On Monadic Theories of Monadic Predicates

Wolfgang Thomas

RWTH Aachen University, Lehrstuhl Informatik 7, 52056 Aachen, Germany
thomas@informatik.rwth-aachen.de

For Yuri Gurevich on the occasion of his 70th birthday

Abstract. Pioneers of logic, among them J.R. Büchi, M.O. Rabin, S. Shelah, and Y. Gurevich, have shown that monadic second-order logic offers a rich landscape of interesting decidable theories. Prominent examples are the monadic theory of the successor structure $\mathcal{S}_1 = (\mathbb{N}, +1)$ of the natural numbers and the monadic theory of the binary tree, i.e., of the two-successor structure $\mathcal{S}_2 = (\{0,1\}^*, \cdot 0, \cdot 1)$. We consider expansions of these structures by a monadic predicate P. It is known that the monadic theory of $(\mathcal{S}_1, P)$ is decidable iff the weak monadic theory is, and that for recursive P this theory is in Δ_3^0, i.e. of low degree in the arithmetical hierarchy. We show that there are structures $(\mathcal{S}_2, P)$ for which the first result fails, and that there is a recursive P such that the monadic theory of $(\mathcal{S}_2, P)$ is Π_1^1-hard.

Key words: Monadic Second-Order Logic, Tree Automata, Decidability

1 Introduction

Over the past century, starting with Löwenheim [16] in 1915, monadic second-order logic has been developed as a framework in which decision procedures can be provided for interesting theories of high expressive power. In building this rich domain of effective logic, two techniques were crucial. The first was based on the correspondence between monadic second-order formulas and finite automata. This "match made in heaven" (cf. Vardi [28]) was first established for weak monadic second-order logic over the successor structure $\mathcal{S}_1 = (\mathbb{N}, +1)$ by Büchi, Elgot, and Trakhtenbrot. Büchi [2] and Rabin [19] extended this to the full monadic second-order theory of $\mathcal{S}_1$ and of the binary tree $\mathcal{S}_2 = (\{0,1\}^*, \cdot 0, \cdot 1)$. The logic-automata connection first led to the decidability of $\mathrm{MT}(\mathcal{S}_1)$ and $\mathrm{MT}(\mathcal{S}_2)$, the monadic second-order theories of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively (or shorter: the "monadic theory" of these structures). The results were extended to many further logical systems and led to new approaches in verification, data base theory, and further areas of computer science.

The second technique, technically more demanding but more general in its scope, is the "composition method" as developed by Shelah [24] (building on earlier work by Ehrenfeucht, Fraïssé, Läuchli, and others). The idea here is to consider finite fragments of a theory and to compose such theory-fragments according to the combination of models. The method has been applied successfully over orderings, trees, and graphs. Over orderings, the "combination" is concatenation. Shelah's work provided a deep analysis of monadic theories of orderings where automata do not help (or at least are hard to imagine), for example over dense orderings.

In both approaches, Yuri Gurevich has played a central role and contributed most influential papers. For the automata theoretic approach, it might suffice to recall his path-breaking work with Harrington [12] on the monadic second-order theory of the binary tree. As an example of his papers involving the composition method, we mention the work [13,14] which explains over which "short" orderings (neither embedding ω_1 nor its reverse) the monadic theory is decidable. For the reader who wants to enter the field, Yuri's survey *Monadic second-order theories* [11] is still the first choice.

In the present paper, a very small mosaic piece is added to this rich picture. We consider the expansions of the binary tree $\mathcal{S}_2$ by recursive monadic predicates P. We study which complexity (on the scale of recursion theory) the monadic second-order theory of such an expansion $(\mathcal{S}_2, P)$ can have, and we compare the weak and the strong monadic second-order theory of the structures $(\mathcal{S}_2, P)$.

As a starting point we take the corresponding results on expansions of the successor structure $\mathcal{S}_1$ by recursive predicates. We recall (in Sect. 2) that for recursive $P \subseteq \mathbb{N}$, the monadic theory of $(\mathcal{S}_1, P)$ belongs to a low level of the arithmetical hierarchy, namely to the class Δ^0_3. It is also known that for any monadic predicate P, the unrestricted monadic theory of $(\mathcal{S}_1, P)$ is decidable iff the weak monadic theory is (where set quantification is restricted to finite sets). In contrast, we show in Sect. 3 that for recursive P the monadic theory of $(\mathcal{S}_2, P)$, which in general is confined to the analytical class Δ^1_2, can be Π^1_1-hard. In Sections 4 and 5 we prove that there is a predicate P such that the weak monadic theory of $(\mathcal{S}_2, P)$ is decidable but the full monadic theory is undecidable. For the proofs, both the automata theoretic and the composition method are useful.[1]

We assume that the reader is familiar with the basics of the subject. We use standard terminology on monadic theories, automata, and recursion theory (see, e.g., [10,11,21,27]).

