Monadic theories of wellorders

Itay Neeman*

February 24, 2008

This article is a partial survey of work on the monadic second order theory of wellorders, concentrating on connections with finite state automata. We present a progression of results, starting with the case of finite wellorders and ending with a general connection between monadic truth and automata on all ordinals. We give proofs and proof sketches at the initial levels, to illustrate some of the ideas in the work connecting automata and monadic truth. At higher levels the proofs are substantially more complicated and beyond the scope of this article. Our exposition follows the most direct mathematical route, and should not be taken as either a complete or an historical account. We refer the reader to Gurevich [7] for a survey on monadic theories, to Khoussainov–Nerode [9] for a comprehensive account of fundamental results on finite state automata, and to the papers by Vardi and Thomas in this volume for specific applications of automata theory in computer science.

Recall that the monadic second order language, monadic language for short below, has two kinds of variables: first order variables which range over elements of the structure, and second order variables which range over subsets of the structure. The atomic formulas in the monadic language are the usual first order atomic formulas (in first order variables), and formulas of the form $v \in U$ where v is a first order variable and U is a second order variable. General formulas are built from atomic formulas using negations, conjunctions, and existential quantifications over both first and second order variables. In many cases the monadic language provides a nice balance of expressivity and feasibility. Feasibility here is a vague term, and can mean many different things, for example that the corresponding theory is decidable, that the theory can be described in terms of a theory in a more limited language, or that definability can be described. It is in proving these kinds of feasibility that we make use of finite state automata.

We are concerned specifically with theories of wellorders, namely of structures of the form $(\alpha; <)$, where $<$ is a wellorder of the set α. Without loss of generality we may assume that α is an ordinal, and $<$ is the membership relation restricted to α.

*This material is based upon work supported by the National Science Foundation under Grant No. DMS-0556223.

Finite ordinals

Consider to begin with the case of finite α. This case serves as a simple illustration of the connection between the monadic theory and finite state automata.

The basic core of a finite state automaton with (finite) alphabet Σ is a finite *set of states* S, a smaller set $I \subseteq S$ of *initial states*, and a *transition table* $T \subseteq S \times \Sigma \times S$. The automaton takes as input a string $X\colon \alpha \to \Sigma$. A *run* of the automaton on X is a string of states $s\colon \alpha+1 \to S$ which satisfies the rules

$$s(0) \in I, \text{ and} \tag{Initial}$$
$$\langle s(\xi), X(\xi), s(\xi+1)\rangle \in T \tag{Succ}$$

for all $\xi < \alpha$.

The automaton is *deterministic* if I is a singleton and the transition table is the graph of a function from $S \times \Sigma$ into S, meaning that for each $b \in S$ and $\sigma \in \Sigma$ there is a unique $b^* \in S$ so that $\langle b, \sigma, b^*\rangle \in T$. Abusing notion we then refer to I as a state and to T as a function. In the case of a deterministic automaton, for finite α at least, conditions (Initial) and (Succ) determine a unique run of the automaton on X. The run is produced by setting $s(0) = I$ and then successively setting $s(n+1) = T(s(n), X(n))$. Non-deterministic automata in contrast may have many runs on an input X, and may also have none.

In addition to the basic core, the automaton has a set F of *accepting final states*. A run $s\colon \alpha+1 \to S$ is *accepting* if $s(\alpha) \in F$. The automaton *accepts* input $X\colon \alpha \to \Sigma$ just in case that there is an accepting run of the automaton on X. Note the existential quantifier that is built into the definition. We shall make good use of it with a non-deterministic automaton soon. But first let us quickly describe a coding of elements and subsets of α by strings which may be taken as inputs for automata.

For a set $A \subseteq \alpha$ define $\chi_A\colon \alpha \to 4$ by $\chi_A(\xi) = 1$ if $\xi \in A$ and $\chi_A(\xi) = 0$ otherwise. For an ordinal $a \in \alpha$ define $\chi_a\colon \alpha \to 4$ by $\chi_a(\xi) = 3$ if $\xi = a$ and $\chi_a(\xi) = 2$ otherwise. For a tuple $\langle e_1, \dots, e_k\rangle$ with each e_i either an element of α or a subset of α, define $\chi_{\langle e_1,\dots,e_k\rangle}\colon \alpha \to {}^k4$ by $\chi_{\langle e_1,\dots,e_k\rangle}(\xi) = \langle \chi_{e_1}(\xi), \dots, \chi_{e_k}(\xi)\rangle$. $\chi_{\langle e_1,\dots,e_k\rangle}$ is then a string of length α in the alphabet $\Sigma = {}^k4$, and codes the tuple $\langle e_1, \dots, e_k\rangle$. The domain α is suppressed in the notation, and is typically understood from the context.

The coding above lets us view tuples of elements and subsets of α as possible inputs for automata. We say that an automaton with alphabet $\Sigma = {}^k4$ *accepts* the tuple $\langle e_1, \dots, e_k\rangle$ iff it accepts $\chi_{\langle e_1,\dots,e_k\rangle}$.

An automaton $\mathcal{A}$ is *equivalent* to a monadic formula $\varphi(v_1, \dots, v_k)$ on structure $(\alpha; <)$ just in case that for every tuple $\langle e_1, \dots, e_k\rangle$ of elements and subsets of α which match the orders of variables of φ, $\mathcal{A}$ accepts $\langle e_1, \dots, e_k\rangle$ iff $(\alpha; <) \models \varphi[e_1, \dots, e_k]$.

Theorem 1. *For every monadic formula φ, there is a deterministic automaton $\mathcal{A}$ which is equivalent to φ on all structures $(\alpha; <)$ with α finite.*

The theorem is part of a large body of work analyzing finite state automata and regular languages. Its proof given below is very direct. For a more complete account which includes the related work we refer the reader to Khoussainov–Nerode [9, Chapter 2].

Proof of Theorem 1. The proof is by induction on the complexity of φ.

If φ is atomic then it is easy to explicitly define an automaton witnessing the theorem. Let us only go over one example, the formula $v_1 \in v_2$, with v_1 a first order variable and v_2 a second order variable. The following automaton is equivalent to this formula on finite structures: The automaton has three states, true, false, and $\mathsf{unknown}$. The initial state is $\mathsf{unknown}$, and the only accepting state is true. $T(\mathsf{unknown}, \langle 3, 2\rangle) = \mathsf{false}$, so that if a ξ is reached so that $\xi = e_1$ and $\xi \notin e_2$ the automaton falls into the state false. $T(\mathsf{unknown}, \langle 3, 1\rangle) = \mathsf{true}$, so that if a ξ is reached so that $\xi = e_1$ and $\xi \in e_2$ the automaton falls into the state true. $T(\mathsf{unknown}, \sigma) = \mathsf{unknown}$ for all other σ, and $T(\mathsf{false}, \sigma) = \mathsf{false}$ and $T(\mathsf{true}, \sigma) = \mathsf{true}$ for all σ.

