THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES

by

Charles Weill Rackoff

February 1975

Research for this paper was supported by the National Science Foundation under research grant number GJ-34671.

THE COMPUTATIONAL COMPLEXITY OF SOME LOGICAL THEORIES

by

Charles Weill Rackoff

Submitted to the Department of Electrical Engineering on September 9, 1974, in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

ABSTRACT

Upper and lower bounds on the inherent computational complexity of the decision problem for a number of logical theories are established.

A general form of Ehrenfeucht game technique for deciding theories is developed which involves analyzing the expressive power of formulas with given quantifier depth. The method allows one to decide the truth of sentences by limiting quantifiers to range over finite sets. In particular for <u>the theory of integer addition</u> an upper bound of space $2^{2^{cn}}$ is obtained; this is close to the known lower bound of nondeterministic time $2^{2^{c'n}}$.

A general development of decision procedures for theories of product structures is presented, which allows one to conclude in most cases that if the theory of a structure is elementary recursive, then the theory of its weak direct power (as well as other kinds of direct products) is elementary recursive. In particular, for <u>the theory of the weak direct power of $\langle N,+\rangle$</u>, and hence for integer multiplication, an upper bound of space $2^{2^{2^{cn}}}$ is obtained. The known lower bound is nondeterministic time $2^{2^{2^{c'n}}}$.

Finally, the complexity of the theories of pairing functions is discussed, and it is shown that no collection of pairing functions has an elementary recursive theory.

THESIS SUPERVISOR: Albert R. Meyer
TITLE: Associate Professor of Electrical Engineering

Acknowledgements

I'd like to thank Albert Meyer for his help and his interest, both personal and professional, during the last two and one half years. Most of my ideas for research were obtained through discussions with him; in particular, he is partly responsible for the main idea of the lower bound construction in Chapter 5.

I'm also grateful to Gregory Cherlin for his interest and suggestions regarding this thesis.

This research was supported by NSF grant GJ-34671.

Table of Contents

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 [Mey73,MS72,SM73,Sto74] and Fischer and Rabin [FiR74] who have shown that most 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 many cases even larger lower bounds have been established. In this thesis we investigate the computational complexity of a number of different logical theories, obtaining decision procedures whose computational complexities roughly meet the known lower bounds and deriving a new lower bound whose complexity is very close to the known upper bound.

Let N be the set of nonnegative integers. Whether a sentence of the first order theory of N under addition is true is decidable according to theorem of Presburger [Pre29]. A more efficient decision procedure given by Cooper [Coo72] has been proved by Oppen [Opp73] to require only $2^{2^{2^{cn}}}$ steps for sentences of length n, where c is some constant. In Chapter 2 we present a fairly general development of Ehrenfeucht games [Ehr61] which allows us to show that <u>space</u> $2^{2^{cn}}$ is sufficient for deciding Presburger arithmetic.

Let N^* be the set of functions from N to N of finite support, i.e.,

$N^* = \{f\colon N \to N \mid f(i) = 0 \text{ for all but finitely many } i \in N\}$.

The structure $< N^+, \cdot >$ of positive integers under multiplication is isomorphic to the structure $< N^*, + >$ (the weak direct power of $< N, + >$) where addition is defined component-wise. The first order theory of this structure is known to be decidable by a theorem of Mostowski [Mos52]. Mostowski's procedure, however, is not elementary recursive 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 Chapter 3 we use the technique of Ehrenfeucht games to derive some general results about the theories of weak direct powers which enable us to obtain a new procedure for deciding whether sentences are true over $< N^*, + >$. Our procedure can be implemented on a Turing machine which uses at most $2^{2^{2^{cn}}}$ tape squares (and hence $2^{2^{2^{c'n}}}$ steps) on sentences of length n. As a corollary we obtain the same upper bound on decision procedures for the first order theory of finite abelian groups. Recent results of Fischer and Rabin [FiR74] show that for some constant $c'' > 0$, any procedure for the first order theory of $< N^*, + >$ requires time $2^{2^{2^{c''n}}}$ even on nondeterministic Turing machines. Thus (see Sections 2 and 3) the worst case behavior of our procedure for $< N^*, + >$ is assymptotically nearly optimal in its computational requirements.

In Chapter 4 we extend the methods of Chapter 3 in order to 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 <u>if</u> (but not only if) the theory of the structure is elementary recursive and <u>if</u> 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.

Chapter 5 is concerned with the computational complexity of pairing function structures. A pairing function is a one-one map $\rho: N \times N \to N$, and the associated structure is $< N, \rho >$. Although the theory of the set of all pairing functions is undecidable and the theories of some individual pairing functions are undecidable, Tenney [Ten74] shows that many commonly used ones have decidable theories. Our main result is that no nonempty collection of pairing functions has an <u>elementary recursive</u> theory. In fact, for some constant $c > 0$, the theory of any nonempty collection of pairing functions requires time $\left.2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\right\}$ height cn to decide.

In Section 2 of this chapter we present the definitions and basic theorems of automata theory needed to clarify the basic notions of upper and lower time and space bounds used in the following chapters. In

Section 3 we discuss the reducibility techniques which allow us to achieve many of the upper and lower bounds. Section 4 consists of a description of the notation and fundamental concepts of mathematical logic which will be needed in the rest of the thesis.

Section 2: Automata Theory Background

We shall consider a version of Turing machines which may be either deterministic or nondeterministic, one tape, one head automata, with a finite tape alphabet Σ. For a rigorous definition of these machines the reader can consult [Sto74, Section 2.2]. For most of our purposes, however, the exact details of the definition chosen do not matter very much, so we provide only an informal description here.

The tape is one-way infinite to the right and the automaton starts in the initial state with its head on the leftmost square of the tape. At any step, depending on the current state and the current contents of the tape square scanned by the head, the automaton can write a new member of Σ on that square, move the head right or left, and go into a new state. The Turing machine is deterministic if its actions at any step are completely determined by its state and by the contents of the square pointed at by the head. If the machine is nondeterministic there may be a finite set of permissible actions possible at any moment. Thus, the deterministic Turing machines form a subset of the nondeterministic ones.

A (deterministic or nondeterministic) Σ-automaton $\mathfrak{M}$ has Σ as the tape alphabet; at any moment, all the symbols on the tape are from the alphabet Σ, $\not{b} \in \Sigma$. Let Σ^* be the set of all finite sequences, or "strings" of elements of Σ and let $\Sigma^+ = \Sigma^* - \{\lambda\}$ where λ is the empty string. If $\gamma \in \Sigma^+$, then $\mathfrak{M}$ accepts γ if there is some sequence of possible steps of $\mathfrak{M}$ with the tape squares initially containing the string $\gamma\not{b}\not{b}\ldots$ and the head scanning the leftmost symbol of γ, that ends with an accepting state. The set $L(M) = \{\gamma \in \Sigma^+ \mid \mathfrak{M} \text{ accepts } \gamma\}$ is called the language recognised by $\mathfrak{M}$.

We now define what we mean by the time and space used by Turing machines. If $\mathfrak{M}$ is a (nondeterministic) Σ-Turing machine which accepts $\gamma \in \Sigma^+$ by some computation containing at most n steps then we say that $\mathfrak{M}$ accepts γ within time n. If $\mathfrak{M}$ accepts γ by some computation during which the head visits at most n different tape squares then we say that $\mathfrak{M}$ accepts γ within space n. Let $L = L(\mathfrak{M})$ and let $f\colon N \to N$. Then we say $\mathfrak{M}$ recognises L within time (space) f(n) if for every $\gamma \in L$, $\mathfrak{M}$ accepts γ within time (space) $f(|\gamma|)$ where $|\gamma|$ is the length of the string γ. NTIME(f(n)) (NSPACE(f(n))) is the set of languages (where by language here we mean a subset of Σ^* for some alphabet Σ) each of which is recognised by some nondeterministic Turing machine within time (space) f(n). DTIME(f(n)) and DSPACE(f(n)) are defined similarly with respect to deterministic machines.

In order to compare the upper and lower bounds for the computational 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 [Sto74].)

Fact 2.1: Let $f\colon N \to N$.

A. Nondeterministic versus deterministic time

a) $\mathrm{DTIME}(f(n)) \subseteq \mathrm{NTIME}(f(n))$

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

B. Nondeterministic versus deterministic space

a) $\mathrm{DSPACE}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$

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

C. Time versus space

a) $\mathrm{DTIME}(f(n)) \subseteq \mathrm{DSPACE}(f(n))$

$\mathrm{NTIME}(f(n)) \subseteq \mathrm{NSPACE}(f(n))$

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

All of Fact 2.1 is relatively straightforward to prove, with the exception of B.b. B.b is proved by Savitch [Sav72]. 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 "$2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}$ 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.

Section 3: Using Reducibilities to Prove Upper and Lower Bounds

Definition 3.1: Let Σ_1 and Σ_2 be finite alphabets and let $L_1 \subseteq \Sigma_1^+$ and $L_2 \subseteq \Sigma_2^+$. Then $\underline{L_1 \leq_{p\ell} L_2}$ if for some function $g: \Sigma_1^+ \to \Sigma_2^+$:

I) for all $\gamma \in \Sigma_1^+$, $\gamma \in L_1 \Leftrightarrow g(\gamma) \in L_2$ and

II) there is some Turing machine which computes g within time a fixed polynomial in the length of the input and within space linear in the length of the input.†

If S is a collection of languages over Σ_1 ($S \subseteq p(\Sigma_1^*)$), then we say $S \leq_{p\ell} L_2$ if $L \leq_{p\ell} L_2$ for all $L \in S$.

We now state Lemma 3.2, 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 3.2: Say that $L_1 \leq_{p\ell} L_2$. Let $f: N \to N$. If

$$L_2 \in \left\{\begin{array}{l} \mathrm{DTIME}(f(n)) \\ \mathrm{DSPACE}(f(n)) \\ \mathrm{NTIME}(f(n)) \\ \mathrm{NSPACE}(f(n)) \end{array}\right., \text{ then } L_1 \in \left\{\begin{array}{l} \mathrm{DTIME}(f(cn) + p(n)) \\ \mathrm{DSPACE}(f(cn) + n) \\ \mathrm{NTIME}(f(cn) + p(n) \\ \mathrm{NSPACE}(f(cn) + n) \end{array}\right.$$

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

---

† A deterministic Turing machine computes g if when it is started with $\gamma b\!\!\!/ b\!\!\!/$ ... on its tape, $\gamma \in \Sigma_1^+$, and its head on the leftmost square, it eventually halts and $g(\gamma)$ is the string on the tape to left of the head.

Contrapositively, if

$$L_1 \notin \left\{\begin{array}{l} \mathrm{DTIME}(f(n)+p(n)) \\ \mathrm{DSPACE}(f(n)+n) \\ \mathrm{NTIME}(f(n)+p(n)) \\ \mathrm{NSPACE}(f(n)+n) \end{array}\right. \text{ then } L_2 \notin \left\{\begin{array}{l} \mathrm{DTIME}(f(cn)) \\ \mathrm{DSPACE}(f(cn)) \\ \mathrm{NTIME}(f(cn)) \\ \mathrm{NSPACE}(f(cn)) \end{array}\right.$$

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 $L_2 \in \mathrm{SPACE}(2^{2^{cn}})$ for some constant c. If $L_1 \le_{p\ell} L_2$ then we can conclude that $L_1 \in \mathrm{SPACE}(2^{2^{cn}})$ for some constant c. If we know that $L_1 \notin \mathrm{NTIME}(2^{2^{c'n}})$ for some constant $c' > 0$, and if $L_1 \le_{p\ell} L_2$, then we can conclude that $L_2 \notin \mathrm{NTIME}(2^{2^{c'n}})$ for some constant $c' > 0$.† This latter idea is often used in conjunction with Lemma 3.3.

Lemma 3.3: (see [Co73, SFM73, Sei74].) Let $f: N \to N$ be one of the functions 2^n, 2^{2^n}, $2^{2^{2^n}}$, or $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}$height n. Then there exists a language L such that $L \in \mathrm{NTIME}(f(n))$ and $L \notin \mathrm{NTIME}(f(n/2))$.

Theorem 3.4: Let $f: N \to N$ be one of the functions $2^n, 2^{2^n}, 2^{2^{2^n}}$ or $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}$height n and let $L_0 \subseteq \Sigma^*$ (for some Σ^*) be such that $\mathrm{NTIME}(f(n)) \le_{p\ell} L_0$. Then for some constant $c > 0$, $L_0 \notin \mathrm{NTIME}(f(cn))$.

---

† It is easy to see that if $L \notin \mathrm{NTIME}(f(n))$, 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.

Proof: Say that $\mathrm{NTIME}(f(n)) \leq_{p\ell} L_0$. By Lemma 3.3, let L be such that $L \notin \mathrm{NTIME}(f(n/2))$ and $L \in \mathrm{NTIME}(f(n))$. So $L \leq_{p\ell} L_0$. By Lemma 3.2, $L_0 \notin \mathrm{NTIME}(f(cn))$ for some constant $c > 0$. □

A typical way Theorem 3.4 is used is the following. Fischer and Rabin [FiR74] show that if TH is the theory of integer addition, then $\mathrm{NTIME}(2^{2^n}) \leq_{p\ell} \mathrm{TH}$, concluding that $\mathrm{TH} \notin \mathrm{NTIME}(2^{2^{cn}})$ for constant c. In Chapter 2 we show that $\mathrm{TH} \in \mathrm{SPACE}(2^{2^{c'n}})$ for some constant c', and hence that $\mathrm{TH} \in \mathrm{DTIME}(2^{2^{2^{c'n}}})$ 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 $\mathrm{TH} \in \mathrm{DTIME}(2^{2^{2^{c\sqrt{n}}}})$ for some constant c'. Since $\mathrm{NTIME}(2^{2^n}) \leq_{p\ell} \mathrm{TH}$, Lemma 3.2 would imply that $\mathrm{NTIME}(2^{2^n}) \subseteq \bigcup_{c \in \mathbb{N}} \mathrm{DTIME}(2^{2^{2^{c\sqrt{n}}}})$, narrowing the gap in Fact 2.1, 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> $\mathrm{DTIME}(c^{f(n)})$ 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.

Section 4: Mathematical Logic Background and Notation:

Most of the notation of mathematical logic that we shall use is fairly standard; the reader can find precise definitions of those concepts not defined here in [Men64].

$\mathcal{L}$ will always represent a language of the first order predicate calculus with a finite number of relational symbols $\underline{\mathfrak{R}}_1, \underline{\mathfrak{R}}_2, \ldots, \underline{\mathfrak{R}}_\ell$ where $\underline{\mathfrak{R}}_i$ will be a t_i-place formal predicate for $1 \le i \le \ell$. For technical convenience, $\mathcal{L}$ will not contain function symbols. Sometimes we will choose $\mathcal{L}$ to have a constant symbol $\underline{e}$ as well. The formal variables of $\mathcal{L}$ are written as $x_0, x_1, x_{10}, x_{11}, \ldots$, that is, the subscripts are written in binary. For expository convenience, we will refer to distinct formal variables as $x, x_0, x_1, x_2, \ldots, y, y_0, y_1, \ldots, z, z_0, z_1, \ldots,$ $w, w_0, w_1, \ldots,$ $x', y', z', \ldots$.

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

$(F_1 \vee F_2)$

$(F_1 \wedge F_2)$

$(F_1 \to F_2)$

$(F_1 \leftrightarrow F_2)$

$\sim F_1$

$\exists v F_1$

$\forall v F_1$

is a formula.† We use the usual notions of an occurrence of a variable in a formula being bound or free, and define a sentence of $\mathcal{L}$ to be a formula in which there are no free occurrences of variables.

A structure for $\mathcal{L}$ is a tuple $\mathcal{S} = \langle S, \mathcal{R}_1, \ldots, \mathcal{R}_\ell \rangle$ where S is a set and $\mathcal{R}_i \subseteq S^{t_i}$ for $1 \le i \le \ell$; if $\mathcal{L}$ has a constant symbol $\underline{e}$, then a structure for $\mathcal{L}$ is $\langle S, \mathcal{R}_1, \ldots, \mathcal{R}_\ell, e \rangle$ where $e \in S$. We call S the domain of $\mathcal{S}$. If F is a sentence of $\mathcal{L}$ we will use the usual notion of F true in $\mathcal{S}$ or $\mathcal{S}$ satisfies F or F holds in $\mathcal{S}$, and we will write this $\mathcal{S} \vdash F$. Sometimes we will say "F is true" or "F holds" or merely assert "F" when $\mathcal{S}$ is understood. $TH(\mathcal{S})$ = the theory of $\mathcal{S}$ = $\{F \mid F$ is a sentence and $\mathcal{S} \vdash F\}$. If $\mathcal{P}$ is a nonempty collection of structures, then define

† When writing formulas we will omit parentheses when it will not lead to confusion.

$TH(\mathcal{P})$ = theory of $\mathcal{P} = \bigcap_{\mathcal{S} \in \mathcal{P}} TH(\mathcal{S})$.

Our language $\mathcal{L}$ would have been just as powerful had we left out much of our logical notation. For instance $x \vee y$ is equivalent to $\sim x \to y$ and $\forall x F$ is equivalent to $\sim \exists x \sim F$. It is only for convenience that we have made $\mathcal{L}$ as large as we have.

We say a formula F is a <u>Boolean combination</u> of subformulas $F_1, F_2, \ldots, F_k$ if F is obtained by combining $F_1, F_2, \ldots, F_k$ using perhaps $\wedge$, $\vee$, $\to$, $\leftrightarrow$, $\sim$ but no quantifiers. Clearly every formula is equivalent to a Boolean combination of formulas, each of which begins with an existential quantifier.

We now define <u>annotated formulas</u> in order to be able to talk about substituting members of a domain for free occurrences of variables, and in order to be able to talk about the relations defined by formulas. Let F be a formula and say that we have a sequence of formal variables containing (not necessarily exclusively) the variables which occur freely in F, say $x_1, x_2, \ldots, x_k$. We define the <u>annotated formula</u> $F(x_1, x_2, \ldots, x_k)$ to be, formally, the ordered pair consisting of F and the sequence $x_1, x_2, \ldots, x_k$. Informally, when we write $F(x_1, x_2, \ldots, x_k)$ we think of ourselves as associating with the formula F the sequence $x_1, x_2, \ldots, x_k$. We will usually use F and $F(x_1, x_2, \ldots, x_k)$ interchangeably, and call them both formulas, as long as this association is understood; we will never associate two different sequences with the same formula.

Say that $F(x_1, x_2, \ldots, x_k)$ is an (annotated) formula and $\mathcal{S}$ is a structure with domain S, and $a_1 \in S$. By $F(a_1, x_2, \ldots, x_k)$ we will mean the formula obtained by substituting a_1 for free occurrences of x_1 in F. Note that this is technically not a formula of $\mathcal{L}$ but rather a (non-annotated) formula in the language $\mathcal{L}'$ obtained by adding constant symbols to $\mathcal{L}$ for every member of S. If $a_1, a_2, \ldots, a_k \in S$, then $F(a_1, a_2, \ldots, a_k)$ is defined similarly, and we write $\mathcal{S} \vdash F(a_1, a_2, \ldots, a_k)$ if $F(a_1, a_2, \ldots, a_k)$ is true in $\mathcal{S}$.

For $k > 0$, we use $\bar{x}_k$ to represent the k-tuple $(x_1, x_2, \ldots, x_k)$, $\bar{a}_k$ to represent $(a_1, a_2, \ldots, a_k)$, $(\bar{a}_k, b)$ to represent $(a_1, a_2, \ldots, a_k, b)$, etc. Thus $F(\bar{x}_k)$ will be used instead of $F(x_1, x_2, \ldots, x_k)$, etc. e^k and $\underline{e}^k$ will stand for the k-tuples $(e, e, \ldots, e)$ and $(\underline{e}, \underline{e}, \ldots, \underline{e})$. S^k is the set of k-tuples of members of S. (S^k is isomorphic to the set of functions from $\{0, 1, 2, \ldots, k-1\}$ to S.) For $k = 0$, S^k is taken to be the singleton set containing the empty set, and $\bar{a}_k$, e^k, etc., denote the empty set. However, we take $(\bar{a}_k, b, c)$ to mean (b, c) when $k = 0$, etc.

If we write $F(\bar{x}_k)$ when $k = 0$, then F is a sentence; $F(\bar{x}_k)$, $F(\bar{a}_k)$, etc., are in this case no different than F itself.

If $\mathcal{S}$ is a structure with domain S and $A \subseteq S^k$ and $F(\bar{x}_k)$ is an annotated formula, then we say F defines A in $\mathcal{S}$ if $A = \{\bar{a}_k \in S^k \mid \mathcal{S} \vdash F(\bar{a}_k)\}$. We say "F defines A" if $\mathcal{S}$ is understood. More generally, say that we are interested in a particular nonempty class of structures $\mathcal{P}$. By a k-place property G we mean a function which assigns to each structure $\mathcal{S} \in \mathcal{P}$ a subset of S^k (where S is the domain of $\mathcal{S}$); we will usually refer to the value of G on $\mathcal{S}$ as the relation G restricted to $\mathcal{S}$. If $\bar{a}_k \in S^k$, then we write $\mathcal{S} \vdash G(\bar{a}_k)$ to mean that $\bar{a}_k \in$ the relation obtained by restricting G to $\mathcal{S}$. When G is a property we sometimes write $G(\bar{x}_k)$ to indicate that G is a k-place property. If $\underline{G}(\bar{x}_k)$ is a formula, we say that $\underline{G}$ defines G in $\mathcal{P}$ if in every $\mathcal{S} \in \mathcal{P}$, $\underline{G}$ defines G restricted to $\mathcal{S}$. We say "$\underline{G}$ defines G" when $\mathcal{P}$ is understood.

Formulas F_1 and F_2 are equivalent in $\mathcal{S}$ if for some sequence $x_1, x_2, \ldots, x_k$ of variables, the free variables of both F_1 and F_2 are from among $x_1, x_2, \ldots, x_k$, and the annotated formulas $F_1(\bar{x}_k)$ and $F_2(\bar{x}_k)$ define the same subset of S^k. F_1 and F_2 are equivalent in $\mathcal{P}$ if they are equivalent in every member of $\mathcal{P}$. We say "F_1 and F_2 are equivalent" to mean with respect to the class of all structures, unless $\mathcal{S}$ or $\mathcal{P}$ is understood.

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

Another usage of the notation $F(x_1, x_2, \ldots, x_k)$ serves to emphasize that the free variables of F are from among $x_1, x_2, \ldots, x_k$. For instance, the more mnemonic notation $\exists x_k F(\bar{x}_k)$ will sometimes be used instead of $\exists x_k F$. If we write $|F(\bar{x}_k)|$ we simply mean $|F|$.

Notation: If α is a string, then $|\alpha|$ is the length of α. If α is a set, then $|\alpha|$ is the cardinality of α. If α is an integer, then $|\alpha|$ is the absolute value of α. N^+ is the set of positive integers. For $i \in N^+$, Q_i will always represent a quantifier, i.e., either $\forall$ or $\exists$. All logarithms are to the base 2.

Definition 4.1: 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 $v_1, v_2, \ldots, v_k$

represent formal variables.

<u>Theorem 4.2</u>: Every formula F is equivalent to a formula G in prenex normal form such that G has at most $|F|$ quantifiers and is of length at most $|F|\cdot\log|F|$. Furthermore, there is a procedure (i.e., Turing machine) which given F computes G within time polynomial in $|F|$.

