Chapter XIII
Monadic Second-Order Theories

by Y. Gurevich

In the present chapter we will make a case for the monadic second-order logic (that is to say, for the extension of first-order logic allowing quantification over monadic predicates) as a good source of theories that are both expressive and manageable. We will illustrate two powerful decidability techniques here—the one makes use of automata and games while the other uses generalized products *à la* Feferman–Vaught. The latter is, of course, particularly relevant, since monadic logic definitely appears to be the proper framework for examining generalized products.

Undecidability proofs must be thought out anew in this area; for, whereas true first-order arithmetic is *reducible* to the monadic theory of the real line R, it is nevertheless not *interpretable* in the monadic theory of R. Thus, the examination of a quite unusual undecidability method is another subject that will be explained in this chapter. In the last section we will briefly review the history of the methods thus far developed and give a description of some further results.

1. Monadic Quantification

Monadic (second-order) logic is the extension of the first-order logic that allows quantification over monadic (unary) predicates. Thus, although binary, ternary, and other predicates, as well as functions, may appear in monadic (second-order) languages, they may nevertheless not be quantified over.

1.1. Formal Languages for Mathematical Theories

We are interested less in monadic (second-order) logic itself than in the applications of this logic to mathematical theories. We are interested in the monadic formalization of the language of a mathematical theory and in monadic theories of corresponding mathematical objects. Before we explore this line of thought in more detail, let us argue that formalizing a mathematical language—not necessarily in monadic logic, but rather in first-order logic or in any other formal logic for that matter—can be useful.

We begin by observing that the first-order Zermelo–Fraenkel set theory stands as a very important case in point, since it provides the most popular way to avoid known paradoxes in set theory. Another excellent example is related to the Lefschetz principle in algebraic geometry. This principle asserts that any algebraic statement that is true for the field of complex numbers is also true for any algebraically closed field of characteristic 0. Tarski proved a meaningful exact version of the Lefschetz principle, namely, that all algebraically closed fields of characteristic 0 are elementarily equivalent.

The task of classifying all mathematical structures of a kind up to isomorphism (or homeomorphism, etc.) may be impossible. For example, nobody can classify all abelian groups up to isomorphism. Formalizing (a portion of) the language may allow classification by properties that are expressible in the formal language. Szmielew [1955] did, in fact, classify all abelian groups up to elementary equivalence. The classification of structures up to indistinguishability in a formal language may indeed be a reasonable alternative to the original classification problem provided, of course, that the formal language expresses enough of the relevant mathematics.

Another impossible task is that of learning everything about a specific structure. For example, nobody can learn all about the binary tree of words in a two-letter alphabet. Formalizing (a portion of) the language may enable us to learn all about the structure that is capable of being expressed in the formal language. It is, of course, a reasonable approach if the formal language is sufficiently rich. Indeed, Rabin [1969] has constructed an algorithm which is capable of recognizing the true statements in the very expressive monadic (second-order) language of the binary tree with two successor functions.

The study of mathematical structures in a formal language may, of course, degenerate to a mere logic exercise if the language is not sufficiently expressive. For example, imagine studying first-order properties of dense linear orders. On the other hand, the study itself may become intractable if the language is over-expressive. For instance, imagine studying second-order properties of dense linear orders. A good formal language has to meet two conflicting demands. It must express an interesting portion of the relevant mathematics, and it must also provide a manageable theory. One of the main aims of this chapter is to demonstrate that the monadic (second-order) logic is a good source of expressive and manageable theories.

1.2. Ordered Abelian Groups and Restricted Monadic Quantification

I began to think in terms of monadic logic while I was working on ordered abelian groups. The original problem I faced was the decision problem for the elementary theory of such groups—a question of Tarski. It appeared, however, that monadic logic gives a better formalization of the informal theory of o.a. groups. The story was an important lesson for me and I will briefly relate it to you.

An o.a. group is a group and a chain, the two structures being connected by the law

$$x < y \to x + z < y + z.$$

Here is a particular example: the additive group of complex numbers ordered thus:

$$a + bi < c + di \quad \text{iff} \quad b < d, \text{or } b = d \text{ and } a < c.$$

The subgroups of an ordered abelian group that form intervals are called *convex subgroups*. For example, the real numbers form a convex subgroup in the o.a. group of complex numbers just described. It is easy to verify that the convex subgroups of any o.a. group are linearly ordered by inclusion. Before proceeding, we should point out that throughout this chapter the terms *chain* and *linear ordering* will be used interchangeably.

The elementary first-order theory of o.a. groups was shown to be decidable in Gurevich [1964], there was proven that two o.a. groups are elementarily equivalent iff their chains of definable convex subgroups with some definable weights are elementarily equivalent. Of course, in that study most of the informal theory of o.a. groups was left aside, such theory tending as it does to deal with convex subgroups. In particular, we note that the o.a. group of complex numbers described above is elementarily equivalent to the naturally ordered additive group of real numbers, although only one of these o.a. groups has a non-trivial convex subgroup.

The elementary language of o.a. groups was expanded in Gurevich [1977a] by adding quantifiable variables that range over arbitrary convex subgroups, and the expanded theory of such groups was there proven to be decidable. You might suspect that the expanded theory is decidable because the expansion did not greatly increase the expressive power, and that the restricted monadic quantification can be essentially eliminated. However, this is not at all the case! Not only does the expansion considerably increase the expressive power, but it is also the *elementary* quantification that can be essentially eliminated in the expanded theory. Two o.a. groups are equivalent in the expanded language iff their chains of convex subgroups with some definable weights are elementarily equivalent. Moreover, the decision procedure is clearer and less cumbersome in the case of the expanded theory. Thus, in the case of o.a. groups, the monadic logic really does provide a better formalization.

Not too much work has yet been done on this kind of algebraic application of *restricted* monadic quantification. In this connection, the reader might consult Kokorin–Pinus [1978], an informative, although somewhat biased, survey. The remainder of this chapter is devoted mainly to *unrestricted* monadic quantification, an area in which some very impressive progress has been made. In the original papers, many of the results on unrestricted monadic quantification are accompanied by restricted monadic quantification results. The work on unrestricted monadic quantification seems to be a natural step in the development of ways that are capable of dealing with the presumably more applicable restricted monadic quantification.

1.3. Monadic Languages

The monadic (second-order) logic is the fragment of the full second-order logic allowing quantification only over elements and monadic predicates. One way to define the monadic version of an elementary language L is to augment L by a sequence of quantifiable set variables and by new atomic formulas $t \in X$, where t is an elementary term and X is a set variable. The intended interpretation here is that $\in$ is the membership relation and the set variables range over all subsets of a structure for L. Observe, however, that in the case of restricted monadic quantification the set variables range only over special subsets; that is to say, they only range over subgroups, or normal subgroups, etc.

The following proposition shows that the monadic theory of a structure may easily be intractable.

1.3.1 Proposition. *Let P be a ternary predicate on a non-empty set S. Suppose that, for every $x, y \in S$, there is $z \in S$, with $(x, y, z) \in P$, and for every $z \in S$ there is at most one pair (x, y) with $(x, y, z) \in P$; such P may be called a pairing predicate. Then the true (full) second-order theory of S is interpretable in the monadic theory of (S, P).*

Proof. The proof is quite clear. First, we code ternary, quaternary, etc., predicates by binary ones. That done, we then code a binary predicate B by a monadic predicate $\{z\colon \text{there is a pair } (x, y) \text{ in } B \text{ with } (x, y, z) \in P\}$. ☐

We will be interested in the monadic theories that are not able to express pairing such as monadic theories of (linear) orders, monadic theories of trees, etc. In these theories it is useful in many cases for us to rid ourselves entirely of elementary variables by coding the original structure on singleton sets. For example, we consider the monadic language of order as the (formally) first-order language whose vocabulary consists of the binary predicate symbols $\subseteq$ and $\leq$. Every chain (that is, every linearly ordered set) gives a standard model: the variables range over all subsets of the chain, $\subseteq$ is the usual inclusion, and $X < Y$ means that there are elements $x < y$ with $X = \{x\}$, $Y = \{y\}$. The (formally) first-order theory of these standard models is, by the definition, the monadic theory of linear order.

2. The Automata and Games Decidability Technique

The first technique for dealing with nontrivial monadic theories originated in the theory of finite automata. In Section 2.1 we will demonstrate this technique on an easy example of the monadic theory of finite chains. Section 2.2 is devoted to the monadic theory of the chain ω of natural numbers, while Section 2.3 is devoted to the central result proven by the technique which is decidability of the monadic theory of the binary tree.

2.1. Monadic Theory of Finite Chains

We define the *monadic language of one successor* as formally the first-order language with binary predicates $\subseteq$ and SUC. It is convenient here for us to view a finite chain as a model for the monadic language of one successor, that is, the variables range over the subsets of the chain, $\subseteq$ is ordinary inclusion, and SUC(X, Y) means that there are points x, y such that $X = \{x\}$, $Y = \{y\}$, and y is the successor of x. The linear order (on singleton sets) is then easily definable.

Throughout this section Σ is an alphabet (all of our alphabets are finite and are not empty). A Σ-*automaton* is a quadruple $A = (S, T, s_{\text{in}}, F)$, where S is the finite *set of states*, $T \subseteq S \times \Sigma \times S$ is the *transition table*, $s_{\text{in}} \in S$ is the *initial state*, and $F \subseteq S$ is the *set of final* (or *accepting*) *states*. A is generally a non-deterministic automaton. It is *deterministic* if T is a total function from $S \times \Sigma$ to S.

