. Continuing in this manner we obtain the following:

Lemma 1.6 For every $\ell \in N$, there are formulas $A_\ell(x,y)$ and $E_\ell(x)$ in L, such that for all $a,b \in N$,

1. $\langle N,' \rangle$ satisfies $E_\ell(a) \iff a=\ell$, and
2. $\langle N,' \rangle$ satisfies $A_\ell(a,b) \iff a = b+\ell$.

Henceforth we let $x = y+\ell$ denote the formula $A_\ell(x,y)$, and $x=\ell$ denote the formula $E_\ell(x)$.

We note also for any $\ell \in N$ that there are obvious formulas $ADD_\ell(x,y,z)$, $MULT_\ell(x,y,z)$, $LE_\ell(x)$, such that for all $a,b,c \in N$.

1. $\langle N,' \rangle$ satisfies $ADD_\ell(a,b,c) \iff a = b+c$ and $c \leq \ell$,
2. $\langle N,' \rangle$ satisfies $MULT_\ell(a,b,c) \iff a = b \cdot c$ and $b,c \leq \ell$, and
3. $\langle N,' \rangle$ satisfies $LE(a) \iff a \leq \ell$.

For instance, given $\ell \in N$, we could let $ADD_\ell(x,y,z)$ be a formula of the form

$$\bigvee_{0\leq i\leq \ell} (x=y+i \wedge z=i);$$

we could let MULT (x,y,z) be similarly expressed

$$\bigvee_{0\leq i,j\leq \ell} (x=i\cdot j \wedge y=i \wedge z=j),$$

and we could let $LE_\ell(x)$ be a formula of the form $\bigvee_{0\leq i\leq \ell} x = i$.

Formulas that represent exponentation of elements of N up to a certain size, and the order relation between pairs of elements up

to a certain size, are also obviously definable in L using these formulas. For instance, if we let $ORD_\ell(x,y)$ denote the formula $LE_\ell(y) \wedge \exists z\ {\sim}(z=0) \wedge ADD_\ell(y,x,z)$, then for all $a,b \in N$, $<N,'>$ satisfies $ORD_\ell(a,b) \Longleftrightarrow a<b\leq\ell$. It is our aim to use such formulas to define the linear bounded functions required in Lemmas 1.4 and 1.5.

Now, given $w \in \Sigma^*$ of length n, it will be necessary to add and multiply numbers of size $\leq k^{n+4}$. However, given $\ell \in N$, the obvious definitions of the formulas ADD_ℓ, $MULT_\ell$, and LE_ℓ, such as the ones given above, are of length as big as $O(\ell^2 \cdot \log_2(\ell))$, and thus their use would not yield linear bounded functions. We show in Lemma 1.7 how one can define formulas equivalent to ADD_ℓ, $MULT_\ell$, etc., whose length is only $O(\log_k(\ell))$; Lemma 1.7 yields Lemma 1.4 directly, and will be used to prove Lemma 1.5.

Lemma 1.7 There is a logspace function mapping $w \in N^+_{k-1}$ to a formula $F_w(x,\bar{x}_3,\bar{y}_3,\bar{z}_2)$ such that if w is of length n

1. $\ell n(F_w(x,\bar{x}_3,\bar{y}_3,\bar{z}_2)) = O(n)$, and
2. for all $a,b_1,b_2,b_3,c_1,c_2,c_3,d_1,d_2 \in N$, $<N,'>$ satisfies $F_w(a,\bar{b}_3,\bar{c}_3,\bar{d}_2) \Longleftrightarrow$
   (a) k-rep(n,a) = w, and
   (b) $b_1 = b_2+b_3$, and $b_3 \leq k^{n+1}$, and
   (c) $c_1 = c_2 \cdot c_3$, and $c_1 \leq k^n$, and
   (d) $d_1 = k^{d_2}$, and $d_2 \leq n$.

Proof We give an inductive procedure which, given $w \in N^+_{k-1}$, constructs F_w.

Clearly, if $\ell n(w) = 1$, that is, if $w \in N_{k-1}$, one can use the formulas of Lemma 1.6 to construct a formula F_w in L satisfying requirement 2 of the lemma.

If $w \in N^+_{k-1}$ is of length n, $\sigma \in N_{k-1}$, we assume we have already shown how to construct F_w.

To define $F_{w\sigma}$, we introduce some abbreviations for formulas. We let

$$FH_w(x)$$

denote any formula of the form

$$\exists \bar{x}_3 \exists \bar{y}_3 \exists \bar{z}_2 \quad F_w(x, \bar{x}_3, \bar{y}_3, \bar{z}_2);$$

we let

$$FADD_w(\bar{x}_3)$$

denote any formula of the form

$$\exists x \; \exists \bar{y}_3 \exists \bar{z}_2 \quad F_w(x, \bar{x}_3, \bar{y}_3, \bar{z}_2);$$

we let

$$FMULT_w(\bar{y}_3)$$

denote any formula of the form

$$\exists x \; \exists \bar{x}_3 \exists \bar{z}_2 \quad F_w(x, \bar{x}_3, \bar{y}_3, \bar{z}_2); \text{ and}$$

we let

$$FPOW_w(\bar{z}_2)$$

denote any formula of the form

$$\exists x \; \exists \bar{x}_3 \exists \bar{y}_3 \quad F_w(x, \bar{x}_3, \bar{y}_3, \bar{z}_2).$$

We let

$$FL_{n+1}(x_1)$$

denote any formula of the form

$$\exists x_2\ \mathrm{FADD}_w(x_1,x_2,x_1), \text{ and}$$

$$\mathrm{FLE}_n(y_1)$$

denote any formula of the form

$$\exists y_2\ \mathrm{FMULT}_w(y_1,y_2,y_1).$$

For $m\in I$, $m\geq 2$ and $i\in N$, we use the notation

$$[x = y_1+\ldots+y_m+i]_w$$

for any formula of the form

$$\exists \bar{x}_{m-1}\ \mathrm{FADD}_w(x_1,y_1,y_2) \wedge \mathrm{FADD}_w(x_2,x_1,y_3) \wedge \mathrm{FADD}_w(x_3,x_2,y_4) \wedge \ldots \wedge \mathrm{FADD}_w(x_{m-1},x_{m-2},y_m) \wedge x = x_{m-1}+i.$$

Also, $[x = my+i]_w$ stands for $[x = \underbrace{y+\ldots+y}_{my's}+i]_w$.

We let

$$[x = \sum_{i,j\in I_m} z_i\cdot y_j]_w$$

stand for any formula of the form

$$\exists \bar{x}_m^1 \ldots \exists \bar{x}_m^m \bigwedge_{i,j\in I_m} \mathrm{FMULT}_w(x_j^i,z_i,y_j) \wedge [x = x_1^1 + x_2^1 + \ldots + x_m^m]_w.$$

We define $F'_{w\sigma}(x,\bar{x}_3,\bar{y}_3,\bar{z}_\alpha)$ to be the formula

$$\exists t\ \exists \bar{t}_k \exists \bar{u}_{2.k} \exists \bar{v}_2 H,$$

where H is the conjunction of (a)-(d)

(a) $FH(t) \wedge [x = kt+\sigma]_w$

(b) $[x_3 = t_1+\ldots+t_k]_w \wedge [x_1 = x_2+t_1+\ldots+t_k]_w$

(c) $[y_2 = u_1+\ldots+u_k]_w \wedge [y_3 = u_{k+1}+\ldots+u_{2\cdot k}]_w \wedge$
$[y_1 = \sum_{i,j\in I_k} u_i \cdot u_{k+j}]_w \wedge FL_{n+1}(y_1)$†

(d) $[z_1 = 1 \wedge z_2 = 0] \vee [FPOW_w(v_1,v_2) \wedge$
$z_2 = v_2+1 \wedge [z_1=kv_1]_w]$.

It should be clear that $F'_{w\sigma}$ so defined satisfies requirement 2 of the lemma.

Finally, we define $F_{w\sigma}(x,\bar{x}_3,\bar{y}_3,\bar{z}_2)$ to be the formula obtained from $F'_{w\sigma}$ by replacing the $2k^2+9k-3$ occurrences of F_w in H by one such occurrence, as in Theorem 2 of Chapter 7, using only a fixed number of variables.

†We remark that it must be verified that if $y_1 = y_2 \cdot y_3$ with $y_1 \leq k^{n+1}$, then u_i, u_{k+j} for $i,j \in I_k$ can be chosen so that for all $i,j \in I_k$, $u_i \cdot u_{k+j}$ is bounded by k^n. (Such terms can trivially be seen to be bounded by k^{n+1}, since $y_1 \leq k^{n+1}$). The proof for the case k=2 proceeds as follows.

Given y_1, y_2, let

$$u_1 = \frac{\lceil y_2 \rceil}{2}, \quad u_2 = \lfloor \frac{y_2}{2} \rfloor,$$

$$u_3 = \frac{\lceil y_3 \rceil}{2}, \quad u_4 = \lfloor \frac{y_3}{2} \rfloor.$$

It clearly suffices to verify

$u_1 \cdot u_3 \leq 2^n$, given $y_2 \cdot y_3 \leq 2^{n+1}$.

Now $u_1 \leq \frac{y_2}{2} + \frac{1}{2}$, and $u_3 \leq \frac{y_3}{2} + \frac{1}{2}$. Thus

$u_1 \cdot u_3 \leq (\frac{y_2}{2} + \frac{1}{2}) \cdot (\frac{y_3}{2} + \frac{1}{2}) = \frac{y_2 y_3}{4} + \frac{y_2}{4} + \frac{y_3}{4} + \frac{1}{4} =$

We must now verify the function we have defined mapping $w \in N^+_{k-1}$ to F_w is linear bounded and in logspace. Since $F_{w\sigma}$ is constructed by concatenating a fixed number of variables and other symbols bounded by a polynomial in k, but independent of w with F_w, clearly, then, the number of variables and other symbols in $F_{w\sigma}$ is equal to $\ell n(F_w)+c_1$, for some $c_1 \in I$. If we choose $c_1 > \ell n(F_\sigma)$ for all $\sigma \in N_{k-1}$, then for all $w \in N^+_{k-1}$, the number of variables and other symbols in $F_w \leq c_1 \ell n(w)$ By judicious reuse of a fixed set of variables we can ensure that for $w \in N^+_{k-1}$, $\ell n(F_w) \leq c_2 \ell n(w)$, for some $c_2 \in I$, as in Theorem 2 of Chapter 7.

We then conclude that the function in question is linear bounded.