If φ is a negation $\neg\psi$, take an automaton $\bar{\mathcal{A}}$ witnessing the theorem for ψ, and define the automaton $\mathcal{A}$ to have the same set of states, the same transition table, the same initial state, and the inverse set of final states, namely $F = S - \bar{F}$. Then $\mathcal{A}$ witnesses the theorem for φ. Notice that it is important here that we are dealing with deterministic automata, so that every input string leads to a final state uniquely determined by the string. $\bar{\mathcal{A}}$ accepts if this final state belongs to $\bar{F}$, and $\mathcal{A}$ accepts if it does not.

If φ is a conjunction $\psi_1 \wedge \psi_2$, take automata $\mathcal{A}_1$ and $\mathcal{A}_2$ witnessing the theorem for ψ_1 and ψ_2, and define an automaton $\mathcal{A}$ which simulates a simultaneous run of $\mathcal{A}_1$ and $\mathcal{A}_2$. The set of states S of $\mathcal{A}$ is $S_1 \times S_2$, the transition function T is defined by $T(\langle b_1, b_2\rangle, \sigma) = \langle T(b_1, \sigma), T(b_2, \sigma)\rangle$, the initial state I is $\langle I_1, I_2\rangle$, and the set of final states F is $F_1 \times F_2$. It is clear that $\mathcal{A}$ accepts X iff both $\mathcal{A}_1$ and $\mathcal{A}_2$ accept X.

Suppose finally that φ is an existential formula $(\exists v_k)\psi(v_1, \dots, v_k)$. Let $\bar{\mathcal{A}}$ witness the theorem for ψ. It is easy, modifying $\bar{\mathcal{A}}$, to define a *non-deterministic* automaton $\mathcal{A}^{nd}$ which is equivalent to φ. The automaton $\mathcal{A}^{nd}$ uses non-determinism to guess the characteristic function of v_k. Suppose for definitiveness that v_k is second order. Define S^{nd} to be $\bar{S}\times\{0, 1\}$ where $\bar{S}$ is the set of states of $\bar{\mathcal{A}}$. Set $\langle\langle b, i\rangle, \sigma, \langle b^*, i^*\rangle\rangle \in T^{nd}$ just in case that $T(b, \sigma^\frown\langle i\rangle) = b^*$. The definition is such that if s^{nd} is a run of $\mathcal{A}^{nd}$ on $\langle e_1, \dots, e_{k-1}\rangle$, then $s^{nd}(\xi)$ has the form $\langle \bar{s}(\xi), i_\xi\rangle$ where, setting $e_k = \{\xi < \alpha \mid i_\xi = 1\}$, $\bar{s}$ is a run of the original automaton $\bar{\mathcal{A}}$ on $\langle e_1, \dots, e_{k-1}, e_k\rangle$. In that sense the part $\langle i_\xi \mid \xi < \alpha\rangle$ of a run of $\mathcal{A}^{nd}$ is a guess by the non-deterministic automaton for a characteristic function of a set that can be substituted for v_k.

Continuing to define $\mathcal{A}^{nd}$, set the initial states to be $\langle \bar{I}, 0\rangle$ and $\langle \bar{I}, 1\rangle$, and let the set of final states F^{nd} be $\bar{F} \times \{0, 1\}$. Recall that existential quantification over runs was built into the definition of acceptance for automata. Inspecting that definition and the definition of $\mathcal{A}^{nd}$ it is easy to check that the non-deterministic $\mathcal{A}^{nd}$ accepts $\langle e_1, \dots, e_{k-1}\rangle$ iff there exists $e_k \subseteq \alpha$ so that $\bar{\mathcal{A}}$

accepts $\langle e_1, \dots, e_{k-1}, e_k\rangle$, namely iff there exists $e_k \subseteq \alpha$ so that $(\alpha;<) \models \psi[e_1, \dots, e_{k-1}, e_k]$. It follows that $\mathcal{A}^{nd}$ is equivalent to $\varphi = (\exists v_k)\psi$.

Of course $\mathcal{A}^{nd}$ is not deterministic. To complete the proof of the theorem we have to convert it to a deterministic automaton, and this can be done using:

Lemma 2 (Rabin–Scott [17])**.** *Every non-deterministic automaton is equivalent to a deterministic automaton on finite domains. Precisely, for every non-deterministic automaton $\mathcal{A}^{nd}$ there is a deterministic automaton $\mathcal{A}$, so that for every input string X of finite length, $\mathcal{A}$ accepts X iff $\mathcal{A}^{nd}$ accepts X.*

Proof. Runs of $\mathcal{A}$ keep track of all possible states which may be reached by runs of $\mathcal{A}^{nd}$, from each initial state. More precisely, states of $\mathcal{A}$ are subsets W of $I^{nd} \times S^{nd}$, the initial state I is the set $\{\langle b_0, b_0\rangle \mid b_0 \in I^{nd}\}$, and the transition function T is defined by $T(W, \sigma) = \{\langle b_0, b^*\rangle \in I^{nd} \times S^{nd} \mid (\exists b \in S^{nd})(\langle b_0, b\rangle \in W \wedge \langle b, \sigma, b^*\rangle \in T^{nd})\}$. With this definition it follows by induction on α that, if s is a run of $\mathcal{A}$ on an input string X of finite length α, then $\langle b_0, b\rangle \in s(\alpha)$ iff there is a run s^{nd} of $\mathcal{A}^{nd}$ on X with $s^{nd}(0) = b_0$ and $s^{nd}(\alpha) = b$. Setting $F = \{W \subseteq I^{nd} \times S^{nd} \mid W \cap (I^{nd} \times F^{nd}) \neq \emptyset\}$ it then easy to check that $\mathcal{A}$ accepts X iff $\mathcal{A}^{nd}$ accepts X. $\square$ (Lemma 2, Theorem 1)

Theorem 1 is constructive, and gives rise to a recursive map $\varphi \mapsto \mathcal{A}_\varphi$ which assigns to each monadic formula φ an equivalent automaton $\mathcal{A}_\varphi$. Already at the level of finite domains this association can be used to prove decidability results, for example:

Corollary 3. *The set of monadic sentences φ so that $(\exists\alpha < \omega)(\alpha;<) \models \varphi$ is decidable.*

For a stronger result, on the decidability of the fragment of the monadic theory of ω involving only finite sets, see Büchi [1] and Elgot [6].