[1] The second result should be attributed to the late Andrei Muchnik; it is stated in a densely written abstract *Automata on infinite objects, monadic theories, and complexity* of the Dagstuhl seminar report [7] of 1992. This abstract, written jointly by A. Muchnik and A.L. Semenov, lists – in a dozen of lines – ten topics and results, among them "an example of predicate on tree for which the weak monadic theory is decidable and the monadic theory undecidable". A manuscript with Muchnik's proof does not seem to exist. The talk itself, which was a memorable scientific event appreciated by all who attended (among them the present author), dealt with a different result, the "Muchnik tree iteration theorem"; see for example [1].

2 The Monadic Theory of Structures $(\mathcal{S}_1, P)$

Let us recall some well-known facts on structures $(\mathcal{S}_1, P)$. First we remark that for recursive P the theory $\mathrm{MT}(\mathcal{S}_1, P)$ may be undecidable:

Proposition 1. *There is a recursive predicate $P \subseteq \mathbb{N}$ such that $\mathrm{MT}(\mathcal{S}_1, P)$ (and even the first-order theory $\mathrm{FT}(\mathcal{S}_1, P)$) is undecidable.*

Proof. Let Q be a non-recursive, recursively enumerable set of natural numbers with effective enumeration $j_0, j_1, j_2, \ldots$. From this enumeration we define P. We present the characteristic sequence χ_P of P (with $\chi_P(i) = 1$ iff $i \in P$, else $\chi_P(i) = 0$):

$$\chi_P \quad = \quad 1\ 0^{j_0}\ 1\ 0^{j_1}\ 1\ 0^{j_2}\ 1 \quad \ldots \ .$$

Clearly χ_P (and hence P) is recursive. We have

$$n \in Q \ \text{ iff } \ (\mathcal{S}_1, P) \ \models \ \exists x (P(x) \ \wedge \ \bigwedge_{i=1}^{n} \neg P(x+i) \ \wedge \ P(x+n+1)) \ ,$$

where $x+i$ indicates the i-fold application of "+1" to x. So Q is 1-reducible even to the first-order order theory of $(\mathcal{S}_1, P)$. Hence also $\mathrm{MT}(\mathcal{S}_1, P)$ is undecidable. □

The set $\mathbb{P}$ of prime numbers gives an interesting example of a predicate P where the status of $\mathrm{MT}(\mathcal{S}_1, P)$ is unknown. Observing that we can express the order relation $<$ over $\mathbb{N}$ in monadic logic over $\mathcal{S}_1$, we note that the (open) twin prime hypothesis is expressible by the sentence

$$\forall x \exists y \ (x < y \ \wedge \ \mathbb{P}(y) \ \wedge \ \mathbb{P}(y+2)) \ .$$

Hence it will be hard to show decidability of $\mathrm{MT}(\mathcal{S}_1, \mathbb{P})$; for a detailed analysis see [4]. On the other hand, no "natural" examples of predicates P are known such that $\mathrm{MT}(\mathcal{S}_1, P)$ is undecidable. The known undecidability results rely on predicates built for the purpose, as in Proposition 1 above.

The conversion of monadic formulas into automata provides nice examples of predicates P where $\mathrm{MT}(\mathcal{S}_1, P)$ is decidable. We use the results of Büchi [2] and McNaughton [17] which together yield a transformation from monadic formulas to deterministic ω-automata: *For each monadic second-order formula $\varphi(X)$ in the monadic second-order language of $\mathcal{S}_1 = (\mathbb{N}, +1)$ one can construct a deterministic Muller automaton $\mathcal{A}_\varphi$ such that for each predicate Q*

$$\mathcal{S}_1 \models \varphi[Q] \ \text{ iff } \ \mathcal{A}_\varphi \text{ accepts } \chi_\mathrm{Q} \ .$$