Finally we note the definition of $F_{w\sigma}$ in terms of F_w is a definition by two-sided recursion of concatenation [Sto74], which implies the function mapping w to F_w is in logspace. □

$\frac{1}{4}(y_2 \cdot y_3 + y_2 + y_3 + 1) \leq \frac{1}{4}(2^{n+1} + y_2 + y_3 + 1)$. Now, to show that $\frac{1}{4}(2^{n+1} + y_2 + y_3 + 1) \leq 2^n$, it suffices to show

$$\frac{1}{4}(y_2 + y_3 + 1) \leq 2^{n-1}.$$

But this is the same as

$$y_2 + y_3 + 1 \leq 2^{n+1},$$

which clearly holds unless one of y_2 or y_3 is equal to 2^{n+1}. However, it is easy to see in this case our choice of u_1, u_3 yields $u_1 \cdot u_3 \leq 2^n$ directly, and so the verification is complete.

To prove Lemma 1.4, then, $w \in N_{k-1}^{+}$ is mapped to $\exists\bar{x}_3\exists\bar{y}_3\exists\bar{z}_2 F_w(x,\bar{x}_3,\bar{y}_3,\bar{z}_2)$. The fact that this function is in logspace and is linear bounded follows directly from Lemma 1.7, and the fact that logspace is closed under concatenation.

Lemma 1.5 will follow directly from the following two lemmas.

Lemma 1.8 For any $\sigma \in N_{k-1}$, there is a logspace function mapping $w \in \Sigma^*$ to a formula $D_{\ell n(w),\sigma}(x,y)$, such that if w is of length n,

1. $\ell n(D_{n,\sigma}(x,y)) = O(n)$, and
2. for all $i,b \in N$,

<N,'> satisfies $D_{n,\sigma}(i,b) \iff i<n+3$ and $b < k^{n+4}$ and the $i+1^{st}$ digit of k-rep(n+3,b) is σ.

Proof Let $\sigma \in N_{k-1}$. Given $w \in \Sigma^*$ of length n, let $\hat{w} = w0000 \in N_{k-1}^{+}$, so $\ell n(\hat{w}) = n+4$. We note that the order relation $x < y \leq k^{n+4}$ is easily expressible using $FADD_{\hat{w}}$ and FLE_{n+4}, as shown in the text immediately preceding Lemma 1.7. The predicate $x = n+3$ is also succintly expressible as

$$FH_u(x)$$

where u = k-rep(n+4,n+3).

We map $w \in \Sigma^*$ to a formula expressing $x < n+3$, $y < k^{n+4}$, and there are z_1,z_2,x_1, where $z_1 < k^{n+1}$, $z_2 < k^{x+1}$, $x_1 < k^x$, such that $y = k^{x+1}\cdot z_1+z_2$, and $z_2 = k^x\cdot\sigma+x_1$. Clearly, such a formula can be constructed using a fixed (independent of n) number of instances of the formula $F_{\hat{w}}$, concatenated with a fixed number of additional symbols. We leave the precise construction of $D_{\ell n(w),\sigma}$ to the interested reader.

The fact that the function mapping $w \in \Sigma^*$ to $D_{\ell n(w),\sigma}(x,y)$ is linear bounded follows easily from Lemma 1.7. Finally, since the function mapping w to $\hat{w}$ is clearly in logspace, and logspace is closed under composition and concatenation, it follows that the function mapping $w \in \Sigma^*$ to $D_{\ell n(w),\sigma}$ is in logspace. □

Lemma 1.9 There is a logspace function mapping $(w_1,w_2) \in (\Sigma^*)^2$ to a formula $S_{\ell n(w_1),\ell n(w_2)}(x,y)$ such that for $w_1 \in \Sigma^*$ of length m, and $w_2 \in \Sigma^*$ of length n,

1. $\ell n(S_{m,n}(x,y)) = O(m+n)$,
2. for all $a,b \in N$ such that $a < k^{n+4}$ and k-rep(n+3,a) is an i.d. of M with the end-marker \$, $\langle N,1\rangle$ satisfies $S_{m,n}(a,b) \iff b<k^{n+4}$ and k-rep(n+3,b) $\in$ $Next_M$(k-rep(n+3,a),c^m).

Our construction of $S_{m,n}$ is based on a similar construction by Stockmeyer [SM73], from which he obtains a space $\sqrt{n}$ lower bound on the theory of equality.

Proof of Lemma 1.9 We give an inductive procedure which, given w_1 of length m and w_2 of length n, constructs $S_{m,n}$.

If $\ell n(w_1) = 0$, we use Lemma 1.5 of Chapter 6, which asserts the existence of a function N_M: $N_{k-1}^3 \to P(N_{k-1}^3)$ with the following property. Let d be any i.d. of M with the end-marker \$ of length ℓ, and let $f \in N_{k-1}^{\ell}$; say $d = d_1 \cdot \ldots \cdot d_\ell$, $f = f_1 \cdot \ldots \cdot f_\ell$, with $d_i, f_i \in N_{k-1}$, for all $i \in I_\ell$. Then

$f \in \text{Next}_M(d)$ iff

$f_{j-1}f_jf_{j+1} \in N_M(d_{j-1}d_jd_{j+1})$, for all $j \in I$, $1 < j < \ell$.

We define $S_{0,n}(x,y)$ as

$$\forall z\ \forall x_1\ \forall x_2[(x_1+1 = z) \wedge (z+1 = x_2) \wedge$$
$$\bigvee_{\sigma_1\sigma_2\sigma_3 \in N^3_{k-1}} [[D_{n,\sigma_1}(x_1,x) \wedge D_{n,\sigma_2}(z,x) \wedge D_{n,\sigma_3}(x_2,x)]$$
$$\rightarrow [\bigvee_{\delta_1\delta_2\delta_3 \in N_M(\sigma_1\sigma_2\sigma_3)} D_{n,\delta_1}(x_1,y) \wedge D_{n,\delta_2}(z,y) \wedge D_{n,\delta_3}(x_2,y)]]]$$

It should be clear $S_{0,n}(x,y)$ satisfies property 2 of the lemma.

We assume $\ell n(w_1) = m+1$, and that we have already shown how to construct $S_{m,n}(x,y)$. We define $S_{m+1,n}(x,y)$ to be that formula equivalent to

$$\exists \bar{z}_{c-1} S_{m,n}(x,z_1) \wedge \bigvee_{i \in I_{c-2}} S_{m,n}(z_i,z_{i+1}) \wedge S_{m,n}(z_{c-1},y),$$

obtained by applying Theorem 2 of Chapter 7; that is, $S_{m+1,n}(x,y)$ is

$$\exists \bar{z}_{c-1} \forall y_1\ \forall y_2\ (y_1 = x \wedge y_2 = z_1) \vee$$
$$\bigvee_{i \in I_{c-2}} (y_1 = z_i \wedge y_2 = z_{i+1}) \vee (y_1 = z_{c-1} \wedge y_2 = y)$$
$$\rightarrow S_{m,n}(y_1,y_2).$$

Thus the definition of $S_{m+1,n}$ from $S_{m,n}$ is by two-sided recursion of concatenation. Clearly, property 2 of the lemma is satisfied.

It remains to show the function mapping $(w_1,w_2) \in (\Sigma^*)^2$ to $S_{\ell n(w_1),\ell n(w_2)}$ is linear bounded and in logspace.

We first show this function is linear bounded. As in Lemma 1.7, it is possible to define $S_{\ell n(w_1),\ell n(w_2)}$ so that it contains only a fixed number of variables, for all $w_1, w_2 \in \Sigma^*$. The details again are left to the reader. Thus for $w_1 \in \Sigma^*$ of length m+1 and $w_2 \in \Sigma^*$ of length n,

$$\ell n(S_{m+1,n}) = \ell n(S_{m,n}) + c_2,$$

for some $c_2 \in I$. Now, the length of $S_{0,n}$ equals $c_3 \cdot n$, for some $c_3 \in I$, by Lemma 1.8. Thus, $\ell n(S_{m,n}) = c_2 \cdot m + c_3 \cdot n = O(m+n)$.

The fact that the function mapping (w_1, w_2) to $S_{\ell n(w_1),\ell n(w_2)}$ is in logspace follows from Lemma 1.8 and the fact that the definition of $S_{m+1,n}$ from $S_{m,n}$ is by two-sided recursion of concatenation. □

Lemma 1.5 follows directly from Lemma 1.9 as follows. Given $w \in \Sigma^*$, we let $S_{\ell n(w)}(x,y)$ be $S_{\ell n(w),\ell n(w)}(x,y)$. Clearly, then, properties 1 and 2 of Lemma 1.5 are satisfied. It remains only to show the function we have defined is in logspace. Since the function g such that $g(w) = w\#w$, for $w \in \Sigma^*$ is clearly in logspace, it follows directly from Lemma 1.9 that the functio mapping $w \in \Sigma^*$ to $S_{\ell n(w)}(x,y)$ is in logspace. This completes the proof of Lemma 1.5 and hence of Theorem 1.1.

Next consider the structure $\langle N, < \rangle$. We can express the predicate $y = x'$ in the language of the structure $\langle N, < \rangle$ by the formula

$$x < y \wedge \forall z[z < y \rightarrow z < x \vee z = x].$$

It then follows easily that $TH(<N,'>) \leq_{\text{log-lin}} TH(<N,<>)$. We then have the following Corollary to Theorem 1.1:

Corollary 1.10 For any function $s:N \to N$ such that $s(n) = o(n)$,

$TH(<N,<>) \notin NSPACE(s(n))$.

Theorem 1.1 and its Corollary should be compared to Theorem 5.32 of Chapter 4, which states $TH(<N,'>) \in DSPACE(n^2)$, and Theorem 5.28 of Chapter 4, which states $TH(<N,<>) \in DSPACE(n^2)$. We remark given Savitch's results discussed in Section 3 of Chapter 1, the "gaps" between the lower and upper bounds are to be expected.

We can also obtain lower bounds for the theories of well order and lexicographical order. We can easily show

$TH(<N,'>) \leq_{\text{log-lin}}$ theory of well order.

Briefly, we map a sentence F in the first theory to a sentence F* expressing F in the language of the second theory, conjuncted with the statement that there is a unique element with no immediate predecessor, and every element has a successor. This latter statement assures the only well orders considered will be ones isomorphic to $<N,<>$. We therefore have

Corollary 1.11 For any function $s:N \to N$ such that $s(n) = o(n)$,

$SAT(\{<\alpha,<>| \alpha \text{ a well order}\}) \notin NSPACE(s(n))$.

The proof of the result for lexicographical order proceeds as the proof for $<N,'>$, since $<N,<>$ is isomorphic to an initial segment of $<\{0,1\}^*, \prec>$.

We therefore obtain:

Corollary 1.12 For any function $s:N \to N$ such that $s(n) = o(n)$.

$TH(<\{0,1\}^*, \prec>) \notin NSPACE(s(n))$.