Proof of Corollary 3. Fix a sentence φ. We describe how to decide whether or not $(\exists\alpha < \omega)(\alpha;<) \models \varphi$. Consider the automaton $\mathcal{A}_\varphi$. Since φ has no free variables, the alphabet of this automaton is ${}^0 4 = \{\emptyset\}$. Its transition function T_φ may therefore be viewed simply as a directed graph. The vertices are states, and the graph has an edge from b to b^* iff the automaton transitions from b to b^*, namely if $T_\varphi(b, \emptyset) = b^*$. The automaton accepts the (unique) input string of length α iff the graph has a path of length α from I_φ to a vertex in F_φ. So $(\exists\alpha < \omega)(\alpha;<) \models \varphi$ iff there is a vertex in F_φ which is reachable from I_φ. The graph is finite, and the question of reachability in finite graphs is decidable. $\square$

Countable ordinals

Büchi [2] discovered that there is a parallel of Theorem 1 to $\alpha = \omega$. Let $\mathcal{A}$ be a finite state automaton with alphabet Σ. Let $X\colon \omega \to \Sigma$ be an input string

of length ω. Conditions (Initial) and (Succ) give rise to a notion of runs of the automaton on X, but of length ω rather than $\omega+1$. A run is a sequence $s\colon \omega \to S$ which satisfies condition (Initial), and satisfies condition (Succ) for each $\xi < \omega$. Since the run does not provide a final state $s(\omega)$, the notion of acceptance requires an additional definition. Büchi equipped each of his automata with a set G of states, which we call *good states*, and defined a run s to be accepting iff $\{\xi < \omega \mid s(\xi) \in G\}$ is infinite. He then proved:

Theorem 4 (Büchi)**.** *For every monadic formula φ, there is a (non-deterministic) Büchi automaton which is equivalent to φ on domain $\alpha = \omega$.*

Proof sketch. Again the proof is by induction on the complexity of φ. The cases of atomic φ and of conjunctions are similar to the corresponding cases in the proof of Theorem 1. Since the automata in Theorem 4 are non-deterministic, the case of existential quantification is easy, similar to the corresponding case in the proof of Theorem 1 but without the need to prove the equivalence in Lemma 2. (This equivalence fails for Büchi automata.) It is the case of negations which is difficult. The proof in this case makes a clever use of Ramsey's theorem.

Say $\varphi = \neg\psi$. Let $\bar{\mathcal{A}}$ be a Büchi automaton witnessing the theorem for ψ, consisting of a set of states $\bar{S}$, a set of initial states $\bar{I}$, a transition table $\bar{T}$, and a set of good states $\bar{G}$. Suppose $X\colon \omega \to \Sigma$ is an input string for $\bar{\mathcal{A}}$. We have to define $\mathcal{A}$ (independently of X) so that $\mathcal{A}$ accepts X iff $\bar{\mathcal{A}}$ does not.

For $n < m < \omega$ let $C_X(n,m)$ be the set of pairs $\langle b, b^*\rangle \in \bar{S}\times\bar{S}$ so that $\bar{\mathcal{A}}$ can get from state b at n to state b^* at m. Precisely, $\langle b, b^*\rangle \in C_X(n,m)$ if there is a sequence $s\colon [n,m] \to \bar{S}$ which satisfies condition (Succ) for $\xi \in [n,m)$, with $s(n) = b$ and $s(m) = b^*$. Let $C^g_X(n,m)$ be the set of pairs $\langle b, b^*\rangle$ so that $\bar{\mathcal{A}}$ can get from state b at n to state b^* at m, with the additional requirement of passing through the set of good states $\bar{G}$. Precisely, $\langle b, b^*\rangle \in C^g_X(n,m)$ if there is s as above with the added requirement that $s(k) \in \bar{G}$ for some $k \in [n,m)$.

C_X and C^g_X are functions from ω^2 into the finite set $\mathcal{P}(\bar{S}\times\bar{S})$. By applications of Ramsey's theorem there is an infinite set $H \subseteq \omega$, and *fixed* D, E, and E^g, so that $C_X(0,n) = D$, $C_X(n,m) = E$, and $C^g_X(n,m) = E^g$ for all $n, m \in H$.

Note that knowledge of D and E^g suffices to determine whether $\bar{\mathcal{A}}$ has an accepting run on X. Such a run exists iff there are states $b_0 \in \bar{I}$ and $b \in \bar{S}$ so that $\langle b_0, b\rangle \in D$, and $\langle b, b\rangle \in E^g$. (Given such b_0 and b one can construct an accepting run $\bar{s}$ of $\bar{\mathcal{A}}$ on X with $\bar{s}(0) = b_0$ and $\bar{s}(n) = b$ for $n \in H$. Conversely, given an accepting run $\bar{s}$, set $b_0 = \bar{s}(0)$ and set b equal to any state which $\bar{s}$ repeats infinitely many time on the infinite set H.)

Let J be the set of all triples $\langle D, E, E^g\rangle \in \mathcal{P}(\bar{S}\times\bar{S})^3$ so that $\langle b_0, b\rangle$ as above do *not* exist. Define a non-deterministic automaton $\mathcal{A}$ so that a run of $\mathcal{A}$ on X does the following: (a) guess, in the very first state, a triple $\langle D, E, E^g\rangle \in J$; (b) guess, during the entire infinite run, a characteristic function χ_H of a set $H \subseteq \omega$; and (c) verify that $C_X(0,n) = D$, $C_X(n,m) = E$, and $C^g_X(n,m) = E^g$ for all $n < m$ both in H. If all three condition can be achieved for input X

with the set H *infinite*, then X is not accepted by the original automaton $\bar{\mathcal{A}}$, and vice versa. This follows from the conclusion of the previous paragraph. Let G, the set of good states for the new automaton $\mathcal{A}$, be the set of states at which $\mathcal{A}$ guesses value 1 for χ_H, so that a run of $\mathcal{A}$ is accepting iff the set H it guesses is infinite. Then $\mathcal{A}$ accepts X iff $\bar{\mathcal{A}}$ does not, completing the proof.