A *run* of the Σ-automaton A on a word $\sigma_1 \ldots \sigma_l$ in Σ is a sequence $s_1 \ldots s_l$ of states such that $(s_{\text{in}}, \sigma_1, s_1) \in T$ and every $(s_i, \sigma_{i+1}, s_{i+1}) \in T$. The automaton *accepts* $\sigma_1, \ldots, \sigma_l$ if there is a run $s_1 \ldots s_l$ on this word with $s_l \in F$.

2.1.1 Theorem. *There is an algorithm that, given an alphabet Σ and a Σ-automaton A, constructs a deterministic Σ-automaton accepting exactly the words accepted by A.*

Proof. See any standard text in automata theory or, for the original proof Rabin–Scott [1959]. □

2.1.2 Theorem. *There is an algorithm that, given an alphabet Σ and a Σ-automaton A, decides whether A accepts at least one non-empty word.*

Proof. Let $A = (S, T, s_{\text{in}}, F)$. First, we construct a singleton alphabet $\Sigma' = \{a\}$ and a Σ'-automaton $A' = (S, T', s_{\text{in}}, F)$ that accepts a non-empty word iff A accepts a non-empty word. Set

$$T' = \{s_1 a s_2 : s_1 \sigma s_2 \in T, \text{ for some } \sigma \in \Sigma\}.$$

Second, we use the algorithm of Theorem 2.1.1 to construct a deterministic Σ'-automaton A'' that accepts exactly the words accepted by A'.

Third, let n be the number of states of A''. Consider now the unique run $s_1 \ldots s_{n+1}$ of A'' on the Σ'-word of length $(n + 1)$. There are $i < j \leq n + 1$ with $s_i = s_j$. Hence, any run of A'' is purely periodic from the ith place on. Thus, A'' accepts a non-empty word iff a final state appears among $s_1, \ldots, s_{j-1}$. □

A finite chain C with n subsets $X_1, \ldots, X_n$ can be considered as a word Word($C, X_1, \ldots, X_n$) of length $|C|$, in the alphabet Σ_n that is the Cartesian product of precisely n copies of $\{0, 1\}$. If $n = 0$, then Σ_0 is a singleton. In case $n > 0$, a letter of Σ_n can be viewed as a column of n zeros and ones. For example, if C is the chain Sunday, ..., Saturday and $X_1 = \{\text{Monday, Thursday}\}$ and $X_2 = \{\text{Monday, Tuesday, Wednesday}\}$, then we have

$$\text{Word}(C, X_1, X_2) = \begin{matrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0. \end{matrix}$$

2.1.3 Theorem. *There is an algorithm that, given n and a Σ_n-automaton A, constructs a formula $\phi(X_1, \ldots, X_n)$ in the monadic language of one successor such that for every finite chain C and any subsets $X_1, \ldots, X_n$ of C, we have that*

$$C \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \text{Word}(C, X_1, \ldots, X_n).$$

Proof. Without loss of generality, C can be taken as the chain $1, \ldots, l$ for some l. Let $s_1, \ldots, s_m$ be the states of A. The desired formula says that there are subsets $Y_1, \ldots, Y_m$ describing an accepting run of A on Word$(C, X_1, \ldots, X_n)$. The intended meaning of Y_k is $\{i: A \text{ is in the state } s_k \text{ after reading the } i\text{th letter}\}$. ∎

2.1.4 Theorem. *There is an algorithm that, given a formula $\phi(X_1, \ldots, X_n)$ in the monadic language of one successor (with free variables as shown), constructs a Σ_n-automaton A such that for every finite chain C and any subsets $X_1, \ldots, X_n$ of C, we have that*

$$C \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \text{Word}(C, X_1, \ldots, X_n).$$

Proof. We will merely sketch the proof. The automaton is built by induction on the formula. The atomic cases and the case of disjunction are quite easy. As to the case in which $\phi = \exists X_{n+1}\psi$, the desired Σ_n-automaton guesses X_{n+1} and mimics the Σ_{n+1}-automaton corresponding to ψ. The case of negation is easy for deterministic automata. We will now use Theorem 2.1.1 and the result will follow. ∎

Theorems 2.1.3 and 2.1.4 together constitute a kind of normal form theorem for the monadic theory of finite chains.

2.1.5 Theorem. *The monadic theory of finite chains is decidable.*

Proof. Given a sentence ϕ, we use the algorithm of Theorem 2.1.4 to find an appropriate automaton. The sentence ϕ is satisfiable iff the automaton accepts at least one non-empty word. Now, using Theorem 2.1.2, the assertion follows immediately. ∎

2.2. Monadic Theory of ω

As usual, ω will denote the chain of natural numbers. We consider it here as a model for the monadic language of one successor: The variables range over the subsets of ω, $\subseteq$ is the usual inclusion, and SUC(X, Y) means that there is a natural number x such that $X = \{x\}$ and $Y = \{x + 1\}$. The monadic theory of ω is known as S1S which is an acronym for second-order (monadic) theory of one successor. Observe that the linear order (on singleton sets) is easily definable in S1S.

A *sequential* Σ*-automaton* is a quadruple $A = (S, T, s_{\text{in}}, F)$, where S is the set of finite *states*, $T \subseteq S \times \Sigma \times S$ is the *transition table*, s_{in} is the *initial state* and F is the set of *final collections* of states. A is generally a non-deterministic automaton. However, it is *deterministic* if T is a total function from $S \times \Sigma$ to S. A *run* of A on a sequence $\sigma_1\sigma_2 \ldots$ is a sequence $s_1 s_2 \ldots$ of states such that $(s_{\text{in}}, \sigma_1, s_1) \in T$, and every $(s_i, \sigma_{i+1}, s_{i+1}) \in T$. It is an *accepting run* if $\{s: s_n = s$ for infinitely many $n\}$ belongs to F. And, finally, A *accepts* a sequence $\sigma_1\sigma_2 \ldots$ if there is an accepting run of A on this sequence.

2.2.1 Theorem. *There is an algorithm that, given an alphabet* Σ *and a sequential* Σ*-automaton A, constructs a deterministic sequential* Σ*-automaton accepting exactly the sequences accepted by A.*

This result is proven in McNaughton [1966]. However, simpler proofs can be found in Rabin [1972], Choueka [1974], Thomas [1981]. □

2.2.2 Theorem. *There is an algorithm that, given an alphabet* Σ *and a sequential* Σ*-automaton A, decides whether A accepts at least one sequence.*

Proof. The argument here is simple, since we only need repeat the proof of Theorem 2.1.2, speaking about sequences rather than words and changing the last sentence to: Thus A'' accepts the unique Σ'-sequence iff the collection $\{s_i, \ldots, s_{j-1}\}$ is final. □

Subsets $X_1, \ldots, X_n$ of ω form a sequence SEQ$(X_1, \ldots, X_n)$ in the alphabet Σ_n. The following three theorems and their proofs are similar to the corresponding theorems and proofs in Section 2.1.

2.2.3 Theorem. *There is an algorithm that, given n and a* Σ_n*-automaton A, constructs a formula* $\phi(X_1, \ldots, X_n)$ *in the monadic language of one successor such that for any subsets* $X_1, \ldots, X_n$ *of* ω,

$$\omega \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \mathrm{SEQ}(X_1, \ldots, X_n). \quad \square$$

2.2.4 Theorem. *There is an algorithm that, given a formula* $\phi(X_1, \ldots, X_n)$ *in the monadic language of one successor (with free variables as shown), constructs a* Σ_n*-automaton A such that for any subsets* $X_1, \ldots, X_n$ *of* ω,

$$\omega \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \mathrm{SEQ}(X_1, \ldots, X_n).$$

2.2.5 Theorem. *The monadic theory of* ω *is decidable.* □

2.3. *Monadic Theory of the Binary Tree*

The *binary tree* is here defined as the set $\{l, r\}^*$ of all words in the alphabet $\{l, r\}$. The empty word e is the root of the tree. The words xl and xr are respectively the *left* and the *right successors* of a word x.

The *monadic language of two successors* is (formally) the first-order language with binary predicates $\subseteq$, Left and Right. We regard the binary tree as a model for the monadic language of two successors: the variables range over the subsets, $\subseteq$ is the usual inclusion, Left(X, Y) means that there is a word x with $X = \{x\}$, $Y = \{xl\}$, and Right(X, Y) means that there is a word x with $X = \{x\}$, $Y = \{xr\}$. The monadic theory of the binary tree is known as S2S which is an acronym for the second-order (monadic) theory of two successors.

S2S is a very expressive theory. The relation "x is the initial segment of y" and "x precedes y lexicographically" are easily expressible (when coded on singleton sets). Rabin [1969] interpreted in S2S the monadic theories of 3, 4, etc. successors, the monadic theory of ω successors, and a good deal more.

A mapping V from the binary tree to an alphabet Σ will be called a Σ-*valuation* or a Σ-*tree*. We say that a *tree* Σ-*automaton* is a quadruple $A = (S, T, T_{\text{in}}, F)$ where S is the finite alphabet of *states*, $T \subseteq S \times \{l, r\} \times \Sigma \times S$ is the *transition table*, $T_{\text{in}} \subseteq \Sigma \times S$ is the *initial state table*, and F is the set of *final collections* of states. In order to describe when the automaton A accepts a Σ-tree V, we introduce a game $\Gamma(A, V)$ between the automaton A and another player called Pathfinder.

A chooses:	Pathfinder chooses:
s_0	
	d_1
s_1	
	d_2
s_2	
	d_3
s_3	
	...