<u>Proof</u>: 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 $|F|$ quantifiers. F has at most $|F|$ 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 $\log |F|$ when put in prenex normal form. This procedure can be checked to operate within polynomial time. $\square$

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 4.3: If F is a formula, we will 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

$\text{q-depth}(F_1 \vee F_2) = \text{q-depth}(F_1 \wedge F_2) = \text{q-depth}(F_1 \rightarrow F_2) = \text{q-depth}(F_1 \leftrightarrow F_2) =$

$\text{Max}\{\text{q-depth}(F_1), \text{q-depth}(F_2)\}$, $\text{q-depth}(\sim F_1) = \text{q-depth}(F_1)$, and

$\text{q-depth}(\exists v F_1) = \text{q-depth}(\forall v F_1) = 1 + \text{q-depth}(F_1)$.

Chapter 2: Ehrenfeucht Games and Decision Procedures

Section 1: Introduction

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. A discussion of game theory also appears in work by Richard Tenney [Ten74,Ten74']. Tenney uses Ehrenfeucht game techniques to decide the theories of certain pairing functions and to decide the second order theory of an equivalence relation. Neither Ehrenfeucht nor Tenney explicitly describes these techniques in generality. We shall present a development in this chapter which, although not completely general, is general enough to handle a wide variety of cases. Where possible we will describe our decision procedures in terms of bounds on quantifiers, so that to decide the truth of a sentence one need only decide the sentence when each quantifier is limited to range over a particular finite set. This idea, which will be carefully described in the next three chapters, is also used by Tenny, Ferrante and Rackoff [FR74], and Ferrante [Fer74]. In addition, as part of our development of the Ehrenfeucht game approach we shall characterize it in terms of the quantifier depth of formulas.

Section 2 of this chapter consists of a general development of Ehrenfeucht games. Our approach is somewhat different from that of Ehrenfeucht or Tenney, but several of the basic theorems and ideas come from these sources. In Section 3 we derive a decision procedure for the first order theory of integer addition as a corollary of our general development. In Section 4 we discuss an important open question relating the complexity of decision procedures to the index of the equivalence relation which characterizes Ehrenfeucht games.

Section 2: The Ehrenfeucht Equivalence Relation and Ehrenfeucht Games

Let $\mathcal{L}$ be a fixed language of the first order predicate calculus with finitely many relational symbols $\underline{\mathcal{R}}_1, \underline{\mathcal{R}}_2, \ldots, \underline{\mathcal{R}}_\ell$ where $\underline{\mathcal{R}}_i$ is a t_i-place formal predicate for $1 \le i \le \ell$. Also, let $\mathcal{L}$ have a single constant symbol $\underline{e}$. Let $\mathcal{S} = \langle S, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_\ell, e\rangle$ be a fixed structure for $\mathcal{L}$. (Actually, the constant symbol $\underline{e}$ plays no important role in this chapter but is included so that we can talk about weak direct powers later.) In addition we will assume we have a norm on $\mathcal{S}$, by which we mean a function $|| \;\; ||: S \to N$, and we will denote the norm of $a \in S$ by $||a||$. If $i \in N$, then we write $a \prec i$ to mean $||a|| \le i$. We introduce this concept of norm in order to describe simple decision procedures which use space efficiently (and without a significant time loss). However the reader should note that many of the theorems below make no mention of the norm and are independent of this notion.

We now define the Ehrenfeucht equivalence relation.

Definition 2.1: For all $n, k \in N$ and all $\bar{a}_k, \bar{b}_k \in S^k$, define $\bar{a}_k \underset{n}{\equiv} \bar{b}_k$ iff for every formula $F(\bar{x}_k)$ of q-depth$\le n$, $F(\bar{a}_k)$ and $F(\bar{b}_k)$ are either both true or both false (in $\mathcal{S}$).

Remark 2.2: For each $n, k \in N$, $\underset{n}{\equiv}$ is an equivalence relation on S^k. Ehrenfeucht originally defined $\underset{n}{\equiv}$ by induction on n; his definition consisted of a combination of our definition of $\underset{0}{\equiv}$ together with what we call Theorem 2.3. We will prove this theorem later.

Theorem 2.3: Let $n, k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$. Then $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k \Leftrightarrow$

1) For each $a_{k+1} \in S$ there exists some $b_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

and 2) For each $b_{k+1} \in S$ there exists some $a_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

Lemma 2.4: Let $n,k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$ such that

1) For each $a_{k+1} \in S$ there exists some $b_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

and 2) For each $b_{k+1} \in S$ there exists some $a_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

Then $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$.

Proof: Say that 1) and 2) hold. Since every formula is equivalent to a Boolean combination of formulas each of which begins with an existential quantifier, it is sufficient to prove, for $F(\bar{x}_k)$ of the form $\exists x_{k+1}\, G(\bar{x}_{k+1})$ where q-depth$(G) \leq n$, that $F(\bar{a}_k) \Leftrightarrow F(\bar{b}_k)$.

So assume that $F(\bar{a}_k)$ holds. Then let $a_{k+1} \in S$ be such that $G(\bar{a}_{k+1})$ holds. By 1), let $b_{k+1} \in S$ be such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$. Since $G(\bar{a}_{k+1})$ is true, $G(\bar{b}_{k+1})$ is true (by definition of $\underset{n}{\equiv}$), so $F(\bar{b}_k)$ is true. By symmetry, $F(\bar{a}_k)$ holds if $F(\bar{b}_k)$ holds. $\square$

Definition 2.5: For each $n,k \in N$, let $M(n,k)$ be the number of equivalence classes of $\underset{n}{\equiv}$ restricted to S^k.

Lemma 2.6: Let $n,k \in N$. Then $M(n,k)$ is finite and for each $\bar{a}_k \in S^k$ there is a formula $F(\bar{x}_k)$ of q-depth n such that for all $\bar{b}_k \in S^k$,

$\mathcal{S} \vdash F(\bar{b}_k) \Leftrightarrow \bar{b}_k \underset{n}{\equiv} \bar{a}_k$ (i.e., F defines the $\underset{n}{\equiv}$ equivalence class of $\bar{a}_k$).

Proof (by induction on n): If $n=0$ and $\bar{a}_k \in S^k$, we can clearly take $F(\bar{x}_k)$ to be a conjunction of atomic formulas and negations of atomic formulas. Since an argument place of an atomic formula can be occupied by either a formal variable or by e, the number of atomic formulas in which at most 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 $\bar{a}_{k+1} \in S^{k+1}$ there exists an i, $1 \le i \le M(n,k+1)$, such that F_i defines the $\underset{n}{\equiv}$ equivalence class of $\bar{a}_{k+1}$.

For each $\bar{c}_k \in S^k$ define

$W(\bar{c}_k) = \{i \mid 1 \le i \le M(n,k+1) \text{ and } \exists x_{k+1} F_i(\bar{c}_k, x_{k+1}) \text{ is true}\}$. We shall show that for all $\bar{b}_k, \bar{c}_k \in S^k$, $\bar{b}_k \underset{n+1}{\equiv} \bar{c}_k \Leftrightarrow W(\bar{b}_k) = W(\bar{c}_k)$. Thus the formula $F(\bar{x}_k) =$

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

defines the $\underset{n+1}{\equiv}$ equivalence class of $\bar{c}_k$.

Clearly if $\bar{b}_k \underset{n+1}{\equiv} \bar{c}_k$, then $W(\bar{b}_k) = W(\bar{c}_k)$ 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.

<u>Claim</u>: If $W(\bar{b}_k) = W(\bar{c}_k)$, then for each $c_{k+1} \in S$ there exists some $b_{k+1} \in S$ such that $\bar{c}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$ (and by symmetry, for each $b_{k+1} \in S$ there exists some $c_{k+1} \in S$ such that $\bar{c}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$).

<u>Proof of Claim</u>: Say that $W(\bar{b}_k) = W(\bar{c}_k)$ and $c_{k+1} \in S$. Let i, $1 \le i \le M(n,k+1)$, be such that $F_i(\bar{x}_{k+1})$ defines the $\underset{n}{\equiv}$ equivalence class of $\bar{c}_{k+1}$. $F_i(\bar{c}_{k+1})$ is true, so $\exists x_{k+1} F_i(\bar{c}_k, x_{k+1})$ is true, so $i \in W(\bar{c}_k)$. So $i \in W(\bar{b}_k)$. This means that $\exists x_{k+1} F_i(\bar{b}_k, x_{k+1})$ is true, and therefore we can find b_{k+1} such that $F_i(\bar{b}_{k+1})$ is true. Since F_i defines the $\underset{n}{\equiv}$ equivalence class of $\bar{c}_{k+1}$, we must have $\bar{c}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

By the Claim and Lemma 2.4, $W(\bar{b}_k) = W(\bar{c}_k) \Leftrightarrow \bar{b}_k \underset{n+1}{\equiv} \bar{c}_k$. Note that the $\underset{n+1}{\equiv}$ equivalence class of $\bar{c}_k$ is determined by $W(\bar{c}_k)$ which is a subset of $\{1,2,\ldots,M(n,k+1)\}$. So $M(n+1,k) \le 2^{M(n,k+1)}$. This and the bound on $M(0,k)$

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

<u>Remark 2.7</u>: There are structures $\mathcal{S}$ such that

$M(n,k) \ge 2^{2^{\cdot^{\cdot^{\cdot^{2^{n+k}}}}}} \Big\}$ height ϵn (for some constant $\epsilon > 0$), so $M(n,k)$ is not in general bounded above by an elementary recursive function. For many structures, however, $M(n,k)$ grows considerably more slowly.

<u>Definition 2.8</u>: Let $H:N^3 \to N$ be a function which is nondecreasing in each argument. Then $\mathcal{S}$ is <u>H-bounded</u> iff for all $n,k \in N$ and all $F(\bar{x}_{k+1})$ of q-depth $\le n$ and all $\bar{a}_k \in S^k$, if $\exists x_{k+1} F(\bar{a}_k, x_{k+1})$ is true in $\mathcal{S}$ then $[\exists x_{k+1} \prec H(n,k,\underset{1\le i\le k}{\mathrm{Max}}\{||a_i||\})]F(\bar{a}_k, x_{k+1})$ is true in $\mathcal{S}$. (We take Max $\emptyset$ to be 0.)

<u>Remark 2.9</u>: If our norm on S is the identically 0 function and $H:N^3 \to N$ is the identically 0 function then clearly $\mathcal{S}$ is H-bounded. This means that often when we have a theorem which involves the concepts of norm and H-boundedness, we can immediately obtain a simpler theorem which doesn't mention those concepts; sometimes, as is the case with Lemma 2.10, this new result is still interesting.

<u>Lemma 2.10</u>: Let $H:N^3 \to N$ be such that $\mathcal{S}$ is H-bounded. Let $n,k \in N$ and let $\bar{a}_k, \bar{b}_k \in S^k$ such that $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$. Then for each $a_{k+1} \in S$ there exists some $b_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$ and such that $||b_{k+1}|| \le H(n,k,\underset{1\le i\le k}{\mathrm{Max}}\{||b_i||\})$.

Proof: Let $\bar{a}_k, \bar{b}_k \in S^k$ such that $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$. Let $a_{k+1} \in S$. By Lemma 2.6 there is a formula $F(\bar{x}_{k+1})$ of q-depth n which defines the $\underset{n}{\equiv}$ equivalence class of $\bar{a}_{k+1}$. Since $\exists x_{k+1} F(\bar{a}_k, x_{k+1})$ is true and $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$, $\exists x_{k+1} F(\bar{b}_k, x_{k+1})$ is true. Since $\mathcal{S}$ is H-bounded, we can choose $b_{k+1} \in S$ such that $F(\bar{b}_{k+1})$ is true and $||b_{k+1}|| \le H(n,k,\underset{1\le i\le k}{\operatorname{Max}}\{||b_i||\})$. But $F(\bar{b}_{k+1})$ implies $\bar{b}_{k+1} \underset{n}{\equiv} \bar{a}_{k+1}$. □

Proof of Theorem 2.3: Theorem 2.3 follows immediately from Lemma 2.10 (keeping in mind Remark 2.9) and Lemma 2.4. □

H-boundedness of a structure guarantees that quantifiers in a formula ranging over all of S can be replaced by quantifiers ranging over elements of S whose norms are bounded by a function determined by H. This is made precise in the following lemma.

Lemma 2.11: Let $H: N^3 \to N$ be such that $\mathcal{S}$ is H-bounded. Let $n,k \in N$ and let $Q_1x_1\, Q_2x_2 \ldots Q_kx_k\, F(\bar{x}_k)$ be a sentence of $\mathcal{L}$ with q-depth $\le n+k$, i.e., q-depth$(F) \le n$. Let $\bar{m}_k \in N^k$ be a sequence such that $m_i \ge H(n+k-i, i-1, \underset{1\le j<i}{\operatorname{Max}}\{m_j\})$ for $1 \le i \le k$.

Then $Q_1x_1\, Q_2x_2 \ldots Q_kx_k\, F(\bar{x}_k)$ is true $\Leftrightarrow$ $(Q_1x_1 \preccurlyeq m_1)(Q_2x_2 \preccurlyeq m_2)\ldots(Q_kx_k \preccurlyeq m_k)F(\bar{x}_k)$ is true.

Proof: Consider the formula $Q_2x_2\, Q_3x_3 \ldots Q_kx_k\, F(\bar{x}_k)$. Because $\mathcal{S}$ is H-bounded, if $m_1 \ge H(n+k-1,0,0)$ then $Q_1x_1(Q_2x_2 \ldots Q_kx_k\, F(\bar{x}_k))$ is equivalent to $(Q_1x_1 \preccurlyeq m_1)(Q_2x_2 \ldots Q_kx_k\, F(\bar{x}_k))$.

Now for each $a \in S$ such that $||a|| \le m_1$, consider the formula $Q_3x_3\, Q_4x_4 \ldots Q_kx_k\, F(a, x_2, x_3, \ldots, x_k)$. Because $\mathcal{S}$ is H-bounded, if $m_2 \ge H(n+k-2, 1, m_1)$ then $Q_2x_2(Q_3x_3 \ldots Q_kx_k\, F(a, x_2, x_3, \ldots, x_k))$ is equivalent to $(Q_2x_2 < m_2)(Q_3x_3 \ldots Q_kx_k\, F(a,x_2,x_3,\ldots,x_k))$. Hence, $(Q_1x_1 < m_1)Q_2x_2 \ldots Q_kx_k\, F(\bar{x}_k)$ is equivalent to $(Q_1x_1 < m_1)(Q_2x_2 < m_2)Q_3x_3 \ldots Q_kx_k\, F(\bar{x}_k)$.

By k-2 additional applications of the H-boundedness of $\mathcal{S}$, we arrive at Lemma 2.11. □

We now demonstrate the existence of a general method of proving H-boundedness.

Lemma 2.12: Let $H:N^3 \to N$ be a function which is nondecreasing in each argument, and say that for each $n,k \in N$ we have an equivalence relation E_n on S^k satisfying the following properties:

1) For all $k \in N$ and all $\bar{a}_k, \bar{b}_k \in S^k$, $\bar{a}_k\ E_0\ \bar{b}_k \Rightarrow \bar{a}_k \underset{0}{\equiv} \bar{b}_k$.

and 2) If $n,k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$ such that $\bar{a}_k\ E_{n+1}\ \bar{b}_k$, then for each $a_{k+1} \in S$ there is some $b_{k+1} \in S$ such that $\bar{a}_{k+1}\ E_n\ \bar{b}_{k+1}$ and such that $||b_{k+1}|| \le H(n,k,\max_{1\le i\le k}\{||b_i||\})$.

THEN

I) For all $n,k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$, $\bar{a}_k\ E_n\ \bar{b}_k \Rightarrow \bar{a}_k \underset{n}{\equiv} \bar{b}_k$.

and II) $\mathcal{S}$ is H-bounded.

Proof:

Proof of I) by induction on n: I) certainly holds if n=0. Assume I) is true for n; we will prove it for n+1.

Say that $\bar{a}_k\ E_{n+1}\ \bar{b}_k$; we wish to show that $\bar{a}_k \underset{n+1}{\equiv} \bar{b}_k$. By Lemma 2.4 and the symmetry of E_{n+1}, it is sufficient to show that for every $a_{k+1} \in S$ there is some $b_{k+1} \in S$ such that $\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$. So choose $a_{k+1} \in S$. By 2) there is some $b_{k+1} \in S$ such that $\bar{a}_{k+1}\ E_n\ \bar{b}_{k+1}$. By the induction hypothesis,

$\bar{a}_{k+1} \underset{n}{\equiv} \bar{b}_{k+1}$.

<u>Proof of II)</u>: Let $F(\bar{x}_{k+1})$ be a formula of q-depth $\leq n$ and let $\bar{a}_k \in S^k$ be such that $\exists x_{k+1} F(\bar{a}_k, x_{k+1})$ is true. Let $a_{k+1} \in S$ be such that $F(\bar{a}_{k+1})$ holds. Since $\bar{a}_k\ E_{n+1}\ \bar{a}_k$, condition 2) implies that we can find some $a'_{k+1} \in S$ such that $\bar{a}_{k+1}\ E_n\ (\bar{a}_k, a'_{k+1})$ and such that $||a'_{k+1}|| \leq H(n,k, \underset{1\leq i\leq k}{\mathrm{Max}}\{||a_i||\})$. But by I), $\bar{a}_{k+1}\ E_n\ (\bar{a}_k, a'_{k+1}) \Rightarrow \bar{a}_{k+1} \underset{n}{\equiv} (\bar{a}_k, a'_{k+1}) \Rightarrow F(\bar{a}_k, a'_{k+1})$ holds. So $\mathcal{S}$ is H-bounded. □

By applying Remark 2.9 to Lemma 2.12 we immediately obtain Lemma 2.12'.

<u>Lemma 2.12'</u>: Say that for each $n,k \in N$ we have an equivalence relation E_n on S^k satisfying the following properties:

1) For all $k \in N$ and all $\bar{a}_k, \bar{b}_k \in S^k$, $\bar{a}_k\ E_0\ \bar{b}_k \Rightarrow \bar{a}_k \underset{0}{\equiv} \bar{b}_k$.

and 2) If $n,k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$ such that $\bar{a}_k\ E_{n+1}\ \bar{b}_k$, then for each $a_{k+1} \in S$ there is some $b_{k+1} \in S$ such that $\bar{a}_{k+1}\ E_n\ \bar{b}_{k+1}$.

<u>THEN</u> for all $n,k \in N$ and $\bar{a}_k, \bar{b}_k \in S^k$, $\bar{a}_k\ E_n\ \bar{b}_k \Rightarrow \bar{a}_k \underset{n}{\equiv} \bar{b}_k$.

We loosely define an "Ehrenfeucht game (abbreviated E-game) decision procedure" for TH($\mathcal{S}$) to be one that involves defining relations E_n and proving that the conditions of Lemma 2.12 or 2.12' hold. This will be made clearer in the examples of Section 3 and Chapter 3. In Section 4 of this chapter we present a general discussion of the computational complexity of E-game decision procedures.

Lemma 2.13 shows how H-boundedness implies bounds on the norms of members of the $\underset{n}{\equiv}$ equivalence classes.

<u>Lemma 2.13</u>: Let $H:N^3 \to N$ be such that $\mathcal{S}$ is H-bounded. Let $n,k \in N$ and let $\bar{m}_k \in N^k$ be a sequence such that $m_i \geq H(n+k-i, i-1, \underset{1\leq j<i}{\mathrm{Max}}\{m_j\})$ for

$1 \le i \le k$. Then for each $\bar{a}_k \in S^k$ there is some $\bar{b}_k \in S^k$ such that $\bar{a}_k \underset{n}{\equiv} \bar{b}_k$ and $||b_i|| \le m_i$ for $1 \le i \le k$.

Proof: Let $n, k, \bar{m}_k$, and $\bar{a}_k$ be as in the statement of the lemma. By Lemma 2.6 there is a formula $F(\bar{x}_k)$ of q-depth n which defines the $\underset{n}{\equiv}$ equivalence class of $\bar{a}_k$. Since $F(\bar{a}_k)$ holds, $\exists x_1 \exists x_2 \ldots \exists x_k F(\bar{x}_k)$ is true. So by Lemma 2.11, $(\exists x_1 \leqslant m_1)(\exists x_2 \leqslant m_2)\ldots(\exists x_k \leqslant m_k)F(\bar{x}_k)$ is true. This means that for some $\bar{b}_k \in S^k$, $F(\bar{b}_k)$ is true and $||b_i|| \le m_i$ for $1 \le i \le k$. □

Section 3: An E-Game Decision Procedure for Integer Addition

We now present some applications of Section 2. For the rest of this section let $\mathcal{L}_1$ be the language of the first order predicate calculus with the formal predicates $v_1 + v_2 = v_3$ and $v_1 \le v_2$ and the constant symbol 0 (where v_1, v_2, v_3 represent formal variables).

Definition 3.0: Let $\mathbb{Z}$ be the structure $< Z, +, \le, 0 >$ where Z is the set of integers and where + and $\le$ are the usual integer addition and order. If $a \in Z$, define $||a|| = |a|$ = absolute value of a.

We will obtain a theoretically efficient decision procedure for TH($\mathbb{Z}$) using results of the previous section. 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 analyzed from a complexity viewpoint by Oppen [Opp73]. We choose this example because it illustrates our thesis that all known quantifier elimination procedures can be converted to E-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 Ferrante and Rackoff [FR74].

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

Definition 3.1: If a, b, c $\in$ Z, then $a \approx b \bmod c$ (a is equivalent to b mod c) if c divides a - b. If A is a nonempty finite set of integers, then lcm A = the least positive integer which every non-zero element of

A divides.[†]

Definition 3.2: Let a, $b \in Z$ and let $d \in N^+$. Then we write $a \underset{d}{=} b$

if either 1) $a = b$

2) $a \geq d$ and $b \geq d$

or 3) $a \leq -d$ and $b \leq -d$.

When we talk about $\underset{d}{=}$ holding between objects one of which is the cardinality of a set, we will often omit the vertical lines indicating cardinality. For instance, if A and B are sets, we will write $A \underset{d}{=} B$ and $A \underset{d}{=} 5$ instead of $|A| \underset{d}{=} |B|$ and $|A| \underset{d}{=} 5$.

Lemma 3.3: Let a, $b \in Z$ and let $d \in N^+$. Then $a \underset{d}{=} b \Leftrightarrow$

for every c, $-d < c \leq d$, $a \geq c \Leftrightarrow b \geq c$.

Proof: Left to the reader. □

Definition 3.4: Define a sequence of sets of integers V_0, V_0', V_1, V_1',... as follows: $V_0 = \{-2, -1, 0, 1, 2\}$. If V_i has been defined, define $V_i' = \{ \frac{\delta}{v} \cdot v' \mid \delta = \text{lcm } V_i;\ v, v' \in V_i;\ v \neq 0\}$ and define $V_{i+1} = V_i \cup \{a + b \mid a, b \in V_i'\}$.

Definition 3.5: Let n, $k \in N$. Then define the equivalence relation E_n on Z^k as follows: Let $\bar{a}_k$, $\bar{b}_k \in Z^k$, let $\delta = \text{lcm } V_n$.

[†] We use this nonstandard notation for equivalence mod c so as not to cause conflict with other notation we use.

Then $\overline{a}_k \; E_n \; \overline{b}_k$ iff for every $\overline{v}_k \in (V_n)^k$:

$$1)\ \sum_{i=1}^{k} v_i a_i \approx \sum_{i=1}^{k} v_i b_i \bmod \delta^2$$

and $$2)\ \sum_{i=1}^{k} v_i a_i \underset{\delta^2}{=} \sum_{i=1}^{k} v_i b_i$$

<u>Lemma 3.6</u>: Let $k \in N$ and let $\overline{a}_k, \overline{b}_k \in Z^k$ such that $\overline{a}_k \; E_0 \; \overline{b}_k$. Then $\overline{a}_k \underset{0}{\equiv} \overline{b}_k$.