As for the actual construction of the automaton $\mathcal{A}$, conditions (a) and (b) are simple, and an automaton whose runs verify the part $C_X(0, n) = D$ in condition (c) can be defined using ideas similar to those in the proof of Lemma 2. The verification that $C_X(n, m) = E$ and $C_X^g(n, m) = E^g$ for all $n < m$ both in H must be done indirectly, since a finite state automaton cannot at stage m keep track of $C_X(n, m)$ and $C_X^g(n, m)$ for unboundedly many $n < m$. One defines the automaton to only verify the simpler requirement that $C_X(n, m) = E$ and $C_X^g(n, m) = E^g$ for n equal to the immediate predecessor of m in H, and adds the initial demand that the guess of E and E^g for condition (a) must satisfy compositional properties which give the full requirement from the simpler one, and conversely follow from the full requirement. For example, the initial guess must satisfy $(\exists b^*)(\langle b, b^*\rangle \in E \wedge \langle b^*, b^{**}\rangle \in E)$ iff $\langle b, b^{**}\rangle \in E$. It is the use of the compositional properties that forces us to involve E and C_X in the definition of the automaton, as the properties for E^g rely on E. $\square$

With Theorem 4 at hand, Büchi obtained the following Corollary. Its proof is similar to that of Corollary 3, relying on Theorem 4 instead of Theorem 1.

Corollary 5 (Büchi)**.** *The monadic theory of* $(\omega; <)$ *is decidable.*

The result can be extended to all countable ordinals. But first let us pass to a class of automata which is more flexible already in the case of inputs of length ω. With the more flexible class we will be able to recover the equivalence between deterministic and non-deterministic automata. The equivalence fails in the case of Büchi automata, and in fact Theorem 4 would fail if "Büchi automaton" were replaced by "deterministic Büchi automaton."

Given $s\colon \gamma \to S$, with γ a limit ordinal, define $\mathrm{cf}(s)$ to be the set of states which occur cofinally along s. Precisely, $b \in \mathrm{cf}(s)$ iff $(\forall \xi < \gamma)(\exists \delta)(\xi < \delta < \gamma \wedge s(\delta) = b)$. $\mathrm{cf}(s)$ is an element of $\mathcal{P}(S)$.

A finite state automaton with a *countable-limit condition* consists of the usual core, S, I, and T as before, and a function $\Psi_{ctbl}\colon \mathcal{P}(S) \to S$. A *run* of the automaton on input $X\colon \alpha \to \Sigma$ of countable length α is a sequence $s\colon \alpha + 1 \to S$ which satisfies conditions (Initial), (Succ) for all $\xi < \alpha$, and

$$s(\gamma) = \Psi_{ctbl}(\mathrm{cf}(s \restriction \gamma)) \qquad \text{(LimCtbl)}$$

for all limit ordinals $\gamma \leq \alpha$.

As usual the automaton is *deterministic* if I is a singleton and the transition table is the graph of a function. In the case of a deterministic automaton, for countable input length α, conditions (Initial), (Succ), and (LimCtbl) determine a unique run of the automaton on X. The run is produced by setting $s(0) = I$, using condition (Succ) to uniquely determine $s(\xi + 1)$ for successor

ordinals $\xi + 1 \leq \alpha$, and using condition (LimCtbl) to uniquely determine $s(\gamma)$ for limit ordinals $\gamma \leq \alpha$.

Reverting to the initial approach to acceptance, we equip the automaton with a set $F \subseteq S$ of accepting final states. A run s on input $X: \alpha \to \Sigma$ is *accepting* if $s(\alpha) \in F$, and the automaton *accepts* X if it has an accepting run on X.

Acceptance in the case of $\alpha = \omega$ is determined via a table on the basis of $\mathrm{cf}(s \restriction \omega)$; accepting runs are those with $\mathrm{cf}(s \restriction \omega) \in \Psi^{-1}(F)$. This method is due to Muller. A Muller automaton consists of the usual core, S, I, and T, and an *acceptance table* $B \subseteq \mathcal{P}(S)$. A run of the automaton on input $X: \omega \to \Sigma$ is a sequence $s: \omega \to \Sigma$ satisfying conditions (Initial) and (Succ) for $n < \omega$. The run is accepting iff $\mathrm{cf}(s) \in B$.

It is clear that every Büchi automaton is equivalent to a Muller automaton. It is not hard to see that the converse is also true, for the non-deterministic case. But deterministic Muller automata are more expressive than deterministic Büchi automata. Indeed, the more flexible acceptance condition in Muller automata allows a determinising construction, due to McNaughton [13], producing a deterministic automaton equivalent to a given non-deterministic automaton.

Theorem 6 (McNaughton)**.** *Every Muller automaton is equivalent to a deterministic Muller automaton.*

The proof is very intricate, substantially more intricate than the proofs given above. We refer the reader to Khoussainov–Nerode [9, §3.8]. The determinism construction has an element of uniformity that is not present in Büchi's original proof of decidability of the monadic theory of $(\omega; <)$. That uniformity allowed Büchi [3] to generalize the construction to all countable domains:

Theorem 7 (Büchi)**.** *Every automaton with countable-limit condition is equivalent to a deterministic automaton with countable-limit condition on countable domains. (The equivalent automaton is independent of the domain.)*

Using Theorem 7 one can very directly imitate the proof of Theorem 1 and obtain:

Theorem 8 (Büchi)**.** *For every monadic formula φ, there is a deterministic automaton with a countable-limit condition, $\mathcal{A}$, which is equivalent to φ on countable domains.*

Corollary 9 (Büchi)**.** *The set of monadic sentences φ so that* $(\exists \alpha < \omega_1)$ $(\alpha; <) \models \varphi$ *is decidable.*

Proof sketch. Similar to the proof of Corollary 9, but this time searching for *generalized paths* in the directed graph of the automaton. A generalized path is a sequence of vertices $\{b_i\}$ so that for each i either (1) the graph has an edge from b_i to b_{i+1}; or (2) $b_i = b_k$ for $k \leq i$ (this gives rise to a *loop*)

and $b_{i+1} = \Psi_{ctbl}(\{b_k, \ldots, b_i\})$. The second condition corresponds to a use of condition (LimCtbl), which allows the automaton to reach state b_{i+1} if it had generated a limit-length sequence of states that repeats $\{b_k, \ldots, b_i\}$. $\square$

With a more careful analysis one can prove further that for each (non-zero) countable ordinal α, the monadic theory of $(\alpha; <)$ is decidable, and depends only on the remainder obtained when dividing α by ω^ω.

The first and second uncountable cardinals

When dealing with monadic theories of uncountable ordinals one has to take into account notions from set theory including clubs, stationarity, and to some extent cofinality.