Section 2: Lower Bounds for the Theory of One Successor with A Monadic Predicate

Let MONN denote the class of structures $\{\langle N,',M\rangle \mid M \subseteq N\}$, where $':N \to N$ is such that $n' = n+1$ for all $n \in N$. Let L be a first-order language with a single binary relation symbol and a single unary relation symbol in it. We use the notation $x' = y$ as in Section 3 of Chapter 4, and give the binary relation symbol in L the standard interpretation implied by this notation. For $\langle N,',M\rangle \in M$, and $n \in N$, we interpret the assertion that the unary relation holds for n as $n \in M$. We note that SAT(MONN) corresponds in an obvious way, to the set of satisfiable sentences in the second-order language of $\langle N,'\rangle$ with a single outermost existential set quantifier.

The aim of this section is to prove:

Theorem 2.1 (Ferrante, Rackoff):

There is a rational $c>0$ such that $\text{SAT(MONN)} \notin \text{NTIME}(2^{c^n})$.

It is of interest that $\text{SAT}(\{\langle N,<,M\rangle \mid M \subseteq N\})$ is decidable, and its decision procedure is not elementary-recursive [Sto74].

For $k \in N$, let S_k denote the satisfiable sentences of one successor with k monadic predicates in a first-order language L_k with only relation symbols. To prove Theorem 2.1, we first establish

Lemma 2.2 For any $k \in I$, $S_k \leq_{p\ell} \text{SAT(MONN)}$.

Proof We show $S_2 \leq_{p\ell} \text{SAT(MONN)}$; the proof when $k>2$ is similar and is left to the reader.

We first show that there is a function mapping a sentence F in L_2 to a sentence $\hat{F}$ in L such that

$F \in S_2 \iff \hat{F} \in SAT(MONN)$.

In order to show the function we define satisfies the property above, and to motivate its definition, we show how to interpret any structure $\langle N,',M_1,M_2\rangle$ in some structure $\langle N,',M(M_1,M_2)\rangle$ where $M_1,M_2,M(M_1,M_2) \subseteq N$.

Define $M(M_1,M_2)$ to be the unique set M satisfying the following:

1. for all $n \in N$, $7n \notin M$ and $7n+1 \notin M$, and $7n+i \in M$ for $i = 2,4,6$,
2. if $n \in M_1$ ($n \notin M_1$), then $7n+3 \in M$ ($7n+3 \notin M$, respectively);
3. if $n \in M_2$ ($n \notin M_2$), then $7n+5 \in M$ ($7n+5 \notin M$, respectively).

We thus interpret n in the structure $\langle N,',M_1,M_2\rangle$ as $7n$ in the structure $\langle N,',M(M_1,M_2)\rangle$. We interpret $n \in M_1$ via the membership of $7n+3$ in $M(M_1,M_2)$, and $n \in M_2$ via the membership of $7n+5$ in $M(M_1,M_2)$. Figure 2.3 presents a pictorial representation of the characteristic functions $M_1(x),M_2(x)$ and $M(M_1,M_2)(x)$ of a typical structure $\langle N,',M_1,M_2\rangle$ and the corresponding structure $\langle N,',M(M_1,M_2)\rangle$.

x =	0	1	2	3	4	5	6	7	8	...
$M_1(x)$ =	1	0	1	1	0	...				
$M_2(x)$ =	0	1	1	...						
$M(M_1,M_2)(x)$ =	0	0	1	$M_1(0)$	1	$M_2(0)$	1	0	0	...
=	0	0	1	1	1	0	1	0	0	...

Figure 2.3 Coding M_1 and M_2 into $M(M_1,M_2)$.

Suppose P_1 and P_2 are the unary relation symbols of L_2, and R is the unary relation symbol of L. WE let DBZ(x) denote a formula expressing $\sim R(x) \wedge \sim R(x+1)$.

Before defining the function mapping F to $\hat{F}$, we first define a function mapping a formula B in L_2 to a formula B^* in L by induction on the structure of B.

$x = y$ is mapped to $x = y$

$x = y'$ is mapped to a formula expressing $x = y + 7$

$P_1(x)$ is mapped to a formula expressing $R(x+3)$

$P_2(x)$ is mapped to a formula expressing $R(x+5)$

$C \vee D$, $C \wedge D$, $C \rightarrow D$, $\sim C$ are mapped to $C^* \vee D^*$, $C^* \wedge D^*$, $C^* \rightarrow D^*$, and $\sim C^*$, respectively.

$\exists x C$ is mapped to $\exists x DBZ(x) \wedge C^*$.

We have

Claim 2.4 For any formula $B(\bar{y}_k)$ in L_2, any structure $\langle N,',M_1,M_2\rangle$, and $\bar{n}_k \in N^k$,

$\langle N,',M_1,M_2\rangle$ satisfies $B(n_1,\ldots,n_k)$ $\Leftrightarrow$

$\langle N,',M(M_1,M_2)\rangle$ satisfies $B^*(7n_1,\ldots,7n_k)$.

The proof of Claim 2.4 is an easy induction argument on the structure of formulas, and is left to the reader.

We now define the function mapping F to $\hat{F}$ as follows. $\hat{F}$ is the sentence expressing

$$[DBZ(0) \wedge \forall x[DBZ(x) \rightarrow DBZ(x+7) \wedge \bigwedge_{i=2,4,6} R(x+i)]] \wedge F^*.$$

Claim 2.5 $F \in S_2 \Leftrightarrow \hat{F} \in SAT(MONN)$.

The proof of Claim 2.5 follows from Claim 2.4 and the fact that $\langle N,',M\rangle$ satisfies $[DBZ(0) \wedge \forall x[DBZ(x) \rightarrow DBZ(x+y) \wedge \bigwedge_{i=2,4,6} R(x+i)]]$ iff $M = M(M_1,M_2)$ for some structure $\langle N,',M_1,M_2\rangle$ as above. The details of the proof are left to the reader.

It is easily verified that the function mapping F in L_2 to $\hat{F}$ in L can be computed within polynomial time and linear space, and is linear bounded. (Of course, to be completely precise, this function must also be defined if F is not a well-formed formula. However, an IOTM computing this function can first check within space $\log_2(n)$ whether F is well-formed, and output a false sentence if not.) □

The proof of Theorem 2.1 will be by efficient arithmetization in S_k (for appropriately chosen k) of simple Turing machines which run in doubly exponential time; we use the following theorem, whose proof we omit:

Theorem 2.6 [Sto74] Let Σ be a finite alphabet. Let $B \subseteq \Sigma^*$ be such that for every (nondeterministic) STM which accepts a set B_M within time $2^{2^n}-2$,† $B_M \leq_{p\ell} B$. Then there is a rational $c > 1$ such that $B \notin NTIME(2^{c^n})$.

To prove Theorem 2.1, then, using Lemma 2.2 and Theorem 2.6, it will be sufficient to show given any nondeterministic STM M with input alphabet Σ which accepts a set $A \subseteq \Sigma^*$ within time $2^{2^n}-2$, there is a $k \in I$ such that $A \leq_{p\ell} S_k$.

To accomplish this goal, we first define a particular deterministic "counting" STM M_c. M_c, started with \$w\$ on its tape in state q_L, where $w \in \{0,1\}^\ell$ for $\ell \in I$, successively adds 1 to the integer whose binary representation is on its tape, returning to state q_L while scanning the leftmost symbol \$ for each new binary word written, until $\$1^\ell\$$ is obtained. M_c then tries to add 1 to 1^ℓ without exceeding the boundaries \$, fails and enters a self-looping state q_d. M_c never halts, and never re-enters state q_ℓ. A table of moves for M_c is given in Definition 2.7 below.

Having defined M_c, we will then show how given $w \in \Sigma^*$ we can construct a succinct formula in L_k (for $k \geq 3$) which

---

† We choose time $2^{2^n}-2$, instead of the more natural time 2^{2^n}, for purely technical reasons. Because STM's possess constant factor speed up [Sto74], there is no difference between the sets accepted by some STM within time 2^{2^n} and those accepted within time $2^{2^n}-2$.

"describes" the computation of M_c on input $\$0^{2^{\ell n(w)+1}}\$$ in the sense that any true interpretation of the first three monadic predicates of the formula then codes the successive i.d.'s of M_c on this input. Given M, and $w \in \Sigma^*$, we will then use this encoding of the computation of M_c on $\$0^{2^{\ell n(w)+1}}\$$ as a "ruler" which imposes an encoding of the computation of M on input w on true interpretations of some additional monadic predicate symbols. (This "ruler" approach to obtaining lower bounds was first used in the work of Stockmeyer [Sto74] and Meyer [Mey75].)

With this motivation in mind, we present our definition of the "counting" machine M_c.

Definition 2.7 We let $M_c = (\{0,1,\$\},\{0,1,\$\},\{q_L,q_R,q_c,q_d,q_f\}, \delta_c,q_L,q_f)$, where $\delta_c: Q_c \times \Gamma_c \to P(Q_c \times \Gamma_c \times \{-1,0,1\})$ is given by the following table.

δ_c	0	1	\$
q_L	$\{(q_L,0,-1)\}$	$\{(q_L,1,-1)\}$	$\{(q_R,\$,1)\}$
q_R	$\{(q_R,0,1)\}$	$\{(q_R,1,1)\}$	$\{(q_c,\$,-1)\}$
q_c	$\{(q_L,1,-1)\}$	$\{(q_c,0,-1)\}$	$\{(q_d,\$,0)\}$
q_d	ϕ	ϕ	$\{(q_d,\$,0)\}$
q_f	ϕ	ϕ	ϕ

q_L is a left-moving state, q_c is a left-moving state which propagates the carry, q_R is a right-moving state, q_d is the self-looping state entered if the carry fails, and q_f is the (never-entered) final state.

We let the reader convince himself that M_c as defined does have the behavior described above. In addition, we let the reader convince herself:

Lemma 2.8 Let $\ell \in I$, $w,w' \in \{0,1\}^{\ell}$. The i.d. $q_L\$w\$$ appears before the i.d. $q_L\$w'\$$ in a computation of M_c on $\$0^{\ell}\$$ $\Longleftrightarrow$ when viewed as binary integers $w < w'$. In addition, the i.d. $q_L\$w\$$ appears exactly once in a computation of M_c on $\$0^{\ell}\$$.

We next show how to code successive i.d.'s of M_c using three monadic predicates. Note that state q_f is never entered on a computation of M_c on <u>any</u> input, and thus to code the computations of M_c it will suffice to code the seven symbols $q_L,q_R,q_c,q_d,0,1,\$$, and the additional symbol $\# \notin \Gamma_c \cup Q_c$.