<u>Proof</u>: Say that $\overline{a}_k \; E_0 \; \overline{b}_k$. We wish to show that for any quantifier free formula $F(\overline{x}_k)$, $Z \vdash F(\overline{a}_k) \Leftrightarrow Z \vdash F(\overline{b}_k)$. Since every quantifier free formula is a boolean combination of atomic formulas, it is sufficient to assume F is atomic. We need only consider the following cases for F:

$x_1 \le x_1$, $x_1 \le x_2$, $x_1 + x_1 = x_1$, $x_1 + x_2 = x_1$, $x_1 + x_1 = x_2$, $x_1 + x_2 = x_3$.

In these cases, in order to show that $Z \vdash F(\overline{a}_k) \Leftrightarrow Z \vdash F(\overline{b}_k)$, it is necessary to show (respectively) that $0 \le 0 \Leftrightarrow 0 \le 0$,

$a_1 - a_2 \le 0 \Leftrightarrow b_1 - b_2 \le 0$, $a_1 = 0 \Leftrightarrow b_1 = 0$, $a_2 = 0 \Leftrightarrow b_2 = 0$,

$2a_1 - a_2 = 0 \Leftrightarrow 2b_1 - b_2 = 0$, $a_1 + a_2 - a_3 = 0 \Leftrightarrow b_1 + b_2 - b_3 = 0$.

But since $0, 1, -1, 2, \in V_0$, all these facts follow from 2) in the definition of E_0. □

<u>Lemma 3.7</u>: For some constant c, $|V_n| \le 2^{2^{cn}}$ and $V_n = \{-a \mid a \in V_n\}$ and $\operatorname{Max} V_n \le 2^{2^{2^{cn}}}$ for all $n \in N$.

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, lcm $V_i \le (\text{Max } V_i)^{|V_i|}$. So

Max $V_{i+1} \le \text{Max}(\text{Max } V_i, 2\cdot\text{Max } V_i') \le 2\cdot\text{lcm } V_i\cdot\text{Max } V_i$
$\le 2\cdot(\text{Max } V_i)^{5^{5^i}}\cdot \text{Max } V_i \le (\text{Max } V_i)^{6^{6^i}}$. So Max $V_n \le 2^{\left(6^{6^n}\right)^n}$.
$|V_n| \le 2^{2^{cn}}$ and Max $V_n \le 2^{2^{2^{cn}}}$ for some constant c and all $n \in N$. □

Theorem 3.8: There exists a constant d such that the following is true: Let $n, k \in N$ and let $\bar{a}_k, \bar{b}_k \in Z^k$ such that $\bar{a}_k \; E_{n+1} \; \bar{b}_k$. Then for each $a_{k+1} \in Z$ there exists some $b_{k+1} \in Z$ such that $\bar{a}_{k+1} \; E_n \; \bar{b}_{k+1}$ and such that

$$|b_{k+1}| \le (1 + \max_{1\le i\le k} \{b_i\})\cdot 2^{2^{2^{d(n+k)}}} .$$

Proof: Say that $\bar{a}_k \; E_{n+1} \bar{b}_k$ and that $a_{k+1} \in Z$. Let δ = lcm V_n and note that δ^2 = lcm V_n' since $1 \in V_n$. Let $T = \{\sum_{i=1}^{k} v_i a_i + v \mid v_i \in V_n'$ for $1 \le i \le k$ and $|v| \le \delta^3\}$ be a nonempty subset of Z. There must exist either a member of T which is $\le \delta a_{k+1}$ or a member of T $\ge \delta a_{k+1}$ (or both); these two cases are symmetrical, so assume without loss of generality that some member of T is $\le \delta a_{k+1}$. Let $\sum_{i=1}^{k} v_i a_i + v$ be the largest member of T which is $\le \delta a_{k+1}$ where $v_i \in V_n'$ for $1 \le i \le k$ and $|v| \le \delta^3$. Consider the sequence

$$\sum_{i=1}^{k} v_i a_i + v,\ \sum_{i=1}^{k} v_i a_i + v + 1,\ \sum_{i=1}^{k} v_i a_i + v + 2, \ldots, \sum_{i=1}^{k} v_i a_i + v + \delta^3.$$

If δa_{k+1} is not equal to any of them, then δa_{k+1} is bigger than all of them and one of them (other than $\sum_{i=1}^{k} v_i a_i + v$) is equivalent to $\delta a_{k+1} \bmod \delta^3$. It is therefore the case that for some u: $|u| \leq \delta^3$, and $\sum_{i=1}^{k} v_i a_i + v + u \approx \delta a_{k+1} \bmod \delta^3$, and $\sum_{i=1}^{k} v_i a_i + v \leq \sum_{i=1}^{k} v_i a_i + v + u \leq \delta a_{k+1}$, and $u = 0 \Leftrightarrow \sum_{i=1}^{k} v_i a_i + v = \delta a_{k+1}$.

<u>Claim</u>: For every $t \in T$, $t \leq \sum_{i=1}^{k} v_i a_i + v + u \Leftrightarrow t \leq \delta a_{k+1}$ and $t \geq \sum_{i=1}^{k} v_i a_i + v + u \Leftrightarrow t \geq \delta a_{k+1}$.

<u>Proof of Claim</u>: If $\sum_{i=1}^{k} v_i a_i + v + u = \delta a_{k+1}$, then the claim is trivial. So assume $\sum_{i=1}^{k} v_i a_i + v + u \neq \delta a_{k+1}$. Then $u \neq 0$, and so $\sum_{i=1}^{k} v_i a_i + v + u$ is strictly between $\sum_{i=1}^{k} v_i a_i + v$ and δa_{k+1}. Since $\sum_{i=1}^{k} v_i a_i + v$ is the largest member of $T \leq \delta a_{k+1}$, we cannot have any $t \in T$ such that $\sum_{i=1}^{k} v_i a_i + v + u \leq t \leq \delta a_{k+1}$; hence the Claim follows.

Now let $\gamma = \operatorname{lcm} V_{n+1}$. Since $0 \in V_n$, $0 \in V_n'$. Therefore $V_n' \subseteq V_{n+1}$. Since $\delta^2 = \operatorname{lcm} V_n'$ and $V_{n+1} \supseteq \{2e \mid e \in V_n'\}$, we have $2\delta^2$ divides γ. Since $\bar{a}_k E_{n+1} \bar{b}_k$, $\sum_{i=1}^{k} v_i a_i \approx \sum_{i=1}^{k} v_i b_i \bmod \gamma^2$, and so $\delta a_{k+1} \approx$

$$\sum_{i=1}^{k} v_i a_i + v + u \approx \sum_{i=1}^{k} v_i b_i + v + u \bmod \delta^3$$

implying that δ divides $\sum_{i=1}^{k} v_i b_i + v + u$. <u>Define</u>

<u>$b_{k+1} = (\sum_{i=1}^{k} v_i b_i + v + u)/\delta$.</u> We will show that $\bar{a}_{k+1} E_n \bar{b}_{k+1}$.

Let $\bar{w}_{k+1} \in (V_n)^{k+1}$. We want to show that $\sum_{i=1}^{k+1} w_i a_i \approx \sum_{i=1}^{k+1} w_i b_i \bmod \delta^2$ and that $\sum_{i=1}^{k+1} w_i a_i =_{\delta^2} \sum_{i=1}^{k+1} w_i b_i$. If $w_{k+1} = 0$, then these facts follow immediately from the fact that $\bar{a}_k E_{n+1} \bar{b}_k$ since $V_n \subseteq V_{n+1}$ and δ^2 divides γ^2. So assume $w_{k+1} \neq 0$.

Since $\bar{a}_k E_{n+1} \bar{b}_k$, we have $\sum_{i=1}^{k} w_i a_i \approx \sum_{i=1}^{k} w_i b_i \bmod \delta^2$. Thus to show that $\sum_{i=1}^{k+1} w_i a_i \approx \sum_{i=1}^{k+1} w_i b_i \bmod \delta^2$, it is sufficient to show that $w_{k+1} a_{k+1} \approx w_{k+1} b_{k+1} \bmod \delta^2$. But $w_{k+1} a_{k+1} \approx w_{k+1} b_{k+1} \bmod \delta^2$ $\Leftrightarrow \delta a_{k+1} \approx \delta b_{k+1} \bmod (\delta/w_{k+1})\delta^2)$. Hence $\sum_{i=1}^{k+1} w_i a_i \approx \sum_{i=1}^{k+1} w_i b_i \bmod \delta^2$.

Next we will show that $\sum_{i=1}^{k+1} w_i a_i =_{\delta^2} \sum_{i=1}^{k+1} w_i b_i$. Since $V_n = \{-a \mid a \in V_n\}$, and since $\sum_{i=1}^{k+1} w_i a_i =_{\delta^2} \sum_{i=1}^{k+1} w_i b_i \Leftrightarrow \sum_{i=1}^{k+1} -w_i a_i =_{\delta^2} \sum_{i=1}^{k+1} -w_i b_i$, we can assume without loss of generality that $w_{k+1} > 0$. By Lemma 3.3 it is sufficient to show that $\sum_{i=1}^{k+1} w_i a_i \geq d \Leftrightarrow \sum_{i=1}^{k+1} w_i b_i$ for every d, $|d| \leq \delta^2$. Therefore fix d, $|d| \leq \delta^2$.

$$\sum_{i=1}^{k+1} w_i a_i \geq d \Leftrightarrow (\sum_{i=1}^{k} (\delta/w_{k+1}) w_i a_i) + \delta a_{k+1} \geq d(\delta/w_{k+1}) \Leftrightarrow$$

$$\delta a_{k+1} \geq (\sum_{i=1}^{k} (-\delta/w_{k+1}) w_i a_i) + d\,(\delta/w_{k+1}).$$

$(-\delta/w_{k+1})w_i \in V_n'$ for $1 \le i \le k$ and $|d(\delta/w_{k+1})| \le \delta^3$, so

$(\sum_{i=1}^{k} (-\delta/w_{k+1})w_i a_i) + d(\delta/w_{k+1}) \in T$. By the above Claim, we can continue: $\delta a_{k+1} \ge (\sum_{i=1}^{k} (-\delta/w_{k+1})w_i a_i) + d(\delta/w_{k+1}) \Leftrightarrow$

$$\sum_{i=1}^{k} v_i a_i + v + u \ge (\sum_{i=1}^{k}(-\delta/w_{k+1})w_i a_i) + d(\delta/w_{k+1}) \Leftrightarrow$$

$$\sum_{i=1}^{k} (v_i + (\delta/w_{k+1})w_i)a_i \ge d(\delta/w_{k+1}) - v - u,$$

$$|d(\delta/w_{k+1}) - v - u| \le 3\delta^3 \le \gamma^2 \text{ (since } 2\delta^2 \text{ divides } \gamma).$$

Because $\bar{a}_k E_{n+1} \bar{b}_k$ we have

$$\sum_{i=1}^{k} (v_i + (\delta/w_{k+1})w_i)a_i \ge d(\delta/w_{k+1}) - v - u \Leftrightarrow$$

$$\sum_{i=1}^{k} (v_i + (\delta/w_{k+1})w_i)b_i \ge d(\delta/w_{k+1}) - v - u \Leftrightarrow$$

$$\sum_{i=1}^{k} (\delta/w_{k+1})w_i b_i + \sum_{i=1}^{k} v_i b_i + v + u \ge d(\delta/w_{k+1}) \Leftrightarrow$$

$$\sum_{i=1}^{k} (\delta/w_{k+1})w_i b_i + \delta b_{k+1} \ge d(\delta/w_{k+1}) \Leftrightarrow \sum_{i=1}^{k+1} w_i b_i \ge d.$$

It remains to calculate the size of b_{k+1}.

$$|b_{k+1}| < |\sum_{i=1}^{k} v_i b_i + v + u| \le k \cdot \operatorname{Max} V_{n+1} \cdot \operatorname*{Max}_{1 \le i \le k} \{b_i\} + \delta^3 + \delta^3$$

$\le k \cdot \operatorname{Max} V_{n+1} \cdot \operatorname*{Max}_{1 \le i \le k} \{b_i\} + 2 \cdot (\operatorname{Max} V_n)^{|V_n| \cdot 3}$. Therefore by Lemma 3.7, we have for some constant d, $|b_{k+1}| \le (1 + \operatorname*{Max}_{1 \le i \le k} \{b_i\}) 2^{2^{2^{d(n+k)}}}$. $\square$

Corollary 3.9: For some constant d, Z is H-bounded where

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

Proof: Immediate from Lemmas 2.12, 3.6 and Theorem 3.8. □

Theorem 3.10: Let F be the sentence of $\mathcal{L}_1$, $Q_1x_1Q_2x_2\ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free. Then for some constant d independent of n, F is equivalent in Z to $(Q_1x_1 < 2^{2^{2^{dn+1}}})(Q_2x_2 < 2^{2^{2^{dn+2}}})\ldots(Q_nx_n < 2^{2^{2^{dn+n}}})G(\bar{x}_n)$.

Proof: Say that Z is H-bounded where $H(n,k,m) = (1+m)2^{2^{2^{d(n+k)}}}$.

Let $m_i = 2^{2^{2^{dn+i}}}$ for $1 \le i \le n$. Applying Lemma 2.11 to Z, we see that since $m_i \ge H(n-i,\ i-1,\ \max_{1\le j<i}\{|m_j|\})$ for $1 \le i \le n$, F is equivalent to $(Q_1x_1 < m_1)\ (Q_2x_2 < m_2)\ \ldots\ (Q_nx_n < m_n)\ G\ (\bar{x}_n)$. □

Corollary 3.11: For some constant c, TH(< Z, +, ≤, 0 >) can be decided within space $2^{2^{cn}}$.

Proof: By Theorem 1.4.2, given a sentence F of $\mathcal{L}_1$, convert it to an equivalent sentence $Q_1x_1Q_2x_2\ \ldots\ Q_nx_n\ G\ (\bar{x}_n)$ where G is quantifier free and of length at most n log n where $n = |F|$. F is equivalent in Z to

$$(Q_1 x_1 < 2^{2^{2^{dn+1}}})(Q_2 x_2 < 2^{2^{2^{dn+2}}}) \ldots (Q_n x_n < 2^{2^{2^{dn+n}}})\ G\ (\bar{x}_n) \text{ for}$$

some constant d (by Theorem 3.10).

F can be decided in Z by setting aside for quantifier Q_i, $1 \le i \le n$, $2^{2^{dn+i}} + 2$ tape squares; every integer $\le 2^{2^{2^{dn+i}}}$ in absolute value 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. ☐

<u>Theorem 3.12</u>: 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 $n \in N$.

See Fischer and Rabin [FiR74] for a proof of Theorem 3.12. Their proof uses the method described in Chapter 1, and hence, for the reasons described in Chapter 1, the upper bound of Corollary 3.11 matches the lower bound of Theorem 3.12 reasonably well.

<u>Definition</u>: Let R be the structure ⟨R, +, ≤, 0⟩ where R is the set of real numbers and + and ≤ are the usual real addition and order.

As above, the upper bound for TH(R) in Theorem 3.13 is close to the lower bound of Theorem 3.14.

Theorem 3.13: For some constant c, TH(R) can be decided in space 2^{cn}.

The proof appears in Ferrante and Rackoff [FR74]. Although part of their proof uses quantifier elimination, it could be rewritten to follow the E-game format used above without loss of efficiency.

Theorem 3.14: For some constant c', any nondeterministic Turing machine which recognizes TH(R) requires time $2^{c'n}$ on some sentence of length n, for infinitely many n.

See Fischer and Rabin [FiR74] for a proof of Theorem 3.14.

Section 4: Complexity of E-Game Decision Procedures.

We have mentioned that an E-game procedure for deciding $TH(S)$ is one which proceeds by defining relations E_n and proving that the conditions of Lemma 2.12 or 2.12' hold. It is then necessary, in order to decide a sentence with n quantifiers, to be able to write down for every i between 0 and n representations of all the E_i equivalence classes on S^{n-i}; this is what is really going on in Lemma 2.11 and the examples of the previous section. Chapters 3 and 4 contain further applications of these ideas.

It is not enough only to be able to write down for every $n, k \in N$ representations of all the E_n equivalence classes on S^k, but this is certainly a necessary part of an E-game decision procedure. Recalling that the E_n classes are at least as numerous as the $\equiv_n$ classes (because of Lemma 2.12), we see that if an E-game procedure (as we have described them) is to be elementary recursive, it is <u>necessary</u> that $M(n,k)$ be bounded above by an elementary recursive function.

Now the only other method we know about for obtaining elementary recursive decision procedures is elimination of quantifiers, and we have stated above that in all known cases a quantifier elimination procedure can be transformed into an E-game procedure without sacrificing (if it was there in the first place) elementary recursiveness. What this means is that in order for a logical theory to be elementary recursively decidable by known methods, it is necessary for $M(n,k)$ to be bounded above by an elementary recursive function. This raises the following important conjecture.

<u>Conjecture 4.1</u>: If $TH(\mathcal{S})$ has an elementary recursive decision procedure, then $M(n,k)$ is bounded above by an elementary recursive function.

Although Conjecture 4.1 is open, its converse is definitely false.

<u>Counterexample to the Converse of Conjecture 4.1</u>:

For the purpose of this counterexample, let $\mathcal{L}$ 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 (although the constant symbol isn't really necessary).

For every nonempty set A of positive integers let $\underset{A}{\sim}$ be an equivalence relation on N such that for every positive integer i

1) if $i \in A$ then there is exactly one $\underset{A}{\sim}$ equivalence class of size i.

and

2) if $i \notin A$ then there are no equivalence classes of size i.

Define the structure $\mathcal{S}_A = < N, =, \underset{A}{\sim}, 0 >$.

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 $TH(\mathcal{S}_A)$ can be decided within time $g(n)$, then A can be decided within time $g(p(n)) + p(n)$ for some polynomial p. Since we can make A arbitrarily hard to decide or arbitrarily nonrecursive, we can make $TH(\mathcal{S}_A)$ arbitrarily hard to decide or arbitrarily nonrecursive.

Now let A be a fixed set of positive integers and consider $M(n,k)$

for $\mathcal{S}_A$; we will show that (no matter what A is) M(n,k) is bounded above by an elementary recursive function, contradicting the converse of Conjecture 4.1.

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) $a_i \underset{A}{\sim} 0 \Leftrightarrow b_i \underset{A}{\sim} 0$, and $a_i = 0 \Leftrightarrow b_i = 0$.

II) $a_i \underset{A}{\sim} a_j \Leftrightarrow b_i \underset{A}{\sim} b_j$, and $a_i = a_j \Leftrightarrow b_i = b_j$.

and

III) $\{a \in N \mid a \underset{A}{\sim} a_i\} \underset{n+k}{=} \{b \in N \mid b \underset{A}{\sim} b_i\}$. It is not difficult to prove Lemma 4.2 using Lemma 2.12'.

<u>Lemma 4.2</u>: $\bar{a}_k \; E_n \; \bar{b}_k \Rightarrow \bar{a}_k \underset{n}{\equiv} \bar{b}_k$.

Since the number of E_n equivalence classes on N^k is bounded above by an elementary recursive function (of n and k), namely $2^{2^{c(n+k)}}$, M(n,k) for $\mathcal{S}_A$ is bounded above by the same function.

Chapter 3: Weak Direct Powers

Section 1: Weak Direct Powers and Ehrenfeucht Games

Let $\mathcal{L}$ be a language of the first order predicate calculus with a finite number of predicate symbols $\underline{\mathcal{R}}_1, \underline{\mathcal{R}}_2, \ldots, \underline{\mathcal{R}}_\ell$ such that $\underline{\mathcal{R}}_i$ is a t_i place formal predicate for $1 \leq i \leq \ell$, and with a constant symbol $\underline{e}$.

Definition 1.1: Let $\mathcal{S} = < S, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_\ell, e >$ be a structure for $\mathcal{L}$. For all $a \in S$, $||a||$ is the norm of a. The weak direct power of $\mathcal{S}$ is the structure $\mathcal{S}^* = < S^*, \mathcal{R}^*_1, \mathcal{R}^*_2, \ldots, \mathcal{R}^*_\ell, e^* >$ where

$S^* = \{f: N \to S \mid f(i) \neq e \text{ for only finitely many } i \in N\}$;

for $1 \leq j \leq \ell$, if $\vec{f}_{t_j} \in (S^*)^{t_j}$, then $\vec{f}_{t_j} \in \mathcal{R}^*_j$ iff $\vec{f}_{t_j}(i) \in \mathcal{R}_j$ 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 $\mathcal{S}^*$ we define, for $f \in S^*$,
$||f|| = \text{Max}(\{i \in N \mid f(i) \neq e\} \cup \{||f(i)|| \mid i \in N\})$. By $f \prec m$ we will mean $||f|| \leq m$.

Mostowski [Mos52] and Feferman and Vaught [FV59] both show that $TH(\mathcal{S})$ decidable $\Rightarrow TH(\mathcal{S}^*)$ decidable. However, their proofs are such that in every case, the decision procedure for $TH(\mathcal{S}^*)$ 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

$TH(\mathcal{S}^*)$ in many cases, and in particular to obtain a procedure for $TH(\mathcal{Z}^*)$ (where $\mathcal{Z}$ is the structure of integer addition defined in Chapter 2) which closely matches the known lower bound. In Chapter 4 we prove even more general theorems which give a condition under which we can conclude $TH(\mathcal{S}^*)$ elementary recursive if $TH(\mathcal{S})$ is elementary recursive.

Now let $H: N^3 \to N$ be such that $\mathcal{S}$ is H-bounded. Let $M(n,k)$ be the function as defined for $\mathcal{S}$ in Chapter 2, definition 2.2.5.

<u>Definition 1.2</u>: 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 $\mu(n,k) = \prod_{i=1}^{n} M(n - i, k + i)$.

<u>Definition 1.3</u>: 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 $\mathcal{S}^*$ is H^*-bounded.

We now prove a combinatorial lemma. $\underset{n}{\equiv}$ is defined in Definition 2.3.2.

<u>Lemma 1.4</u>: Let N_1 and N_2 be sets and let $n, m \in N^+$ such that $N_1 \underset{n\cdot m}{\equiv} N_2$. Let $A_1, A_2, \ldots, A_n$ be a sequence of (possibly empty) pairwise disjoint subsets of N_1 such that $\bigcup_{i=1}^{n} A_i = N_1$.

<u>Then</u> 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 $A_i \underset{m}{\equiv} B_i$ for $1 \le i \le n$.

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 S^k and $(S^*)^k$ as in Chapter 2, Definition 2.2.1.

Definition 1.5: Let n, $k \in N$ and $\vec{f}_k, \vec{g}_k \in (S^*)^k$. Then we say $\vec{f}_k \ E_n \ \vec{g}_k$ iff for all $\vec{a}_k \in S^k$, $\{i \in N \mid \vec{f}_k(i) \underset{n}{\equiv} \vec{a}_k\} \underset{\mu(n,k)}{=} \{i \in N \mid \vec{g}_k(i) \underset{n}{\equiv} \vec{a}_k\}$.

Lemma 1.6: For all $k \in N$, $\bar{f}_k, \bar{g}_k \in (S^*)^k$, if $\bar{f}_k \ E_0 \ \bar{g}_k$ then $\bar{f}_k \underset{0}{\equiv} \bar{g}_k$.

Proof: Say that $\overline{f}_k\ E_0\ \overline{g}_k$. We wish to show that for every quantifier free formula $F(\overline{x}_k)$, $\mathcal{S}^* \vdash F(\overline{f}_k) \Leftrightarrow \mathcal{S}^* \vdash F(\overline{g}_k)$. It is clearly sufficient to prove this for the case where F is atomic. By symmetry, it is sufficient to show that $F(\overline{f}_k)$ false in $\mathcal{S}^* \Rightarrow F(\overline{g}_k)$ false in $\mathcal{S}^*$.