Recall that an ordinal β is a *limit point* of a set of ordinals A if $A \cap \beta$ is unbounded in β, meaning that $(\forall \xi < \beta)(\exists \delta)(\xi < \delta < \beta \wedge \delta \in A)$. A set $C \subseteq \kappa$ is *closed unbounded* in an ordinal κ, club in κ or simply club for short, if: (a) C is unbounded in κ, and (b) C is closed in κ, meaning that every limit point $\beta < \kappa$ of C is an element of C.

A function on ordinals $f \colon \tau \to \kappa$ is *cofinal* in κ if its range $\{f(\xi) \mid \xi < \tau\}$ is unbounded in κ. The *cofinality* of κ, denoted $\mathrm{cof}(\kappa)$, is the smallest ordinal τ so that there is a function $f \colon \tau \to \kappa$ cofinal in κ. An ordinal τ is *regular* if $\mathrm{cof}(\tau) = \tau$. Regular ordinals are in fact cardinals, and if $\tau = \mathrm{cof}(\kappa)$ then τ is regular. Thus the cofinality of an ordinal κ is always a cardinal.

For κ of uncountable cofinality, any two club subsets of κ have non empty, and in fact club, intersection. For such κ we say that a set $A \subseteq \kappa$ is *stationary* in κ if it meets—meaning it has non-empty intersection with—every club subset of κ. The notion is non trivial as every club in κ is stationary. Using the axiom of choice every stationary set can be split into two disjoint sets which are both stationary, so there are stationary sets which are not club.

For $\tau \geq \omega_1$, if κ has cofinality $\geq \tau$ then every club in κ has points of all cofinalities $< \tau$, and vice versa. In the next paragraph we write $(*)$ to refer to this equivalence.

The notions limit point, club, and stationary are clearly expressible in the monadic language. Cofinality need not be expressible in the monadic language, since its definition uses functions. But using the equivalence $(*)$, and the fact that $\mathrm{cof}(\kappa) \geq \omega_0$ iff κ is a limit ordinal, it is easy to define by recursion, for finite n, monadic formulas $\varphi_{\mathrm{cof} \geq \omega_n}$ so that $(\mathrm{On}; <) \models \varphi_{\mathrm{cof} \geq \omega_n}[\kappa]$ iff $\mathrm{cof}(\kappa) \geq \omega_n$.

Work with a cardinal κ of cofinality $\geq \omega_1$. For sets $A, B \subseteq \kappa$ define $A \sim B$ iff A and B are equal on a club, meaning that there is a club $C \subseteq \kappa$ so that $A \cap C = B \cap C$. Since the intersection of two club subsets of κ is itself club, $\sim$ is an equivalence relation. We write $[A]$ to denote the equivalence class of A. Let $\mathcal{B}_\kappa$ be the set of equivalence classes of $\sim$. The basic operations on sets, union, intersection, and difference, extend naturally to operations on the equivalence class. Abusing notation slightly we use the same symbols to denote these

operations on classes, writing for example $[A] \cap [B] = [C]$ if $A \cap B = C$. The structure $(\mathcal{B}_\kappa; \cup, \cap, -, \emptyset, \kappa)$ is a Boolean algebra, and since the notions used to define it are all expressible in the monadic language, the first order theory of $(\mathcal{B}_\kappa; \cup, \cap, -, \emptyset, \kappa)$ is computable from the monadic theory of $(\kappa; <)$.

In fact much more may be computed from the monadic theory. For $A \subseteq \kappa$ define $R(A)$ to be the set of $\alpha < \kappa$ so that $A \cap \alpha$ is stationary in α. Such α are *reflection points* of A. R is trivial in the case of $\kappa = \omega_1$, but its behavior is highly non-trivial, and indeed independent of ZFC, already at $\kappa = \omega_2$. R extends to act on $\sim$ equivalence classes, and it is clear that the theory of $(\mathcal{B}_\kappa; R, \cup, \cap, -, \emptyset, \kappa)$ is also computable from the monadic theory of $(\kappa; <)$.

Shelah [18] used intricate model theoretic arguments to provide converses to these observations. He showed that the monadic theory of $(\omega_1; <)$ can be reduced to the first order theory of $(\mathcal{B}_\kappa; \cup, \cap, -, \emptyset, \kappa)$, and the monadic theory of $(\omega_2; <)$ can be reduced to the first order theory of $(\mathcal{B}_\kappa; R, \cup, \cap, -, \emptyset, \kappa)$. His results are more general, and reduce the monadic theories of higher cardinals κ to first order theories of structures that extend $(\mathcal{B}_\kappa; \cup, \cap, -, \emptyset, \kappa)$ with operations additional to R, collecting more operations as κ increases.

$(\mathcal{B}_{\omega_1}; \cup, \cap, -, \emptyset, \omega_1)$ is an atomless Boolean algebra and its theory is decidable. Thus it follows from Shelah's reduction that the monadic theory of $(\omega_1; <)$ is decidable. This had been proved previously by Büchi [4], using automata that he defined acting on sequences of length ω_1. In later work Büchi–Zaiontz [5] proved the following theorem, and in fact characterized completely the monadic theories of ordinals below ω_2.

Theorem 10 (Büchi–Zaiontz)**.** *For every $\alpha < \omega_2$, the monadic theory of $(\alpha; <)$ is decidable.*

At ω_2 matters change drastically. Shelah's reduction shows that the monadic theory of $(\omega_2; <)$ and the first order theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$ are each computable from the other. But the complexity of the first order theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$ is independent from ZFC.

It is helpful to divide ω_2 into two parts, C_0 and C_1, with C_i consisting of the ordinals $\alpha < \omega_2$ of cofinality ω_i. Both are stationary. Subsets of C_1 do not reflect, so in analyzing the operation R on $\mathcal{B}_{\omega_2}$ we are concerned only with its behavior on subsets of C_0.

Assuming mild large cardinals Magidor [12] showed it is consistent to have $R(A) = C_1$ (modulo the equivalence relation $\sim$) for every $A \subseteq C_0$. In this case R is trivial on $\mathcal{B}_{\omega_2}$, and the first order theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$ is decidable.

There are other behaviors of R that result in a decidable theory. Shelah [19] shows, assuming just the consistency of ZFC, that it is consistent that for every $A \subseteq C_0$ and every stationary $\hat{B}, \hat{C}$ with $\hat{B} \cup \hat{C} = R(A)$, there are stationary $B, C \subseteq A$ so that $R(B) = \hat{B}$ and $R(C) = \hat{C}$. (Again equality is modulo $\sim$.) From this principle too it follows that the theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$ is decidable, though R is not trivial.