Definition 2.9 Let $\Delta = \{S_0,\ldots,S_{\ell}\}$ be a finite alphabet. Let $M_i \subseteq N$ for $i \in I_r$. $n \in N$ $\bar{M}_r$-codes S_i for $i \in N_{\ell}$ if

$$\text{brep}(r,i) = M_1(n)\cdot\ldots\cdot M_r(n).$$

Now if $r,j \in I$, with $j \geq r$, and Δ a finite alphabet with $|\Delta| \leq 2^r$, it should be clear how to define formulas of L_j, $\text{Code}_{\sigma}(x)$, for each $\sigma \in \Delta$, such that for any structure $\langle N,',\bar{M}_j\rangle$ and any $n \in N$,

$\langle N,',\bar{M}_j\rangle$ satisfies $\text{Code}_{\sigma}(n)$ $\Longleftrightarrow$ n $\bar{M}_r$-codes σ.

We therefore assume for each symbol $\sigma \in \{q_L,q_R,q_c,q_d,0,1,\$,\#\}$ there is a formula $\text{Code}_{\sigma}(x)$ in L_k (for $k \geq 3$) which satisfies the above.

Given these fixed coding formulas, we define:

Definition 2.10 Let $M_i \subseteq N$ for $i = 1,2,3$, and $\ell \in I$. $\bar{M}_3$ codes the computation of M_c on $\$0^{\ell}\$$ if

1. the sequence $0,\ldots,\ell+3$ $\bar{M}_3$-codes $\#,q_L,\$,0,\ldots,0,\$$, respectively, and
2. for all $m \in N$, if the sequence $m,\ldots,m+\ell+3$ codes $\#d$, where d is an i.d. which appears in the computation of M_c on input $\$0^{\ell}\$$, then $m+\ell+4,\ldots,m+2\ell+7$ $\bar{M}_3$-codes $\#d'$, where $d' \in Next_{M_c}(d)$.

Suppose $\bar{M}_3$ codes the computation of M_c on $\$0^{\ell}\$$. Then for each $j \in N$, it follows from Definition 2.10 that there exists a unique $m \in N$ such that $m \leq j \leq m+\ell+3$ and m $\bar{M}_3$-codes $\#$. Furthermore, the sequence $m,m+1,\ldots,m+\ell+3$ encodes $\#d$ for some i.d., d, of M_c. Define $id(j) = d$.

Our goal now is to write a sentence F_{ℓ} of size $O(\log(\ell))$ such that

$\langle N,',\bar{M}_k\rangle$ satisfies $F_{\ell} \iff \bar{M}_3$ codes the computation of M_c on $\$0^{\ell}\$$.

Lemma 2.11 There is a polylin function mapping $w \in \Sigma^*$ to a formula $Add_w(x,y,z)$ in L_0 such that for all $a,b,c \in N$,

1. $\ell n(Add_w) = O(\ell n(w))$, and
2. $\langle N,'\rangle$ satisfies $Add_w(a,b,c) \iff a = b+c$ and $c \leq 2^{\ell n(w)+1}$.

The proof of Lemma 2.11 is contained in Lemma 1.7 and is not repeated here.

Lemma 2.12 Let $k \geq 3$. There is a polylin function mapping $w \in \Sigma^*$ to a sentence A_w in L_k such that

1. $\ell n(A_w) = O(\ell n(w))$, and
2. for any structure $\langle N,', \bar{M}_k \rangle$, $\langle N,', \bar{M}_k \rangle$ satisfies $A_w \Leftrightarrow \bar{M}_3$ codes the computation of M_c on $\$0^{2^{\ell n(w)+1}}\$$.

Proof We use Lemma 1.5 of Chapter 6, which states if $\Delta = \Gamma_c \cup Q_c \cup \{\#\} - \{q_f\}$ there is a function $N'_{M_c} : \Delta^3 \to P(\Delta^3)$ with the following property:

> If d is any i.d. of M_c of length ℓ, $d = d_1 \cdot \ldots \cdot d_\ell$ with $d_i \in \Delta$ for $i \in I_\ell$, and if $f = d_{\ell+2} \cdot \ldots \cdot d_{2\ell+1}$, with $d_i \in \Delta$ for $\ell+2 \leq i \leq 2\ell+1$, and $\#d\#f\# = d_0 \ldots d_{2\ell+2}$, then
>
> $f \in Next_{M_c}(d)$ iff
>
> $d_{\ell+j} d_{\ell+j+1} d_{\ell+j+2} \in N'_{m_c}(d_{j-1} d_j d_{j+1})$
>
> for $1 \leq j \leq \ell$.

Suppose $w \in \Sigma^*$ is of length n. We construct A_w to be a formula expressing the conjunction of 1-3 below, using the fixed coding of Δ by the first three monadic predicates of L_k.

1. expresses the fact that the initial i.d. is coded by $0, \ldots, 2^{n+1}+3$, 2. that end-markers align properly, and 3. that the computation, as it proceeds, is properly encoded.

1. $\mathrm{Code}_{\#}(0)$ and $\mathrm{Code}_{q_L}(1)$ and $\mathrm{Code}_{\$}(2)$, and for every x such that $2 < x \leq 2^{n+1}+2$, $\mathrm{Code}_0(x)$, and $\mathrm{Code}_{\$}(2^{n+1}+3)$, and
2. for every z such that $\mathrm{Code}_{\#}(z)$ we have $\mathrm{Code}_{\#}(z+2^{n+1}+4)$, and
3. for every y such that $\sim\mathrm{Code}_{\#}(y)$,

$$\bigvee_{\sigma_1\sigma_2\sigma_3\in\Delta^3}[(\mathrm{Code}_{\sigma_1}(y-1) \wedge \mathrm{Code}_{\sigma_2}(y) \wedge \mathrm{Code}_{\sigma_3}(y+1))$$

$$\rightarrow \bigvee_{\delta_1\delta_2\delta_3\in N'_{M_c}(\sigma_1\sigma_2\sigma_3)} (\mathrm{Code}_{\delta_1}((y-1)+2^{n+1}+4) \wedge$$

$$\mathrm{Code}_{\delta_2}(y+2^{n+1}+4) \wedge \mathrm{Code}_{\delta_3}((y+1)+2^{n+1}+4)].$$

We remind the reader that the order relation between elements of N bounded by 2^{n+1} can be expressed using Add_w, where w is of length n. It is then easy to see how to express the three preceding conditions by formulas involving a fixed (independent of w) number of instances of the formula Add_w. The rest of the proof follows directly from Lemma 1.6 and the definition of N'_{M_c}. □

We proceed to utilize these results to code the computation of STM M on input $w \in \Sigma^*$.

Now, for $w \in \Sigma^*$ of length n, if $k \geq 3$, and $\langle N,',\bar{M}_k\rangle$ satisfies A_w, then there are exactly $2^{2^{n+1}}$ elements m of N such that m $\bar{M}_3$-codes # and m+1 $\bar{M}_3$-codes q_L. (This follows directly from definitions and Lemma 1.5). These are the

elements of $\mathbb{N}$ we will use to code (via the use of additional monadic predicates) the computation of M on input w. We first show there are "enough" elements of this kind.

Now since M accepts $A \subseteq \Sigma^*$ within time $2^{2^n}-2$, it clearly accepts A within space $2^{2^n}-2$. Thus, the i.d.'s of M in an accepting computation with input $w\not\!\!\beta^{2^{2^n}-2-n}$ are of length $2^{2^n}-1$, where $w \in \Sigma^*$ is of length n and $\not\!\!\beta$ is the blank tape symbol of M. If $\not\!\!\xi$ is a symbol not in $\Gamma \cup S$, (where Γ is the tape alphabet of M, S the state set of M), then $\not\!\!\xi$ concatenated with such i.d.'s yields words of length exactly 2^{2^n} Since M accepts A within time $2^{2^n}-2$, there are at most $2^{2^n}-2$ such i.d.'s. Thus since $(2^{2^n}-2)(2^{2^n}) \leq 2^{2^{n+1}}$, space $2^{2^{n+1}}$ is sufficient to write the entire accepting computation of M on input w, with successive i.d.'s separated by the end-marker $\not\!\!\xi$. Thus there are "enough" elements m of $\mathbb{N}$ such that m $\bar{M}_3$-codes # and m+1 $\bar{M}_3$-codes q_L to code the computation in this way. We now prove some technical lemmas which will allow us to carry this through.

Definition 2.13 Let $k \geq 3$. Point (x) denotes any formula of the form

$$\text{Code}_{\#}(x) \wedge \exists z[(z = x+1) \wedge \text{Code}_{q_L}(z)]$$

in L_k.

Definition 2.14 Let d be an i.d. of M_c of the form q'\$x\$ where $x \in \{0,1\}^+$. We define bin (d) to be the integer k such that $\text{brep}(\ell n(x),k) = x$.

Lemma 2.15 Let $k \geq 3$. There is a polylin function mapping $w \in \Sigma^*$ to a formula $Ruler_w(x,y)$ in L_k such that if w is of length n,

1. $\ell n(Ruler_w) = O(n)$, and
2. if $\langle N,',\bar{M}_k\rangle$ satisfies A_w, and if $a,b \in N$, $\langle N,',\bar{M}_k\rangle$ satisfies $Ruler_w(a,b)$ $\langle\Rightarrow\rangle$ $\langle N,',\bar{M}_k\rangle$ satisfies Point (a) $\wedge$ Point (b), and there are exactly $2^{2^n}-1$ distinct elements c, $a < c < b$, such that $\langle N,',\bar{M}_k\rangle$ satisfies Point (c).

Proof Let $w \in \Sigma^*$ be of length n. Given Lemma 2.8, it will be sufficient to define $Ruler_w$, to assert Point (x) $\wedge$ Point (y), and $bin(id(y)) = bin(id(x)) + 2^{2^n}$. Figure 2.16 provides a pictorial representation of the symbols of id(x) and id(y) as encoded by elements of N "close" to x and y.

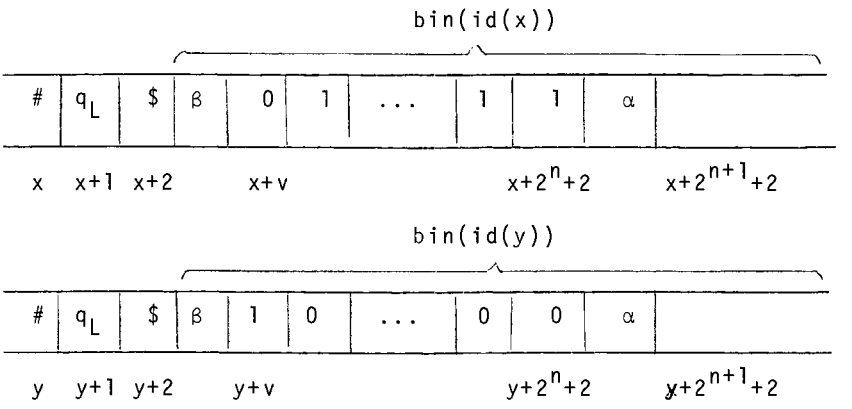

Figure 2.16 $bin(id(y)) = bin(id(x)) + 2^{2^n}$.