Thus assume that $F(\overline{f}_k)$ is false in $\mathcal{S}^*$. By definition of the relations of $\mathcal{S}^*$ we can choose $i_0 \in N$ such that $F(\overline{f}_k(i_0))$ is false in $\mathcal{S}$. Since $\overline{f}_k\ E_0\ \overline{g}_k$, we have that $\{i \in N \mid \overline{f}_k(i) \underset{0}{\equiv} \overline{f}_k(i_0)\} \underset{\mu(0,k)}{=}$

$\{i \in N \mid \overline{g}_k(i) \underset{0}{\equiv} \overline{f}_k(i_0)\}$. Since $\mu(0, k) = 1$, we have

$|\{i \in N \mid \overline{g}_k(i) \underset{0}{\equiv} \overline{f}_k(i_0)\}| \geq 1$. So let $i_1 \in N$ be such that

$\overline{g}_k(i_1) \underset{0}{\equiv} \overline{f}_k(i_0)$. By definition of $\underset{0}{\equiv}$, $F(\overline{f}_k(i_0))$ false in

$\mathcal{S} \Rightarrow F(\overline{g}_k(i_1))$ false in $\mathcal{S}$. So $F(\overline{g}_k)$ is false in $\mathcal{S}^*$. $\square$

Lemma 1.7: Let $n, k \in N$ and $\overline{f}_k, \overline{g}_k \in (S^*)^k$ such that $\overline{f}_k\ E_{n+1}\ \overline{g}_k$. Then for each $f_{k+1} \in S^*$ there exists some $g_{k+1} \in S^*$ such that

1) $\overline{f}_{k+1}\ E_n\ \overline{g}_{k+1}$

and

2) $||g_{k+1}|| \leq H^*(n,k, \underset{1\leq i\leq k}{\text{Max}}\{||g_i||\})$.

<u>Proof</u>: Let $\bar{f}_k, \bar{g}_k \in (S^*)^k$ be such that $\bar{f}_k \; E_{n+1} \; \bar{g}_k$. Let $m = \underset{1 \le i \le k}{\text{Max}} \{||g_i||\}$ and let $f_{k+1} \in S^*$. 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 $\underset{n}{\equiv}$ equivalence classes on S^{k+1}. Our goal is to find $g_{k+1} \in S^*$ such that if $1 \le j \le M(n, k+1)$, then $\{i \in N \mid \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} \{i \in N \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$; we also want $||g_{k+1}|| \le H^*(n,k,m)$. Instead of defining g_{k+1} simultaneously on all of N, we will define it separately on various pieces of N.

For each $\bar{a}_k \in S^k$ define $N_1(\bar{a}_k) = \{i \in N \mid \bar{f}_k(i) \underset{n+1}{\equiv} \bar{a}_k\}$ and $N_2(\bar{a}_k) = \{i \in N \mid \bar{g}_k(i) \underset{n+1}{\equiv} \bar{a}_k\}$. We claim it is sufficient to define g_{k+1} on each $N_2(\bar{a}_k)$ such that

I) $\{i \in N_1(\bar{a}_k) \mid \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} \{i \in N_2(\bar{a}_k) \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$

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 S^* will show that II) and III) together imply $||g_{k+1}|| \le H^*(n,k,m)$. Since $\{N_1(\bar{a}_k) \mid \bar{a}_k \in S^k\}$ and $\{N_2(\bar{a}_k) \mid \bar{a}_k \in S^k\}$ 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

$$\Big(\bigcup_{\bar{a}_k \in S^k} \{i \in N_1(\bar{a}_k) \mid \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}\Big) \underset{\mu(n,k+1)}{=} \Big(\bigcup_{\bar{a}_k \in S^k} \{i \in N_2(\bar{a}_k) \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}\Big),$$

i.e., $\{i \in N \mid \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} \{i \in N \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$.

So now let $\bar{a}_k \in S^k$ 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 $N_3 = \{i \in N_2 \mid i \le m + \mu(n+1, k)\}$.

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 $m < i \le m + \mu(n+1,k)$; if $\bar{a}_k \underset{n+1}{\not\equiv} e^k$ 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 $1 \le j \le M(n,k+1)$, $A_j = \{i \in N_1 \mid \bar{f}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$.

$A_1, A_2, \ldots, A_{M(n,k+1)}$ 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 1.4 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 $\overline{f}_k(i_0) \underset{n+1}{\equiv} \overline{a}_k \underset{n+1}{\equiv} \overline{g}_k(i)$. By Lemma 2.2.10 we can define $g_{k+1}(i)$ such that $\overline{f}_{k+1}(i_0) \underset{n}{\equiv} \overline{g}_{k+1}(i)$ and

$||g_{k+1}(i)|| \leq H(n,k,\mathrm{Max}\{||g_1(i)||,||g_2(i)||,\ldots,||g_k(i)||\}) \leq H(n,k,m)$.

Clearly III) above holds. Since $i_0 \in A_j$, $\overline{f}_{k+1}(i_0) \underset{n}{\equiv} \overline{b}^j_{k+1}$. So $\overline{g}_{k+1}(i) \underset{n}{\equiv} \overline{b}^j_{k+1}$. Thus, we have defined $g_{k+1} \in S^*$ so that for $1 \leq j \leq M(n,k+1)$,

$$\{i \in N_3 \mid \overline{g}_{k+1}(i) \underset{n}{\equiv} \overline{b}^j_{k+1}\} = B_j \underset{\mu(n,k+1)}{=} A_j = \{i \in N_1 \mid \overline{f}_{k+1}(i) \underset{n}{\equiv} \overline{b}^j_{k+1}\}.$$

To complete the proof of Lemma 1.7, we must show I), i.e.,

$$\{i \in N_2 \mid \overline{g}_{k+1}(i) \underset{n}{\equiv} \overline{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} A_j \text{ when } 1 \leq j \leq M(n,\ k+1).$$

So fix j, $1 \le j \le M(n, k + 1)$. If

$\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} = \{i \in N_3 \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$ we are done, so assume

$\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \neq \{i \in N_3 \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\}$. Since

$N_3 = \{i \in N_2 \mid i \le m + \mu(n + 1, k)\}$, there must exist some $i > m + \mu(n + 1, k)$

such that $i \in N_2$ (hence $\bar{g}_k(i) \underset{n+1}{\equiv} \bar{a}_k$) and $\bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}$. But since

$i > m + \mu(n + 1, k)$ implies $\bar{g}_{k+1}(i) = e^{k+1}$, this means that $\bar{a}_k \underset{n+1}{\equiv} e^k$

and $\bar{b}^j_{k+1} \underset{n}{\equiv} e^{k+1}$. Hence, both A_j and $\{i \in N_2 \mid \bar{g}_{k+1}(i) \equiv \bar{b}^j_{k+1}\}$ are

infinite, so $\{i \in N_2 \mid \bar{g}_{k+1}(i) \underset{n}{\equiv} \bar{b}^j_{k+1}\} \underset{\mu(n,k+1)}{=} A_j$. □

<u>Theorem 1.8</u>: $\mathcal{S}^*$ is H^*-bounded. Also, for every $n, k \in N$ and

$\bar{f}_k, \bar{g}_k \in (S^*)^k$, $\bar{f}_k \; E_n \; \bar{g}_k \Rightarrow \bar{f}_k \underset{n}{\equiv} \bar{g}_k$.

<u>Proof</u>: This follows immediately from Lemmas 2.2.12, 1.6, and 1.7. □

Section 2: Applications

We now present some applications of the material in Section 1. Let $\mathcal{L}_1$ be the language of Chapter 2.

Let $Z = \langle Z, +, \le, 0 \rangle$ be the structure of Chapter 2 and let $Z^* = \langle Z^*, +, \le, 0^* \rangle$ be the weak direct power of Z. As before, for $a \in Z$ let $||a|| = |a|$ and, following Definition 1.1, for $f \in Z^*$ let $||f|| = \mathrm{Max}\ (\{i \in N \mid f(i) \ne 0\} \cup \{|f(i)| \mid i \in N\})$.

Lemma 2.1: There exists a constant e such that Z^* is $(1+m)\cdot 2^{2^{2^{e(n+k)}}}$-bounded.

Proof: By Corollary 2.3.9, 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 Z. Letting $m_i = 2^{2^{2^{d(n+k)+i}}}$ for $1 \le i \le k$, we see that $m_i \ge H(n+k-i,\ i-1,\ \mathrm{Max}_{1\le j\le i}\{|m_j|\})$ for $1 \le i \le k$. So by Lemma 2.2.13, for each $\bar{a}_k \in Z^k$ there is some $\bar{b}_k \in Z^k$ such that $\bar{a}_k \equiv_n \bar{b}_k$ and $|b_i| \le m_i$ for $1 \le i \le k$. Hence, since $m_i \le m_k$, we certainly have $M(n,k) \le (2\cdot 2^{2^{2^{d(n+k)+k}}} + 1)^k$. So $\mu(n,k) = \prod_{i=1}^{n} M(n-i,\ k+1) \le 2^{2^{2^{d'(n+k)}}}$ for some constant d'.

So for some constant e, $H^*(n,k,m) = \mathrm{Max}\{H(n,k,m),\ m + \mu(n+1,\ k),\ 0\} \le (1+m)\cdot 2^{2^{2^{e(n+k)}}}$.

By Theorem 1.8, Z^* is $(1 + m)\cdot 2^{2^{2^{e\cdot(n+k)}}}$-bounded. □

<u>Theorem 2.2</u>: Let F be the sentence of $\mathcal{L}_1$, $Q_1x_1Q_2x_2 \ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free. Then for some constant e independent of n, F is equivalent in Z^* to $(Q_1x_1 \prec 2^{2^{2^{en+1}}})(Q_2x_2 \prec 2^{2^{2^{en+2}}}) \ldots (Q_nx_n \prec 2^{2^{2^{en+n}}})G(\bar{x}_n)$.

<u>Proof</u>: Theorem 2.2 follows from Lemma 2.1 exactly as Theorem 2.3.10 follows from Corollary 2.3.9. □

<u>Corollary 2.3</u>: For some constant c, $TH(< Z, +, \leq, 0 >^*)$ can be decided within space $2^{2^{2^{cn}}}$.

<u>Proof</u>: By Theorem 1.4.2 it is sufficient to consider the sentence F of $\mathcal{L}_1$ 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 Theorem 2.2, F is equivalent to

$$(Q_1x_1 \prec 2^{2^{2^{en+1}}})(Q_2x_2 \prec 2^{2^{2^{en+2}}}) \ldots (Q_nx_n \prec 2^{2^{2^{en+n}}})G(\bar{x}_n) \text{ for some}$$

constant e.

Now if $f \in Z^*$ and $f \prec 2^{2^{2^{en+i}}}$, 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 $(2^{2^{cn+1}} + 2)\cdot 2^{2^{cn+1}}$ tape squares. So a procedure like the one outlined in Corollary 2.3.11 would decide $TH(Z^*)$ in space $2^{2^{2^{cn}}}$ for some constant c. □

<u>Definition 2.4</u>: Let N^* be the structure $< N^*, +, \leq, 0^* >$, i.e, the weak direct power of the nonnegative integers (under + and $\leq$).

<u>Remark 2.5</u>: The structure $< N^*, + >$ is isomorphic to the structure $< N^+, \cdot >$ (i.e., the positive integers under multiplication). So an upper bound on the complexity of $TH(N^*)$ is an upper bound on $TH(< N^+, \cdot >)$.

<u>Corollary 2.6</u>: $TH(N^*)$ can be decided in space $2^{2^{2^{cn}}}$ for some constant c.

<u>Proof</u>: Since $x \geq 0$ is a formula of $\mathcal{L}_1$, it is easy to see that $TH(N^*) \leq_{p\ell} TH(Z^*)$. So Corollary 2.6 follows from Lemma 1.3.2. □

The upper bound of Corollary 2.3 and Corollary 2.6 matches the lower bound of Theorem 2.7 reasonably well.

<u>Theorem 2.7</u>: (Fischer and Rabin [FiR74]) For some constant c' > 0, any nondeterministic Turing machine which recognizes $TH(Z^*)$ (or $TH(N^*)$)

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(N^*) and conclude

<u>Theorem 2.8</u>: The first order theory of finite abelian groups can be decided within space $2^{2^{2^{cn}}}$ for some constant c.

There is still a significant gap between the upper bound of Theorem 2.8 and the known lower bound of Theorem 2.9.

<u>Theorem 2.9</u> (Fischer and Rabin [FiR74]): For some constant $c' > 0$, any nondeterministic Turing machine which recognizes the theory of finite abelian groups requires time $2^{2^{2^{c'n}}}$ on some sentence of length n, for infinitely many n.

The language of groups, $\mathcal{L}_2$, merely contains the formal predicate $v_1 + v_2 = v_3$. We are interested in deciding which sentences of $\mathcal{L}_2$ are true of every finite abelian group. Recall that every finite abelian

---

[†] This topic is also discussed in Chapter 4 from a slightly different viewpoint.

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 [Fis73]) is to think of every nonzero $f \in N^*$ as representing an FAG, G_f. This is made precise in the following definition.

<u>Definition 2.10</u>: 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

$$m_f(j) = \text{the } j^{th} \text{ smallest member of } \{i \in N \mid f(i) \neq 0\} \text{ for } 1 \leq j \leq \ell_f.$$

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 $f \in N^*$, $f \neq 0^*$.

<u>Definition 2.11</u>: 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 $\mathcal{F}_1$ by formulas interpreted over N^*.

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

$$MEM(x_1,x_2) \wedge MEM(x_1,x_3) \wedge MEM(x_1,x_4) \wedge \forall x \forall x_1' \forall x_2' \forall x_3' \forall x_4' [$$

$$(PICK(x,x_1',x_1) \wedge PICK(x,x_2',x_2) \wedge PICK(x,x_3',x_3) \wedge PICK(x,x_4',x_4)) \rightarrow$$

$$(x_2' + x_3' = x_4' \vee x_2' + x_3' = x_4' + x_1')].$$

<u>Proof of Theorem 2.8</u>: 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 $\mathcal{L}_2$ to a sentence F' of $\mathcal{L}_1$, such that F is true of every FAG $\Leftrightarrow F' \in TH(N^*)$. So $TH(FAG) \leq_{p\ell} TH(N^*)$. Theorem 2.8 therefore follows from Corollary 2.6 and Lemma 1.3.2. $\square$

Chapter 4: Some General Results about the Complexity of Direct Products

Section 1: Introduction.

Let $\mathcal{L}$, $\mathcal{S}$, and $\mathcal{S}^*$ be defined as in Chapters 2 and 3, and let $M(n,k)$ be defined for $\mathcal{S}$ as in Definition 1.2.5.

Theorem 1.1: If $TH(\mathcal{S})$ is elementary recursive and if $M(n,k)$ is bounded above by an elementary recursive function, then $TH(\mathcal{S}^*)$ is elementary recursive.

Theorem 1.1 can be proven by modifying either Mostowski's or Feferman and Vaught's decision procedure for $TH(\mathcal{S})^*$ [Mos52, FV59], but we present a different approach in Section 2 and prove there a quantitative version of Theorem 1.1. In Section 3 we present some similar results for other notions of direct products (besides weak direct powers).

The converse to Theorem 1.1 is false.

Counterexample to the Converse to Theorem 1.1:

Let $\mathcal{L}$ be the language used in the counterexample to Conjecture 2.4.1. For every nonempty set $A \subseteq N^+$ define $\mathcal{S}_A$ as in Chapter 2 to be

$< N, =, \underset{A}{\sim}, 0 >$. As in Chapter 2, by varying A we can make $\mathcal{S}_A$ arbitrarily

hard to decide. Let A be a fixed set such that $1 \notin A$, i.e., there are no $\underset{A}{\sim}$ equivalence classes of size 1.

Claim: $\mathcal{S}_A^*$ consists of an infinite collection of infinite equivalence classes.

<u>Proof of Claim</u>: Since 0 is not in an equivalence class of size 1, there exists some number, say 1, such that $1 \underset{A}{\sim} 0$. Since $A \neq \emptyset$, there exists some finite $\underset{A}{\sim}$ class, and hence at least two $\underset{A}{\sim}$ classes. So there exists some number, say 2, such that it is <u>not</u> true that $2 \underset{A}{\sim} 0$.

Thinking of every member of N^* as an infinite sequence of members of N, we see that the strings $0,0,0,\ldots$; $2,0,0,\ldots$; $2,2,0,0\ldots$; ... form an infinite set of pairwise inequivalent members of N^*. So S_A^* has an infinite number of equivalence classes.

Let $\gamma,0,0,\ldots$ be any member of N^*, where γ is a finite sequence of members of N. The strings $\gamma,1,0,0,\ldots$; $\gamma,1,1,0,0,\ldots$; ... form an infinite set of elements equivalent to $\gamma,0,0,\ldots$. So each equivalence class of S_A^* is infinite, proving the claim. □

From the above claim,† it is not hard to see that a sentence of $\mathcal{L}$ with n quantifiers will be true in S_A^* iff it is true in a domain of size n^2 consisting of exactly n equivalence classes of size n. Therefore, $TH(S_A^*)$ can be decided in polynomial space, even though $TH(S_A)$ may be arbitrarily difficult to decide.

† and Lemma 2.4.2.

Section 2: Complexity of Weak Direct Powers.

Our goal in this chapter is to prove Theorem 1.1; actually, we shall prove a quantitative version of Theorem 1.1, which relates the complexity of $TH(\mathcal{S}^*)$ to the complexity of $TH(\mathcal{S})$ and $M(n,k)$.

To begin with, let $\mathcal{S} = < S, \mathcal{R}_1, \ldots, \mathcal{R}_\ell, e >$ be a structure as before and let $\mathcal{L}$ be the corresponding first order language. $\mathcal{S}$ and $\mathcal{L}$ are fixed for the rest of this chapter. Let $\underset{n}{\equiv}$ be defined on S^k for each $n, k \in N$ as in Chapter 2, Definition 2.2.1. Let $C_{n,k}$ be the set of equivalence classes determined by $\underset{n}{\equiv}$ on S^k and let $M(n,k) = |C_{n,k}|$ as before. For $\bar{a}_k \in S^k$, let $[\bar{a}_k]_n$ be the equivalence class of $\bar{a}_k$ determined by $\underset{n}{\equiv}$. By Lemma 2.2.6, for every $\bar{a}_k \in S^k$ there is a formula $F(\bar{x}_k)$ defining $[\bar{a}_k]_n$. What we are now interested in is how much time is needed, as a function of n and k, to write down all such formulas.

Remark: Here is the motivation behind what we will be doing. Using a decision procedure for $TH(\mathcal{S})$ we will obtain (efficient) representations of the members of $C_{n,k}$. This will allow us to use results of Chapter 3 to obtain efficient representations of the $\underset{n}{\equiv}$ classes on $(S^*)^k$. We will then decide the truth of sentences in $\mathcal{S}^*$ by limiting quantifiers to range over appropriate sets of these representations.

Definition 2.1: We will define for every $n, k \in N$ a collection of formulas, $\mathcal{F}_{n,k}$, such that in every member of $\mathcal{F}_{n,k}$ exactly $x_1, x_2, \ldots, x_k$ occur freely. Firstly, for every $k \in N$ define

$\mathfrak{S}_k = \{F(\bar{x}_k) \mid F$ is an atomic formula$\}$; for every $W \subseteq \mathfrak{S}_k$ define $F_{0,k,W}(\bar{x}_k)$ to be the formula $(\bigwedge_{F \in W} F) \wedge (\bigwedge_{F \in \mathfrak{S}_k - W} \sim F)$; define

$\mathcal{F}_{0,k} = \{F_{0,k,W} \mid \mathcal{S} \vdash \exists x_1 \exists x_2 \ldots \exists x_k\, F_{0,k,W}(\bar{x}_k)\}$.

Assuming $\mathcal{F}_{n,k+1}$ has been defined such that in every member exactly $x_1, x_2, \ldots, x_{k+1}$ occur freely, we now define $\mathcal{F}_{n+1,k}$. For every $W \subseteq \mathcal{F}_{n,k+1}$ define $F_{n+1,k,W}(\bar{x}_k)$ to be the formula $(\bigwedge_{F \in W} \exists x_{k+1} F) \wedge (\bigwedge_{F \in \mathcal{F}_{n,k+1} - W} \sim \exists x_{k+1} F)$.

Define $\mathcal{F}_{n+1,k} = \{F_{n+1,k,W}(\bar{x}_k) \mid \mathcal{S} \vdash \exists x_1 \exists x_2 \ldots \exists x_k F_{n+1,k,W}\}$. Clearly exactly $x_1, x_2, \ldots, x_k$ occur freely in each member of $\mathcal{F}_{n+1,k}$.†

Lemma 2.2: Let $n, k \in \mathbb{N}$. Then

$\{A \subseteq S^k \mid$ some member of $\mathcal{F}_{n,k}$ defines $A\} = C_{n,k}$. Furthermore, every member of $C_{n,k}$ is defined by a unique member of $\mathcal{F}_{n,k}$.

Proof: Lemma 2.2 follows immediately from the proof of Lemma 2.2.6. □

We next wish to calculate how long it takes as a function of n and k for a Turing machine to write down the set $\mathcal{F}_{n,k}$ on its tape when implementing Definition 2.1. In order for a Turing machine to do this at all it is necessary that TH($\mathcal{S}$) be decidable, so for the rest of this section assume that there is some decision procedure for TH($\mathcal{S}$) which

† Every $F \in \mathcal{F}_{n,k}$ is considered to implicitly contain the annotation $x_1, x_2, \ldots, x_k$.

operates within time $T_1'(n)$. In order to simplify the calculations to follow, instead of working with the function $T_1'(n)$ we will use instead some non-decreasing function $T_1(n) \geq \text{Max}\ \{T_1'(n),\ 2^n\}$. It will similarly make things simpler below if we define the function

$T_2(n) = \text{Max}(\{M(n-k,\ k)\ |\ 0 \leq k \leq n\} \cup \{n\})$.† The reader may note that at many places in the calculations below we make gross over-estimates. This is because we are ultimately interested in the amount of nesting of exponentials in the complexity of our decision procedures, and our over-estimates do not affect this, whereas they do have the advantage of shortening the expressions we obtain.

We first define $L(n,k)$ to be the length of the longest formula of the form $F_{n,k,W}$.

To calculate $L(0,k)$, note that (as in the proof of Lemma 2.2.6)

$|\mathfrak{S}_k| = \sum_{i=1}^{\ell} (k+1)^{t_i}$ (where $\underline{R}_i$ is a t_i-place relation for $1 \leq i \leq \ell$).

As k increases, the length of the longest member of $\mathfrak{S}_k$ will increase since longer subscripts of formal variables will have to be written; however, for every $k \geq 0$ the length of the longest member of $\mathfrak{S}_k$ will be $\leq c_1 \cdot (k+1)$ for some constant c_1 independent of k. Everything of the form $\mathcal{F}_{0,k,W}$ looks like a concatenation of the members of $\mathfrak{S}_k$, with some additional logical symbols, and is of length $\leq$ twice the length of the concatenation of the members of $\mathfrak{S}_k$. That is,

---

†It is easy to see that T_2 is nondecreasing.

$L(0,k) \leq 2 \cdot c_1 \cdot (k+1) \cdot \sum_{i=1}^{\ell} (k+1)^{t_i} \leq (k+2)^{c_2}$ for some constant c_2 independent of k.

Everything of the form $F_{n+1,k,W}$ looks like a concatenation of the members of $\mathcal{F}_{n,k+1}$, with some additional symbols; for some constant c_3 they are each of length $\leq c_3 \cdot (k+1) \cdot$ the length of the concatenation of the members of $\mathcal{F}_{n,k+1}$. That is,

$L(n+1, k) \leq c_3 \cdot (k+1) \cdot L(n, k+1) \cdot |\mathcal{F}_{n,k+1}| \leq$

$c_3 \cdot (k+1) \cdot L(n, k+1) \cdot T_2(n+k+1)$. Since

$L(0,k) \leq (k+2)^{c_2}$ and $T_2(n+k) \geq n+k$ we can calculate that

$L(n,k) \leq (\text{Max}\{T_2(n+k), 2\})^{c_4(n+1)}$ for every $n,k \in N$ and for some constant c_4.