Here each $s_n \in S$ and each $d_n \in \{l, r\}$. The choices of A are restricted by the following conditions:

$$(V(e), s_0) \in T_{\text{in}} \quad \text{and} \quad (s_n, d_{n+1}, V(d_1 \ldots d_{n+1}), s_{n+1}) \in T.$$

We would like to avoid the possibility of the automaton not being able to make the next move. One way to do this is to provide our automata with an additional state FAILURE in such a way that a transition into FAILURE is always possible, but a transition from a FAILURE to another state is never possible. Of course, the singleton set {FAILURE} will not be a final collection.

The automaton A *wins* a play $s_0 d_1 s_1 d_2 \ldots$ if $\{s \in S: s_n = s \text{ for infinitely many } n\}$ belongs to F. Otherwise, Pathfinder *wins*. The automaton A *accepts* V if it has a winning strategy in $\Gamma(A, V)$. Otherwise, it *rejects* V. The notion of strategy is clarified below.

A *position* in $\Gamma(A, V)$ is a word in the alphabet $S \cup \{l, r\}$ that is an initial segment of some play $s_0 d_1 s_1 d_2 \ldots$. The *last appearance record* LAR(p) in a position p is the string of last appearances of states in p. Consider the following example:

A	Pathfinder	Position	LAR
		e	e
a		a	a
	l	al	a
b		alb	ab
	r	$albr$	ab
a		$albra$	ba
	l	$albral$	ba
c		$albralc$	bac
	r	$albralcr$	bac
c		$albralcrc$	bac
	l	$albralcrcl$	bac
a		$albralcrcla$	bca

Here is an inductive definition of the last appearance record LAR(p). If p is the empty word e (that is, the initial position), then LAR(p) is empty. If $p = ql$ or $p = qr$, then LAR(p) = LAR(q). Suppose now that $p = qs$, u = LAR(q) and u' is obtained from u by erasing all appearances of s. Then LAR(p) = $u's$. Every last appearance record is a word in alphabet S, where each state appears at most once.

A (deterministic) *strategy* for the automaton A in the game $\Gamma(A, V)$ is a function assigning a legal state to every position of even length. A (deterministic) *strategy* for Pathfinder is a function assigning a direction l or r to each position of odd length.

Unfortunately, deterministic tree automata are too weak and Theorem 2.1.1 cannot be generalized to them. That theorem played a key role in Section 2.1; and in the case of tree automata the proper form of determinacy will play an analogous role.

2.3.1 Theorem (Forgetful Determinacy Theorem). *One of the players has a winning strategy f in $\Gamma(A, V)$ such that if p, q are two positions, where the winner makes moves and p, q define the same residual game (that is, they have the same continuation) and have the same last appearance records, then $f(p) = f(q)$.*

Proof. See Gurevich and Harrington [1982]. ∎

A strategy f for a player in $\Gamma(A, V)$ will be called *forgetful* if $f(p) = f(q)$, for all positions p, q such that the player makes moves in p, q and p, q define the same residual games, and moreover, the last appearance records in p and in q coincide. The reason for this term is that any value $f(p)$ depends on the residual game and an only limited information about the history. Thus, in brief, we may say that a forgetful strategy "forgets" most of the history.

2.3.2 Theorem. *There is an algorithm that, given an alphabet Σ and a tree Σ-automaton A, decides whether A accepts at least one Σ-tree.*

Proof. As in the proof of Theorem 2.1.2, we first reduce the problem to the case of a singleton alphabet. Thus, suppose that Σ is a singleton and V is the unique Σ-tree. By the forgetful determinacy theorem, one of the players has a forgetful strategy winning $\Gamma(A, V)$. List all forgetful strategies $f_1, \ldots, f_m$ for the automaton A and all forgetful strategies $g_1, \ldots, g_n$ for Pathfinder. It is possible to check each f_i against each g_j because the play eventually becomes periodic. This way we can find the desired winning strategy and determine whether or not A accepts V. □

Subsets $X_1, \ldots, X_n$ of the binary tree give a Σ_n-tree that will be called TREE($X_1, \ldots, X_n$), where Σ_n is as in Section 2.1.

2.3.3 Theorem. *There is an algorithm that, given n and a tree Σ_n-automaton A, constructs a formula $\phi(X_1, \ldots, X_n)$ in the monadic language of two successors such that for any n subsets $X_1, \ldots, X_n$ of the binary tree,*

$$\{l, r\}^* \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \mathrm{TREE}(X_1, \ldots, X_n).$$

Proof. A *run* of a tree Σ-automaton A on a Σ-tree V is a function R from the binary tree to the set of states of A such that every sequence

$$R(e)d_1R(d_1)d_2R(d_1d_2)\ldots$$

is a legal play in $\Gamma(A, V)$. If A wins all these plays then the run R is *accepting*.

The desired formula says that there are subsets Y_s, where s ranges over the states of the given tree Σ_n-automaton A, that describe an accepting run R of A on TREE($X_1, \ldots, X_n$). The intended meaning of Y_s is

$$\{x \in \{l, r\}^* : R(x) = s\}. \quad \square$$

2.3.4 Theorem. *There is an algorithm that, given a formula $\phi(X_1, \ldots, X_n)$ in the monadic language of two successors, constructs a tree Σ_n-automaton A in such a way that for any n subsets $X_1, \ldots, X_n$ of the binary tree,*

$$\{l, r\}^* \models \phi(X_1, \ldots, X_n) \quad \textit{iff} \quad A \textit{ accepts } \mathrm{TREE}(X_1, \ldots, X_n).$$

Proof. The argument here is similar to that given for Theorem 2.1.4, except for the case of negation which is treated in Theorem 2.3.6 below. □

2.3.5 Theorem. *The monadic theory of the binary tree is decidable.*

Proof. The argument here is similar to that given for Theorem 2.1.5. ☐

2.3.6 Theorem (Complementation Theorem). *There is an algorithm that, given an alphabet Σ and a tree Σ-automaton A, constructs a tree Σ-automaton accepting exactly the Σ-trees rejected by A.*

Proof. Let V be a Σ-tree rejected by A. By the forgetful determinacy theorem, Pathfinder has a forgetful strategy f winning $\Gamma(A, V)$. If p is a position in $\Gamma(A, V)$, let Node(p) be the string of even letters in p. For example, if $p = albralcrcla$ then Node(p) $= lrlrl$. If p, q are two positions of odd length, Node(p) = Node(q), and A is in the same state in p, q (that is to say, p, q have the same last letter), then p, q define the same residual game. This allows us to code f by an appropriate valuation of the binary tree.

Let RECORDS be the set of words u in the alphabet of states of A such that every state appears at most once in u. Elements of RECORDS will be called records. Let Σ' be the set of functions assigning a letter l or r to each record. There is a Σ'-tree V' such that for every position p in $\Gamma(A, V)$ we have

$$f(p) = (V'(\text{Node } p))(\text{LAR } p).$$

Since f is winning, every path

$$e, d_1, d_1d_2, d_1d_2d_3, \ldots$$

through the binary tree $\{l, r\}^*$ satisfies the following condition:

(*) There are no sequences $s_0s_1s_2\ldots$ and $u_0u_1u_2\ldots$ such that $s_0d_1s_1d_2\ldots$ is a play with respect to f and $u_0, u_1, u_2, \ldots$ are corresponding last appearance records and $\{s\colon$ for every i there is $j > i$ with $s_j = s\}$ is a final collection of states.

Clearly (*) abbreviates a formula in the monadic language of one successor whose parameters code the path $e, d_1, d_1d_2, d_1d_2d_3, \ldots$ and the corresponding sequences $V(e), V(d_1), V(d_1d_2), \ldots$ and $V'(e), V'(d_1), V'(d_1d_2), \ldots$. By Theorem 2.2.4 there is a sequential automaton $A' = (S', T', s'_{\text{in}}, F')$ over the alphabet $(\Sigma \times \Sigma') \cup (\{l, r\} \times \Sigma \times \Sigma')$ that accepts a sequence

$$V(e)V'(e), d_1V(d_1)V'(d_1), d_2V(d_1d_2)V'(d_1d_2), \ldots$$

iff it satisfies (*).

Let $A'' = (S', T'', T'_{\text{in}}, F')$ be the deterministic tree $\Sigma \times \Sigma'$-automaton with $T''(s, d, \sigma\sigma') = T'(s, d\sigma\sigma')$ and $T''_{\text{in}}(\sigma\sigma') = T'(s'_{\text{in}}, \sigma\sigma')$. A'' mimics A' and accepts the $\Sigma \times \Sigma'$-tree $V \times V'$ given by V and V'. Finally, let $\bar{A}$ be the Σ-automaton that guesses V' and mimics A''. Note that each successor in the row $A, A', A'', \bar{A}$ is computable from the predecessor. Evidently $\bar{A}$ accepts V.