We therefore define $\text{Ruler}_w(x,y)$ to be a formula expressing the conjunction of a) - c) below:

a) $\text{Point}(x) \wedge \text{Point}(y)$

b) for all z, $2^n + 2 < z \leq 2^{n+1} + 2$,
$\text{Code}_\sigma(x+z) \Longleftrightarrow \text{Code}_\sigma(y+z)$
for all $\sigma \in \{0,1\}$,

c) there is v, $3 \leq v \leq 2^n + 2$ such that
i) $\text{Code}_0(x+v) \wedge \text{Code}_1(y+v)$, and
ii) for all u, $3 \leq u < v$,
$\text{Code}_\sigma(x+u) \Longleftrightarrow \text{Code}_\sigma(y+u)$
for all $\sigma \in \{0,1\}$, and
iii) for all z, $v < z \leq 2^n + 2$,
$\text{Code}_1(x+z) \wedge \text{Code}_0(y+z)$.

We let the reader convince himself Ruler_w so defined satisfies requirement 2 of the lemma.

As in Lemma 2.12, one can construct the formula Ruler_w directly from a), b) and c) above using a fixed number (independent of w) of instances of the formula Add_w, concatenated with a fixed number of additional symbols. □

Lemma 2.17 Let $k \geq 3$. There is a polylin function mapping $w \in \Sigma^*$ to a formula $\text{Ord}_w(x,y)$ such that if w is of length n,

1. $\ell n(\text{Ord}_w) = O(n)$, and
2. for all $\langle N, ', \bar{M}_k \rangle$ which satisfies A_w, and $a,b \in N$,
$\langle N, ', \bar{M}_k \rangle$ satisfies $\text{Ord}_w(a,b)$ $\Longleftrightarrow$
$\langle N, ', \bar{M}_k \rangle$ satisfies $\text{Point}(a) \wedge \text{Point}(b)$, and $a < b$.

Proof Let $w \in \Sigma^*$ be of length n. Suppose $\langle N,', \bar{M}_k \rangle$ satisfies A_w.

Now, if $a,b \in N$ such that Point (a) $\wedge$ Point (b), then because of the way M_c "counts", we have that $a < b \iff bin(id(a)) < bin(id(b))$. Accordingly, we define $Ord_w(x,y)$ to be a formula expressing

Point (x) $\wedge$ Point (y) $\wedge$ there is v, $3 \leq v \leq 2^{n+1} + 2$ such that v is the smallest number such that x+v and y+v code different symbols, and for this v, x+v codes 0 and y+v codes 1.

Clearly, then, Ord_w satisfies requirement 2 of the lemma.

We can construct Ord_w using only a fixed number of instances of Add_w, concatenated with a fixed number of additional symbols. □

We are finally in a position to write a sentence that succintly describes the computation of STM M on input w and asserts that the accepting i.d. of appropriate length appears in the computation. (In fact, by our conventions the accepting state is entered only on inputs that are accepted, so we need only assert that the accepting state is entered).

Suppose $M = (\Sigma,\Gamma,S,\delta,q_i,q_a)$, with $\Gamma \cap S = \phi$. Clearly by choosing $k > 3$ sufficiently large, we can assume there are formulas $Code_\sigma(x)$ for $\sigma \in \Gamma \cup S \cup \{\not\in\}$ such that for $n \in N$

$\langle N,',\bar{M}_k\rangle$ satisfies $\mathrm{Code}_\sigma(n)$ $\Longleftrightarrow$

n $(M_4,\ldots,M_k)$-codes σ.

Throughout the remainder of this section, both k and the formulas $\mathrm{Code}_\sigma(x)$, for $\sigma \in \Gamma \cup S \cup \{\not\in\}$, are fixed.

Definition 2.18 Let $\langle N,',\bar{M}_k\rangle$ satisfy A_w. The points of N are those $n \in N$ such that $\langle N,',\bar{M}_k\rangle$ satisfies $\mathrm{Point}(n)$.

Lemma 2.19 There is a polylin function mapping $w \in (\Gamma \cup S \cup \{\not\in\})^+$ to a formula $H_w(x)$ so that

1. $\ell n(H_w) = O(\ell n(w))$, and
2. if $\langle N,',\bar{M}_k\rangle$ satisfies A_w, and $a \in N$, $\langle N,',\bar{M}_k\rangle$ satisfies $H_w(a)$ $\Longleftrightarrow$ a is the $\ell n(w)^{th}$ point (in order) of $\langle N,',\bar{M}_k\rangle$, and the first $\ell n(w)$ points $(M_4,\ldots,M_k)$-code the successive symbols of w.

Proof We sketch an inductive procedure for constructing a formula $D_w(x,x_1,x_2,x_3)$ such that if $\langle N,',\bar{M}_k\rangle$ satisfies A_w, and $a,a_1,a_2,a_3 \in N$, $\langle N,',\bar{M}_k\rangle$ satisfies $D_w(a,a_1,a_2,a_3)$ $\Longleftrightarrow$ a is the $\ell n(w)^{th}$ point of $\langle N,',\bar{M}_k\rangle$, and the first $\ell n(w)$ points code the successive symbols of w, and $a_1 = a_2 + a_3$ and $a_3 \leq 2^{\ell n(w)+1}$.

Once D_w is constructed, we simply define $H_w(x)$ to be $\exists \bar{x}_3 D_w(x,\bar{x}_3)$.

If $w \in \Gamma \cup S \cup \{\not\in\}$, we let $D_w(x,x_1,x_2,x_3)$ be a formula expressing

x codes w and $[\bigvee_{1\leq i\leq 4} x_1 = x_2+i \wedge x_3 = i]$.

If $w \in (\Gamma \cup S \cup \{\not\in\})^+$, $\sigma \in \Gamma \cup S \cup \{\not\in\}$, we assume $D_w(x,x_1,x_2,x_3)$ has already been defined.

We define $D'_{w\sigma}(x,x_1,x_2,x_3)$ to be a formula expressing

1. x is a point, and $\exists z[\exists \bar{x}_3 D_w(z,\bar{x}_3)]$ and $\mathrm{bin}(\mathrm{id}(x))$ is the result of adding 1 to $\mathrm{bin}(\mathrm{id}(z))$, and x $(M_4,\ldots,M_k)$-codes σ, and
2. $\exists \bar{t}_2$ with $x_3 = t_1 + t_2$ with $t_1,t_2 \leq 2^{\ell n(w)+1}$, and $x_1 = (x_2 + t_1) + t_2$.

That $\mathrm{bin}(\mathrm{id}(x)) = \mathrm{bin}(\mathrm{id}(y))+1$ can be defined from $[a = b+c \wedge c \leq 2^{n+1}]$ follows as in Lemmas 2.15 and 2.17. It follows that $D'_{w\sigma}$ can be expressed using a fixed number of occurrences of D_w; we then define $D_{w\sigma}(x,\bar{x}_3)$ to be the formula obtained by replacing the occurrences of D_w in $D'_{w\sigma}$ by one such occurrences, as in Theorem 2 of Chapter 7. The details of the construction and the remainder of the proof are similar to those in Lemmas 2.15, 2.17, and 1.7. □

Lemma 2.20 There is a polylin function mapping $w \in \Sigma^*$ to a sentence Acc_w in L_k such that

1. $\ell n(\mathrm{Acc}_w) = O(\ell n(w))$, and
2. w is accepted by $M \iff \mathrm{Acc}_w \in S_k$.

Clearly, once we prove the lemma, Theorem 2.1 follows immediately.

Proof of Lemma 2.20 By Lemma 1.5 of Chapter 6, there is a function N'_M: $\Delta^3 \to P(\Delta^3)$ with the following property:

If d is any i.d. of M of length ℓ, $d = d_1 \cdot \ldots \cdot d_\ell$ with $d_i \in \Delta$ for $i \in I_\ell$, and if $f = d_{\ell+2} \cdot \ldots \cdot d_{2\ell+1}$ with $d_i \in \Delta$ for $\ell+2 \leq i \leq 2\ell+1$, and $\not{c}d\not{c}f\not{c} = d_0 \cdot \ldots \cdot d_{2\ell+2}$, then $f \in \text{Next}_M(d)$ iff $d_{\ell+j}d_{\ell+j+1}d_{\ell+j+2} \in N'_M(d_{j-1}d_jd_{j+1})$, for $1 \leq j \leq \ell$.

We construct Acc_w to say that successive points of N encode a computation which starts correctly, proceeds correctly, and enters the accepting state, as follows.

We construct sentences expressing

a) A_w, and there is x such that $H_{\not{c}q_iw}(x)$, and if z is a point larger than x but smaller than or equal to the 2^nth point, then z codes $\not{b}$.

b) If z is a point which codes $\not{c}$ and z' is the 2^nth point after z, z' codes $\not{c}$.

c) If z_1, z_2, z_3 are three successive points, such that z_2 does not code $\not{c}$, and y_i is the 2^nth point after z_i, $i = 1,2,3$, and if z_1, z_2, z_3 code $\sigma_1, \sigma_2, \sigma_3$ respectively, then for some $\delta_1\delta_2\delta_3 \in N'_M(\sigma_1\sigma_2\sigma_3)$, we must have y_1, y_2, y_3 code $\delta_1, \delta_2, \delta_3$ respectively.

d) There is a point y such that y codes $\not{c}$ and y+1 codes q_a.

We define Acc_w so that it is equivalent to the conjunction of a), b), c) and d).

Clearly the only interpretation satisfying a), b) and c) is one which codes the computation of M on input w. In this case, d) is satisfied iff M accepts w. Thus

$w \in A \iff$ for some $\langle N,',\bar{M}_k\rangle$,

$\langle N,',\bar{M}_k\rangle$ satisfies Acc_w. Thus property 2 of the lemma is satisfied.

Each of the sentences a), b), c) and d) can obviously be written using a fixed number of instances of A_w, $H_{\not\in q_i w}$, $Ruler_w$ and Ord_w, concatenated with a fixed number of additional symbols.

In addition, we can show

Corollary 2.21: There is a rational $c > 0$ such that $SAT(1\text{1FM}) \notin NTIME(2^{2^{cn}})$.