Now define $T(n,k)$ to be the time which a Turing machine implementing Definition 2.1 takes to write down $\mathcal{F}_{n,k}$ on its tape. We first calculate an upper bound on $T(n+1, k)$ in terms of $T(n, k+1)$.

To compute $\mathcal{F}_{n+1,k}$ we begin by computing $\mathcal{F}_{n,k+1}$ within time $T(n,k+1)$. We next write down beside $\mathcal{F}_{n,k+1}$ on the tape the set $\{F_{n+1,k,W} \mid W \subseteq \mathcal{F}_{n,k+1}\}$. Then for each $W \subseteq \mathcal{F}_{n,k+1}$ we write down the sentence

$\exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}$, and then use our decision procedure for $TH(\mathcal{S})$ to decide for each $W \subseteq \mathcal{F}_{n,k+1}$ if $\mathcal{S} \vdash \exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}$.

We lastly consolidate all the material on the tape (i.e. erasing $F_{n+1,k,W}$ for cases where it is not true that $\mathcal{S} \vdash \exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}$) so that next to $\mathcal{F}_{n,k+1}$ we have written $\mathcal{F}_{n+1,k}$.

For each $W \subseteq \mathcal{F}_{n,k+1}$, we know that $x_1, x_2, \ldots, x_k$ occur in $F_{n+1,k,W}$, so that $|\exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}| \leq 3 \cdot |F_{n+1,k,W}| \leq 3 \cdot L(n + 1, k)$. The decision procedure for TH($\mathcal{S}$) decides whether or not $\mathcal{S} \vdash \exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}$ within time and space $T_1(3 \cdot L(n+1,k))$; actually in order to decide if $\exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W} \in TH(\mathcal{S})$ and return the Turing machine head (which started on the leftmost $\exists$) to its original position requires time $\leq 2T_1(3 \cdot L(n + 1, k))$. So when computing $\mathcal{F}_{n+1,k}$, the total amount of time used in deciding membership in TH($\mathcal{S}$) is $\leq 2T_1(3 \cdot L(n + 1, k)) \cdot 2^{|\mathcal{F}_{n,k+1}|} \leq 2T_1(3 \cdot L(n + 1, k)) \cdot 2^{T_2(n+k+1)}$.

We lastly calculate how much time is used in computing $\mathcal{F}_{n+1,k}$ which is not used in either computing $\mathcal{F}_{n,k+1}$ or in deciding membership in TH($\mathcal{S}$). The total amount of space used in this way is the space on which $\mathcal{F}_{n,k+1}$ is written plus the space to write $F_{n+1,k,W}$ for every $W \subseteq \mathcal{F}_{n,k+1}$ plus the space to write $\exists x_1 \exists x_2 \ldots \exists x_k\, F_{n+1,k,W}$ for every $W \subseteq \mathcal{F}_{n,k+1}$; this is $\leq (L(n, k + 1)) \cdot |\mathcal{F}_{n,k+1}| + (L(n + 1, k)) \cdot 2^{|\mathcal{F}_{n,k+1}|} + (3 \cdot L(n + 1, k)) \cdot 2^{|\mathcal{F}_{n,k+1}|} \leq 2^{|\mathcal{F}_{n,k+1}|} \cdot 5 \cdot L(n + 1, k) \leq 5 \cdot 2^{T_2(n+k+1)} \cdot L(n + 1, k)$.

The time our Turing machine uses (aside from computing $\mathcal{F}_{n,k+1}$ or membership in TH($\mathfrak{S}$)) is spent in having the head go back and forth in this space doing the necessary amount of copying; the reader can verify for himself that this is $\leq (5 \cdot 2^{T_2(n+k+1)} \cdot L(n+1, k))^{c_5}$ for some constant c_5.†

So the total amount of time used in computing $\mathcal{F}_{n+1,k}$

$$= T(n+1, k) \leq T(n, k+1) + 2T_1(3 \cdot L(n+1, k)) \cdot 2^{T_2(n+k+1)} + (5 \cdot 2^{T_2(n+k+1)} \cdot L(n+1,k))^{c_5}$$

Since $T_2(n) \geq n$ and $T_1(n) \geq 2^n$ for all $n \in N$, we can calculate that for some constant c_6,

$$T(n+1, k) \leq T(n, k+1) + [T_1((T_2(n+k+2))^{c_6(n+k+1)})]^{c_6}.$$

It can also be seen that the time needed to write down $\mathfrak{S}_k$ is polynomial in the space needed, and therefore $\leq (L(0, k))^{c_7}$ for some constant c_7. Obtaining $\mathcal{F}_{0,k}$ from $\mathfrak{S}_k$ is certainly quicker than obtaining $\mathcal{F}_{1,k}$ from $\mathcal{F}_{0,k+1}$, so we have $T(0,k) \leq (L(0,k))^{c_7} + [T_1((T_2(k+2))^{c_6(k+1)})]^{c_6}$.

Doing some final calculations we can conclude that

---

† We are using the fact that we can simultaneously use space for two different purposes. For instance, some of the space on which sentences are written down is also used for deciding truth of sentences in $\mathfrak{S}$.

$T(n, k) \leq [T_1((T_2(n+k+2))^{c(n+k+1)})]^c$ for some constant c and all $n,k \in N$.

Lemma 2.3: For some constant c, there is a procedure which given n, writes down the sequence $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}$ within time $[T_1((T_2(n+2))^{c(n+1)})]^c$; the length of this sequence is $\leq (T_2(n+2))^{c(n+1)}$.

Proof: When we were calculating above the time to write down $\mathcal{F}_{n,0}$, we were calculating as well the time to write down the sequence $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}$. The length of the sequence is $\leq (n+1)(T_2(n)) \cdot L(n,0) \leq (T_2(n+2))^{c(n+1)}$ for some constant c. □

Remark 2.4: Note that every member of $\mathcal{F}_{n,0}$ must be a true sentence and hence define the set whose sole member is the empty set. Therefore, Lemma 2.2 implies that $M(n,0) = |\mathcal{F}_{n,0}| = 1$.

Definition 2.5: For every n, $k \in N$, let $F_{n,k}$ be that member of $\mathcal{F}_{n,k}$ which defines $[e^k]_n$. That is, $F_{n,k}$ is the unique member of $\mathcal{F}_{n,k}$ such that $\mathcal{S} \vdash F_{n,k}(\underline{e}^k)$ (where $F(\underline{e}^k)$ is the formula (of $\mathcal{L}$) obtained by replacing free occurrences of x_i by $\underline{e}_i$ for $1 \leq i \leq k$.)

<u>Lemma 2.6</u>: For some constant c there is a procedure which given n, writes down the sequence

$\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}, F_{0,n}, F_{1,n-1}, \ldots, F_{n,0}$ within time

$[T_1((T_2(n+2))^{c(n+1)})]^c$; the length of the sequence is $\le (T_2(n+2))^{c(n+1)}$.

<u>Proof</u>: First compute the sequence $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}$ as in Lemma 2.3. Then for each k, $0 \le k \le n$, and for each $F \in \mathcal{F}_{n-k,k}$, write out the formula $F(\underline{e}^k)$; each of these formulas will be of length $\le L(n,0)$ and there are at most $(T_2(n))\cdot(n+1)$ of them. Then use the decision procedure for $\mathcal{S}$ to decide each of the sentences $F(\underline{e}^k)$, and then consolidate the information on the tape.

The time used in deciding each sentence $F(\underline{e}^k)$ (and returning the head) is $\le 2T_1(L(n,0))$, so the total time used in deciding truth of sentences in $\mathcal{S}$ is $\le (2T_1(L(n,0)))\cdot(T_2(n))\cdot(n+1)$.

So the time to write down $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}$ plus the time used in deciding truth of sentences is $\le$

$[T_1((T_2(n+2))^{c(n+1)})]^c + (2T_1(L(n,0)))\cdot(T_2(n))\cdot(n+1)$ for c as in Lemma 2.3. As in the proof of Lemma 2.3, the remaining time used is polynomial in the space in which $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}$ and all the sentences $F(\underline{e}^k)$ are written, which is $\leq 2(T_2(n+2))^{c(n+1)}$.

$L(n,0) \leq (T_2(n+2))^{c(n+1)}$ and so we calculate that for some other constant c, the sequence $\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}, F_{0,n}, F_{1,n-1}, \ldots, F_{n,0}$ can be computed within time $[T_1((T_2(n+2))^{c(n+1)})]^c$ and its length is $\leq (T_2(n+2))^{c(n+1)}$.

<u>Definition 2.7</u>: For all n, k $\in$ N and every $F \in \mathcal{F}_{n,k}$, define W(F) to be the set such that $F = F_{n,k,W(F)}$.

<u>Remark 2.8</u>: If n,k $\in$ N and $F \in \mathcal{F}_{n+1,k}$ and $F' \in \mathcal{F}_{n,k+1}$ and $\bar{a}_k \in S^k$ such that $\mathcal{S} \vdash F(\bar{a}_k)$, <u>then</u>

$$F' \in W(F) \Leftrightarrow \text{for some } a_{k+1} \in S,\ \mathcal{S} \vdash F'(\bar{a}_{k+1}).$$

We are now ready to consider the structure $\mathcal{S}^* = < S^*, \mathcal{R}_1^*, \ldots, \mathcal{R}_\ell^*, e >$ as defined in Chapter 3. For each n,k $\in$ N let $\equiv_n$ be defined on S^k and on $(S^*)^k$ as in Chapter 2 and let E_n be defined on $(S^*)^k$ as in definition 3.1.5 and let $\mu(n,k)$ be defined as in definition 3.1.2.

<u>Definition 2.9</u>: For each n,k $\in$ N, define

$$\mathcal{F}^*_{n,k} = \{V: N \to \mathcal{F}_{n,k} \mid \text{for all but finitely many } i \in N,\ V(i) = F_{n,k}\}.$$

For every $V \in \mathcal{F}^*_{n,k}$, define $||V|| = \text{Min}\{i \in N|$ for all $j \geq i$, $V(j) = F_{n,k}\}$ = the <u>norm of V</u>. For every $\bar{f}_k \in (S^*)^k$, let $V_{n,\bar{f}_k}$ be the unique member of $\mathcal{F}^*_{n,k}$ such that $V_{n,\bar{f}_k}(i)$ defines $[\bar{f}_k(i)]_n$ for all $i \in N$.

<u>Remark 2.10</u>: For every $V \in \mathcal{F}^*_{n,k}$ there exists some $\bar{f}_k \in (S^*)^k$ such that $V = V_{n,\bar{f}_k}$. Also note that if $k = 0$, we have $|\mathcal{F}_{n,0}| = 1$ and so $|\mathcal{F}^*_{n,0}| = 1$.

<u>Lemma 2.11</u>: Let $n,k \in N$ and $\bar{f}_k, \bar{g}_k \in (S^*)^k$ such that $V_{n,\bar{f}_k} = V_{n,\bar{g}_k}$. Then $\bar{f}_k \underset{n}{\equiv} \bar{g}_k$.

<u>Proof</u>: If $V_{n,\bar{f}_k} = V_{n,\bar{g}_k}$ then for every $i \in N$, $[\bar{f}_k(i)]_n = [\bar{g}_k(i)]_n$, meaning that $\bar{f}_k(i) \underset{n}{\equiv} \bar{g}_k(i)$. This implies that $\bar{f}_k \; E_n \; \bar{g}_k$. By Theorem 3.1.8, $\bar{f}_k \underset{n}{\equiv} \bar{g}_k$. $\square$

<u>Definition 2.12</u>: Let n, $k \in N$ and $V \in \mathcal{F}^*_{n,k}$ and let $F(\bar{x}_k)$ be a formula of q-depth $\leq n$. Let $\bar{f}_k \in (S^*)^k$ be such that $V = V_{n,\bar{f}_k}$. Then we say $V \vdash F$ iff $\mathcal{S}^* \vdash F(\bar{f}_k)$. By Lemma 2.11, this notation is well defined.

<u>Remark 2.13</u>: If $n \in N$ and $V \in \mathcal{F}^*_{n,0}$ and F is a sentence of q-depth $\leq n$,

then $V \vdash F$ iff $\mathcal{S}^* \vdash F$.

Definition 2.14: Let $n,k \in N$. Define the map EX: $\mathcal{F}^*_{n+1,k} \to P(\mathcal{F}^*_{n,k+1})$ (where EX stands for extension and P(A) is the set of subsets of A) as follows: If $V \in \mathcal{F}^*_{n+1,k}$ and $V' \in \mathcal{F}^*_{n,k+1}$, then $V' \in EX(V)$ iff

a) for each $i \in N$, $V'(i) \in W(V(i))$.

and

b) $||V'|| \le ||V|| + \mu(n + k + 1, 0)$.

Lemma 2.15: Let $F(\overline{x}_{k+1})$ be a formula of q-depth $\le n$ and let $V \in \mathcal{F}^*_{n+1,k}$. Then $V \vdash \exists x_{k+1}\ F(\overline{x}_{k+1})^\dagger \Leftrightarrow$ for some $V' \in EX(V)$, $V' \vdash F(\overline{x}_{k+1})$.

Proof of $\Leftarrow$:

Say that V is $V_{n+1,\overline{f}_k}$ where $\overline{f}_k \in (S^*)^k$, and that V' is $V_{n,\overline{g}_{k+1}}$ where $\overline{g}_{k+1} \in (S^*)^{k+1}$ and say that $V_{n,\overline{g}_{k+1}} \in EX(V_{n+1,\overline{f}_k})$ and $V_{n,\overline{g}_{k+1}} \vdash F(\overline{x}_{k+1})$ where q-depth $(F) \le n$.

Let $i \in N$. We have $V_{n+1,\overline{f}_k}(i)$ defines $[\overline{f}_k(i)]_{n+1}$ and $V_{n,\overline{g}_{k+1}}(i)$ defines $[\overline{g}_{k+1}(i)]_n$ and $V_{n,\overline{g}_{k+1}}(i) \in W(V_{n+1,\overline{f}_k}(i))$. By Remark 2.8 we can choose $\overline{f}_{k+1}(i) \in S$ such that $[\overline{f}_{k+1}(i)]_n = [\overline{g}_{k+1}(i)]_n$.

So $V' = V_{n,\overline{f}_{k+1}}$ and $V' \vdash F(\overline{x}_{k+1})$. By Definition 2.12

† where we assume $\exists x_{k+1} F(\overline{x}_{k+1})$ is annotated by $x_1, x_2, \ldots, x_k$.

$\mathcal{S}^* \vdash F(\bar{f}_{k+1})$, and therefore $\mathcal{S}^* \vdash \exists x_{k+1} F(\bar{f}_k, x_{k+1})$. So

$V_{n+1,\bar{f}_k} \vdash \exists x_{k+1} F(\bar{x}_{k+1})$.

Proof of $\Rightarrow$:

Say that $V \in \mathcal{F}^*_{n+1,k}$ such that $V \vdash \exists x_{k+1} F(\bar{x}_{k+1})$ where q-depth $(F) \leq n$. For $i \geq ||V||$, $V(i)$ defines $[e^k]_{n+1}$. Therefore there exists $\bar{f}_k \in (S^*)^k$ such that $\bar{f}_k(i) = e^k$ for $i \geq ||V||$ and such that $V(i)$ defines $[\bar{f}_k(i)]_{n+1}$ for $i \in N$. So $V = V_{n+1,\bar{f}_k}$.

Since $\mathcal{S}^* \vdash \exists x_{k+1} F(\bar{f}_k, x_{k+1})$, we can find $f \in S^*$ such that $\mathcal{S}^* \vdash F(\bar{f}_k, f)$. $\bar{f}_k \; E_{n+1} \; \bar{f}_k$, so the proof of Lemma 3.1.7 shows that we can find $f_{k+1} \in S^*$ such that $(\bar{f}_k, f) \; E_n \; \bar{f}_{k+1}$ and such that $f_{k+1}(i) = e$ whenever $i \geq ||V_{n+1,\bar{f}_k}|| + \mu(n+1, k)$. By Lemma 3.1.8,

$(\bar{f}_k, f) E_n \bar{f}_{k+1} \Rightarrow (\bar{f}_k, f) \underset{n}{\equiv} \bar{f}_{k+1} \Rightarrow \mathcal{S} \vdash F(\bar{f}_{k+1})$.

$V_{n,\bar{f}_{k+1}}$ has norm

$\leq ||V_{n+1,\bar{f}_k}|| + \mu(n+1,k) \leq ||V_{n+1,\bar{f}_k}|| + \mu(n+k+1,0)$, and clearly $V_{n,\bar{f}_{k+1}} \vdash F(\bar{x}_{k+1})$. For each $i \in N$, $V_{n+1,\bar{f}_k}(i)$ defines $[\bar{f}_k(i)]_{n+1}$ and $V_{n,\bar{f}_{k+1}}$ defines $[\bar{f}_{k+1}(i)]_n$ implying (by Remark 2.8) that $V_{n,\bar{f}_{k+1}}(i) \in W(V_{n+1,\bar{f}_k}(i))$. So $V_{n,\bar{f}_{k+1}} \in EX(V_{n+1,\bar{f}_k})$. □

Lemma 2.16: Let F be the formula $Q_1x_1Q_2x_2 \ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free. Let $V_0 \in \mathcal{F}^*_{n,0}$. Then

$$\mathcal{S}^* \vdash F \Leftrightarrow (Q_1V_1 \in EX(V_0))(Q_2V_2 \in EX(V_1))\ldots(Q_nV_n \in EX(V_{n-1}))(V_n \vdash G).$$

Proof: $\mathcal{S}^* \vdash F \Leftrightarrow V_0 \vdash F$. By n applications of Lemma 2.15 we have

$$V_0 \vdash F \Leftrightarrow (Q_1V_1 \in EX(V_0))(V_1 \vdash Q_2x_2Q_3x_3 \ldots Q_nx_nG(\bar{x}_n))$$

$$\ldots \Leftrightarrow (Q_1V_1 \in EX(V_0)) \ldots (Q_nV_n \in EX(V_{n-1}))(V_n \vdash G(\bar{x}_n)).$$ □

Theorem 2.17: Say that $T_1: N \to N$ is such that $TH(\mathcal{S})$ 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 such that $T_2(k + k') \geq M(k,k')$ and $T_2(k) \geq k$ for all $k,k' \in N$. (Assume T_1 is nondecreasing.)

Then there exists an algorithm for deciding $TH(\mathcal{S}^*)$ which operates within time $[T_1((T_2(n + 2))^{dn})]^d$ for some constant d.

Proof: By Theorem 1.4.2 it is sufficient to consider the sentence F of the form $Q_1x_1Q_2x_2 \ldots Q_nx_nG(\bar{x}_n)$ where G is quantifier free and of length at most n log n. The decision procedure proceeds in three steps.

Step 1: Compute the sequence

$\mathcal{F}_{0,n}, \mathcal{F}_{1,n-1}, \ldots, \mathcal{F}_{n,0}, F_{0,n}, F_{1,n-1}, \ldots, F_{n,0}$. By Lemma 2.6

this can be done within time $[T_1((T_2(n+2))^{c(n+1)})]^c$ and the length of the sequence is $\leq (T_2(n+2))^{c(n+1)}$.

Step 2: Compute $\mu(n,0)$ (say, in unary).

$\mu(n,0) = |\mathcal{F}_{0,n}| \cdot |\mathcal{F}_{1,n-1}| \cdot \ldots \cdot |\mathcal{F}_{n-1,1}| \leq (T_2(n))^n$. So $\mu(n,0)$ can be computed and written down using at most $(T_2(n))^n$ more tape squares than those containing the sequence computed in Step 1.

Step 3: Say that $\mathcal{F}^*_{n,0} = \{V_0\}$. We want to decide if

$(Q_1V_1 \in EX(V_0))(Q_2V_2 \in EX(V_1)) \ldots (Q_nV_n \in EX(V_{n-1}))(V_n \vdash G)$.

To do this we have to have a way of writing down representations of members of $\mathcal{F}^*_{n-i,i}$ for $0 \leq i \leq n$. Our convention is as follows: if $V \in \mathcal{F}^*_{j,i}$, then REP(V) is the sequence $V(0), V(1), \ldots, V(||V||)$.

Now if $V_0, V_1, \ldots, V_n$ is a sequence such that $V_{i+1} \in EX(V_i)$ for $0 \leq i < n$ (where $V_0 \in \mathcal{F}^*_{n,0}$), then since $||V_0|| = 0$ and $||V_{i+1}|| \leq ||V_i|| + \mu(n,0)$ we see that $||V_i|| \leq i \cdot \mu(n,0)$ for $0 \leq i \leq n$. So for each Q_i, $1 \leq i \leq n$, set aside $(L(n,0)) \cdot (1+i \cdot \mu(n,0))$ additional tape squares; this is enough space to write down the representation of any member of $\mathcal{F}^*_{n-i,i}$ of norm $\leq i \cdot \mu(n,0)$ (since $L(n,0) \geq L(n-i, i)$).

<u>Claim</u>: There exists a procedure which given

$\mathcal{F}_{n,0}, \mathcal{F}_{n-1,1}, \ldots, \mathcal{F}_{0,n}, F_{n,0}, F_{n-1,1}, \ldots, F_{0,n}, \gamma, \gamma'$ as input, where

$\gamma = \mathrm{REP}(V)$ for some $V \in \mathcal{F}^*_{n-i,i}$, $0 \le i < n$, determines, using

no more space than the input takes up, whether or not $\gamma' \in \mathrm{EX}(\gamma)$.

<u>Proof of Claim</u>: Say that γ is the sequence $\gamma(0), \gamma(1), \ldots, \gamma(J)$

and γ' is the sequence $\gamma'(0), \gamma'(1), \ldots \gamma'(J')$ for some $J, J' \in N$.

We first calculate i (say in unary) such that $\gamma(0)$ has free variables

exactly $x_0, x_1, \ldots, x_i$; that is $\gamma = \mathrm{REP}(V)$ for some $V \in \mathcal{F}^*_{n-i,i}$.

Assuming $i < n$, in order to ensure that $\gamma' \in \mathrm{EX}(\gamma)$ we need only check that

1) γ' is a sequence of members of $\mathcal{F}_{n-i-1,i+1}$, and $J' \le J + \mu(n,0)$.

2) $\gamma'(J') = F_{n-i-1,i+1}$ and if $J' > 0$, then $\gamma'(J'-1) \ne F_{n-i-1,i+1}$.

and

3) for every $j \ge 0$ such that $j \le J$ and $j \le J'$, we have $\gamma'(j) \in W(\gamma(j))$.

For every j such that $J < j \le J'$, we have $\gamma'(j) \in W(\gamma(J))$.

1), 2), and 3) can be checked using no additional space, and so the Claim is proved.

Now to decide F, cycle through each quantifier space appropriately. That is, use the space set aside for Q_1 to cycle through the representatives of members of $EX(V_0)$, obtaining different values for $REP(V_1)$, the space set aside for Q_2 to cycle through the representatives of the members of each $EX(V_1)$, etc. For every particular value of $V_n \in \mathcal{F}^*_{0,n}$ looked at, we have to decide from $REP(V_n)$ if $V_n \vdash G(\bar{x}_n)$. It is sufficient to be able to test if $V_n \vdash G_0(\bar{x}_n)$ for each atomic formula $G_0(\bar{x}_n)$ occurring in G. But recall that for every $i \in N$, $V_n(i)$ is simply a conjunction of atomic formulas or negations of atomic formulas. So $V_n \vdash G_0(\bar{x}_n)$ iff for every formula $F \in \mathcal{F}_{0,n}$ of the sequence $REP(V_n)$, $G_0 \in W(F)$. So $TH(\mathcal{S}^*)$ is decidable. Testing if $V_n \vdash G$ uses only the space on which G and $REP(V_n)$ are written.