On the other hand Gurevich–Magidor–Shelah [8] construct models where the theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$ has arbitrary Turing degree, assuming mild large cardinals. Lifsches–Shelah [10] construct models, assuming just the consistency of ZFC, where the theory is arbitrarily complicated.

In short, the theory of $(\mathcal{B}_{\omega_2}; R, \cup, \cap, -, \emptyset, \omega_2)$, and equivalently the monadic theory of $(\omega_2; <)$, cannot be determined from ZFC, and:

Theorem 11 (Shelah, Gurevich–Magidor–Shelah, Lifsches–Shelah)**.** *The decidability of the monadic theory of* $(\omega_2; <)$ *is independent of* ZFC.

Reflection principles may affect not just decidability, but also definability. Using the monadic formulas $\varphi_{\text{cof} \geq \omega_n}$ defined above it is easy to see that each of the cardinals ω_n, for finite n, is definable by a monadic formula over $(\text{On}; <)$: ω_n is the least ordinal of cofinality $\geq \omega_n$. Magidor [12] construct from large cardinals a model in which ω_{n+1} is also definable. In Magidor's model every stationary subset of $\omega_{\omega+1}$ reflects. This universal reflection fails on the cardinals ω_n (the set of points of cofinality ω_{n-1} does not reflect), and since the property is expressible in the monadic language, one can write a monadic formula which in Magidor's model defines $\omega_{\omega+1}$. Then using the equivalence $(*)$ above and a recursive definition similar to the definition of the formulas $\varphi_{\text{cof} \geq \omega_n}$, it follows that $\omega_{\omega+1+n}$ is definable for each finite n.

One can build on this argument to construct models where other regular cardinals are definable. Note that the argument skips the singular cardinal ω_ω. We shall see in the next section that singular cardinals can never be defined by a monadic formula over $(\text{On}; <)$.

Automata capturing monadic truth on all domains

The *almost-all* language, which we shall use below, is obtained from the monadic language by removing the first order quantifiers, and adding instead the quantifiers $(\forall^* \xi)$ and $(\forall^* \xi < \delta)$. The semantics of the first quantifier is given by the following condition: $(\alpha; <) \models (\forall^* \xi)\varphi(\xi)$ iff

1. $\text{cof}(\alpha) \geq \omega_1$; and
2. There is a club $C \subseteq \alpha$ so that $(\alpha; <) \models \varphi[\xi]$ for all $\xi \in C$.

(In writing the condition we suppressed the instantiated variables of $(\forall^* \xi)\varphi(\xi)$, for notational simplicity.) The semantics of the second quantifier is given by a similar condition requiring that $\text{cof}(\delta) \geq \omega_1$, and that $(\alpha; <) \models \varphi[\xi]$ on a club $C \subseteq \delta$.

We saw already that the properties of having uncountable cofinality and of being club can be expressed in the monadic langauge. It follows from this that the almost-all language is a fragment of the monadic language. The almost-all language is strictly less expressive than the monadic language. For example, the truth value of an almost-all formula about sets $A_1, \ldots, A_k \subseteq \alpha$ depends only on the restriction of these sets to a club in α. In other words

the language cannot distinguish between tuples $\langle A_1, \ldots, A_k \rangle$ and $\langle A'_1, \ldots, A'_k \rangle$ provided $A_i = A'_i$ on a club. In particular the almost-all language cannot express equality on sets. (In the full monadic language equality can be expressed: $A = B$ iff $(\forall \xi)(\xi \in A \leftrightarrow \xi \in B)$.) Nonetheless it is expressive enough to capture an important essence of the monadic truth. For example, the results in the previous section may be viewed as reducing full monadic truth on ω_1 and ω_2 to truth in the almost-all language, and then reasoning about the decidability or undecidability of this almost-all truth.

More uniformly, almost-all truth can be used as a foundation for decision making at limit stages, in a class of automata that capture monadic truth on all domains. In this section we define the class, and see how it is connected to monadic truth on ordinals.

We begin with some preliminary definitions. The notation $f : \alpha \rightharpoonup S$ indicates that f is a *partial* function from α into S. Assuming S is finite the function can be coded by a tuple of subsets of α as follows. Let $b_0, \ldots, b_{n-1}$ be the elements of S, and let $A_i = \{\xi < \alpha \mid f(\xi) = b_i\}$ for $i < n$. Then $\langle A_0, \ldots, A_{n-1} \rangle$ codes f. When we write $\varphi(\ldots, f, \ldots)$ below, with φ an almost-all formula and $f : \alpha \rightharpoonup S$, we mean $\varphi(\ldots, A_0, \ldots, A_{n-1}, \ldots)$.

Define $\mathrm{cd}(f) : \alpha \rightharpoonup \mathcal{P}(S)$, the *cofinal-state derivative* of f, by $\mathrm{cd}(f)(\gamma) = \mathrm{cf}(f \restriction \gamma)$ for each limit ordinal $\gamma < \alpha$, and $\mathrm{cd}(f)(\gamma)$ undefined otherwise. $\mathrm{cd}(f)$ too can be coded by a tuple $\langle D_0, \ldots, D_{n-1} \rangle$ of n subsets of α, setting $\gamma \in D_i$ iff $b_i \in \mathrm{cd}(f)(\gamma)$ iff $b_i \in \mathrm{cf}(f \restriction \gamma)$. When we write $\varphi(\ldots, \mathrm{cd}(f), \ldots)$, with φ an almost-all formula and $f : \alpha \rightharpoonup S$, we mean $\varphi(\ldots, D_0, \ldots, D_{i-1}, \ldots)$.

Given a finite sequence of formulas $\vec{\psi} = \langle \psi_0, \ldots, \psi_{l-1} \rangle$ in the almost-all language, and functions $s : \alpha \to S$ and $r : \alpha \rightharpoonup S$, define $\mathrm{Tf}_{\vec{\psi}}(s, r)$, the $\vec{\psi}$ *fragment of the almost-all truth table of s,* $\mathrm{cd}(s)$*, and* r, to be the set of $i < l$ so that $(\alpha; <) \models \psi_i[s, \mathrm{cd}(s), r]$. This is the restriction of the set of almost-all formulas which are true of $\langle s, \mathrm{cd}(s), r \rangle$, to the finite fragment specified by $\vec{\psi}$. Given further a function $\Psi : \mathcal{P}(l) \to S$, define $(\Psi \oplus \vec{\psi})(s, r)$ to be $\Psi(\mathrm{Tf}_{\vec{\psi}}(s, r))$.