Proof Using the results of section 3 of chapter 4, we can show $SAT(MONN) \leq_{log\text{-}lin} SAT(1\text{1FM})$. Briefly, we map a sentence F of length n in the language of the class MONN to a sentence which asserts

(1) there is a unique origin, and

(2) there are no finite chains of length $< 2^{2^n}$, and

(3) F*

where F* is F with each occurrence of a subformula x' = y replaced by f(x) = y. □

Section 3: Lower Bounds for the Theory of Two Successors

Let T denote the structure $\langle \{0,1\}^*, r_0, r_1 \rangle$, where for $a \in \{0,1\}^*$,

$r_0(a) = a \cdot 0$, and

$r_1(a) = a \cdot 1$.

Let L be a first-order language with two binary relation symbols in it. For any $a,b \in \{0,1\}^*$, we interpret the assertion in the language L that the first binary relation holds for the pair a,b as $r_0(a) = b$, and we interpret the assertion in the language L that the second binary relation holds for the pair a,b as $r_1(a) = b$.

The aim of this section is to prove:

Theorem 3.1 There is $c \in Q$, $c > 1$, such that $TH(T) \notin NTIME(c^n)$

We actually prove:

Theorem 3.2 $INEQ(\{0,1\}\text{-}\{U,\cdot,^2\}) \leq_{p\ell} TH(T)$.

Using Theorem 2.3 of Chapter 6, Theorem 3.1 then follows immediately.

Proof of Theorem 3.2 Given a $\{0,1\}\text{-}\{U,\cdot,^2\}$ expression E, we construct a formula $F_E(v,x,y,z)$ such that

1. $\ell n(F_E) = O(\ell n(E))$, and
2. for all $w,a,b,c \in \{0,1\}^*$, T satisfies

   $F_E(w,a,b,c) \iff w \in L(E)$ and $a = b \cdot c$ for $\ell n(c) \leq$ the length of the longest word in $L(E)$.

F_F is constructed inductively on the structure of E.

If E is (0), we let $F_E(v,x,y,z)$ be a formula expressing

$$v = 0 \wedge \bigvee_{\sigma \in \{0,1,\lambda\}} (x = y\sigma \wedge z = \sigma)$$

If E is (1), F_E is identical to $F_{(0)}$, with "v=0" replaced by "v=1". Clearly, these formulas satisfy requirement 2 above.

Before proceeding, we introduce some abbreviations for formulas. If $F_E(v,x,y,z)$ has already been defined, we let

$$Mem_E(v)$$

denote any formula of the form

$$\exists x\ \exists y\ \exists z\ F_E(v,x,y,z);$$

we let

$$Concat_E(x,y,z)$$

denote any formula of the form

$$\exists v\ F_E(v,x,y,z).$$

Now, if E is $E_1 \cup E_2$, then the length of the longest word in L(E) is the maximum of the lengths of the longest words in $L(E_1)$ and $L(E_2)$. Therefore, we define $F'_E(v,x,y,z)$ to be

$$[Mem_{E_1}(v) \vee Mem_{E_2}(v)] \wedge$$
$$[Concat_{E_1}(x,y,z) \vee Concat_{E_2}(x,y,z)].$$

Clearly F'_E so defined satisfies requirement 2 above. We let $F_E(v,x,y,z)$ be that formula obtained from F'_E by replacing the two occurrences of F_{E_i} by one such occurrence, for i = 1,2, as in Theorem 2 of Chapter 7.

If E is $E_1 \cdot E_2$, then the length of the longest word in L(E) is the sum of the lengths of the longest words in $L(E_1)$ and $L(E_2)$. Accordingly, we define $F'_E(v,x,y,z)$ to be

$$\exists v_1\ \exists v_2\ \mathrm{Mem}_{E_1}(v_1) \wedge \mathrm{Mem}_{E_2}(v_2) \wedge$$
$$\mathrm{Concat}_{E_2}(v,v_1,v_2) \wedge \exists z_1\ \exists z_2\ \exists x_1\ \mathrm{Concat}_{E_1}(x_1,y,z_1) \wedge$$
$$\mathrm{Concat}_{E_2}(x,x_1,z_2) \wedge \mathrm{Concat}_{E_2}(z,z_1,z_2).$$

Clearly, F'_E so defined satisfies requirement 2 above. We let $F_E(v,x,y,z)$ be that formula obtained from F'_E by replacing the two occurrences of F_{E_1} by a single occurrence, and the four occurrences of F_{E_2} by a single occurrence, as in Theorem 2 of Chapter 7.

If E is A^2, then the length of the longest word in L(E) is twice the length of the longest word in L(A). We therefore define $F'_E(v,x,y,z)$ as

$$\exists v_1\ \exists v_2\ \mathrm{Mem}_A(v_1) \wedge \mathrm{Mem}_A(v_2) \wedge$$
$$\mathrm{Concat}_A(v,v_1,v_2) \wedge \exists z_1\ \exists z_2\ \exists x_1\ \mathrm{Concat}_A(x_1,y,z_1) \wedge$$
$$\mathrm{Concat}_A(x,x_1,z_2) \wedge \mathrm{Concat}_A(z,z_1,z_2).$$

Again, it is clear F'_E satisfies requirement 2 above. We let $F_E(v,x,y,z)$ be that formula obtained from F'_E be replacing the six occurrences of F_A by a single occurrence, as in Theorem 2 of Chapter 7.

Now, as in Lemma 1.7, it is possible to define F_E so that it contains only a fixed number of variables, for all E. It then follows that $\ell n(F_F) = O(\ell n(E))$.

To show that $INEQ(\{0,1\} - \{U,\cdot,^2\}) \leq_{p\ell} TH(T)$, suppose first (E_1,E_2) is a pair of well-formed $\{0,1\}$-$\{U,\cdot,^2\}$ expressions. We define $Ineq_{E_1,E_2}$ to be the sentence

$$\exists v \sim [Mem_{E_1}(v) \Longleftrightarrow Mem_{E_2}(v)].$$

Thus, $L(E_1) \neq L(E_2) \Longleftrightarrow Ineq_{E_1,E_2} \in TH(T)$.

If (E_1,E_2) is not a pair of well-formed $\{0,1\}$-$\{U,\cdot,^2\}$ expressions, we define $Ineq_{E_1,E_2}$ to be some fixed false sentence.

Then the fact that function mapping (E_1,E_2) to $Ineq_{E_1,E_2}$ is linear bounded and in polylin follows easily from the above. □

Section 4: Lower Bounds for the Theory of a 1-1 Unary Function

We finally consider SAT(11F), the set of satisfiable sentences of a 1-1 unary function in a first-order language L with a single binary relation symbol, first considered in Section 2 of Chapter 4. We use here some of the notation already developed in Section 2 of Chapter 4. The aim of this section is to prove

Theorem 4.1 There is a $c \in Q$, $c > 1$, such that $SAT(11F) \notin NTIME(c^n)$

The proof will be by efficient arithmetization, in the language L, of simple Turing machines which run in exponential time; more specifically, we use the following theorem, whose proof we omit:

Theorem 4.2 [Sto74] Let Σ be a finite alphabet. Let $B \subseteq \Sigma^*$ be such that

$$\{A \mid A \text{ is accepted by a (nondeterministic) STM within time } 2^n\} \leq_{\text{log-lin}} B.$$

Then for some $c \in Q$, $c > 1$, $B \notin NTIME(c^n)$.

To prove Theorem 4.1, then, using Theorem 4.2, it will be sufficient to prove that given any (nondeterministic) STM M with input alphabet Σ which accepts a set $A \subseteq \Sigma^*$ within time 2^n, there is a function $f: \Sigma^* \to \{\text{the sentences of L}\}$ such that

1. for all $w \in \Sigma^*$, $[w \in A \iff f(w) \in SAT(11F)]$,
2. f is linear bounded, and
3. f is in logspace.

Clearly, if M accepts A within time 2^n, M accepts A within space 2^n and thus within space $2^{n+1}-2$. Now given any element $w \in \Sigma^*$ of length n, our conventions require that if M accepts w, every i.d. of M appearing in an accepting computation of M on input $w\not\!\!\beta^{2^{n+1}-n-2}$ be of length $2^{n+1}-1$, where $\not\!\!\beta$ is the blank tape symbol of M. Since M accepts A within time 2^n, we need consider only 2^n such i.d.'s. If we leave space for an end-marker to separate i.d.'s, we need space $(2^{n+1})\cdot 2^n = 2^{2n+1}$ to code such a computation. Our method of encoding the symbols of such a computation will be via the <u>number</u> of loops of a particular size. We proceed as follows:

<u>Lemma 4.3</u> There is a logspace function mapping $w \in \Sigma^*$ to a formula $Str_{\ell n(w)}(\bar{x}_2,\bar{y}_2,\bar{z}_3)$ in L, such that if $w \in \Sigma^*$ is of length n,

1. $\ell n(Str_n) = O(n)$, and
2. for any $A \in \mathrm{IIF}$ and any $a_1,a_2,b_1,b_2,c_1,c_2,c_3 \in A$, A satisfies $Str_n(\bar{a}_2,\bar{b}_2,\bar{c}_3)$ $\Leftrightarrow$ there are $\ell,k \in I_{2^n}$ such that ℓ is the smallest positive integer such that $f^k(b_1) = b_2$, $f^{\ell+k}(c_1) = c_2$, and for all $i \in I_{\ell+k-1}$, $f^i(c_1) \neq c_3$.

<u>Proof</u> We give an inductive procedure which constructs $Str_{\ell n(w)}$.

If $\ell n(w) = 0$, that is if $w = \lambda$, we define $Str_0(\bar{x}_2,\bar{y}_2,\bar{z}_3)$ to be any formula expressing

$f(x_1) = x_2$, $f(y_1) = y_2$,
$f^2(z_1) = z_2$, and $f(z_1) \neq z_3$.

Clearly, then, Str_0 satisfies requirement 2 of the lemma.

Let $w \in \Sigma^*$ be of length n, $\sigma \in \Sigma$, and suppose we have already shown how to construct Str_n. We define $Str'_{n+1}(\bar{x}_2,\bar{y}_2,\bar{z}_3)$ to be a formula of the form

$$\exists x' \; Str_n(x_1,x',x',x_2,x_1,x_2,x_1) \wedge$$
$$\exists y' \; Str_n(y_1,y',y',y_2,y_1,y_2,y_1) \wedge$$
$$\exists z'[Str_n(x_1,x',x',x_2,z_1,z',z_3) \wedge$$
$$Str_n(y_1,y',y',y_2,z',z_2,z_3) \wedge \sim z' = z_3].$$

It should be clear that Str'_{n+1} satisfies requirement 2 of the lemma.

We then let Str_{n+1} be the formula obtained from Str'_{n+1} by replacing the four occurrences of Str_n by a single occurrence, as in Theorem 2 of Chapter 7.