The total space used in Steps 2 and 3, including the output of Step 1, is

$$\le \underbrace{(T_2(n+2))^{c(n+1)}}_{\text{output of Step 1}} + \underbrace{(T_2(n))^n}_{\text{Step 2}} + \underbrace{n\cdot(L(n,0))\cdot(1+n\cdot\mu(n,0))}_{\text{Step 3}}$$

(the n log n space on which G is written is insignificant). The time used by Steps 2 and 3 is at most exponential in this bound. Since $\mu(n,0) \le (T_2(n))^n$ and $L(n,0) \le (T_2(n+2))^{c(n+1)}$, we have that the total time used in all three steps is $\leq [T_1((T_2(n+2))^{dn})]^d$ for some constant d (since the length of a sentence is > 0). □

<u>Corollary 2.18</u>: Let $s_1, s_2, c \in N$, $s_1 \geq 1$ and $s_2 \geq 2$, such that TH($\mathfrak{S}$) can be decided within time $\left.2^{2^{\cdot^{\cdot^{2^{cn}}}}}\right\}$ height s_1 and such that $M(n,k) \leq \left.2^{2^{\cdot^{\cdot^{2^{c(n+k)}}}}}\right\}$ height s_2 for all $n,k \in N$.

<u>Then</u> TH($\mathfrak{S}^*$) can be decided within time $\left.2^{2^{\cdot^{\cdot^{2^{c'n}}}}}\right\}$ height $s_1 + s_2$ for some constant c'.

<u>Proof</u>: Immediate from Theorem 2.17. □

Section 3: 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 Chapter 3 and the preceeding parts of this Chapter.

Definition 3.1: Let I be a nonempty set, and let $(\mathcal{S}^{(i)} \mid i \in I)$ be a collection of structures for $\mathcal{L}$, indexed by I; say that $\mathcal{S}^{(i)} = < S^{(i)}, \mathcal{R}_1^{(i)}, \mathcal{R}_2^{(i)}, \ldots, \mathcal{R}_\ell^{(i)}, e^{(i)} >$ for all $i \in I$. Let $D = \{f: I \to \bigcup_{i\in I} S^{(i)} \mid f(i) \in S^{(i)}$ for $i \in I\}$. For each j, $1 \le j \le \ell$, define $\mathcal{R}_j \subseteq D^{t_j}$ as follows: if $\bar{f}_{tj} \in D^{t_j}$, then $\bar{f}_{tj} \in \mathcal{R}_j$ iff $\bar{f}_{tj}(i) \in \mathcal{R}_j^{(i)}$ 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 $(\mathcal{S}^{(i)} \mid i \in I)$ by $\mathrm{STRONG}(\mathcal{S}^{(i)} \mid i \in I) = < D, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_\ell, e >$.

Let $D' \subseteq D$ be the set $\{f \in D \mid$ for all but finitely many $i \in I$, $f(i) = e^{(i)}\}$, and let $\mathcal{R}_j'$ be the relation $\mathcal{R}_j$ restricted to $(D')^{t_j}$ for $1 \le j \le \ell$. Define the weak direct product of the system $(\mathcal{S}^{(i)} \mid i \in I)$ by $\mathrm{WEAK}(\mathcal{S}^{(i)} \mid i \in I) = < D', \mathcal{R}_1', \mathcal{R}_2', \ldots, \mathcal{R}_\ell', e >$.

If I is finite, then $\mathrm{STRONG}(\mathcal{S}^{(i)} \mid i \in I) = \mathrm{WEAK}(\mathcal{S}^{(i)} \mid i \in I)$.

If we take I to be N and $\mathcal{S}^{(i)} = \mathcal{S}$ for some fixed structure $\mathcal{S}$ and all $i \in N$, then we denote $\mathrm{STRONG}(\mathcal{S}^{(i)} \mid i \in N)$ by $\mathcal{S}^{\omega}$ and call it the <u>strong direct power</u> of $\mathcal{S}$; $\mathrm{WEAK}(\mathcal{S}^{(i)} \mid i \in N)$ is $\mathcal{S}^{*}$, the weak direct power of $\mathcal{S}$, which was defined earlier. If $\mathcal{P}$ is a nonempty collection of structures, then <u>$\mathrm{STRONG}(\mathcal{P})$</u> is the class $\{\mathrm{STRONG}(\mathcal{S}^{(i)} \mid i \in I) \mid I \text{ is a set and } \mathcal{S}^{(i)} \in \mathcal{P} \text{ for } i \in I\}$ and <u>$\mathrm{WEAK}(\mathcal{P})$</u> is the class

$$\{\mathrm{WEAK}(\mathcal{S}^{(i)} \mid i \in I) \mid I \text{ is a set and } \mathcal{S}^{(i)} \in \mathcal{P} \text{ for } i \in I\}.$$

Mostowski shows that if $\mathrm{TH}(\mathcal{S})$ is decidable, then $\mathrm{TH}(\mathcal{S}^{\omega})$ is decidable. Feferman and Vaught show that

$\mathrm{TH}(\mathrm{STRONG}(\mathcal{P})) = \mathrm{TH}(\{\mathrm{STRONG}(\mathcal{S}^{(i)} \mid i \in I) \mid I \text{ is a } \underline{\text{finite}} \text{ set and } \mathcal{S}^{(i)} \in \mathcal{P} \text{ for } i \in I\})$,

and if $\mathrm{TH}(\mathcal{P})$ is decidable, then $\mathrm{TH}(\mathrm{STRONG}(\mathcal{P}))$ and $\mathrm{TH}(\mathrm{WEAK}(\mathcal{P}))$ are decidable. We can prove stronger versions of these theorems.

<u>Theorem 3.1</u>: Let $\mathcal{S}$ be a structure and let $M(n,k)$ be defined as before (Definition 2.2.5). Say that $T_1 \colon N \to N$ is such that $\mathrm{TH}(\mathcal{S})$ 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 such that $T_2(k + k') \geq M(k,k')$ and $T_2(k) \geq k$ for all $k, k' \in N$. (Assume T_1 is nondecreasing.)

Then there exists an algorithm for deciding $TH(\mathcal{S}^{\omega})$ which operates within time $[T_1((T_2(n+2))^{dn})]^d$ for some constant d.

<u>Definition 3.2</u>: If $\mathcal{P}$ is a collection of structures, let

$INFSTRONG(\mathcal{P}) = \{STRONG(\mathcal{S}^{(i)} \mid i \in I) \mid I$ is an <u>infinite</u> set and $\mathcal{S}^{(i)} \in \mathcal{P}$ for $i \in I\}$.

Let $INFWEAK(\mathcal{P}) = \{WEAK(\mathcal{S}^{(i)} \mid i \in I) \mid I$ is an <u>infinite</u> set and $\mathcal{S}^{(i)} \in \mathcal{P}$ for $i \in I\}$.

<u>Theorem 3.3</u>: Let $\mathcal{P}$ be a nonempty collection of structures and for each $\mathcal{S} \in \mathcal{P}$, let $M_{\mathcal{S}}(n,k)$ be defined for $\mathcal{S}$ as before (Definition 2.2.5). Say that $T_1: N \to N$ is such that $TH(\mathcal{P})$ 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 such that $T_2(k + k') \geq M_{\mathcal{S}}(k, k')$ and $T_2(k) \geq k$ for all $k, k' \in N$ and all $\mathcal{S} \in \mathcal{P}$. (Assume T_1 is nondecreasing.)

Then there exists algorithms for deciding $TH(STRONG(\mathcal{P}))$, TH(INFSTRONG($\mathcal{P}$)), TH(WEAK($\mathcal{P}$)), and TH(INFWEAK($\mathcal{P}$)) which operate within time $[T_1(2^{(T_2(n+2))^{dn}})]^d$ for some constant d.

It is important to note that in Theorems 2.17, 3.1 and 3.3, the decision procedure that is produced is obtained effectively from the one that is given. For instance, in Theorem 3.3 TH(STRONG($\mathcal{P}$)) is completely determined by TH($\mathcal{P}$).

Now let $\mathcal{P}$ 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($\mathcal{P}$)). TH($\mathcal{P}$) is decidable, and we could have used the technique involved in proving Theorem 3.3 to prove Theorem 3.2.8. Every finitely generated abelian group is isomorphic to a finite direct product of cyclic groups [MB68]. So if $\mathcal{P}'$ is the collection of cyclic group structures, then TH(STRONG($\mathcal{P}'$)) is the first order theory of finitely generated abelian groups. But using results of [Szm55] it can be shown that TH($\mathcal{P}$) = TH($\mathcal{P}'$), and so by Theorem 3.2.8 we see that TH(STRONG($\mathcal{P}'$)) can also be decided within space $2^{2^{2^{cn}}}$ for some constant c.

Chapter 5: 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 $\mathcal{L}$ 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 $\mathcal{L}$ in the structure $< N, \rho >$ in the obvious way; by a P-structure we shall mean a pair $< N, \rho >$ where ρ is a pairing function. Let $\mathcal{P}$ be the collection of all P-structures. Note that although equality is not a formal predicate of $\mathcal{L}$, we can define equality in $\mathcal{P}$ 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($\mathcal{P}$) 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.†

---

† In an earlier version of Tenney's work [Ten72] he presented some elementary recursive algorithms which were supposed to be decision procedures for some theories of pairing functions. We pointed out to him that this was impossible, and he has since written a corrected version [Ten74] in which all the algorithms presented are non-elementary recursive.

The major result of this chapter will be that this is an <u>intrinsic</u> 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.

<u>Definition 1.1</u>: Define $f: N \to N$ by $f(i) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}\Big\}\text{height } i$. That is, $f(0) = 1$ and $f(i + 1) = 2^{f(i)}$ for $i \geq 0$.

<u>Theorem 1.2</u>: Let C be a nonempty collection of P-structures. Then $\mathrm{NTIME}(f(n)) \leq_{p\ell} \mathrm{TH}(C)$.

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.

<u>Corollary 1.3</u>: For some constant $c > 0$, the following is true: Let C be a nonempty collection of P-structures and let $\mathfrak{M}$ be a nondeterministic Turing machine which recognizes TH(C). Then for infinitely many n, there is a sentence in TH(C) which $\mathfrak{M}$ 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.

<u>Definition 1.4</u>: Let n be an integer > 2. Then an <u>n-ling function</u> is a one-one map $\rho: N^n \to N$. $\mathcal{L}_n$, 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 $< N, \rho >$ where ρ is an n-ling function.

<u>Corollary 1.5</u>: Let $n > 2$ and let C be a nonempty collection of n-structures. Then TH(C) has no elementary recursive decision procedure.

<u>Proof</u>: Assume for convenience that $n = 3$; the other cases are handled similarly. If ρ is a 3-ling function 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 $< N, \rho >$ and any $a \in N$,

$$< N, \rho > \vdash F'(a) \Leftrightarrow < N, \rho_a > \vdash F.$$

Now let C' be a nonempty collection of 3-structures and define $C = \{< N, \rho_a > \mid < N, \rho > \in C' \text{ and } a \in N\}$; C is a nonempty collection of P-structures. Let F be a sentence of $\mathcal{L}$. Then $C \vdash F \Leftrightarrow$ for every $< N, \rho > \in C'$ and $a \in N$, $< N, \rho_a > \vdash F \Leftrightarrow$ for every $< N, \rho > \in C'$ and

$a \in N$, $< N,\rho > \vdash F'(a) \Leftrightarrow C' \vdash \forall x F'(x)$. An elementary recursive decision procedure for TH(C') would therefore yield an elementary recursive procedure for TH(C), contradicting Corollary 1.3. $\square$

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 author, Jeanne Ferrante, 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 $\mathcal{S} = \langle N,\rho \rangle$ is a P-structure, define

$$REL(\mathcal{S}) = \{(a_1,a_2) \in N^2 \mid \mathcal{S} \vdash F_{REL}(a_1,a_2)\}.$$

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), ... 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), ... where the numbers grow sufficiently slowly to ensure that $b_i \neq 2i$.) We will now define the sequence $\rho(a_1,b_1), \rho(a_2,b_2)$, ...

Let $n \in N^+$ and assume that $\rho(a_i,b_i)$ has been defined for $0 < i < n$; we now define $\rho(a_n,b_n)$.

<u>Case 1</u>: $(a_n,b_n) \in R$. Define $\rho(a_n,b_n) = 2n$.

<u>Case 2</u>: $b_n = 2i + 1$ and $a_n = 2i$ and $(a_i,b_i) \in R$. Define $\rho(a_n,b_n) = b_n$.

<u>Case 3</u>: 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 $\rho(a_j,b_j)$ and $\rho(a_k,b_k)$ must have been define 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 $S = \{(a,b) \in N^2 \mid b \in N_e \text{ and } a \notin N_e \text{ and } b \neq m\}$. ρ cannot have been defined on any member of S via Case 1 or Case 2, so ρ must have been defined on every member of S via Case 3. Since S is infinite, $\{(a,b) \mid \rho$ is defined on (a,b) 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 $REL(< N,\rho >) = R$. 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 $\rho(a_i,b_i) \in R$. $\square$

<u>Definition 2.3</u>: Let $\mathcal{L}_1$ be the language of the first order predicate

calculus with only a 2-place formal predicate <u>REL</u>. Define the class of structures for $\mathcal{L}_1$, $C = \{< D,R > \mid R \subseteq D^2$ and $D = $ domain $R\}$ (where domain R for a 2-place relation R means $\{a \mid$ for some b, $(a,b) \in R$ or $(b,a) \in R\})$.

<u>Lemma 2.4</u>: (Kalmar [cf. Ch56]). TH(C) is undecidable.†

<u>Theorem 2.5</u>: a) TH($\mathcal{P}$) is undecidable.

b) There exist particular P-structures with undecidable theories.

<u>Proof</u>: If F is a sentence of $\mathcal{L}_1$, let F' be the sentence of $\mathcal{L}$ 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 $\underline{REL}(v_1,v_2)$ 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 $\mathcal{S} \in \mathcal{P}$ and sentence F of $\mathcal{L}_1$, $<$ domain(REL($\mathcal{S}$)), REL($\mathcal{S}$) $> \vdash F \Leftrightarrow \mathcal{S} \vdash F'$.

---

† Actually, the theorem as stated by Church as TH($\{< D,R > \mid R \subseteq D^2\}$) is undecidable, but Lemma 2.4 follows immediately from the proof.

Proof of (a): We will show that $C \vdash F \Leftrightarrow \mathcal{P} \vdash F'$.

$C \vdash F \Rightarrow$ for all $< D, R > \in C$, $< D, R > \vdash F \Rightarrow$
for all $\mathcal{S} \in \mathcal{P}$, $<$ domain(REL($\mathcal{S}$)), REL($\mathcal{S}$) $> \vdash F \Rightarrow$
for all $\mathcal{S} \in \mathcal{P}$, $\mathcal{S} \vdash F' \Rightarrow \mathcal{P} \vdash F'$.

Conversely, $\mathcal{P} \vdash F' \Rightarrow$ for all $\mathcal{S} \in \mathcal{P}$, $\mathcal{S} \vdash F' \Rightarrow$
for all $\mathcal{S} \in \mathcal{P}$, $<$ domain(REL($\mathcal{S}$)), REL($\mathcal{S}$) $> \vdash F \Rightarrow$
(by Lemma 2.2)
for all $< D,R > \in C$ such that $D \subseteq N_e$, $< D,R > \vdash F$.
By the Skolem-Löwenheim theorem [cf. Men64], this implies that for __every__ $< D,R > \in C$, $< D, R > \vdash F$, implying $C \vdash F$. So $C \vdash F \Leftrightarrow \mathcal{P} \vdash F'$.

Hence, a decision procedure for TH($\mathcal{P}$) would yield one for TH(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($< N_e,R >$) (in $\mathcal{L}_1$) is undecidable. (We can, for example, choose R to be an equivalence relation so as to make TH($< N_e,R >$) undecidable, as described in Section 4 of Chapter 2.) By Lemma 2.2 we can find $\mathcal{S} = < N,\rho >$ such that REL($\mathcal{S}$) = R. Then for any sentence F of $\mathcal{L}_1$ we have $< N_e,R > \vdash F \Leftrightarrow \mathcal{S} \vdash F'$. So TH($\mathcal{S}$) is undecidable. □

Remark 2.6: Let $\mathcal{P}' = \{< N,\rho > \in \mathcal{P} \mid \rho \text{ is onto}\}$. The proof of Theorem 2.5 shows that (a) TH($\mathcal{P}'$) is undecidable and (b) TH($\mathcal{S}$) is undecidable for some $\mathcal{S} \in \mathcal{P}'$.

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(C) for any nonempty collection C of P-structures. We shall do this as follows: Let $\mathfrak{M}$ be a nondeterministic Turing machine over the alphabet Σ. Then for every $w \in \Sigma^+$ we will produce a sentence F_w of $\mathcal{L}$, such that for any P-structure $\mathcal{S}$, $\mathcal{S} \vdash F_w \Leftrightarrow \mathfrak{M}$ accepts w within time $f(|w|)$; furthermore, the time it takes to produce F_w will be polynomial in $|w|$, and the space needed will be linear in $|w|$. If $\mathfrak{M}$ operates within time f(n) and C is a nonempty collection of P-structures, then we have $C \vdash F_w \Leftrightarrow \mathfrak{M}$ accepts w within time $f(|w|) \Leftrightarrow \mathfrak{M}$ accepts w, and hence NTIME(f(n)) $\leq_{p\ell}$ TH(C).

The way F_w will "say" that $\mathfrak{M}$ accepts w within time $f(|w|)$ is as follows: We regard the instantaneous configuration of a computation of $\mathfrak{M}$ on w at any time as a string of length $f(|w|)$, and hence the concatenation of the first $(f(|w| + 1) / f(|w|))$ (which is $\geq f(|w|)$) successive instantaneous configurations is a string of length $f(|w| + 1)$. F_w will "say" roughly that there exists such a string of length $f(|w| + 1)$ which contains an accepting configuration. In order to write such sentences as F_w, we will first have to be able to write down formulas of $\mathcal{L}$ 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 [Mey73], 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 $< N,\rho >$ 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.

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 $\mathcal{S} \in \mathcal{P}$ and any $a,b \in N$, $\mathcal{S} \vdash F_\ell(a,b)$ iff $\ell(a) = b$ and $\mathcal{S} \vdash F_r(a,b)$ 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.

<u>Definition 3.1</u>: Let $\preccurlyeq$ be the reverse lexicographical ordering on $\{r,\ell\}^*$. That is, $\sigma_1 \preccurlyeq \sigma_2$ 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 σ_1', σ_2', $\sigma \in \{r,\ell\}^*$.
$\sigma_1 \prec \sigma_2$ means $\sigma_1 \preccurlyeq \sigma_2$ and $\sigma_1 \neq \sigma_2$.

All the properties mentioned in this chapter will be with respect to $\mathcal{P}$.

<u>Definition 3.2</u>: For each $n \in N$, we define the property $ORD_n(x,y_1,y_2)$ as follows: let $< N,\rho > \in \mathcal{P}$, let a, b_1, $b_2 \in N$. Then $< N,\rho > \vdash ORD_n(a,b_1,b_2)$ iff there exists $\sigma_1,\sigma_2 \in \{r,\ell\}^*$ such that

(I) $|\sigma_1| = |\sigma_2| = f(n)$

(II) $\sigma_1 \preccurlyeq \sigma_2$

(III) $\sigma_1 a = b_1$ and $\sigma_2 a = b_2$

<u>Remark 3.3</u>: $< N,\rho > \vdash ORD_n(a,b,b)$ iff for some $\sigma \in \{r,\ell\}^*$, $|\sigma| = f(n)$ and $\sigma a = b$. Clearly $|\{b| < N,\rho > \vdash ORD_n(a,b,b)\}| \leq 2^{f(n)} = f(n+1)$.

<u>Definition 3.4</u>: For $n \in N$ we define the property $FULL_n(x)$ as follows: let $< N,\rho > \in \mathcal{P}$, let $a \in N$. Then $< N, \rho > \vdash FULL_n(a)$ iff

$|\{b \mid \langle N,\rho \rangle \vdash \mathrm{ORD}_n(a,b,b)\}| = f(n+1)$.

<u>Lemma 3.5</u>: 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 $\prec$) 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. <u>Then</u> there exists $a \in N$ such that $\sigma_i a = b_i$ for $1 \le i \le 2^n$.

<u>Proof</u>: (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 $1 \le i \le 2^n$. 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 $1 \le i \le 2^n$. By definition of $\prec$, $\ell\sigma_1, r\sigma_1, \ell\sigma_2, r\sigma_2, \ldots, \ell\sigma_{2^n}, r\sigma_{2^n}$ 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. $\square$

<u>Lemma 3.6</u>: Let $< N,\rho > \in \mathcal{P}$ and let a, $n \in N$. Then the following two statements are equivalent.

(I) $< N,\rho > \vdash FULL_n(a)$

(II) For every $a' \in N$, <u>if</u> $[(ORD_n(a,b,b) \Rightarrow ORD_n(a',b,b))$ for all $b \in N]$

<u>then</u> $[(ORD_n(a',b,b) \Rightarrow ORD_n(a,b,b))$ for all $b \in N]$

<u>Proof</u>:

(I ⇒ II): Say that $FULL_n(a)$ holds in $< N,\rho >$ and that $a' \in N$ has the property that for all $b \in N$, $< N,\rho > \vdash ORD_n(a,b,b) \Rightarrow < N,\rho > \vdash ORD_n(a',b,b)$. We have $f(n+1) = |\{b| < N,\rho > \vdash ORD_n(a,b,b)\}| \leq |\{b| < N,\rho > \vdash ORD_n(a',b,b)\}| \leq f(n+1)$. Hence $< N,\rho > \vdash ORD_n(a'b,b) \Rightarrow < N,\rho > \vdash ORD_n(a,b,b)$.

(II ⇒ I): Say that II is true. Let $A \subseteq N$ be a set of cardinality $f(n+1)$ such that $\{b \mid < N,\rho > \vdash ORD_n(a,b,b)\} \subseteq A$. By Lemma 3.5 we can choose $a' \in N$ such that $\{b \mid < N,\rho > \vdash ORD_n(a',b,b)\} = A$, so $\{b \mid < N,\rho > \vdash ORD_n(a,b,b)\} \subseteq \{b \mid < N,\rho > \vdash ORD_n(a',b,b)\}$. So by II, $\{b \mid < N,\rho > \vdash ORD_n(a,b,b)\} = \{b \mid < N,\rho > \vdash ORD_n(a',b,b)\} = A$. Hence, $|\{b \mid < N,\rho > \vdash ORD_n(a,b,b)\}| = |A| = f(n+1)$ and so $< N,\rho > \vdash FULL_n(a)$. $\square$

<u>Remark 3.7</u>: If $< N,\rho > \vdash FULL_n(a)$, 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 $\{(b_1,b_2) \mid <N,\rho> \vdash ORD_n(a,b_1,b_2)\}$ is a linear ordering on the set $\{b \mid <N,\rho> \vdash ORD_n(a,b,b)\}$ 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 $<N,\rho> \in \mathcal{P}$ and let $a,b_1,b_2 \in N$ Lemma 3.8 says that $<N,\rho> \vdash ORD_{n+1}(a,b_1,b_2)$ 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 $\sigma_1 a = b_1$ and $\sigma_2 a = b_2$ and $\sigma_1 \prec \sigma_2$. To see how this coding is done, examine Figure 1. Every node in the tree in Figure 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 $\sigma_1 \prec \sigma_2$, 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.