$\bar{A}$ is the desired Σ-automaton complementing A. For, suppose that $\bar{A}$ accepts a Σ-tree V. There is a Σ'-tree V' such that A'' accepts $V \times V'$. Then A' accepts every sequence

$$V(e)V'(e),\, d_1 V(d_1)V'(d_1),\, d_2 V(d_1 d_2)V'(d_1 d_2), \ldots .$$

Thus, every path e, d_1, $d_1 d_2, \ldots$ through the binary tree satisfies $(*)$, where f is the strategy for Pathfinder defined by

$$f(p) = (V'(\text{Node } p))(\text{LAR } p).$$

Evidently f is winning. Hence, A rejects V. ∎

3. The Model-Theoretic Decidability Technique

The most important tools for dealing with monadic theories are *composition theorems*. The term "composition" here means generalized products in the sense of Feferman–Vaught [1959]. The Feferman–Vaught theorem reduces the first-order theory of the given composition to the first-order theories of the parts (summands, factors) and the monadic (!) theory of the index structure. Monadic composition theorems reduce the monadic theory of the given composition to the monadic theory of the parts and the monadic theory of the index structure (see, for example, the monadic composition theorem for chains in Section 3.2). Thus, monadic composition theorems appear to be more natural. Moreover, the interplay of monadic theories opens absolutely new and unexpected approaches to the decision problem. One of these approaches is demonstrated in Section 3.3 by a model-theoretic proof of decidability of the monadic theory of ω. Limited by the size of this chapter, we have chosen in the present section to explain only an easy part of the model-theoretic technique for proving decidability of monadic theories and to make this exposition as comprehensible as possible. We hope that this discussion—selective though it may be—will assist the interested reader in examining the more comprehensive exposition to be found in either Shelah [1975e] or in the papers Gurevich [1979a] and Gurevich–Shelah [1979].

3.1. Bounded Theories

Recall that the prefix of a prenex first-order formula is a word in the alphabet $\{\forall, \exists\}$. Blocks of universal quantifiers alternate with blocks of existential quantifiers in a prefix. The *alternation type* of a prefix is the sequence of lengths of the quantifier blocks. For example the alternation type of both $\forall^3\exists^4\forall^5$ and $\exists^3\forall^4\exists^5$ is 3, 4, 5. Clearly, the alternation type of the empty prefix is the empty sequence. Letters ξ and η (without subscripts) will be used to denote alternation types. We use the symbol $\frown$ to denote concatenation of sequences. Thus, if ξ is 3, 4, 5 then $\xi \frown 8$ is 3, 4, 5, 8.

Let L be a first-order language. For every n, indistinguishability by prenex sentences with prefix of length n gives an equivalence relation on structures for L. The n-step Ehrenfeucht game was introduced to provide a convenient sufficient condition for this equivalence relation to hold. Indistinguishability by prenex sentences with prefix of a given alternation type is also an equivalence relation on structures for L. We generalize Ehrenfeucht games to provide convenient sufficient conditions for these new equivalence relations to hold.

Proviso 1. *The vocabulary of L consists of finitely many relation symbols and individual constants.*

Let M and N be structures for L and ξ be an alternation type $\xi_1 \ldots \xi_n$. The game $\xi - \Gamma(M, N)$ is played between players I and II in n steps. On the kth step, player I chooses a structure M or N and a tuple of ξ_k elements of the chosen structure; and, in response, player II chooses a tuple of ξ_k elements of the remaining structure. Let $a_1, \ldots, a_m$ be the tuple of all $\xi_1 + \cdots + \xi_n$ elements chosen in M; the ξ_1-tuple of the first step concatenated with the ξ_2-tuple of the second step, etc. Let $b_1, \ldots, b_m$ be the corresponding tuple of elements chosen in N. Player II wins if the quantifier-free type of $a_1, \ldots, a_m$ in M coincides with the quantifier-free type of $b_1, \ldots, b_m$ in N, otherwise player I wins.

3.1.1 Theorem. *If player* II *has a winning strategy in* $\xi - \Gamma(M, N)$, *then* M *and* N *are indistinguishable by prenex sentences with prefix of type* ξ.

Proof. Any distinguishing prenex sentence of type ξ gives a winning strategy for player I. □

We will say that L-structures M and N are *ξ-equivalent* if player II has a winning strategy in $\xi - \Gamma(M, N)$.

By induction on the length of ξ, we define the ξ-theory of an L-structure M with a tuple of additional elements. $0 - \mathrm{Th}(M, a_1, \ldots, a_l)$ is the quantifier-free type of $a_1, \ldots, a_l$ in M. If ξ is $\eta \frown k$ then $\xi - \mathrm{Th}(M, a_1, \ldots, a_l)$ is the set of all $\eta - \mathrm{Th}(M, a_1, \ldots, a_l, b_1, \ldots, b_k)$ where $b_1, \ldots, b_k \in M$.

3.1.2 Theorem. *Two structures for L are ξ-equivalent iff they have the same ξ-theory.*

Proof. The proof is simple and we will omit it here. □

The usual n-step Ehrenfeucht game corresponds to the case when ξ is a sequence of n ones. This sequence will be denoted 1^n. 1^n-equivalent structures are called usually n-equivalent. The 1^n-theory of a structure is called usually the n-theory.

It is important for us that our bounded theories—in particular, quantifier-free types—are finite objects. This explains Proviso 1. This proviso is, however, too restrictive for applications. Is there any way to have finite quantifier-free types in a situation when Proviso 1 fails? The answer is Yes. In fact, consider the first-order theory of boolean algebras. There are infinitely many terms in a given finite set of variables, but only finitely many of these terms are in disjunctive normal form and each term is equal to one in disjunctive normal form.

Proviso 2. *L may have function symbols but it has only finitely many relation symbols. T is a theory in L, T allows a definition of normal terms in such a way that*:

(i) *there are only finitely many normal terms for any given finite set of variables*; *and*
(ii) *every term is equal in T to a normal term* (*in the same variables*).

An atomic formula $P(\tau_1, \ldots, \tau_k)$ will be called *standard* if the terms $\tau_1, \ldots, \tau_k$ are normal. We identify the quantifier-free type of a tuple $(a_1, \ldots, a_l)$ in a model M of T with the set of standard atomic formulas $\phi(v_1, \ldots, v_l)$ such that $M \models \phi(a_1, \ldots, a_l)$. Now we can simply repeat the definition of ξ-theories. Proviso 2 will suffice for our purposes here. A more liberal proviso can be found in Gurevich [1979a].

3.1.3 Theorem. *T is decidable if there is an algorithm computing $\{\xi - \mathrm{Th}(M): M \models T\}$ from ξ. T is decidable if there is an algorithm computing $\{1^n - \mathrm{Th}(M): M \models T\}$ from n.*

Proof. As in the case of Theorem 3.1.2, the proof of this result is simple and will not be given here. ▯

Even if T is not decidable, there is often an algorithm which computes a box including $\{\xi - \mathrm{Th}(M): M \models T\}$ from ξ. We define these boxes by induction on the length of ξ. The 0-l-Box is

$$\{0 - \mathrm{Th}(M, a_1, \ldots, a_l): M \models T \text{ and } a_1, \ldots, a_l \in M\}.$$

If ξ is $\eta \frown k$, then the ξ-l-Box is the power-set of the η-$(l + k)$-Box. We now turn our attention to

3.1.4 Proposition. *If $M \models T$ and $a_1, \ldots, a_l \in M$ then*

$$\xi - \mathrm{Th}(M, a_1, \ldots, a_l) \in \xi\text{-}l\text{-Box}.$$

Proof. Again, the argument for this result is obvious and is omitted here. ▯

It will be convenient to view elements of every ξ-l-Box as ordered in a standard manner. For example, the order may be lexicographical.

3.2. Monadic Composition Theorem for Chains

To fit this section into the framework of Section 3.1, we should say what the language L and the theory T are. Let BOOL be the first-order language of boolean algebras containing all the usual boolean operations and the equality predicate. L is the monadic language of order that is obtained from BOOL by adding the predicate $X \leq Y$. Every chain gives a standard model for L in the following way: We consider the boolean algebra of subsets and define $X \leq Y$ iff there are points $x \leq y$ with $X = \{x\}$ and $Y = \{y\}$. T is the monadic theory of order in L. In other words, T is simply the first-order theory of the described standard models for L. L and T satisfy Proviso 2 and we can freely use ξ-theories as well as other notions defined in Section 3.1.

Suppose that M is the lexicographic sum

$$L\Sigma\langle M_i : i \in I\rangle$$

of chains M_i with respect to a chain I. This means that M is itself a chain, the chains M_i are disjoint, the universe of M is the union of the universes of the chains M_i, and a point $x \in M_i$ precedes in M a point $y \in M_j$ iff $i < j$ or $i = j$ and $x < y$ in M_i.

Let X be an l-tuple $X_1, \ldots, X_l$ of subsets of M. For $i \in I$, the l-tuple $X_1 \cap M_i, \ldots, X_l \cap M_i$ will be denoted $X | M_i$. For every alternation type ξ and every $t \in \xi$-l-Box, let

$$P(\xi, X, t) = \{i : \xi - \text{Th}(M_i, X | M_i) = t\}.$$

Furthermore, let $P(\xi, X)$ be the sequence $\langle P(\xi, X, t) : t \in \xi\text{-}l\text{-Box}\rangle$ that partitions I.

3.2.1 Lemma. *There is an algorithm that computes* $0 - \text{Th}(M, X)$ *from* $0 - \text{Th}(I, P(0, X))$ *when I, M and X are varied.*

Proof. Let $P = P(0, X)$ and $P_t = P(0, X, t)$. If τ is a boolean term in variables $v_1, \ldots, v_l$, then we let $\tau^* = \tau(X_1, \ldots, X_l)$, where the complements are taken in M. It is easy to check that

$$\tau^* \cap M_i = \tau(X_1 \cap M_i, \ldots, X_l \cap M_i),$$

where the complements are taken in M_i.

In order to compute $0 - \text{Th}(M, X)$ it suffices to compute the truth values of statements $\sigma^* = \tau^*$ and $\sigma^* \leq \tau^*$, where σ and τ are in disjunctive normal form.