We can use the left-hand side for a *fixed* predicate P, replacing in $\varphi(X)$ each occurrence of X by the predicate constant P. Then we have for each *sentence* φ of the monadic second-order language of the structure $(\mathcal{S}_1, P)$:

$$(\mathcal{S}_1, P) \models \varphi \ \text{ iff } \ \mathcal{A}_\varphi \text{ accepts } \chi_P \ .$$

This reduces the decision problem for the theory $\mathrm{MT}(\mathcal{S}_1, P)$ to the following acceptance problem Acc_P: *Given a Muller automaton $\mathcal{A}$ over the input alphabet $\{0,1\}$, does $\mathcal{A}$ accept χ_P?*

This reduction can be exploited in a concrete way, regarding example predicates P, and also in a general way, regarding the recursion theoretic complexity of theories $\mathrm{MT}(\mathcal{S}_1, P)$.

Concrete examples of predicates P such that $\mathrm{MT}(\mathcal{S}_1, P)$ is decidable were first proposed by Elgot and Rabin [9], namely, the set of factorial numbers, the set of k-th powers and the set of powers of k, for each $k > 1$. The idea is to solve the acceptance problem Acc_P as follows: A given automaton $\mathcal{A}$ accepts χ_P iff $\mathcal{A}$ accepts a modified sequence χ' where the distances between successive letters 1 are contracted below a certain length (a contracted 0-segment just should induce the same state transformation as the original one and should cause the automaton to visit the same states as the original one). In each of the cases mentioned above (factorials, k-th powers, powers of k), the contracted sequence χ' turns out to be ultimately periodic (where phase and period depend on $\mathcal{A}$). So one can decide whether $\mathcal{A}$ accepts χ' and hence whether it accepts χ_P. The method has been extended to further predicates (see e.g. [8]), and criteria for the decidability of $\mathrm{MT}(\mathcal{S}_1, P)$ have been developed in [23,4,22].

For the general aspect we analyze the acceptance problem Acc_P for a Muller automaton $\mathcal{A} = (S, \Sigma, s_0, \delta, \mathcal{F})$ in more detail. As usual, we write S for the set of states, Σ for the input alphabet, s_0 for the initial state, δ for the transition function from $S \times \Sigma$ to S, and $\mathcal{F} \subseteq 2^S$ for the acceptance component; recall that $\mathcal{A}$ accepts an input word α if the set of states visited infinitely often in the unique run of $\mathcal{A}$ on α coincides with a set in $\mathcal{F}$. Let us write $\delta(s_0, \alpha[0,j])$ for the state reached by $\mathcal{A}$ after processing the initial sement $\alpha(0)\dots\alpha(j)$. Then, taking $\alpha = \chi_P$, the automaton $\mathcal{A}$ accepts χ_P iff the following condition holds:

$$(*)_{\mathcal{A},P} \qquad \bigvee_{F\in\mathcal{F}} \Big(\bigwedge_{s\in F} (\forall i \exists j > i \;\; \delta(s_0, \chi_P[0,j]) = s) \\ \wedge \bigwedge_{s\in S\setminus F} (\neg \forall i \exists j > i \;\; \delta(s_0, \chi_P[0,j]) = s) \Big) \; .$$

Assuming that P is recursive, we obtain a reduction of the decision problem for $\mathrm{MT}(\mathcal{S}_1, P)$ to Boolean combinations of conditions that are in Π^0_2; note that the condition $\delta(s_0, \chi_P[0,j]) = s$ can be decided if P is recursive. By relativization, and using recursion theoretic terminology, we obtain for arbitrary $P \subseteq \mathbb{N}$:

$$\mathrm{MT}(\mathcal{S}_1, P) \;\le_{\mathrm{tt}}\; P'' \; .$$

Here $\le_{\mathrm{tt}}$ is truth-table reducibility and P'' is the second jump of P. (In [25] it is shown that the slightly sharper bounded truth-table reducibility does not suffice.) We conclude the following fact, first noted in [3]:

Proposition 2. ([3]) *For each recursive $P \subseteq \mathbb{N}$, the theory $\mathrm{MT}(\mathcal{S}_1, P)$ belongs to the class Δ^0_3 of the arithmetical hierarchy.*

In particular, it is not possible to show the undecidability of a theory $\mathrm{MT}(\mathcal{S}_1, P)$ by a reduction of true first-order arithmetic to it.