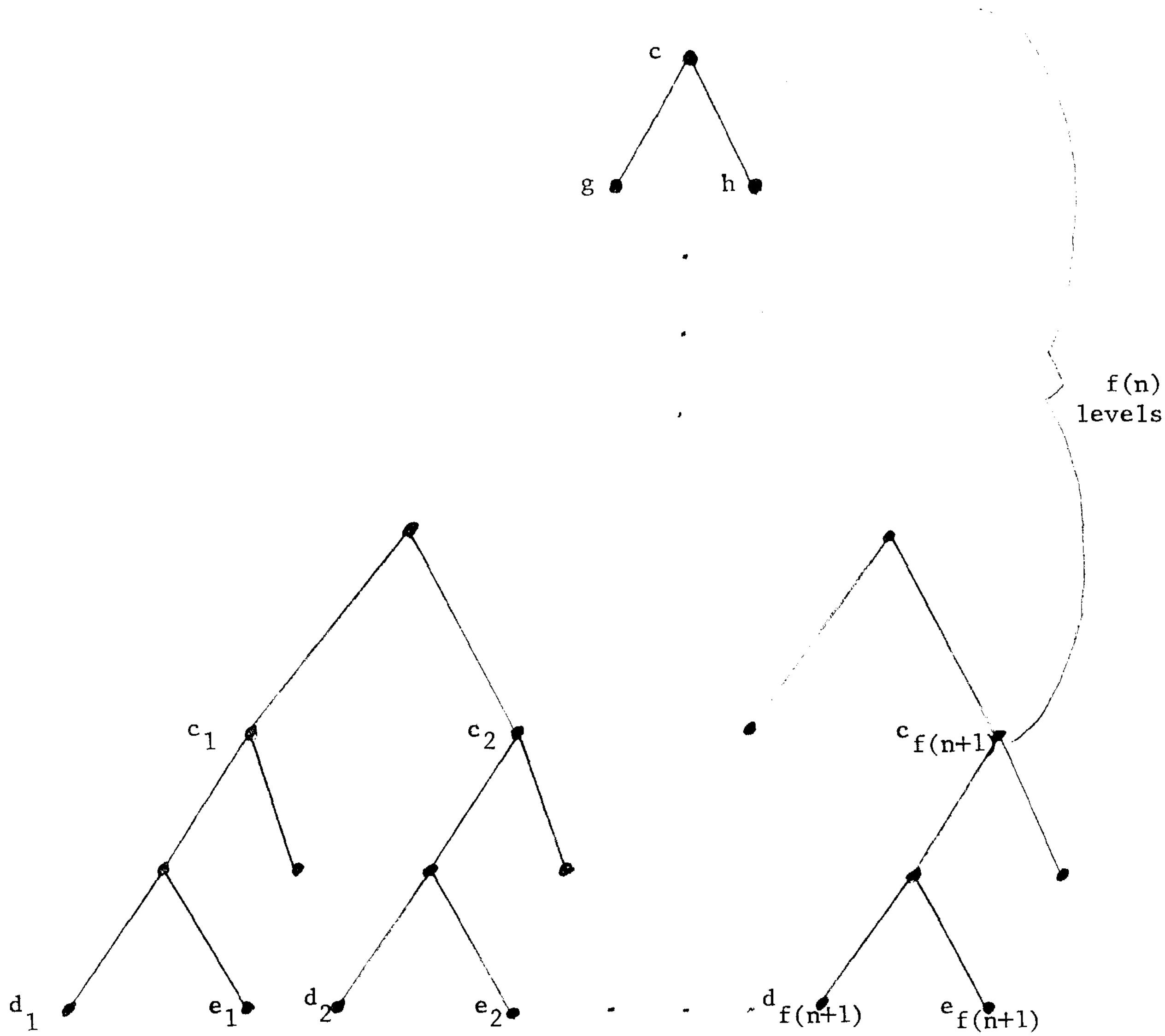

Figure 1:

Illustrating Lemma 5.3.8

<u>Lemma 3.8</u>: Let $< N,\rho > \in P$, let $n \in N$, let $a,b_1,b_2 \in N$. Then $< N,\rho > \vdash ORD_{n+1}(a,b_1,b_2)$ if and only if there exists $c \in N$ such that the following four facts hold.

1) $< N,\rho > \vdash FULL_n(c)$.

Let $\leq$ be the linear order imposed on the set $\{b \mid < N,\rho > \vdash ORD_n(c,b,b)\}$ by ORD_n. Let $c_1, c_2, \ldots, c_{f(n+1)}$ be the elements ordered by $\leq$ listed in increasing order (with respect to $\leq$).

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).

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 $< N,\rho >$, n,a,b_1,b_2.

(If): Say that for some $c \in N$, 1) through 4) hold. If $0 < i \le f(n+1)$, define $\gamma_i = \ell$ if $d_i = \ell d_{i-1}$, and $\gamma_i = r$ if $d_i = rd_{i-1}$ and $d_i \ne \ell d_{i-1}$. If $0 < i \le f(n+1)$, define $\delta_i = r$ if $e_i = re_{i-1}$, and $\delta_i = \ell$ if $e_i = \ell e_{i-1}$ and $e_i \ne re_{i-1}$. 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 $\sigma_1 \prec \sigma_2$. If $\sigma_1 \ne \sigma_2$, then for some i we have $\gamma_j = \delta_j$ when $0 < j < i$, and $\gamma_i \ne \delta_i$. 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 $d_i \ne e_i$, then by 4.2) $d_i = \ell d_{i-1}$. So $\gamma_i = \ell$ and $\sigma_1 \prec \sigma_2$.

(Only if): Say that $< N,\rho > \vdash ORD_{n+1}(a,b_1,b_2)$. Let $\sigma_1,\sigma_2 \in (r,\ell)^*$ be such that $\sigma_1 \prec \sigma_2$ 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 $0 < i \le f(n + 1)$. 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 $0 < i \le f(n + 1)$. 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 $0 < i \le f(n + 1)$. 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 $1 \le i \le f(n + 1)$. 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 $e_i = r\ell c_i$ 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 $e_i = re_{i-1}$, so 4) holds also. □

<u>Lemma 3.9</u>: There exists a sequence of formulas of $\mathcal{L}$ $\underline{\mathrm{ORD}}_0(x,y_1,y_2)$, $\underline{\mathrm{ORD}}_1(x,y_1,y_2)$, ... such that

(I) $\underline{\mathrm{ORD}}_n(x,y_1,y_2)$ defines the property ORD_n for $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{\mathrm{ORD}}_n$ within time polynomial in n and space linear in n.

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 $\{b \mid ORD_n(c,b,b)\}$)†.

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 proportional to n^2. We can, however, use a result due to Fischer and Meyer [cf. FiR74] to obtain (using Lemma 3.8) a formula $\underline{ORD}_n$ of length proportional to n which defines ORD_n for all $n \in N^+$. This result is stated formally and is proven in Appendix 1. Thus by Theorem A.2 of Appendix 1, we can conclude Lemma 3.9. ┐

---

† It is at first difficult to see how to use Lemma 3.8 to write $\underline{ORD}_{n+1}$ using $\underline{ORD}_n$ as a subformula, since the free variables of $\underline{ORD}_n$ are fixed and we might wish to use formulas similar to $\underline{ORD}_n$ but with different free variables at different places in $\underline{ORD}_{n+1}$. One way is by understanding the phrase "using $\underline{ORD}_n$ as a subformula" to mean using formulas like $\underline{ORD}_n$ but with the variable names changed. Another way is by the following trick: Say we have a formula F(x,y) and we wish to have a formula G(y,z) such that F and G define the same property. We can let G be $\forall x_1, \forall x_2((x_1 = y \wedge x_2 = z) \rightarrow \forall x \forall y((x = x_1 \wedge y = y_2) \rightarrow F(x,y)))$.

<u>Corollary 3.10</u>: There exists a sequence of formulas of $\mathcal{L}$,

$\underline{\mathrm{FULL}}_0(x)$, $\underline{\mathrm{FULL}}_1(x)$, ... such that

(I) $\underline{\mathrm{FULL}}_n(x)$ defines the property FULL_n for all $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{\mathrm{FULL}}_n$ within time polynomial in n and within space linear in n.

<u>Proof</u>: Use Lemma 3.6 to express $\underline{\mathrm{FULL}}_n$ using $\underline{\mathrm{ORD}}_n$ for $n \in N$.

<u>Lemma 3.11</u>: There exists a sequence of formulas of $\mathcal{L}$,

$\underline{\mathrm{DIST}}_0(x,y_1,y_2)$, $\underline{\mathrm{DIST}}_1(x,y_1,y_2)$, ... such that

I) If $\mathcal{S} \in \mathcal{P}$ and $n,a,b_1,b_2 \in N$, then $\mathcal{S} \vdash \underline{\mathrm{DIST}}_n(a,b_1,b_2) \Leftrightarrow$

(1) $\mathcal{S} \vdash \mathrm{FULL}_n(a)$

(2) $\mathcal{S} \vdash \mathrm{ORD}_n(a,b_1,b_2)$

(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 $\underline{\mathrm{DIST}}_n$ within time polynomial in n and space linear in n.

<u>Proof</u>: Let $\underline{\mathrm{DIST}}_0$ be $\rho(y_1,y_2) = x \wedge y_1 \neq y_2$.

Let $\mathcal{S} \in \mathcal{P}$, $n \in N^+$, $a, b_1, b_2 \in N$. We wish to say that $\mathcal{S} \vdash \mathrm{FULL}_n(a)$ and $|\{c \in N \mid c \neq b_1, \text{ and } \mathcal{S} \vdash \mathrm{ORD}_n(a,b_1,c) \text{ and } \mathcal{S} \vdash \mathrm{ORD}_n(a,c,b_2)\}| = f(n)$. (This implies that $\mathcal{S} \vdash \mathrm{ORD}_n(a,b_1,b_2)$.) But by Lemma 3.5, this will be true iff $\mathcal{S} \vdash \mathrm{FULL}_n(a)$ and there is some $c' \in N$ such that $\mathcal{S} \vdash \mathrm{FULL}_{n-1}(c')$ and such that for all $c \in N$,

$(\mathcal{S} \vdash \mathrm{ORD}_{n-1}(c',c,c)) \Leftrightarrow (c \neq b_1$ and $\mathcal{S} \vdash \mathrm{ORD}_n(a,b_1,c)$ and $\mathcal{S} \vdash \mathrm{ORD}_n(a,c,b_2))$.

We can therefore write down a formula $\underline{\mathrm{DIST}}_n(x,y_1,y_2)$ for $n \in N$ (by using $\underline{\mathrm{FULL}}_n$, $\underline{\mathrm{ORD}}_n$, $\underline{\mathrm{FULL}}_{n-1}$ and $\underline{\mathrm{ORD}}_{n-1}$) such that (I) and (II) are satisfied. $\square$

<u>Definition 3.12</u>: For all $n \in N$, let $\mathrm{SET}_n(x,y_1,y_2)$ be the property such that for $\mathcal{S} \in \mathcal{P}$ and $n,a,b_1,b_2 \in N$, $\mathcal{S} \vdash \mathrm{SET}_n(a,b_1,b_2) \Leftrightarrow$ $\mathcal{S} \vdash \mathrm{FULL}_n(a)$ and $\mathcal{S} \vdash \mathrm{ORD}_n(a,b_2,b_2)$ and $\mathcal{S} \vdash \mathrm{ORD}_n(b_1,b_2,b_2)$.

<u>Lemma 3.13</u>: Let $\mathcal{S} \in \mathcal{P}$ and let $n,a \in N$ such that $\mathcal{S} \vdash \mathrm{FULL}_n(a)$. Let $A \subseteq \{b \mid \mathcal{S} \vdash \mathrm{ORD}_n(a,b,b)\}$. Then for some $b_1 \in N$, $A = \{b_2 \mid \mathcal{S} \vdash \mathrm{SET}_n(a,b_1,b_2)\}$.

<u>Proof</u>: Say that $\mathcal{S} \vdash \mathrm{FULL}_n(a)$ and $A \subseteq \{b \mid \mathcal{S} \vdash \mathrm{ORD}_n(a,b,b)\}$. Let $A' \subseteq N$ be such that $0 < |A'| \leq f(n+1)$ and $A = A' \cap \{b \mid \mathcal{S} \vdash \mathrm{ORD}_n(a,b,b)\}$. By Lemma 3.5 we can find some $b_1 \in N$ such that $A' = \{b_2 \mid \mathcal{S} \vdash \mathrm{ORD}_n(b_1,b_2,b_2)\}$. Hence, $A = \{b_2 \mid \mathcal{S} \vdash \mathrm{SET}_n(a,b_1,b_2)\}$. $\square$

<u>Lemma 3.14</u>: There exists a sequence of formulas of $\mathcal{L}$, $\underline{\mathrm{SET}}_0(x,y_1,y_2)$, $\underline{\mathrm{SET}}_1(x,y_1,y_2)$, ... such that

(I) $\underline{\mathrm{SET}}_n(x,y_1,y_2)$ defines the property SET_n for $n \in N$.

(II) There is a procedure which given $n \in N^+$ computes $\underline{\mathrm{SET}}_n$ within 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 $n = |\gamma|$, $\mathcal{S} \in \mathcal{P}$, $a,b,c \in N$, then $\mathcal{S} \vdash START_\gamma(a,b,c)$ iff

1) $\mathcal{S} \vdash FULL_n(a)$

Let $\leq$ be the ordering determined on $\{b' \mid \mathcal{S} \vdash ORD_n(a,b',b')\}$ by ORD_n.

Let α be the characteristic sequence (with respect to $\leq$) of the set

$\{b' \mid \mathcal{S} \vdash SET_n(a,b,b')\} = \{b' \mid \mathcal{S} \vdash ORD_n(a,b',b')$ and $\mathcal{S} \vdash ORD_n(b,b',b')\}$,

i.e., α is the binary string of length $f(n + 1)$ determined by b, a and $\mathcal{S}$.

2) $\alpha = \gamma \cdot 0^{f(n)-n} \cdot \delta$ for some $\delta \in \{0,1\}^*$ of length $f(n+1) - f(n)$.

3) c is the n + 1 smallest member (with respect to $\leq$) of the set $\{b' \mid S \vdash ORD_n(a,b',b')\}$.

<u>Lemma 3.16</u>: Let $\gamma \in \{0,1\}^*$, $|\gamma| = n$, and let $i \in \{0,1\}$. Let $S \in P$ and let $a,b,c \in N$. Then $S \vdash START_{\gamma i}(a,b,c) \Leftrightarrow$ the following eight properties hold for some $a',b',c' \in N$.

1) $S \vdash FULL_{n+1}(a)$.

2) $S \vdash FULL_n(a')$

Let $\leq$ be the ordering determined on $\{c'' \mid S \vdash ORD_{n+1}(a,c'',c''\}$ by ORD_{n+1}. Say that $c_1, c_2, \ldots, c_{f(n+1)}$ are the first $f(n+1)$ elements in increasing order (with respect to $\leq$). Let $\leq'$ be the ordering determined on $\{c'' \mid S \vdash ORD_n(a',c'',c'')\}$ by ORD_n.

3) $\{c'' \mid S \vdash ORD_n(a',c'',c'')\} = \{c_1, c_2, \ldots, c_{f(n+1)}\}$. Furthermore, $c_j \leq' c_{j+1}$ for $1 \leq j < f(n+1)$.

4) $S \vdash SET_{n+1}(a,b,c_j) \Leftrightarrow S \vdash SET_n(a',b',c_j)$ for $1 \leq j \leq f(n+1)$.

5) $S \vdash START_\gamma(a',b',c')$.

6) $S \vdash SET_{n+1}(a,b,c') \Leftrightarrow i = 1$.

7) c is the immediate successor of c' in the ordering $\leq$

8) S does not satisfy $SET_{n+1}(a,b,c'')$ for any c'', $c \leq c'' \leq c_{f(n+1)}$.

Proof: (3) says that the ordered set of size $f(n + 1)$ determined by ORD_n and a' (and $\mathcal{S}$) is the same as the first $f(n + 1)$ elements of the ordered set determined by ORD_{n+1} and a. 4) therefore says that the binary sequence of size $f(n + 1)$ determined by SET_n and a' and b' is the same as the first $f(n + 1)$ elements of the binary sequence of size $f(n + 2)$ determined by SET_{n+1} and a and b; 5) and 6) say that this sequence of length $f(n + 1)$ begins with γi and 8) says that the rest of it is 00... . 7) says that c is the n + 2 smallest member of the ordered set determined by ORD_{n+1} and a. □

Lemma 3.17: For every $\gamma \in \{0,1\}^*$ there exists a formula of $\mathcal{L}$ $\underline{START}_\gamma(x,y,z)$ such that

(I) $\underline{START}_\gamma(x,y,z)$ defines the property $START_\gamma$ for $\gamma \in \{0,1\}^*$.

(II) There is a procedure which given $\gamma \in \{0,1\}^+$ computes $\underline{START}_\gamma$ within time polynomial, in $|\gamma|$ and space linear in $|\gamma|$.

Proof: Let $\underline{START}_\lambda(x,y,z)$ be the formula $\exists z'(\rho(z,z') = x \wedge z \neq z')$,

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, and $DIST_{n+1}$ where $n = |\gamma|$. All of these latter properties can be expressed in a fixed way from ORD_n, and so $START_{\gamma i}$ can be expressed in a fixed way from $START_\gamma$ and ORD_n. In order to conclude Lemma 3.17, we have to use a more powerful theorem from Appendix 1 than that used in the proof of Lemma 3.9. 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 A.9 in Appendix 1 (in which $\underline{F}'_0 = \underline{F}'_1$) to conclude Lemma 3.17. $\square$

Remark 3.18: For $\gamma \in \{0,1\}^*$ let $START'_\gamma(x,y)$ be the property such that $\mathcal{S} \vdash START'_\gamma(a,b) \Leftrightarrow$ for some c, $\mathcal{S} \vdash START_\gamma(a,b,c)$. We will really only use the fact that we can write short formulas defining the properties $START'_\gamma$; the reason we have dealt with the more complicated $START_\gamma$ was in order to be able to express these properties inductively.

Section 4: Using Formulas to Simulate Turing Machines

In this section we will use the formulas $\underline{FULL}_n$, $\underline{ORD}_n$, $\underline{DIST}_n$, $\underline{SET}_n$, $\underline{START}_\gamma$ 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(C) for any nonempty collection C of P-structures.

Proof: Let $\mathfrak{M}$ be a nondeterministic Σ-Turing machine which operates within NTIME(f(n)). In order to prove Theorem 1.2 we specify in detail (partly reviewing from Chapter 1) the nature of our Turing machine. The tape alphabet is Σ, $\not{b} \in \Sigma$, and $\mathfrak{M}$ has one head and one tape where the tape is one-way infinite to the right; initially the head is on the leftmost square of the tape and $\mathfrak{M}$ never tries to read off the tape. If $w \in \Sigma^+$, then we input w to $\mathfrak{M}$ by having the initial tape contents be $w\not{b}\not{b}\ldots$. Let the state set of $\mathfrak{M}$ be $\{1,2, \ldots, k\}$ where 1 is the initial state and k is the accepting state. $\mathfrak{M}$ accepts w if there is some computation starting on $w\not{b}\not{b}\ldots$ such that $\mathfrak{M}$ eventually enters state k. Let us assume that after entering state k, $\mathfrak{M}$ thereafter stays in state k without moving the head or changing the tape contents. Since $\mathfrak{M}$ operates within NTIME(f(n)), if $\mathfrak{M}$ accepts w then there is some computation of $\mathfrak{M}$ on w which enters state k within $f(|w|)$ steps and hence without leaving the first $f(|w|)$ tape squares.

Let $w \in \Sigma^+$, $|w| = n$. Let $g(n) = f(n + 1)/f(n)$; $g(n) \geq f(n)$, so if $\mathfrak{M}$ accepts w there is some computation which accepts w within $g(n)$ steps. Consider now a particular computation of $\mathfrak{M}$ on w which goes for $g(n)$ steps without leaving the first $f(n)$ squares. Let $W_i \in \Sigma^*$ of length $f(n)$ be the contents of the first $f(n)$ tape squares at time i (where $\mathfrak{M}$ begins at time 0). Let $U_i \in \{0, 1, 2, \ldots, k\}^*$ of length $f(n)$ be such that $U_i = 0^q\ j\ 0^{f(n)-q-1}$ where at time i, $\mathfrak{M}$ is in state j and the head is pointing at square q (where the leftmost tape square is square 0). Let $W = W_0 \cdot W_1 \cdot \ldots \cdot W_{g(n)-1}$ and $U = U_0 \cdot U_1 \cdots \cdot U_{g(n)-1}$ so that $|W| = |U| = f(n + 1)$. Define the <u>marking string</u> $M \in \{0,1\}^*$ of length $f(n + 1)$ by $M = (1\ 0^{f(n)-1})^{g(n)}$. We will call (W,U,M) the <u>computation triple</u> of the computation (on w). (W,U,M) is an <u>accepting computation triple</u> if k appears in U. Clearly $\mathfrak{M}$ accepts w if and only if there is an accepting computation triple for w.

Let (W,U,M) be a computation triple for $w \in \Sigma^+$, $|w| = n$. For any string γ, let $\gamma(i)$ be the $i + 1$ member of γ so that $W = W(0) \cdot W(1) \cdot \ldots \cdot W(f(n + 1) - 1)$, etc. For every j, $0 \leq j < g(n)$, and every i, $0 \leq i < f(n)$, the values of $W(j \cdot f(n) + i)$ and $U(j \cdot f(n) + i)$ tell us the contents of square i and whether or not the head is pointing at square i (and if so, then the state of $\mathfrak{M}$), at instant j. The rules (of the finite state control) of $\mathfrak{M}$ together with the fact that we only consider computations which do not leave the first $f(n)$ tape squares put constraints on the values of W, U, and M around place $j \cdot f(n) + i + f(n)$ (if $j \cdot f(n) + i + f(n) < f(n + 1)$), depending on the values of W and U at $j \cdot f(n) + i$.

For instance, say that $0 \le k < k + f(n) < f(n + 1)$. Say that $W(k) = 0$ and $U(k) = 5$ and say that if $\mathfrak{M}$ is in state 5 with the head pointing to a square containing 0, then the machine is allowed to print 1 and move the head to the right and transfer to state 7; it is permissible therefore that: $W(k + f(n)) = 1$ and $U(k + f(n)) = 0$ and $U(k + f(n) + 1) = 1$ and $M(k + f(n) + 1) \neq 1$. If $U(k) = 0$, then we must have $W(k + f(n)) = W(k)$. The point is that there are only certain values of $(W(k), U(k), W(k + f(n)), U(k + f(n) - 1), U(k + f(n)),$

$U(k + f(n) + 1), M(k + f(n) + 1))$

which are permissible, i.e., consistent with $\mathfrak{M}$. These ideas are developed rigorously in [Sto74, Section 2.2].

Lemma 4.1: Let $W \in \Sigma^*$, $U \in \{0, 1, 2, \ldots, k\}^*$, $M \in \{0, 1\}^*$ be strings of length $f(n + 1)$. Then (W,U,M) is an accepting computation for $w \in \{0,1\}^*$, $|w| = n$, if and only if

1) $M \in 1 \cdot \{0,1\}^*$ and every contiguous $f(n)$ symbol of M contains exactly one 1.

2) $W \in w \cdot \not{b}^{f(n)-n} \cdot \Sigma^*$.

3) $U \in 1 \cdot 0^{f(n)-1} \cdot \{0,1, \ldots, k\}^*$.

4) For $0 \le i < f(n + 1)$, if $M(i) = 1$, then exactly one of the numbers $U(i), U(i + 1), \ldots, U(i + f(n) - 1)$ is nonzero.

5) For all i such that $1 \leq i < i + f(n) < f(n+1)$, the value of the 7-tuple $(W(i), U(i), W(i + f(n)), U(i + f(n) - 1), U(i + f(n)), U(i + f(n) + 1), M(i + f(n) + 1))$ is <u>consistent with $\mathfrak{M}$</u>.

and

6) U contains an occurrence of k.