$\sigma^* = \tau^*$ iff $\sigma^* \cap M_i = \tau^* \cap M_i$, for every $i \in I$, iff for every $t \in 0$-l-Box, we have that either $P_t = 0$ or t implies $\sigma = \tau$. Given $0 - \text{Th}(I, P)$, we can check the last necessary and sufficient condition.

Note that $\tau \leq \tau$ means that τ is a singleton set. τ^* is a singleton iff there is $s \in 0$-l-Box such that P_s is a singleton, s implies $\tau \leq \tau$ and for every other $t \in 0$-l-Box, we have that either $P_t = 0$ or t implies $\tau = 0$. Given $0 - \text{Th}(I, P)$, we can check the necessary and sufficient condition.

Finally $\sigma^* \leq \tau^*$ iff both σ^* and τ^* are singleton and either

(i) there are distinct $s, t \in 0$-l-Box such that $P_s \leq P_t$ and s implies $\sigma \neq 0$, t implies $\tau \neq 0$; or

(ii) there is $t \in 0$-l-Box such that $P_t \neq 0$, and t implies $\sigma \leq \tau$.

Given $0 - \text{Th}(I, P)$, we can check the necessary and sufficient condition. □

3.2.2 Definition. If ξ is empty, then for every k, $H(\xi, k)$ is the empty alternation type. If ξ is $\eta \frown j$, then $H(\xi, k) = H(\eta, k + j) \frown p$, where p is the cardinality of η-$(k + j)$-Box.

3.2.3 Theorem. *There is an algorithm* COMP *that computes* $\xi - \text{Th}(M, X)$ *from* $H(\xi, l)$-$\text{Th}(I, P(\xi, X))$, *when* I, M, X *and* ξ *are varied.*

Proof. By induction on n, we construct algorithms COMP_n such that every COMP_n computes $\xi - \text{Th}(M, X)$ from $H(\xi, l) - \text{Th}(I, P(\xi, X))$, for every ξ of length n. The construction is uniform in n and results in the desired algorithm COMP.

The case $n = 0$ was treated in Lemma 3.2.1. Suppose that COMP_n is already constructed. Instead of defining COMP_{n+1} formally, we will simply explain how it works.

Let ξ be an alternation type of length n. $\xi \frown k - \text{Th}(M, X)$ is the set

$$\text{S1} = \{\xi - \text{Th}(M, X \frown Y): lh(Y) = k\},$$

where Y ranges over tuples of k subsets of M. COMP_n will compute S1 from

$$\text{S2} = \{\eta - \text{Th}(I, P(\xi, X \frown Y)): lh(Y) = k\},$$

where $\eta = H(\xi, l + k)$. S2 is computable from

$$\text{S3} = \{\eta - \text{Th}(I, P(\xi \frown k, X), P(\xi, X \frown Y)): lh(Y) = k\}.$$

From the other side, $H(\xi \frown k, l) - \text{Th}(I, P(\xi \frown k, X))$ is the set

$$\text{S4} = \{\eta - \text{Th}(I, P(\xi \frown k, X) \frown Q): lh(Q) = |\xi\text{-}(l + k)\text{-Box}|\},$$

where η is again $H(\xi, l + k)$. Evidently, S3 is included into S4. We give a verifiable necessary and sufficient condition for an element $u = \eta - \text{Th}(I, P(\xi \frown k, X) \frown Q)$ of S4 to belong to S3:

The sequence

$$Q = \langle Q_t: t \in \xi\text{-}(l + k)\text{-Box}\rangle$$

partitions I, and $t \in s$ whenever Q_t meets $P(\xi \frown k, X, s)$.

The argument for necessity is obvious. To prove the sufficiency, suppose that u satisfies the condition. We need to find a tuple Y of k subsets of M such that $P(\xi, X \frown Y) = Q$. For every $i \in I$, there are $s \in \xi \frown k$-l-Box and $t \in \xi$-$(l + k)$-Box such that $i \in P(\xi \frown k, X, s) \cap Q_t$. Then $t \in s$; that is to say, $t \in \xi \frown k - \mathrm{Th}(M_i, X | M_i)$. Hence, $t = \xi - \mathrm{Th}(M_i, (X | M_i) \frown Y^i)$, for some tuple Y^i of k subsets of M_i. Now choose Y such that $Y | M_i = Y^i$, for $i \in I$. ▯

3.3. Monadic Theory of Countable Ordinals

3.3.1 Theorem. *There is an algorithm* PLUS *such that if M is the lexicographic sum $M_1 + M_2$ of chains M_1 and M_2 and if X is a tuple of subsets of M, then for every alternation type ξ,*

$$\xi - \mathrm{Th}(M, X) = \mathrm{PLUS}(\xi - \mathrm{Th}(M_1, X | M_1), \xi - \mathrm{Th}(M_2, X | M_2)).$$

Proof. Simply take $I = \langle 1, 2 \rangle$ in the composition theorem and the result follows. ▯

We write $t = t_1 + t_2$ if $t = \mathrm{PLUS}(t_1, t_2)$. The induced addition of bounded theories is obviously associative.

3.3.2 Theorem. *The monadic theory of finite chains is decidable.*

Proof. By Section 3.1, it suffices to show that $\{1^n - \mathrm{Th}(M)\colon M \text{ is a finite chain}\}$ is computable from n. Given n, we compute the 1^n-theory t_1 of singleton chains. We thus compute $t_2 = t_1 + t_1$, $t_3 = t_2 + t_1$, etc., stopping when we find $i < j$ with $t_i = t_j$. The set $\{t_1, \ldots, t_{j-1}\}$ is equal to $\{1^n - \mathrm{Th}(M)\colon M \text{ is finite}\}$. ▯

3.3.3 Theorem. *There is an algorithm* MULT *satisfying the following condition. Let M be the lexicographical sum of chains M_i with respect to a chain I, and let X be a tuple of l subsets of M. If $\xi - \mathrm{Th}(M_i, X | M_i) = s$ for every i and $\eta = H(\xi, l)$, then*

$$\xi - \mathrm{Th}(M, X) = \mathrm{MULT}(\eta - \mathrm{Th}(I), s).$$

Proof. The algorithm COMP computes $\xi - \mathrm{Th}(M, X)$ from $\eta - \mathrm{Th}(I, P(\xi, X))$ which is itself computable from $\eta - \mathrm{Th}(I)$ and s, because $P(\xi, X, s) = I$ and any other $P(\xi, X, t) = 0$. ▯

We write $s' = t \cdot s$ if $s' = \mathrm{MULT}(t, s)$.

3.3.4 Theorem. *The monadic theory of ω is decidable.*

Proof. By induction on n, we construct an algorithm f_n such that, given an alternation type ξ of length n and a natural number l, f_n computes $\{\xi - \mathrm{Th}(\omega, X)\colon X \text{ is an } l\text{-tuple of subsets of } \omega\}$. The construction is uniform in n and provides an algorithm which will subsume every f_n. By Section 3.1, we know that this is enough for decidability.

Case $n = 0$ is easy. Suppose that $n > 0$ and f_{n-1} is already constructed. Given ξ and l, we compute $\eta = H(\xi, l)$ which is equal to $\tilde{\eta} \frown k$, for some alternation type $\tilde{\eta}$ of length $n - 1$ and some k. Also, we compute

$$t = \eta - \text{Th}(\omega) = \{\tilde{\eta} - \text{Th}(\omega, Y)\colon Y \text{ is a } k\text{-tuple of subsets of } \omega\}$$
$$= f_{n-1}(\tilde{\eta}, k).$$

Using the decision procedure for the monadic theory of finite chains, we compute $A = \{\xi - \text{Th}(M, X)\colon M$ is a finite chain and X is an l-tuple of subsets of $M\}$. And, finally, using the algorithms PLUS and MULT, we compute $B = \{s_0 + t \cdot s\colon s_0, s \in A\}$.

Evidently, $B \subseteq C = \{\xi - \text{Th}(\omega, X)\colon X$ is an l-tuple of subsets of $\omega\}$. We prove that $B = C$, which fact allows us to compute C.

Given an l-tuple X of subsets of ω color every non-empty interval $[i, j)$ of natural numbers by the "color" $\xi - \text{Th}([i, j), X \mid [i, j))$. By the Ramsey theorem, there is an infinite sequence $0 < n_1 < n_2 < \cdots$ such that all intervals $[n_i, n_{i+1})$ have the same color s. If s_0 is the color of $[0, n_1)$, then $\xi - \text{Th}(\omega, X) = s_0 + t \cdot s \in B$. □

3.3.5 Theorem. *The monadic theory of countable ordinals is decidable.*

Proof. We explain how to compute $\{1^n - \text{Th}(\alpha)\colon \alpha$ is a countable ordinal$\}$ from a given number n. First, we use the algorithm of Theorem 3.3.4 to compute $t = \eta - \text{Th}(\omega)$, where $\eta = H(1^n, 0)$. By Theorem 3.3.3 $1^n - \text{Th}(\alpha \cdot \omega) = t \cdot (1^n - \text{Th}(\alpha))$, for any α. Second, compute the minimal set S of 1^n-theories which contains the 1^n-theory of singleton chains and which is also closed under addition and under multiplication by t. It is easy to see that S is the desired $\{1^n - \text{Th}(\alpha)\colon \alpha$ is a countable ordinal$\}$. □

4. The Undecidability Technique

The *monadic topology* of a topological space U is the first-order theory of the structure $\langle \text{PS}(U), \subseteq, \text{OPEN}\rangle$, where $\text{PS}(U)$ is the power-set of U, $\subseteq$ is the usual inclusion and OPEN is the unary predicate "X is open." In this section, we will describe a proof of undecidability of the monadic topology of the Cantor discontinuum CD. The monadic topology of CD is easily interpretable in the monadic theory of the real line R. In this way, we get undecidability of the monadic theory of R. We could, of course, deal directly with the monadic theory of R—it would be practically the same proof. Undecidability of the monadic topology of CD seems to be even more mysterious and more difficult to prove.