A second consequence of the formulation $(*)_{\mathcal{A},P}$ is a reduction of the strong monadic language over $(\mathcal{S}_1, P)$ to the weak monadic language. For this we observe that the condition $(*)_{\mathcal{A},P}$ from above can be formalized in the weak monadic language over $(\mathcal{S}_1, P)$; note that the statement "$\delta(s_0, \chi_P[0, x]) = s$" involves only a finite run (up to position x) and hence can be expressed by a weak monadic formula $\psi_s(y)$. This shows that for each monadic sentence φ one can construct an equivalent weak monadic sentence φ' such that $(\mathcal{S}_1, P) \models \varphi$ iff $(\mathcal{S}_1, P) \models \varphi'$ (in φ' we use a definition of $<$ in weak monadic logic over $\mathcal{S}_1$). So we obtain:

Proposition 3. *For each $P \subseteq \mathbb{N}$: $\mathrm{MT}(\mathcal{S}_1, P)$ is decidable iff $\mathrm{WMT}(\mathcal{S}_1, P)$ is decidable.*[2]

Our aim in the subsequent sections is to show that both propositions fail when we consider the binary tree $\mathcal{S}_2$ instead of the successor structure $\mathcal{S}_1$.

3 A Recursive Predicate Where $\mathrm{MT}(\mathcal{S}_2, P)$ is Π^1_1-Hard

In the same way as described above for theories $\mathrm{MT}(\mathcal{S}_1, P)$, the automata theoretic approach can be applied to study the complexity of the monadic theory of an expansion $(\mathcal{S}_2, P)$ of the binary tree. Here we identify a structure $(\mathcal{S}_2, P)$ with a $\{0, 1\}$-labelled tree t_P which has label 1 at node u iff $u \in P$. We know from Rabin's Tree Theorem [19] that for each monadic sentence φ in the language of $(\mathcal{S}_2, P)$ one can construct a Rabin tree automaton $\mathcal{A}_\varphi$ such that

$$(\mathcal{S}_2, P) \models \varphi \quad \text{iff} \quad \mathcal{A}_\varphi \ \text{ accepts } \ t_P \ .$$

For recursive P, the right-hand side is a Σ^1_2-statement of the form $\exists X \forall Y \psi(X, Y)$ with first-order formula ψ, namely, "there is an $\mathcal{A}_\varphi$-run on t_P such that each infinite path of this run satisfies the Rabin acceptance condition". Since Rabin automata are closed under complement, the statement can also be phrased in Π^1_2-form. This proves the first statement of the following result:

Theorem 1. *For recursive $P \subseteq \{0, 1\}^*$, the theory $\mathrm{MT}(\mathcal{S}_2, P)$ belongs to the class Δ^1_2, and there is a recursive $P \subseteq \{0, 1\}^*$ such that $\mathrm{MT}(\mathcal{S}_2, P)$ is Π^1_1-hard.*

For the proof of the second statement we have to find a recursive P such that a known Π^1_1-complete set is reducible to $\mathrm{MT}(\mathcal{S}_2, P)$. As Π^1_1-complete set we use a coding of *finite-path trees* (cf. [21, Ch. 16.3]). We work with the infinitely branching tree $\mathcal{S}_\omega$ whose nodes are sequences $(n_1, \dots n_k)$ of natural numbers. The empty sequence is the root, and the nodes $(n_1, \dots, n_k, i)$ are the successors of $(n_1, \dots, n_k)$. Paths in $\mathcal{S}_\omega$ are defined accordingly. We say that a subset S of $\mathcal{S}_\omega$ *defines a finite-path tree* if S is closed under taking predecessors and if it does not contain an infinite path. For a recursion theoretic treatment, we use a computable bijective coding of the finite sequences over $\mathbb{N}$ by natural

[2] Although this Proposition is very close to Proposition 2, a result of [3], it was left as an open problem in [3]. In a more general context an answer was then given in [26].

numbers, writing $\langle n_1, \ldots, n_k \rangle$ for the code of $(n_1, \ldots, n_k)$. Furthermore, we refer to a standard numbering of the partial recursive functions; we write f_e for the function with number e. A function $f : \mathbb{N} \to \mathbb{N}$ is the characteristic function of a finite-path tree if

1. f is total and has only values 0 or 1,
2. the set $\{(n_1, \ldots, n_k) \mid f(\langle n_1, \ldots, n_k \rangle) = 1\}$ defines a finite-path tree.