We are ready now to define our final class of finite state automata. We work as usual with a an alphabet Σ. A finite state automaton with *full-limit condition*, $\mathcal{A}$, consists of the following objects. The objects in conditions (A1) and (A2) are similar to objects we have seen before, for the basic core of an automaton, and for countable limits which in our context will generalize to all limits of countable cofinality. The objects in condition (A3) will be used to determine the state $s(\lambda)$ for limit λ of uncountable cofinality. The objects in condition (A4) will be used to determine the extra component r of a run of $\mathcal{A}$.

A1. A finite set of states S, a set of initial states $I \subseteq S$, and a transition table $T \subseteq S \times \Sigma \times S$.

A2. A lower-limit function $\Psi_{lo} : \mathcal{P}(S) \to S$.

A3. A finite sequence of formulas $\vec{\psi} = \langle \psi_0, \ldots, \psi_{l-1} \rangle$ in the almost-all language, and a higher-limit function $\Psi_{hi} : \mathcal{P}(l) \to S$.

A4. A finite set P of *pebbles*, and two functions which will be used in placing and removing pebbles, $u\colon S \to \{U \mid U \subsetneq P\}$, and $h\colon S \to P$ with $h(b) \in P - u(b)$ for all $b \in S$.

A *run* of the automaton on input $X\colon \alpha \to \Sigma$, is a pair of functions $s\colon \alpha + 1 \to S$ and $r\colon \alpha \rightharpoonup S$ satisfying the following conditions. The first three we have seen before, and they will now apply to the initial state, successor states, and all limit states of countable cofinality. The fourth condition will apply to limits of uncountable cofinality, and the fifth will determine the additional component r of the run. We shall say more about these rules below.

$$s(0) \in I \tag{Initial}$$

$$\langle s(\xi), X(\xi), s(\xi+1)\rangle \in T \tag{Succ}$$

$$s(\lambda) = \Psi_{lo}(\mathrm{cf}(s\restriction\lambda)) \tag{LimCtbl}$$

$$s(\lambda) = (\Psi_{hi} \oplus \vec{\psi})(s\restriction\lambda, r\restriction\lambda) \tag{LimHi}$$

If there exists some $\gamma > \xi$ so that $h(s(\xi)) \notin u(s(\gamma))$ then $r(\xi) = s(\gamma)$ (Peb)
for the least such γ, and otherwise $r(\xi)$ is undefined.

As usual the automaton is deterministic if I is a singleton and T is the graph of a function. It is conceptually easier to explain how the conditions above govern the behavior of a deterministic automaton, so we do this first, and comment on the natural extension to non-deterministic automata later.

A deterministic automaton $\mathcal{A}$ should be viewed as running over input $X\colon \alpha \to \Sigma$ and producing a run $\langle s, r\rangle$ through a transfinite sequence of stages. In the initial stage the automaton sets $s(0)$ equal to the unique element of I. In each subsequent stage β the automaton determines $s(\beta)$ through one of the conditions (Succ), (LimCtbl), and (LimHi), depending on whether β is a successor, a limit of countable cofinality, or a limit of uncountable cofinality. If β is a successor, say $\xi+1$, then the automaton sets $s(\xi+1) = T(s(\xi), X(\xi))$, determining the state $s(\xi+1)$ on the basis of the state $s(\xi)$ and input $X(\xi)$, as usual. If β is a limit ordinal of countable cofinality, then the automaton sets $s(\beta) = \Psi_{lo}(\mathrm{cf}(s\restriction\beta))$, determining $s(\beta)$ on the basis of the cofinal set of the run $s\restriction\beta$ produced so far. Finally, if β is a limit of uncountable cofinality then the automaton sets $s(\beta) = (\Psi_{hi} \oplus \vec{\psi})(s\restriction\beta, r\restriction\beta)$, determining $s(\beta)$ on the basis of a fragment of the almost-all theory of the run $s\restriction\beta$ produced so far, its cofinal-state derivative $\mathrm{cd}(s\restriction\beta)$, and the auxiliary sequence $r\restriction\beta$. The sequence $\vec{\psi}$ defines the window of formulas to be consulted, and Ψ_{hi} converts the resulting fragment of the theory into a state.

The auxiliary sequence r is determined using the objects P, u, and h in condition (A4), subject to condition (Peb). We think of P as a finite set of pebbles. The functions h and u are used to place and remove pebbles as follows. Having determined the state $s(\beta)$, the automaton places a pebble $p = h(s(\beta))$ on the ordinal β. The pebble p remains on β until a later stage β^* is reached with $p \notin u(s(\beta^*))$. At the first such stage β^* the automaton removes the pebble from β, and sets $r(\beta) = s(\beta^*)$. This is expressed precisely

in condition (Peb). $r(\beta)$ remains undefined until the pebble placed on β is removed, and may indeed remain undefined throughout, if the pebble is not removed at all during the run. The use of pebbles therefore introduces a delay into part of the construction of a run. This delay is essential in the proofs of Theorems 12 and 13 below.

The value of $r\restriction\beta$ known by stage β, call it $(r\restriction\beta)^{\text{local}}$, is not the same as the final value $r\restriction\beta$ known by the end of the run, after stage α, as there may be ordinals $\xi < \beta$ so that the pebble $h(s(\xi))$ placed on ξ is removed at a stage $\gamma \geq \beta$. But there may only be finitely many such ordinals, since the number of pebbles is finite and since no pebble is ever located on two ordinals at the same stage (to see this use the restriction $h(b) \notin u(b)$ in condition (A4)). Thus $(r\restriction\beta)^{\text{local}}$ and $r\restriction\beta$ may only differ on a finite set.

When reaching a limit stage β the automaton looks at the value of $r\restriction\beta$ known by stage β, setting $s(\beta)$ equal to $(\Psi_{hi} \oplus \vec{\psi})(s\restriction\beta, (r\restriction\beta)^{\text{local}})$. This assignment satisfies condition (LimHi) since $(r\restriction\beta)^{\text{local}}$ and $r\restriction\beta$ differ only on a finite set, and the almost-all theory cannot distinguish such a difference.

Runs of non-deterministic automata are governed by conditions (LimCtbl), (LimHi), and (Peb) in exactly the way described above. Deterministic and non-deterministic automata differ only in the initial and successor stages, where conditions (Initial) and (Succ) require a non-deterministic automaton to make a choice.