In Section 4.1 we will give a rough idea how one can talk about natural numbers in the monadic topology of CD—explaining the details would require more space. However, the details can be found in Gurevich–Shelah [1982]. There is a serious restriction on how much we can say about natural numbers in the monadic topology of CD: true first-order arithmetic is not interpretable (in the usual sense of this word, for example Monk [1976]) in the monadic theory of R, see Gurevich–Shelah [1981a]. In Section 4.2, we show that whatever we can say about natural numbers in the monadic topology of CD is enough to reduce true first-order arithmetic to the monadic topology of CD. Actually, a much stronger result is proven in Section 4.2.

4.1. *How Can One Speak About Natural Numbers in the Monadic Topology of the Cantor Discontinuum?*

The idea is to slice a countable everywhere dense set D into everywhere dense slices $S_0, S_1, \ldots$ and to code this decomposition by parameters. First, we choose an everywhere subset D^0 of D such that $D - D^0$ is everywhere dense also. Then, we slice D in such a way that the sets $A_0 = S_0 \cap D^0$, $A_1 = S_1 \cap D^0$, $A_2 = S_2 \cap D^0, \ldots$ are disjoint as well as everywhere dense. We then prove that there is a parameter W such that a certain monadic formula $\phi(X)$ with parameters D, D^0, W defines the slices locally: that is, every S_n satisfies ϕ and if some X satisfies ϕ, then every non-empty open set G has a non-empty open subset H where X coincides with one of the slices S_n. We have not said anything about sets $S_0 - A_0$, $S_1 - A_1, \ldots$. They can be used to code additional information. In particular, a pairing function can be coded.

The coding described is best explained in Gurevich–Shelah [1982]. Here we can only summarize results of the coding in a convenient form. There are monadic topological formulas Premise($\bar{u}$), Share($\bar{u}$, v_0) and Pairing($\bar{u}$, v_0, v_1, v_2, v_3) which satisfy the following conditions. Both $\bar{u}$ and (v_0, v_1, v_2, v_3) are sequences of (set) variables. The formulas Premise, Share, and Pairing do not have any free variables except those shown. Premise($\bar{u}$) is satisfiable in CD. If t is a sequence of point sets and Premise(t) holds in CD then there is a sequence $\langle A_i : i < \omega \rangle$ of disjoint subsets of CD which satisfy the conditions C0–C2 below:

- C0. Each A_n is everywhere dense and each intersection $A_i \cap A_j$, with $i \neq j$, is empty.
- C1. Share(t, X) holds iff every non-empty open set G has a non-empty open subset H such that $X \cap H$ is equal to some $A_n \cap H$.

We will say that X is a *t-share* if Share(t, X) holds. We order the ordered pairs of natural numbers first by the maximum and then lexicographically:

$$(0, 0), (0, 1), (1, 0), (1, 1), (0, 2), (1, 2), (2, 1), \ldots .$$

Let P be the set of triples (i, j, k) of natural numbers such that (i, j) is the kth pair (when (0, 0) is pair number 0).

- C2. Suppose that X, Y, Z are t-shares and G is a non-empty open set. Then, Pairing(t, X, Y, Z, G) holds iff, for every non-empty open $G_1 \subseteq G$, there is a triple $(i, j, k) \in P$ and a nonempty open $H \subseteq G_1$ with $X \cap H = A_i \cap H$, $Y \cap H = A_j \cap H$, $Z \cap H = A_k \cap H$.

Before we go on to discuss reduction, let us recall that an open subset G of a topological space is called *regular* if the interior of the closure of G coincides with G. The following propositions is well known.

4.1.1 Proposition. *The regular open subsets of any topological space U form a complete boolean algebra with*:

(i) $G \cdot H = G \cap H$;
(ii) $G + H = \text{Interior}(\text{Closure}(G \cup H))$;
(iii) $-G = \text{Interior}(U - G)$; *and*
(iv) $1 = U$, and $0 = \varnothing$.

4.2. Reduction

Models of ZFC, the Zermelo–Fraenkel set theory with the axiom of choice, will be called *worlds*. In this discussion we will work in a world V. By sets is meant elements of V. For every complete boolean algebra B (in the world V) a standard construction provides a B-valued world V^B (see Jech [1978]). If ϕ is a formula in the language of ZFC with possible parameters from V^B, then the boolean value of ϕ will be denoted as usual $\|\phi\|$. Some simple but useful facts about V^B are summarized in the following

4.2.1 Proposition. (a) *Suppose that $\{b_i : i \in I\}$ is an antichain in B (which means that $b_i \cdot b_j = 0$ for $i \neq j$). For every $\{\sigma_i \in V^B: i \in I\}$ there is $\sigma \in V^B$ such that $b_i \leq \|\sigma_i = \sigma\|$ for $i \in I$.*

(b) *Let $\psi(v)$ be a formula in the language of* ZFC *with exactly one free variable and perhaps some parameters from V^B, then there is $\sigma \in V^B$ such that $\|\psi(\sigma)\| = \|\exists v \psi(v)\|$.*

(c) *Let $\psi(v)$ be as above and $\tau \in V^B$. Suppose $\|\exists v(v \in \tau)\| = 1$, then there is $\sigma \in V^B$ such that $\|\sigma \in \tau\| = 1$, and $\|\psi(\sigma)\| = \|(\exists v \in \tau)\psi(v)\|$.*

Proof. For the proof of (a), see Lemma 18.5 in Jech [1978]. As to part (b), see Lemma 18.6 in Jech [1978]. Turning now to part (c), we let $b = \|(\exists v \in \tau)\psi(v)\|$. By part (b), there are σ_0 and σ_1 such that $\|\sigma_0 \in \tau\| = 1$ and $\|\sigma_1 \in \tau$ and $\psi(\sigma_1)\| = b$. Moreover, by part (a), there is σ such that $(-b) \leq \|\sigma = \sigma_0\| \leq \|\sigma \in \tau\|$, and then $b \leq \|\sigma = \sigma_1\| \leq \|\sigma \in \tau\| \cdot \|\psi(\sigma)\|$. σ is the desired element of V^B. ☐

In the remainder of this subsection B is the boolean algebra of regular open subsets of the Cantor discontinuum CD (in V). An element $\sigma \in V^B$ will be called a *quasi-element* (of ω) if $\|\sigma \in \omega\| = 1$. It will be called a *quasi-set* (of natural numbers) if $\|\sigma \subseteq \omega\| = 1$. Hereafter, we ignore the difference between an element of V and the canonical name for it in V^B.

Let t be a sequence of subsets of CD satisfying Premise(t). We will say that a t-share X *represents* a quasi-element σ if

$$\Sigma\{b \in B: X \cap b = A_n \cap b\} = \|\sigma = n\| \quad \text{for} \quad n < \omega.$$

Subsets of CD will be called *point sets*, and we will say that a point-set Y *represents* a quasi-set τ if

$$\Sigma\{b \in B: A_n \cap b \subseteq Y\} = \|n \in \tau\| \quad \text{for} \quad n < \omega.$$

4.2.2 Proposition. (a) *Every t-share represents some quasi-element, and every quasi-element is represented by some t-share.*

(b) *Suppose that t-shares X_0, X_1, X_2 represent quasi-elements $\sigma_0, \sigma_1, \sigma_2$. For every $b \in B$,* Pairing(t, X_0, X_1, X_2, b) *holds in* CD *iff* $b \leq \|(\sigma_0, \sigma_1, \sigma_2) \in P\|$.

(c) *Every point set represents some quasi-set, and every quasi-set is represented by some point set.*

(d) *Suppose that a t-share X represents a quasi-element σ, and a point set Y represents a quasi-set τ. Then*

$$\|\sigma \in \tau\| = \Sigma\{b \in B: X \cap b \subseteq Y\}.$$

Proof. (a) Given a t-share X let

$$b_n = \Sigma\{b \in B: X \cap b = A_n \cap b\} \quad \text{for} \quad n < \omega.$$

By condition C0, distinct regular open sets b_n are disjoint. Moreover, by condition C1, they partition CD. By Proposition 4.2.1, there is σ with $\|\sigma = n\| \geq b_n$, for all n. σ is the desired quasi-element. Conversely, if σ is a quasi-element, then the desired t-share is

$$X = \bigcup \{A_n \cap \|\sigma = n\| : n < \omega\}.$$

For the proof of part (b) we use condition C2.

Turning now to part (c), we see that if Y is a point set, then the desired quasi-set τ is a function from ω to B with

$$\tau(n) = \Sigma\{b \in B: A_n \cap b \subseteq Y\} \quad \text{for all} \quad n.$$

Conversely, if τ is a quasi-set, then the desired point set is

$$Y \doteq \bigcup \{A_n \cap \|n \in \tau\| : n < \omega\}.$$

We now consider part (d). To prove $\subseteq$, we will suppose that $0 < a \leq \|\sigma \in \tau\|$. It then suffices to show that there is $0 < b \leq a$ with $X \cap b \subseteq Y$. Since σ is a quasi-element and τ is a quasi-set, there are n and $0 < a_1 \leq a$ such that $a_1 \leq \|\sigma = n\|$ and $a_1 \leq \|n \in \tau\|$. Since X represents σ, there is $0 < a_2 \leq a_1$ such that $X \cap a_2 = A_n \cap a_2$. Since Y represents τ, there is $0 < b \leq a_2$ such that $A_n \cap b \subseteq Y$. Thus, $X \cap b \subseteq Y$.