As in previous examples, one can easily show that the function we have defined mapping $w \in \Sigma^*$ to $Str_{\ell n(w)}$ is linear bounded and in logspace. □

Lemma 4.4 There is a logspace function mapping $w \in \Sigma^*$ to a formula $LA_{\ell n(w)}(x,y,z)$ such that for $w \in \Sigma^*$ of length n,

1. $\ell n(LA_n) = O(n)$, and
2. for any $A \in 11F$, and $a,b,c \in A$, A satisfies $LA_n(a,b,c) \iff |C_a|,|C_b| \leq 2^n$, and $|C_a|+|C_b| = |C_c$

Proof Let $w \in \Sigma^*$ be of length n. We define LA_n to be $Str_n(x,x,y,y,z,z,z)$. □

Now, using the formula LA_{2n+1}, we can write a formula that will allow us to quantify over elements in loops up to size 2^{2n+1}, and we can write a formula asserting there are loops of every size up to 2^{2n+1} in a structure. (The construction of such formulas appears below.) This suggests using the number of loops of a particular size to code a finite alphabet.

That is, if $M = (\Sigma,\Gamma,Q,\delta,q_0,q_a)$, we assume $\Gamma \cap Q = \phi$, $\# \notin \Gamma \cup Q$, and $\Gamma \cup Q \cup \{\#\} = I_k$ for some $k \in I$. For $j \in I_k$, if x is on a "small" loop, x will code j if there are exactly j loops of size $|C_x|$. We proceed to construct the formulas that will allow us to carry through this encoding.

Lemma 4.5 There is a logspace function mapping $w \in \Sigma^*$ to a formula $LOOP_{\ell n(w)}(x)$ in L such that for $w \in \Sigma^*$ of length n,

1. $\ell n(LOOP_n) = O(n)$, and
2. for any $A \in 11F$ and $a \in A$, A satisfies $LOOP_n(a) \iff |C_a| \leq 2^n$.

Proof For $w \in \Sigma^*$ of length n, we let $LOOP_n(a)$ be $\exists y\, \exists z\, LA_n(x,y,z)$. □

Lemma 4.6 There is a logspace function mapping $w \in \Sigma^*$ to a sentence $LOTS_{\ell n(w)}$ in L such that for $w \in \Sigma^*$ of length n,

1. $\ell n(LOTS_n) = O(n)$, and
2. for any $A \in 11F$,
   A satisfies $LOTS_n \iff$ for all $j \in I_{2^{n+1}}$,
   A has at least one loop of size j.

Proof Let $w \in \Sigma^*$ be of length n. We let $LOTS_n$ be a sentence expressing

there is a loop of size 1, and for all y, z such that $LOOP_n(y)$ and $LOOP_n(z)$, there is a z' such that $LA_n(y,z,z')$.

□

Before proving the lemma that will provide the symbol-encoding we require, we note that if x is such that $LOOP_n(x)$, then $|C_x| = |C_y|$ is expressed by any formula of the form

$$\forall \bar{v}_2[LA_n(\bar{v}_2,x) \iff LA_n(\bar{v}_2,y)].$$

We also note that if x is such that $LOOP_n(x)$, then $y \in C_x$ is expressed by any formula of the form

$$\exists \bar{y}_2 \; \exists \bar{z}_3 \; Str_n(x,y,\bar{y}_2,\bar{z}_3).$$

Lemma 4.7 For any $j \in I_k$, there is a logspace function mapping $w \in \Sigma^*$ to a formula $Code_{j,\ell n(w)}(x)$ such that if $w \in \Sigma^*$ is of length n,

1. $\ell n(Code_{j,n}) = O(n)$, and
2. for any $A \in 11F$, and any $a \in A$, A satisfies $Code_{j,n}(a) \iff |C_a| \leq 2^n$, and there are exactly j loops in A of size $|C_a|$.

Proof Let $w \in \Sigma^*$ be of length n. We define $Code_{j,n}(x)$ to be a formula expressing $LOOP_n(x)$, and there are $x_1,\ldots,x_j$ such that

1. $|C_x| = |C_{x_i}|$ for all $i \in I_j$,
2. $x_i \notin C_{x_\ell}$ for all $i,\ell \in I_j$, with $i \neq \ell$, and
3. if x_{j+1} is such that $|C_{x_{j+1}}| = |C_x|$, then $x_{j+1} \in C_{x_i}$ for some $i \in I_j$.

Clearly $Code_{j,n}$ so defined satisfies requirement 2 of the lemma.

It should also be clear from the remarks preceding the lemma that $Code_{j,n}$ can be constructed using only a fixed number (independent of n) of instances of $LOOP_n$, LA_n, and Str_n, concatenated with a fixed number of additional symbols. We leave the precise construction of $Code_{j,n}$ to the reader.

□

Definition 4.8 Let $A \in 11F$, $n \in N$, and suppose A satisfies $LOTS_n$. For $a,b \in A$ such that $|C_a| < |C_b| \leq 2^n$ we define $WORD(a,b)$ by induction on $|C_b|-|C_a|$. If $|C_b|-|C_a| = 1$,

$$WORD(a,b) = \begin{cases} j & \text{if } A \text{ satisfies } Code_{j,n}(a) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

If $|C_b|-|C_a|>1$, then since A satisfies $LOTS_n$ there is $c \in A$ such that $|C_b|-|C_c| = 1$. We define inductively,

$$WORD(a,b) = WORD(a,c)\cdot WORD(c,b).$$

We need one additional lemma before completing the proof of Theorem 4.1.

Lemma 4.9 There is a logspace function mapping $w' \in I_k^+$ to a formula $Const_{w'}(x,y)$ such that if w' is of length n,

1. $\ell n(Const_{w'}) = O(n)$, and

2. for any $A \in 11F$ such that A satisfies $LOTS_n$, and any $a,b \in A$,

A satisfies $Const_{w'}(a,b) \iff |C_a| = 1$ and $|C_b| = n+1$ and $WORD(a,b) = w'$.

Proof We give an inductive procedure for defining the formula $Const_{w'}$.

If $w' \in I_k$, that is, if $\ell n(w') = 1$, we let $Const_{w'}(x,y)$ be any formula expressing

x is a loop of size 1,

y is a loop of size 2, and $Code_{w',1}(x)$.

Then clearly $Const_{w'}$ satisfies requirement 2 of the lemma.

If $w \in I_k^*$ is of length n, and $\sigma \in I_k$, we assume $Const_w$ has already been defined.

We define $Const_{w\sigma}(x,y)$ to be any formula expressing

$\exists z\ Const_w(x,z)$, and $Code_{\sigma,n+1}(z)$, and

$|C_z|+1 = |C_y|$.

Clearly, $Const_{w\sigma}(x,y)$ so defined satisfies requirement 2 of the lemma. It should also be clear by the proof of Lemmas 4.3, 4.4 and 4.5 and the remarks proceeding Lemma 4.7 that $Const_{w\sigma}$ can be defined from $Const_w$ and a fixed number (independent of n) of instances of Str_{n+1}, since we can express both $Code_{\sigma,n+1}$ and $|C_z|+1 = |C_y|$, given that $|C_z| \leq n$, using only a fixed number of instances of Str_{n+1}.

However, the use of even a fixed number of instances of Str_{n+1} in our inductive definition of $Const_{w'}$, where w' is of length n, does not yield a formula of length O(n), but of length $O(n^2)$. However, as in Theorem 8 of Chapter 7, we can give an inductive procedure for defining a formula $C_{w'}(x,y,\bar{v}_2,\bar{y}_2,\bar{z}_3)$ equivalent to $Const_{w'}(x,y) \wedge Str_{\ell n(w')+1}(\bar{v}_2,\bar{y}_2,\bar{z}_3)$ such that $C_{w\sigma}$ is the concatenation of $C_{w\sigma}$ and a fixed number of additional symbols. Thus for $w \in I_k^+$, $\ell n(C_w) = O(\ell n(w))$. We could then define $Const_{w'}(x,y)$ to be

$$\exists \bar{v}_2 \; \exists \bar{y}_2 \; \exists \bar{z}_3 C_{w'}(x,y,\bar{v}_2,\bar{y}_2,\bar{z}_3).$$

We let the reader convince himself that this strategy can indeed be carried out, and that the function mapping $w' \in I_k^+$ to $Const_{w'}$ so defined is linear bounded and in logspace. □

Proof of Theorem 4.1 As stated previously, it will be sufficient to show there is a logspace function mapping $w \in \Sigma^*$ to a sentence Acc_w in L such that

1. $\ell n(Acc_w) = O(\ell n(w))$, and
2. $w \in A \iff Acc_w \in SAT(1\text{IF})$.

We use Lemma 1.5 of Chapter 6, which implies there is a function $N'_M: I_k^3 \to P(I_k^3)$ which characterizes the computation of M.

We wish to express the fact that the model in question encodes enough of a computation of M on input $w\emptyset^{2^{n+1}-n-2}$ to

include the accepting i.d. $acc(2^{n+1}-2)$. That is, we wish to state via a sentence Acc_w that the symbols encoded by elements on "small" loops, taken in turn, represent successive i.d.'s of M, separated by the end-marker #, on input $w\emptyset^{2^{n+1}-n-2}$

Before proceeding with the construction of Acc_w, we remark that using a fixed number of instances of LA_{n+1}, we can express the predicate $|C_x| = 2^{n+1}$, and using a fixed number of instances of LA_{2n+1}, we can express the predicate $|C_x| < |C_y| \leq 2^{2n+1}$. The details are similar to those in Lemma 2.17 and the text preceding Lemma 1.7 and are left to the reader.

We now define Acc_w to be a formula expressing the conjunction of 1) - 4) below:

1) $LOTS_{2n+1}$, and there are x,y such that $Const_{\#q_0w}(x,y)$, and for all z such that $|C_y| \leq |C_z| < 1+2^{n+1}$ it must be that $Code_{\emptyset,2n+1}(z)$.

2) For every z' such that $Code_{\#,2n+1}(z')$, and any y' such that $|C_{y'}| = |C_{z'}|+2^{n+1} \leq 2^{2n+1}$, we have $Code_{\#,2n+1}(y')$.

3) For every y_2 such that $|C_{y_2}| \leq 2^{2n+1}$ and $\sim Code_{\#,2n+1}(y_2)$, and any y_1,y_3 such that $|C_{y_1}|+1 = |C_{y_2}|$, and $|C_{y_2}|+1 = |C_{y_3}|$, we have

$$\bigvee_{\sigma_1\sigma_2\sigma_3 \in \Delta^3} [Code_{\sigma_1,2n+1}(y_1) \wedge Code_{\sigma_2,2n+1}(y_2) \wedge Code_{\sigma_3,2n+1}(y_3) \rightarrow \exists \bar{z}_3 \text{ with } |C_{z_i}| = |C_{y_i}| + 2^{n+1}$$

for i = 1,2,3, and such that

$$\delta_1\delta_2\delta_3 \bigvee_{\sigma_1\sigma_2\sigma_3 \in N'_M} [Code_{\delta_1,2n+2}(z_1) \wedge Code_{\delta_2,2n+2}(z_2) \wedge Code_{\delta_3,2n+2}(z_3)]].$$

4) $\exists z\ Code_{q_a,2n+1}(z)$.