As usual we equip the automaton with a set F of *accepting final states.* A run $\langle s, r\rangle$ on input $X\colon \alpha \to \Sigma$ is *accepting* iff its last state $s(\alpha)$ belongs to F. As usual the automaton *accepts* X iff there is an accepting run of the automaton on X.

The main result connecting this class of automata to monadic truth, and the purpose behind the definition of the class, is the following theorem:

Theorem 12 (Neeman)**.** *For every monadic formula φ, there is a deterministic automaton with a full-limit condition, $\mathcal{A}$, which is equivalent to φ on all ordinal domains.*

The theorem is a corollary to the following result on determinism, in much the same way that Theorem 1 is a corollary to Lemma 2.

Theorem 13 (Neeman)**.** *Every automaton with full-limit condition is equivalent to a deterministic automaton with full-limit-condition. The equivalence holds on all ordinal domains, and the equivalent automaton is independent of the domain.*

Theorem 13 extends the work of Büchi [3] and Büchi–Zaiontz [5] to automata acting on inputs of lengths ω_2 and greater. The specific details of the definition of automata with full-limit condition above are of course important to the proof of the theorem. The proof, and the uses of the various aspects of the definition, are beyond the scope of this paper. We refer the reader to Neeman [14].

Since automata may refer to the almost-all theory of the run constructed to set limit states, Theorem 12 may be viewed as reducing questions about the monadic theory to questions about the almost-all theory. This kind of reduction can also be made using model theoretic techniques on individual cardinals, see Shelah [18], with increasing complexity as one moves to higher cardinals. What makes Theorem 12 particularly useful is its uniformity. To each monadic formula φ it assigns a single deterministic automaton which performs the reduction to the almost-all theory on *all* domains. This uniformity, and properties of the almost-all theories of ordinals, for example the fact that the almost-all theory of $(\alpha; <)$ depends only on the cofinality of α, allow deriving the following result from Theorem 12.

Theorem 14 (Neeman)**.** *No singular cardinal is definable over* $(\mathrm{On}; <)$ *by a monadic formula.*

The result was extended in Neeman [15], again using Theorem 12, to show further:

Theorem 15 (Neeman)**.** *An ordinal is definable over* $(\mathrm{On}; <)$ *by a monadic formula iff it can be obtained, using ordinal addition and multiplication, from regular cardinals which are definable over* $(\mathrm{On}; <)$ *by monadic formulas.*

The uniform reduction from monadic theory to almost-all theory given by Theorem 12 may also help obtain results on the monadic theory of ordinals in future forcing extensions which manipulate the almost-all theory of all regular cardinals. At the moment though we only know how to force useful almost-all theories (useful for the purpose of connections with monadic theories) at low cardinals. The following theorem of Neeman [16] reaches ω_3. The model involved satisfies $2^{\omega_2} = \omega_4$, and the construction does not generalize to higher cardinals even if one were willing to let 2^{ω_2} rise further.

Theorem 16 (Neeman)**.** *It is consistent (assuming the consistency of* ZFC*) that the monadic theory of* $(\omega_3; <)$ *is decidable.*

Much more remains to be discovered on the monadic theory of ordinals. Is it consistent that for every ordinal α the monadic theory of α is decidable? Is it consistent that $\aleph_{\omega+1}$ is not definable by a monadic formula over $(\mathrm{On}; <)$? By Theorem 12, both questions are in fact questions about the almost-all theories of the mentioned ordinals.

Theorem 12 uses the axiom of choice. In fact already at the level of ω_1 known proofs of decidability of the monadic theory use some fragment of the axiom of choice, see Litman [11]. What happens when this fragment of the axiom of choice fails? In particular, is the monadic theory of $(\omega_1; <)$ decidable under the axiom of determinacy? Very little is known that may help with this question.

Second order theories are typically very complicated. But within second order theories the monadic theories should be relatively manageable, and it is not unreasonable to hope that further research should shed light on the questions above.

References

[1] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.

[2] J. Richard Büchi. On a decision method in restricted second order arithmetic. In *Logic, Methodology and Philosophy of Science (Proc. 1960 Internat. Congr .)*, pages 1–11. Stanford Univ. Press, Stanford, Calif., 1962.

[3] J. Richard Büchi. Decision methods in the theory of ordinals. *Bull. Amer. Math. Soc.*, 71:767–770, 1965.

[4] J. Richard Büchi. The monadic second order theory of ω_1. In *The monadic second order theory of all countable ordinals (Decidable theories, II)*, pages 1–127. Lecture Notes in Math., Vol. 328. Springer, Berlin, 1973.

[5] J. Richard Büchi and Charles Zaiontz. Deterministic automata and the monadic theory of ordinals $< \omega_2$. *Z. Math. Logik Grundlag. Math.*, 29(4):313–336, 1983.

[6] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–51, 1961.

[7] Yuri Gurevich. Monadic second-order theories. In *Model-theoretic logics*, Perspect. Math. Logic, pages 479–506. Springer, New York, 1985.

[8] Yuri Gurevich, Menachem Magidor, and Saharon Shelah. The monadic theory of ω_2. *J. Symbolic Logic*, 48(2):387–398, 1983.

[9] Bakhadyr Khoussainov and Anil Nerode. *Automata theory and its applications*, volume 21 of *Progress in Computer Science and Applied Logic*. Birkhäuser Boston Inc., Boston, MA, 2001.

[10] Shmuel Lifsches and Saharon Shelah. The monadic theory of $(\omega_2, <)$ may be complicated. *Arch. Math. Logic*, 31(3):207–213, 1992.

[11] Ami Litman. On the monadic theory of ω_1 without A.C. *Israel J. Math.*, 23(3-4):251–266, 1976.

[12] Menachem Magidor. Reflecting stationary sets. *J. Symbolic Logic*, 47(4):755–771 (1983), 1982.

[13] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

[14] Itay Neeman. Finite state automata and monadic definability of singular cardinals. To appear, J. of Symbolic Logic, June 2008.

[15] Itay Neeman. Monadic definability of ordinals. To appear, Computational Prospects of Infinity, Part II: Presented Talks, Vol. 15, Lecture Notes Series, Institute of Mathematical Sciences, National University of Singapore.

[16] Itay Neeman. Monadic theory of ω_3. To appear.

[17] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. Develop.*, 3:114–125, 1959.

[18] Saharon Shelah. The monadic theory of order. *Ann. of Math. (2)*, 102(3):379–419, 1975.

[19] Saharon Shelah. A weak generalization of MA to higher cardinals. *Israel J. Math.*, 30(4):297–306, 1978.