To prove $\supseteq$, we will suppose that $a > 0$ and $X \cap a \subseteq Y$. It then suffices to show that there is $0 < b \leq a$ with $b \leq \|\sigma \in \tau\|$. Since σ is a quasi-element, there are n and $0 < a_1 \leq a$ with $a_1 \leq \|\sigma = n\|$. Since X represents σ, there is $0 < b \leq a_1$

such that $X \cap b = A_n \cap b$ and, therefore, $A_n \cap b \subseteq Y$. Since Y represents τ, we have $b \leq \|n \in \tau\|$. Thus, $b \leq \|\sigma \in \tau\|$. ▯

4.2.3 Theorem. *The full second-order theory of $\aleph_0$ in the world V^B is reducible to the monadic topology (in the world V) of the Cantor discontinuum. In other words, there is an algorithm (not depending on the choice of the ground world V) that assigns a sentence ϕ^* in the language of monadic topology to every second-order sentence ϕ in such a way that* CD $\models \phi^*$ *iff* $\|\omega \models \phi\| = 1$. ▯

This theorem tells us that the monadic topology of CD is very complicated. In particular, true first-order arithmetic is reducible to the monadic topology of CD. For, V and V^B share the same true first-order arithmetic. Moreover, there is an algorithm interpreting true first-order arithmetic in (and therefore reducing it to) the full second-order theory of $\aleph_0$ in any world. This algorithm, in conjunction with the algorithm of Theorem 4.2.3, reduces true first-order arithmetic to the monadic topology of CD.

Proof of Theorem 4.2.3. The algorithm of Proposition 1.3.1 interprets the full second-order V^B-theory of ω in the monadic V^B-theory of the structure (ω, P), where P is the pairing predicate defined in Section 4.1. Let L be the monadic language of (ω, P). We will view individual variables (respectively set variables) of L as variables ranging over quasi-elements (respectively quasi-sets). Thus, we view L as a sublanguage of the language of ZFC. If ϕ is a sentence that is an L-formula with parameters, we will write $\|\phi\|$ instead of $\|\omega \models \phi\|$.

Let t be a tuple of point sets such that Premise(t) holds in CD. By induction on L-formulas $\phi(u_1, \ldots, u_m, V_1, \ldots, V_n)$, we define a formula

$$(w \leq \|\phi(u_1, \ldots, u_m, V_1, \ldots, V_n)\|)_t$$

in the language of monadic topology in such a way that if t-shares $X_1, \ldots, X_m$ represent quasi-elements $\sigma_1, \ldots, \sigma_m$, and point sets $Y_1, \ldots, Y_n$ represent quasi-sets $\tau_1, \ldots, \tau_n$ and $b \in B$, then

$$(*) \qquad \text{CD} \models (b \leq \|\phi(X_1, \ldots, X_m, Y_1, \ldots, Y_n)\|)_t \quad \text{iff } b \leq \|\phi(\sigma_1, \ldots, \sigma_m, \tau_1, \ldots, \tau_n)\|.$$

In the case $m = n = 0$, $b = 1$ we will have the desired:

$$\text{Premise}(t) \to (1 \leq \|\phi\|)_t \text{ holds in CD} \quad \text{iff} \quad \|\phi\| = 1.$$

Case 1. ϕ is $(u_0, u_1, u_2) \in P$. Let $(w \leq \|\phi\|)_t$ be Pairing(t, u_0, u_1, u_2, w), and use Proposition 4.2.2(b).

Case 2. ϕ is $u \in V$. Let $(w \leq \|\phi\|)_t$ be a formula saying that $u \cap w - V$ is nowhere dense, and use the result of Proposition 4.2.2(d).

Case 3. ϕ is ϕ_1 & ϕ_2. Set

$$(w \leq \|\phi\|)_t = (w \leq \|\phi_1\|)_t \,\&\, (w \leq \|\phi_2\|)_t.$$

Case 4. ϕ is $\sim\psi$. Let $(w \leq \|\phi\|)_t$ be a formula saying that there is no $0 < w' \leq w$ satisfying $(w' \leq \|\psi\|)_t$. To check (∗), we suppose for simplicity that ϕ is a sentence. Then $(b \leq \|\phi\|)_t$ holds iff there is no $0 < a \leq b$ with $a \leq \|\psi\|$ iff $b \leq \|\phi\|$.

Case 5. ϕ is $\exists u\psi(u)$. Let $(w \leq \|\phi\|)_t$ be a formula saying that there is a t-share u satisfying $(w \leq \|\psi(u)\|)_t$. To check (∗) assume for simplicity that ϕ is a sentence. We first suppose that $b \leq \|\phi\|$. By the results of Proposition 4.1.1(c), there is a quasi-element σ with $\|\psi(\sigma)\| = \|\phi\| \geq b$. If a t-share X represents σ, then by the induction hypothesis $(b \leq \|\psi(X)\|)_t$ holds. Hence, $(b \leq \|\phi\|)_t$ holds. Next, we suppose that some t-share X satisfies $(b \leq \|\psi(X)\|)_t$. It represents some quasi-element σ. By the induction hypothesis, $b \leq \|\psi(\sigma)\|$. Hence, we have $b \leq \|\phi\|$.

Case 6. ϕ is $\exists V\psi(V)$. Let $(w \leq \|\phi\|)_t$ be a formula asserting that there is a point set V which satisfies $(w \leq \|\psi(V)\|)_t$. To check (∗) in this situation is similar to the task of checking in Case 5. ▯

5. *Historical Remarks and Further Results*

We will first review very briefly the history of the method of automata and games. We will also mention delimiting undecidability results and some other closely related results obtained by model-theoretic methods. In Section 5.2 we will, very briefly review the history of the model-theoretic methods used to deal with monadic theories. Some later results use model-theoretic methods as well as the method of automata and games. It seems to make no real sense to divide the two approaches too sharply, however.

5.1. *Emphasizing the Method of Automata and Games*

Church [1963] gave "a summary of recent work in the application of mathematical logic to finite automata." Exploring connections between logic and finite automata proved fruitful indeed; but the most interesting applications appeared to be applications of finite automata to the decision problems for monadic second-order theories. Decidability of the monadic theory of finite chains could have been the first, the most natural and the easiest example—but it was not. I only just made up this particular application and inserted it into Section 2 for expository purposes. Arithmetic was too much on the minds of those who first explored the connections between logic and finite automata. The first results were related to the weak monadic theory of ω with the successor relation. This theory was called weak second-order arithmetic. (Let us recall that the weak monadic theory of a structure is the theory of that structure in the monadic second-order language when the set variables range over finite sets of elements.) We will not speak about weak monadic theories here. A survey of the results in this area can be found in Thatcher–Wright [1968]. Let us note merely that the game technique given in Section 2 can be used to give an alternative (and relatively simple) proof of decidability of the weak monadic theory of the binary tree. We should also note that the decidability schema of Section 2, a schema that is based on correspondence between monadic formulas and automata, had already taken shape in the work on weak monadic theories.

Decidability of the monadic theory S1S of ω with the successor relation was proved by Büchi [1962]. He established a correspondence between S1S formulas and Büchi automata. These machines are ordinary finite automata $A = (S, T, s_{in}, F)$ with $F \subseteq S$ that work on sequences. A is said to accept a sequence $\sigma_1\sigma_2 \ldots$ in the input alphabet of A if there is a run $s_1 s_2 \ldots$ of A on the given sequence (which means, of course, that $(s_{in}, \sigma_1, s_1) \in T$ and every $(s_i, \sigma_{i+1}, s_{i+1}) \in T$) such that for every i there is $j > i$ with $s_j \in F$. Büchi also solved the emptiness problem for Büchi automata. Unfortunately, a non-deterministic Büchi automaton may be not equivalent to any deterministic Büchi automaton, and Büchi used the Ramsey theorem to solve the complementation problem for Büchi automata. Our sequential automata were introduced by Muller [1963] in order to prove Theorem 2.2.1. However, the first correct proof of that theorem was published by McNaughton [1966]. Simplifications of McNaughton's proof can be found in Rabin [1970], Choueka [1974], Thomas [1981].

Decidability of the monadic theory S2S of the binary tree with two successor relations was proven by Rabin [1969]. He established a correspondence between S2S formulas and Rabin automata that are somewhat different from our tree automata, and his proof of the complementation theorem is an extremely difficult induction on countable ordinals. He used the same technique to solve the emptiness problem for Rabin automata, although Rackoff [1972] found a simple reduction of the emptiness problem for Rabin automata to the emptiness problem for automata on finite binary trees. Our simple proof of the decidability of S2S follows Gurevich and Harrington [1982].

The idea of using games had been exploited earlier however. Büchi–Landweber [1969] used a strong determinacy of more special games to prove the following: Suppose that a sentence $\forall X\, \exists Y \phi(X, Y)$ holds in S1S where X, Y are tuples of variables. Then there is a deterministic sequential automaton which outputs an appropriate output Y when reading X. In particular, there is an S1S formula $\phi^*(X, Y)$ uniformizing ϕ; that is, ϕ^* implies ϕ in S1S and, for every X, there is a unique Y such that $\phi^*(X, Y)$ holds in S1S. Büchi [1977] sketched a reduction of the complementation problem for Rabin automata to a strong determinacy for boolean-F_σ games. This determinacy result was proven independently in Gurevich–Harrington [1982] and in the manuscript Büchi [1981]. The latter solution, however, is much more complicated (and it still uses an induction on countable ordinals).