1) expresses the fact that the encoding starts with the correct initial i.d., 2) that end-markers align properly, 3) that the computation proceeds properly, and 4) that it is an accepting computation. Clearly, the only structures which satisfy 1), 2) and 3) are those which encode the first 2^n steps of the computation of M on input w. In this case, 4) is satisfied by the structure iff $w \in A$. Thus

$w \in A \iff$ for some $A \in \mathbb{F}$,

A satisfies Acc_w.

Thus property 2 of the lemma is satisfied.

Acc_w can be defined using a fixed number of instances of $LOTS_{2n+1}$, $Const_{\#q_0w}$, LA_{n+1}, LA_{2n+1}, $Code_{\sigma,2n+1}$ and $Code_{\sigma,2n+2}$ for $\sigma \in I_k$ concatenated with a fixed number of additional symbols.

Now let δ be a fixed element of Σ. The functions mapping $w \in \Sigma^*$ of length n to $w\delta^{n+1}$, $w \in \Sigma^*$ of length n to $w\delta^{n+2}$, and $w \in \Sigma^*$ of length n to $w\delta$ are all in logspace, as is the function mapping $w \in \Sigma^*$ to $\#q_0w$. It then follows from Lemmas 4.4, 4.6, 4.7 and 4.9 that the function mapping $w \in \Sigma^*$ to Acc_w is in logspace.

The fact that this function is linear bounded also follows directly from Lemmas 4.4, 4.6, 4.7 and 4.9.

References

[Ber77] Berman, L., "Precise bounds for Presburger arithmetic and the reals with addition: preliminary report", Proc. 18th Annual Symp. on Foundations of Computer Science, 1977.

[Ch56] Church, A., Introduction to Mathematical Logic, I, Princeton, 1956.

[Co71] Cook, S.A., "Characterization of pushdown machines in terms of time bounded computers", JACM 18, 1971, 4-18.

[Co73] Cook, S.A., "A hierarchy for nondeterministic time complexity", J. Comput. Syst. Sci. 7, 4, 1973, 343-353.

[Cob64] Cobham, A., "The intrinsic computational difficulty of functions", Proc. Internat. Congr. Logic, Method. and Philos. Sci., 1964, 24-30.

[Col75] Collins, George E., "Quantifier elimination for real closed fields by cylindrical algebraic decomposition", Springer-Verlag Lecture Notes in Computer Science 33, Proc. 2 G.I. Fachtagung Automatentheorie und Formale Sprachen, 1975.

[Coo72] Cooper, D.C., "Theorem-proving in arithmetic without multiplication", Machine Intelligence 7, Univ. Edinburgh Press, 1972, 91-100.

[Ehr59] Ehrenfeucht, A., "Decidability of the theory of one function", Notices AMS 6, 1959, 268.

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

[Eich57] Eichholz, T., "Semantische Untersuchungen zur Entscheidbarkeit im Prädikatenkalkül mit Funktionsvariablen", Archiv Math. Logik Grundl. 3 , 1957, 19-28.

[ELTT65] Ershov, Y.L., Lavrov, I.A., Taimanov, A.D. and Taitslin, M.A., "Elementary theories", Russian Math. Surveys, 20, 1965, 35-105.

[ER66] Elgot, C.E., Rabin, M.O., "Decidability and undecidability of extensions of second (first) order theory of (generalized) successor", J. Symbolic Logic 31, 1966, 169-181.

[Fer74] Ferrante, J., "Some upper and lower bounds on decision procedures in logic", Doctoral Thesis, Dept. of Math., MIT, Cambridge, Mass., Project MAC TR-139, 1974.

[FeGe77] Ferrante, J. and Geiser, J., "An efficient decision procedure for the theory of rational order", Theoretical Computer Science 4, 2, 1977, 227-234.

[FiR74] Fischer, M.J. and Rabin, M.O., "Super-exponential complexity of Presburger arithmetic", Proc. AMS Symp. on Complexity of Real Computational Processes, vol. VII, 1974.

[FMS76] Fleischmann, K., Mahr, B., Siefkes, D., "Bounded concatenation theory as a uniform method for proving lower complexity bounds", in Logic Colloquium 76, ed. R.O. Gandy and J.M.E. Hyland, North-Holland 1977, 471-490.

[FMS76'] Fleischmann, Mahr, Siefkes, "Complexity of decision problems: notes of a summer course of Abert R. Meyer", Tech. Report, Technische Universität Berlin, Fachbereich Informatik, 1976.

[FR75] Ferrante, J. and Rackoff, C., "A decision procedure for the first order theory of real addition with order", SIAM J. Comput. 4, 1, 1975, 69-76.

[FV59] Feferman, S. and Vaught, R.L., "The first order properties of products of algebraic systems", Fund. Math. 47, 1959, 57-103.

[Hop-Ul69] Hopcroft, J.E., and Ullman, J.D., Formal Languages and Their Relation to Automata, Addison-Wesley, Reading, Ma., 1969.

[Kam50] Kamke, E., Theory of Sets, Dover, N.Y., N.Y., 1950.

[Lad77] Ladner, R., "Applications of model-theoretic games to discrete linear orders and finite automata", Information and Control 33, 4, 1977, 281-303.

[Lad77'] Ladner, R., "The computational complexity of provability in systems of modal propositional logic", SIAM J. on Computing 6, 1977, 467-480.

[LeTour69] Le Tourneau, J.J., "Decision problems related to the concept of operation", Doctoral Thesis, Dept. of Mathematics, University of California at Berkeley, 1969.

[Lind74] Lind, J., "Computing in logarithmic space", Project MAC Technical Memo 52, MIT, Cambridge, Mass., September 1974.

[MB68] Maclane, S. and Birkhoff, G., Algebra, Macmillan, 1968.

[Men64] Mendelson, E., Introduction to Mathematical Logic, Van Nostrand Reinhold, 1964.

[Mey75] Meyer, A.R., "Weak monadic second order theory of successor is not elementary-recursive", Boston Univ. Logic Colloquium Proc., Springer-Verlag, 1975, 132-154.

[Monk75] Monk, L., "Elementary recursive decision procedures", Ph.D. Thesis, University of California at Berkeley, 1975.

[Mos52] Mostowski, A., "On direct powers of theories", J. Symb. Logic 17, 1952, 1-31.

[MS72] Meyer, A.R., and Stockmeyer, L.J., "The equivalence problem for regular expressions with squaring requires exponential space", Proc. 13 IEEE Symp. on Switching and Automata Theory, 1973, 125-129.

[MT49] Mostowski, A., Tarski, A., "Arithmetical classes and types of well-ordered systems", Bull. AMS 55, 65 (abstract).

[Opp73] Oppen, D.C., "Elementary bounds for Presburger arithmetic", Proc. 5th ACM Symp. on Theory of Computing, 1973, 34-37, to appear (under different title), JACM.

[Pet67] Péter, R., Recursive Functions, Academic Press, 1967.

[Pre29] Presburger, M., "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchen die Addition als einzige Operation hervortritt", Comptes Rendus, I. Congrès des Math. des Pays Slaves, Warsaw, 1929, 192-201, 395.

[Rab69] Rabin, M.O., "Decidability of second-order theories and automata on infinite trees", Transactions AMS 141, 1969, 1-35.

[Rack75] Rackoff, C.W., "Complexity of some logical theories", Doctoral Thesis, Dept. of Electrical Engineering, MIT, Cambridge, Mass., Project MAC TR-144, 1975.

[Rack75'] Rackoff, C.W., "The complexity of theories of the monadic predicate calculus", IRIA Report 136, Roquencourt, France, 1975.

[Rack76] Rackoff, C.W., "On the complexity of the theories of weak direct powers", Jour. of Symb. Logic 41, 1976, 561-573.

[Rob74] Robertson, E.L., "Structure of complexity in the weak monadic second-order theories of the natural numbers", Research Report CS-73-31, Dept. of Applied Analysis and Computer Science, Univ. of Waterloo (Dec. 1973); also Proc. 6th ACM Symp. on Theory of Computing, 1974, 161-171.

[Rit63] Ritchie, R.W., "Classes of predictably computable functions", Trans. AMS 106, 1963, 139-173.

[Sav70] Savitch, W.J., "Relationships between nondeterministic and deterministic tape complexities", J. Comput. Syst. Sci. 4, 2, 1970, 177-192.

[Sei74] Seiferas, J., "Nondeterministic time and space complexity classes", Doctoral Thesis, Dept. of Elect. Eng., MIT, Project MAC TR-137, 1974.

[Sko31] Skolem, T., "Über einige Satzfunktionen in der Arithmetik", Skrifter Norske Vid. Akad. Oslo I. Klasse 1930, no.7, Oslo.

[SFM73] Seiferas, J., Fischer, M.J., and Meyer, A.R., "Refinements of the nondeterministic time and space hierarchies", Proc. 14th IEEE Symp. on Switching and Automata Theory, 1973, 130-137.

[SFM77] Seiferas, J., Fischer, M.J., and Meyer, A.R., "Separating nondeterministic time complexity classes", JACM 25, 1978, 146-167.

[SM73] Stockmeyer, L.J., and Meyer, A.R., "Word problems requiring exponential time: preliminary report", Proc. 5th ACM Symp. on Theory of Computing, 1973, 1-9.

[Sho67] Shoenfield, J.R., Mathematical Logic, Addison-Wesley, Reading, Ma., 1967.

[Smu68] Smullyan, R.M., First-Order Logic, Springer-Verlag, N.Y., N.Y., 1968.

[Sto74] Stockmeyer, L.J., "The complexity of decision problems in automata theory and logic", Project MAC Tech. Report 133, 1974.

[Szm55] Szmielew, W., "Elementary properties of abelian groups", Fund. Math. 41, 1955, 203-271.

[Ten72] Tenney, R.L., "Decidable pairing functions", Dept. of Computer Science, Cornell Univ., Technical Report 72-136, 1972.

[Ten74] Tenney, R.L., "Decidable pairing functions", manuscript, 1974.

[Ten74'] Tenney, R.L., "Second-order Ehrenfeucht games and the decidability of the second order theory of an equivalence relation", Jour. of the Australian Math. Soc., 20, Series A, 1975, 323-331.

SUBJECT AND NOTATION INDEX