<u>Proof</u>: 1) through 6) say roughly that W and U begin with the right configuration, that the transition between any two successive configurations of length f(n) (marked off by M) are permitted by the rules of $\mathfrak{M}$, and that the accepting state appears in U. These are necessary and sufficient conditions for (W,U,M) to be an accepting computation for w. □

<u>Completion of the proof of Theorem 1.2</u>: Let $w \in \Sigma^+$, $|w| = n$. 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 $f(n + 1)$ over $\{0, 1\}$. Every sequence $\gamma_1, \gamma_2, \ldots, \gamma_v$ of v strings over $\{0, 1\}$ of length $f(n + 1)$ represents a string of length $f(n + 1)$ over the alphabet $\{0,1\}^v$ (the set of v-tuples containing just 1 and 0), namely the string γ where

$\gamma(i) = (\gamma_1(i), \gamma_2(i), \ldots, \gamma_v(i))$ for $0 \leq i < f(n + 1)$; if $|\Sigma \cup \{0, 1, 2, \ldots, k\}| = 2^v$, we can think of $\gamma_1, \gamma_2, \ldots, \gamma_v$ as representing a string of length $f(n + 1)$ over the alphabet $\Sigma \cup \{0, 1, \ldots, k\}$ by coding $\Sigma \cup \{0, 1, \ldots, k\}$ into $\{0, 1\}^v$. Say that $\not{b}$ is coded as $\underbrace{(0,0,\ldots,0)}_{v \text{ times}}$. Then the string $w\not{b}^{f(n)-n}$ will be represented by

$\gamma_1 0^{f(n)-n}$, $\gamma_2 0^{f(n)-n}$, ..., $\gamma_v 0^{f(n)-n}$ where $\gamma_i \in \{0,1\}^*$ and is of length n for $1 \le i \le v$.

Therefore 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 exists (W,U,M) satisfying conditions 1) through 6) in Lemma 4.1. That is, for any $\mathcal{S} \in \mathcal{P}$, F_w will be true in $\mathcal{S}$ if and only if $\mathfrak{M}$ accepts w. Hence, if C is a nonempty collection of P-structures, $F_w \in \mathrm{TH}(C) \Leftrightarrow \mathfrak{M}$ accepts w.

So $L(\mathfrak{M}) \le_{p\ell} \mathrm{TH}(C)$. □

References

[Ch56] Church, A., Introduction to Mathematical Logic, I, Princeton, 1956.

[Co73] Cook, S.A.,"A hierarchy for nondeterministic time complexity," J. Comput. Syst. Sci. 7, 4 (Aug. 1973), 343-353.

[Cob72] Cobham, A., "The intrinsic computational difficulty of functions," Proc. Internat. Congr. Logic, Method. and Philos. Sci., 1964, 24-30.

[Coo72] Cooper, C.D., "Theorem-proving in arithmetic without multiplication," Comp. and Logic Group Memo 16,U.C. of Swansea, (April 1972).

[Ehr61] Ehrenfeucht, A, "An application of games to the completeness problem for formalized theories," Fund. Math. 49, 1961, 129-141.

[ELTT65] Ershov, Y. L., Lavrov, I.A., Taimanov, A.D. and Taitslin, M.A., "Elementary theories," Russian Math. Surveys, 20, 1965, 35-105.

[Fer74] Ferrante, J., "Some upper and lower bounds on decision procedures in logic,"Doctoral Thesis, Dept. of Math., M.I.T., to appear 1974.

[FiR74] Fischer, M.J. and Rabin M.O., "Super-exponential complexity of Presburger arithmetic," Proc. AMS. Symp. on Complexity of Real Computational Processes , 1974, to appear; also, MAC Tech. Memo. 43, M.I.T., 1974.

[Fis73] Fischer, M.J., Personal communication.

[FR74] Ferrante, J. and Rackoff, C.,"A decision procedure for the first order theory of real addition with order," SIAM J. for Computing,1974, to appear; also MAC Tech. Memo. 33, (May 1973).

[FV59] Feferman, S. and Vaught, R.L., "The first order properties of products of algebraic systems," Fund. Math.47, 1959, 57-103.

[MB68] Maclane S. and Birkhoff, G., Algebra, Macmillan, 1968.

[Men 64] Mendelson, E., Introduction to Mathematical Logic, Van Nostrand Reinhold, 1964.

[Mey73] Meyer, A.R., "Weak monadic second order theory of successor is not elementary-recursive," Boston Univ. Logic Colloquium Proc., to appear 1974; also MAC Tech. Memo 38, M.I.T., 1973.

[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.

[Opp73] Oppen, D.C., "Elementary bounds for Presburger arithmetic," Proc. 5th ACM Symp. on Theory of Computing, 1973, 34-37.

[Pet67] Péter, R., Recursive Functions, Academic Press, 1967.

[Pre29] Presburger, M., "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt," Comptes Rendus, I Congrès des Math. des Pays Slaves, Warsaw, 1929, 192-201, 395.

[Rob73] 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 (April 1970), 177-192.

[Sei74] Seiferas, J., "Nondeterministic time and space complexity classes," Doctoral Thesis, Dept. of Elect. Eng., M.I.T., to appear 1974.

[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.

[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.

[Sol73] Solovay, R., Personal communication.

[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. Comp. Sci., Cornell Univ., Tech. Report 72-136, 1972.

[Ten74] Tenney, R.L.,"Decidable pairing functions," to appear, 1974.

[Ten74'] Tenney, R.L., "Second-order Ehrenfeucht games and the decidability of an equivalence relation," to appear, 1974.

Appendix 1: Writing Short Formulas for Inductively Defined Properties

Let $\mathcal{L}$ 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 $\mathcal{P}$ be a class of structures for $\mathcal{L}$. Henceforth all properties and all equivalences between formulas of $\mathcal{L}$ will be with respect to $\mathcal{P}$. The purpose of this appendix is to prove that one can construct short formulas defining certain inductively described properties.

Theorem A.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 $\mathcal{L}$ and such that G_{i+1} can be expressed in a fixed way (independent of i) from G_i using the language $\mathcal{L}$, then for every $i > 0$ there is a formula of $\mathcal{L}$ of length proportional to i which defines the property G_i.

We assume for convenience that equality is definable in $\mathcal{P}$, and hence for convenience assume that $v_1 = v_2$ is an atomic formula of $\mathcal{L}$. We also assume that every structure in $\mathcal{P}$ has a domain of cardinality ≥ 2.

Now let $k \in N$ be fixed and let $\mathcal{L}'$ be the language of the first order predicate calculus which is the same as $\mathcal{L}$ except that a k-place formal predicate $\underline{R}$, has been added.†

---

† Two formulas of $\mathcal{L}'$ are <u>equivalent</u> if they are equivalent in any structure obtained by adding to a structure from $\mathcal{P}$ an interpretation for $\underline{R}$.

<u>Definition A.1</u>: Let $\underline{F}(\bar{x}_k)$ be a formula of $\mathcal{L}'$ and let $G(\bar{x}_k)$ be a property. We define an infinite sequence of properties, $G_0(\bar{x}_k)$, $G_1(\bar{x}_k)$, ... as follows: Let $G_0(\bar{x}_k)$ be $G(\bar{x}_k)$. For every $i \in N$ and for every structure $\mathcal{S} \in \mathcal{P}$ with domain S and for every $\bar{a}_k \in S^k$, we say that $\mathcal{S} \vdash G_{i+1}(\bar{a}_k)$ iff $\mathcal{S} \vdash \underline{F}(\bar{a}_k)$ when the formal predicate $\underline{R}$ is interpreted in $\mathcal{S}$ as G_i (restricted to $\mathcal{S}$).

<u>Theorem A.2</u>: Let $\underline{F}(\bar{x}_k)$ be a formula of $\mathcal{L}'$ and let $\underline{G}(\bar{x}_k)$ be a formula of $\mathcal{L}$ defining the property $G(\bar{x}_k)$. Let $G_0(\bar{x}_k)$, $G_1(\bar{x}_k)$, ... be the properties defined in Definition A.1. Then there exists a sequence $\underline{G}_0(\bar{x}_k)$, $\underline{G}_1(\bar{x}_k)$, ... of formulas of $\mathcal{L}$ such that

(I) $\underline{G}_i$ defines the property G_i for each $i \in N$.

(II) There is a procedure which given $i \in N^+$ computes $\underline{G}_i$ within time a fixed polynomial in i and space linear in i.

Theorem A.2 is due to Fischer and Meyer [cf. FiR74], working from earlier ideas of Stockmeyer [SM73]. A key part of the proof will be Lemma A.3.

Lemma A.3: Let $\underline{F}$ be a formula of $\mathcal{L}'$. Then there exists a formula $\underline{F}'$ of $\mathcal{L}'$ equivalent to $\underline{F}$ such that $\underline{F}'$ has exactly one occurrence of the predicate letter $\underline{\mathfrak{R}}$; this occurs in an atomic formula in which all the k formal variables are distinct.

Proof: Let $\underline{F}$ be a formula of $\mathcal{L}'$. Since any formula of $\mathcal{L}'$ can trivially be extended to an equivalent one with at least one occurrence of $\underline{\mathfrak{R}}$, assume that $\underline{F}$ contains at least one occurrence of $\underline{\mathfrak{R}}$. Assume $\underline{F}$ is in prenex normal form so that $\underline{F}$ looks like $Q_1v_1Q_2v_2 \ldots Q_Jv_J\underline{A}$ where $\underline{A}$ is a quantifier free formula containing $m \geq 1$ occurrences of the symbol $\underline{\mathfrak{R}}$ 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 $\underline{\mathfrak{R}}$ occurs, from left to right are $\underline{\mathfrak{R}}(v_{11}, v_{12}, \ldots, v_{1k}), \underline{\mathfrak{R}}\ (v_{21}, v_{22}, \ldots, v_{2k}), \ldots, \underline{\mathfrak{R}}(v_{m1}, v_{m2}, \ldots, v_{mk})$ 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 A. Let A' be the formula obtained from A by replacing $\underline{\mathfrak{R}}(v_{i1}, v_{i2}, \ldots, v_{ik})$ by $y_i = y_i'$ for $1 \leq i \leq m$. Since in every structure of $\mathcal{P}$ 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 A is equivalent to

$$\exists y_1 \exists y_1' \exists y_2 \exists y_2' \ldots \exists y_m \exists y_m' (A' \wedge \bigwedge_{1 \le i \le m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]).$$

Now let $y, y', z_1, z_2, \ldots, z_k$ be distinct formal variables not occurring in $\bigwedge_{1 \le i \le m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]$.

$\bigwedge_{1 \le i \le m} [(y_i = y_i') \leftrightarrow \underline{R}(v_{i1}, v_{i2}, \ldots, v_{ik})]$ is equivalent to

$$\forall y \forall y' \forall z_1 \ldots \forall z_k [(\bigvee_{1 \le i \le m} (y = y_i \wedge y' = y_i' \wedge z_1 = v_{i1} \wedge z_2 = v_{i2} \wedge \ldots \wedge z_k = v_{ik})) \rightarrow$$

$$((y = y') \leftrightarrow \underline{R}(z_1, z_2, \ldots, z_k))].$$

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 $\underline{R}(\bar{z}_k)$. □

<u>Definition A.4</u>: Let $\underline{F}(\bar{x}_k)$ be a formula of $\mathcal{L}$ 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, $1 \le i \le k$, replace v by x_i; if $v = x_i$ for some i,

$1 \le i \le k$, replace v by z_i.

<u>Definition A.5</u>: If <u>F</u> is a formula of $\mathcal{L}$, define the <u>size</u> of <u>F</u>, s(<u>F</u>), to be the length of <u>F</u> 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.

<u>Lemma A.6</u>: Let $\underline{F}(\bar{x}_k)$ and $\underline{F}^{(\bar{z}_k|\bar{x}_k)}(\bar{z}_k)$ be as in Definition A.4. <u>Then</u> $s(\underline{F}) = s(\underline{F}^{(\bar{z}_k|\bar{x}_k)})$, and $\underline{F}(\bar{x}_k)$ and $\underline{F}^{(\bar{z}_k|\bar{x}_k)}(\bar{z}_k)$ define the same property.

<u>Proof of Theorem A.2</u>: Let $\underline{F}(\bar{x}_k)$ be a formula of $\mathcal{L}'$ and let $\underline{G}(\bar{x}_k)$ be a formula of $\mathcal{L}$ defining the property $G(\bar{x}_k)$. By Lemma A.3 assume that <u>F</u> contains exactly one occurrence of $\underline{\mathcal{R}}$; the proof of Lemma A.3 assures us in fact that we can insist that the atomic formula in which $\underline{\mathcal{R}}$ occurs is $\underline{\mathcal{R}}(\bar{z}_k)$ 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 $\underline{G}_0(\bar{x}_k)$, $\underline{G}_1(\bar{x}_k)$, ... of formulas of $\mathcal{L}$ as follows. Let $\underline{G}_0$ be <u>G</u>. For all $i \in N$, let $\underline{G}_{i+1}$ be the formula obtained by substituting $\underline{G}_i^{(\bar{z}_k|\bar{x}_k)}$ for $\underline{R}(\bar{z}_k)$ in $\underline{F}$. It is easy to see by induction (using Lemma A.6) that $\underline{G}_i(\bar{x}_k)$ defines $G_i(\bar{x}_k)$ for each $i \in N$.

For $c_0 = |\underline{F}|$ we have $s(\underline{G}_{i+1}) \leq c_0 + s(\underline{G}_i^{(\bar{z}_k|\bar{x}_k)}) = c_0 + s(\underline{G}_i)$ 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 $|\underline{G}_i| \leq c_1 \cdot s(\underline{G}_i) \leq c_1 \cdot (s(\underline{G}) + i \cdot c_0) \leq c \cdot i$ for $i \in N^+$ 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. □

<u>Remark A.7</u>: Theorem A.2 can be improved in a number of ways. Firstly, we can obtain our result even without the restrictions that equality be definable in $\mathcal{P}$ and that every structure in $\mathcal{P}$ have a domain of cardinality ≥ 2. In addition, using a trick suggested by Solovay [Sol73] we can obtain the same result even if our language of the predicate calculus doesn't contain $\leftrightarrow$.

Theorem A.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 $\mathcal{L}''$ be the language of the first order predicate calculus which is the same as $\mathcal{L}$ except that we have added two new formal k-place predicates: $\underline{R}$ and $\underline{R}'$ for some fixed $k \in N$.

<u>Definition A.8</u>: Let $\underline{F}_0(\overline{x}_k)$, $\underline{F}_1(\overline{x}_k)$, $\underline{F}_0'(\overline{y}_k)$, $\underline{F}_1'(\overline{y}_k)$ be formulas of $\mathcal{L}''$. Let $G(\overline{x}_k)$ and $G'(\overline{y}_k)$ be properties. For every $\gamma \in \{0,1\}^*$ we let $G_\gamma(\overline{x}_k)$ and $G_\gamma'(\overline{y}_k)$ 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

$\mathcal{S} \in \mathcal{P}$ with domain S and every $\overline{a}_k \in S^k$ we say

$\mathcal{S} \vdash G_{\delta i}(\overline{a}_k)$ (where $i \in \{0,1\}$) iff $\mathcal{S} \vdash \underline{F}_i(\overline{a}_k)$ when $\underline{\mathcal{R}}$ is interpreted as G_δ (restricted to $\mathcal{S}$) and $\underline{\mathcal{R}}'$ is interpreted as G_δ'; we say $\mathcal{S} \vdash G_{\delta i}'(\overline{a}_k)$ iff $\mathcal{S} \vdash \underline{F}_i'(\overline{a}_k)$ when $\underline{\mathcal{R}}$ is interpreted as G_δ and $\underline{\mathcal{R}}'$ is interpreted as G_δ'.

<u>Theorem A.9</u>: Let $\underline{F}_0,\underline{F}_1,\underline{F}_0',\underline{F}_1'$ be formulas of $\mathcal{L}''$ and let $\underline{G}(\overline{x}_k)$, $\underline{G}'(\overline{y}_k)$ be formulas of $\mathcal{L}$ defining, respectively, the properties $G(\overline{x}_k)$ and $G'(\overline{y}_k)$. For each $\gamma \in \{0,1\}^*$, let $G_\gamma(\overline{x}_k)$ and $G_\gamma'(\overline{y}_k)$ be as in Definition A.8. Assume that for any $\mathcal{S} \in \mathcal{P}$, the relations obtained by restricting G_γ and G_γ' to $\mathcal{S}$ are both nonempty. Then for each $\gamma \in \{0,1\}^*$ there exist formulas $\underline{G}_\gamma(\overline{x}_k)$, $\underline{G}_\gamma'(\overline{y}_k)$ 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 $|\gamma|$ and space linear in $|\gamma|$.

Proof: The basic idea of this proof 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 $\mathcal{S} \in \mathcal{P}$ with domain S and $\bar{a}_k, \bar{b}_k \in S^k$, then we say $\mathcal{S} \vdash H_\gamma(\bar{a}_k,\bar{b}_k) \Leftrightarrow \mathcal{S} \vdash G_\gamma(\bar{a}_k)$ and $\mathcal{S} \vdash G'_\gamma(\bar{b}_k)$. 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 $\mathcal{S} \in \mathcal{P}$ with domain S and any $a_k \in S^k$,

$\mathcal{S} \vdash G_\delta(\bar{a}_k) \Leftrightarrow$ for some $\bar{b}_k \in S^k$, $\mathcal{S} \vdash H_\delta(\bar{a}_k,\bar{b}_k)$, and

$\mathcal{S} \vdash G'_\delta(\bar{a}_k) \Leftrightarrow$ for some $\bar{b}_k \in S^k$, $\mathcal{S} \vdash H_\delta(\bar{b}_k,\bar{a}_k)$,† we see that G_δ and G'_δ can be expressed from H_δ.

Proceeding more formally, let $\mathcal{L}''_0$ be the language of the first order

---

† since the relations obtained by restricting G_δ and G'_δ to $\mathcal{S}$ are nonempty.

predicate calculus obtained from $\mathcal{L}$ 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 $\underline{\mathcal{F}}_i(\bar{x}_k)$ be the formula of $\mathcal{L}''_0$ 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 $\underline{\mathcal{R}}(\bar{v}_k)$ every time $\underline{\mathcal{R}}$ 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 $\underline{\mathcal{R}}'(\bar{v}_k)$ every time $\underline{\mathcal{R}}'$ appears; obtain $\underline{\mathcal{F}}'_i(\bar{y}_k)$ 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 $\mathcal{L}''_0$ as $\underline{\mathcal{F}}_i(\bar{x}_k) \wedge \underline{\mathcal{F}}'_i(\bar{y}_k)$. One can now see that for $\delta \in \{0,1\}^*$, $i \in \{0,1\}$, $\mathcal{S} \in \mathcal{P}$ with domain S, and $\bar{a}_k,\bar{b}_k \in S^k$, we have $\mathcal{S} \vdash G_{\delta i}(\bar{a}_k) \Leftrightarrow \mathcal{S} \vdash \underline{\mathcal{F}}_i(\bar{a}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to $\mathcal{S}$, $\mathcal{S} \vdash G'_{\delta i}(\bar{b}_k) \Leftrightarrow \mathcal{S} \vdash \underline{\mathcal{F}}'_i(\bar{b}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to $\mathcal{S}$, and therefore $\mathcal{S} \vdash H_{\delta i}(\bar{a}_k,\bar{b}_k) \Leftrightarrow \mathcal{S} \vdash \underline{T}_i(\bar{a}_k,\bar{b}_k)$ when $\underline{U}$ is interpreted as H_δ restricted to $\mathcal{S}$.

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 T_0 and 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 $\mathcal{L}''_0$ 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 $\mathcal{L}$ 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 A.2, we can check that $|\underline{H}_\gamma| \le c|\gamma|$ for $|\gamma| > 0$. 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 Lemma A.9 hold. $\square$

Appendix 2: Notation

$\emptyset$	The empty set.
A-B	$\{x \mid x \in A$ and $x \notin B\}$ (set difference).
P(A)	The set of all subsets of the set A.
$\|A\|$	The cardinality of the set A.
$\|\alpha\|$	The length of the string α.
$\|n\|$	The absolute value of the integer n.
Σ^*	The set of all strings over Σ if Σ is a finite alphabet.
λ	The empty string.
Σ^+	$\Sigma^* - \{\lambda\}$.
$\alpha\gamma$ or $\alpha\cdot\gamma$	Concatenation of the strings α and γ.
$\alpha(i)$	The $i + 1$ (from the left) member of the string α.
α^k	If α is a string, then $\alpha\cdot\alpha\cdot \ldots \cdot\alpha$ (k times) if $k > 0$ and λ if $k = 0$.
S^k	If S is a set, then $S \times S \times \ldots \times S$ (k times) if $k > 0$ and $\emptyset$ if $k = 0$.
$\bar{a}_k$	$(a_1, a_2, \ldots, a_k)$ if $k > 0$ and $\emptyset$ if $k = 0$.
e^k	$(e, e, \ldots, e)$ (length k) if $k > 0$ and $\emptyset$ if $k = 0$.
$\underline{e}^k$	$(\underline{e}, \underline{e}, \ldots, \underline{e})$ (length k) if $k > 0$ and $\emptyset$ if $k = 0$.
Max A	Maximum of the set A.
Min A	Minimum of the set A. Min A = 0 if $A = \emptyset$.
log n	$\log_2 n$.
f is one-one	$f(a) = f(b) \Rightarrow a = b$.
f is onto B	For all $b \in B$ there is some a such that $f(a) = b$.
N	The set of nonnegative integers.
Z	The set of integers.
R	The set of real numbers.

$\mathbb{N}$	The structure $< N, +, \leq, 0 >$.
$\mathbb{Z}$	The structure $< Z, +, \leq, 0 >$.
$\mathbb{R}$	The structure $< R, +, \leq, 0 >$.
$\mathcal{S}$	A logical structure with domain S.
$\mathcal{S}^*$	The weak direct power of $\mathcal{S}$.
S^*	The domain of $\mathcal{S}^*$.
$\mathcal{S}^\omega$	The strong direct power of $\mathcal{S}$.
$\underset{n}{\equiv}$	The Ehrenfeucht equivalence relation (definition 2.2.1).
$\underset{n}{=}$	Equal up to size n (definition 2.3.2).
$\approx$ mod k	Equivalence mod k.
M(n,k)	The number of $\underset{n}{\equiv}$ equivalence classes on S^k.
TH($\mathcal{S}$)	The set of sentences true in $\mathcal{S}$.
TH($\mathcal{P}$)	The set of sentences true in every structure in the set $\mathcal{P}$.
$\mathcal{S} \models F$	F is true in $\mathcal{S}$.
$\lVert a \rVert$	The norm of the element a of a logical structure.
FAG	Finite abelian group.
$\mathfrak{M}$	A (one tape, one head) Turing machine.
L($\mathfrak{M}$)	A language recognized by $\mathfrak{M}$.
$\leq_{p\ell}$	Polynomial time, linear space reducibility.
DTIME(f(n))	The set of languages recognizable within time f(n) by a deterministic Turing machine.
NTIME(f(n))	The set of languages recognizable within time f(n) by a non-deterministic Turing machine.
DSPACE(f(n))	The set of languages recognizable within space f(n) by a deterministic Turing machine.
NSPACE(f(n))	The set of languages recognizable within space f(n) by a non-deterministic Turing machine.

Biographical Note

The author was born on November 26, 1948, in New York City. He attended Forest Hills High School from 1963 to 1966, where he became captain of the math team as well as a member in certain of those strange high school honor societies.

He entered M.I.T. in 1966, and in 1972 he received an S.B. in mathematics and an S.M. in electrical engineering. During that time he was elected to Sigma Xi and awarded an N.S.F. fellowship; for the last two years he has been a research assistant at Project MAC.

He plans to spend the next year as a research associate at I.R.I.A., in France.