Let me add a few words about Rabin's uniformization problem for S2S. Suppose that a sentence $\forall X\, \exists Y \phi(X, Y)$ holds in S2S (where for the sake of simplicity, X, Y are just single variables). Is there an S2S formula $\phi^*(X, Y)$ such that ϕ^* implies ϕ in S2S and, for every X, there is a unique Y such that $\phi^*(X, Y)$ holds in S2S? Using model-theoretic methods and forcing Gurevich–Shelah [1983b] solved this problem negatively. Their counterexample $\phi(X, Y)$ asserts that if X is not empty, then Y is a singleton subset of X.

Rabin [1969] proved the decidability of many interesting theories by interpreting them in S2S. Among those theories we find the monadic theory of countable chains and the theory of the real line with quantification over countable sets. More direct model-theoretic proofs of these two results as well as delimiting undecidability results can be found in Gurevich–Shelah [1979]. For more on this the reader may also see Section 5.2. Finally, we note that Rabin [1969] also proved that S2S allows us to quantify over F_σ subsets of (infinite) branches of the binary tree. (Basic open sets of the topology in question are sets of branches through a given node.)

Open Question. *If we augment the language of* S2S *by allowing quantification of arbitrary Borel sets over branches, is the resulting theory of the binary tree in the augmented language decidable?*

Shelah [1975e] states the reducibility of the monadic theory of a tree of height ω with a given structure S on the successors of each node to the monadic theory of S. The details appear in Stupp [1975]. Their proof uses Rabin's technique. The game technique of Gurevich–Harrington [1982] gives the generalized result fairly easily.

Büchi [1973] used automata to prove decidability of the monadic theory of ω_1 (with the order). See also Litman [1972], Büchi–Siefkes [1973], Büchi–Zaiontz [1983] for additional results about monadic theories of ordinals of cardinality at most $\aleph_1$. There is a good reason why these results cannot be generalized to ω_2. Using model-theoretic methods and assuming the existence of a weakly compact cardinal, Gurevich, Magidor, and Shelah [1983] prove:

(i) for any given $S \subseteq \omega$, there is a forcing extension of the given set-theoretic world, where the monadic theory of ω_2 has the Turing degree of S; and
(ii) there is a forcing extension of the given set-theoretic world, where the monadic theory of ω_2 and the full second-order theory of ω_2 are reducible each to the other.

5.2. Model-Theoretic Methods

The paper Shelah [1975e] represented a breakthrough in the study of monadic theories of chains. Shelah developed the model-theoretic decidability method, which we illustrated in Section 3, and proved all known decidability results about monadic theories of chains in a uniform way. Assuming the continuum hypothesis, he reduced true first-order arithmetic to the monadic theory of the real line. This was the first undecidability result in the area.

Shelah's decidability method was rooted in achievements of his predecessors. In this connection, let me mention Feferman–Vaught [1959], Ehrenfeucht [1961], and Läuchli [1968]. Working on well-orderings, Shelah used ideas of Büchi and

Rabin. For more on this, see the references in Shelah [1975e]. A detailed version of the model-theoretic decidability method, a version which prepared the ground for stronger results, is given in Gurevich [1979a]. Shelah's undecidability method was absolutely new. Actually, he wanted to prove decidability of the monadic theory of the real line. He was developing and sharpening the decidability method to achieve this goal when he discovered the undecidability. Later, he reduced true first-order arithmetic to the monadic theory of the real line just in ZFC, without making any additional set-theoretic assumptions. See Gurevich–Shelah [1982] in this connection.

Sometimes model-theoretic analysis is less informative than is the automaton-theoretic. For example, the decision procedure in Section 2 for the monadic theory of ω gives more than the corresponding decision procedure in Section 3: It establishes the correspondence between monadic formulas and deterministic sequential automata. In many other cases, however, the model-theoretic analysis is more informative. For example, Shelah answered negatively a question posed by Rabin, a question asking whether or not countable orders can be characterized in the monadic theory of chains.

Let us examine the monadic theory of countable chains a bit further. Shelah [1975e] conjectured that the monadic theory of countable chains can be finitely axiomatizable in the monadic theory of chains. However, Gurevich [1977b] refuted this conjecture. He provided a certain axiomatization of the monadic theory of countable chains. A chain is *short* if it embeds neither ω_1 nor ω_1^*, where ω_1^* is the dual of ω_1. A chain without jumps (that is, a densely ordered chain) is *perfunctorily n-modest* if for all everywhere dense subsets $X_1, \ldots, X_n$, there is a perfect subset Y without jumps such that $Y \subseteq X_1 \cup \cdots \cup X_n$ and every $X_i \cap Y$ is dense in Y. A chain is *n-modest* if all its subchains without jumps are perfunctorily n-modest. A chain is *modest* if it is n-modest, for every n. It appears that a chain is monadically equivalent to a countable chain iff it is short and modest. Rabin [1969] proved decidability of the monadic theory of countable chains. Thus, the monadic theory of short modest chains is decidable. Gurevich–Shelah [1979] proved directly decidability of short modest chains.

The situation is very different for non-modest chains. Assuming the continuum hypothesis, Gurevich–Shelah [1979] reduced true first-order arithmetic to the monadic theory of any nonmodest chain. The use of the continuum hypothesis was removed in Gurevich–Shelah [1982]. The reader may also consult Gurevich–Shelah [1979] for a model-theoretic analysis of the theory of the real line with quantification over countable subsets.

In order to discuss undecidability results, we need to clarify the terminology. A *reduction* of a theory T to a theory T^* is an algorithm associating a sentence ϕ^* in the language of T^* with each sentence ϕ in the language of T in such a way that ϕ^* holds in T^* iff ϕ holds in T. An *interpretation* of one theory in another is a special case of reduction when models of T are defined inside models of T^*. An exact definition of interpretation can be found in Monk [1976] for example.

As we mentioned above, Shelah [1975e] reduced true first-order arithmetic to the monadic theory of the real line. In Section 4 we did not say much about the undecidability method of Shelah [1975e]. This method was augmented in Gurevich

[1977b] by a technique of towers, a technique that has been exploited extensively in subsequent papers. Confirming Shelah's conjecture, Gurevich [1979b] reduced true third-order arithmetic to the monadic theory of the real line (in fact, to the monadic theory of any short non-modest chain) in Gödel's constructive universe. The converse reduction is obvious. Only during the Jerusalem Logic Year 1980–81 we—Saharon Shelah and I—realized that our reductions are really a kind of interpretation of (in terms of Section 4) theories in the "next world" V^B in theories in "this world" V. Subsuming all mentioned undecidability results, Gurevich-Shelah [1981a] managed:

(i) reduce true second-order arithmetic in V^B to the monadic V-theory of any short non-modest chain; and also
(ii) to reduce true third-order arithmetic in V^B to the monadic V-theory of any short non-modest chain if the continuum hypothesis holds in V.

In contrast to this, Gurevich-Shelah [1981a] proved that true first-order arithmetic is not interpretable in the monadic theory of the real line.

Gurevich-Shelah [1983a] reduce true second-order logic to the monadic theory of (linear) order under very weak set-theoretical assumptions. This gives the complexity of the monadic theory of order. It does not mean, however, that the monadic theory of order is as un-manageable as second-order logic. From a model-theoretical point of view, there is an enormous difference between these two theories (reflected somewhat in different Löwenheim and Hanf numbers). This topic is, however, beyond the scope of this chapter and the reader may see Chapter 12 in this connection.

A few words about topology. Grzegorczyk [1951] introduced the monadic topology (see Section 4) and interpreted (in a simple and natural way) true first-order arithmetic in the monadic topology of the Euclidean plane. It does not take much more sophistication to verify that the monadic topology of the Euclidean plane and true third-order arithmetic are interpretable, each in the other. For more on this, the reader may see Gurevich [1980]. Grzegorczyk's question about the decision problem for the monadic topology of the real line was, however, long open. Reading the paper Shelah [1975e], I noted that Shelah had solved negatively the question of Grzegorczyk under the continuum hypothesis. Several papers—especially Gurevich-Shelah [1981c]—give undecidability results about the monadic topology. In particular, all mentioned above undecidability results about the monadic theory of the real line apply to the monadic topology of the Cantor discontinuum. For a positive result on monadic topology see Gurevich [1982].

Gurevich-Shelah [1981b] use both model-theoretic methods and the method of automata and games to construct a decision procedure for the theory of trees (all trees, not necessarily well-founded) with quantification over maximal branches.

Finally, let us mention some results that are not directly related to decision problems. Gurevich [1977b] proved (thus refuting Shelah's conjecture) that the predicate "X is countable" is expressible in the monadic theory of the real line if the continuum hypothesis holds. Gurevich [1979b] also proved (and thus partly refuted and partly confirmed Shelah's conjectures) that the monadic theory of the real line can be finitely axiomatizable (in the monadic theory of chains) and categorical under natural set-theoretic assumptions. By "Shelah's conjectures" here, we mean the collection of conjectures that are given in Shelah [1975e]. Almost all of these conjectures have been decided by now, and a majority of those decided are true. Thus, the program sketched in Shelah [1975e] is essentially fulfilled. Moreover, I have an impression that an important and natural phase in the study of monadic second-order theories is now completed.