Let

$$\mathrm{FPT} = \{e \in \mathbb{N} \mid f_e \text{ is characteristic function of a finite-path tree}\} .$$

We use the following fact (see [21, Ch. 16.3]):

Proposition 4. FPT *is a Π^1_1-complete set of natural numbers.*

Proof of Theorem 1: It suffices to define a recursive set P of nodes of the *binary* tree $\mathcal{S}_2$ such that for each number e we can construct a monadic second-order sentence φ_e with

$$e \in \mathrm{FPT} \text{ iff } (\mathcal{S}_2, P) \models \varphi_e .$$

We build the structure $(\mathcal{S}_2, P)$ as a sequence of $\{0,1\}$-labelled trees $t_0, t_1, \ldots$ attached to the rightmost branch of $\mathcal{S}_2$. So the root of t_e is the node $r_e := 1^e0$. In the tree t_e we obtain a copy of $\mathcal{S}_\omega$: Its node $(n_1, \ldots, n_k)$ is coded by $r_e1^{n_1+1}01^{n_2+1} \ldots 1^{n_k+1}0$. The predicate P will only apply to nodes of the leftmost branch starting in such a node. We define P by attaching labels 0 and 1 to the nodes $r_e1^{n_1+1}01^{n_2+1} \ldots 1^{n_k+1}0^i$ for $i = 1, 2, 3 \ldots$. All other nodes get label 0 by default.

In order to define the labelling, we imagine an effective procedure $\mathcal{P}$ that computes, in a dovetailed fashion, the values $f_e(\langle n_1, \ldots, n_k \rangle)$ of all functions f_e simultaneously. So the procedure treats each pair $(e, \langle n_1, \ldots, n_k \rangle)$ again and again, and when dealing with this pair it progresses with the computation of $f_e(\langle n_1, \ldots, n_k \rangle)$ for one further step (unless a value has been computed already). Consider the i-th step of $\mathcal{P}$ $(i = 1, 2, 3, \ldots)$. It will determine the bit label attached to all the nodes $r_e1^{n_1+1}01^{n_2+1} \ldots 1^{n_k+1}0^i$, reporting on the current status of the computation of $f_e(\langle n_1, \ldots, n_k \rangle)$ at $\mathcal{P}$-step i. If the i-th $\mathcal{P}$-step produces the value $f_e(\langle n_1, \ldots, n_k \rangle)$ then we attach label 1 to the node $r_e1^{n_1+1}0 \ldots 1^{n_k+1}0^i$, for all other nodes $r_{e'}1^{m_1+1}01^{m_2+1} \ldots 1^{m_{k'}+1}0^i$ we attach label 0. In fact, when we find a value for $f_e(\langle n_1, \ldots, n_k \rangle)$, we attach to the nodes $r_e1^{n_1+1}0 \ldots 1^{n_k+1}0^j$ for $j = i, i+1, i+2$ the labels 100, respectively 110, respectively 111, depending on whether the computation of $f_e(\langle n_1, \ldots, n_k \rangle)$ produced value 0, 1, or > 1, respectively. After such a block of letters 1 on the path $r_e1^{n_1+1}0 \ldots 1^{n_k+1}0^\omega$, all subsequent labels will be 0.

Clearly this attachment of labels defines a recursive predicate over $\mathcal{S}_2$. ¿From the labels on the 0^ω-parts of the paths $r_e1^{n_1+1}0 \ldots 1^{n_k+1}0^\omega$ (for fixed e) we can infer whether f_e is a characteristic function, i.e., whether for all tuples $(n_1, \ldots, n_k)$ the value $f_e(\langle n_1, \ldots, n_k \rangle)$ is defined and either 0 or 1: This happens if for all $(n_1, \ldots, n_k)$, on the 0^ω-part of $r_e1^{n_1+1}0 \ldots 1^{n_k+1}0^\omega$ precisely one or two labels 1 occur. (Let us call such a path associated to $(n_1, \ldots, n_k)$ "once 1-labelled", respectively "twice 1-labelled".) So, using P, we can easily express in monadic logic for any given e whether f_e is a characteristic function. The function f_e is the characteristic function *of a finite-path tree* if moreover the nodes $r_e 1^{n_1+1} 0 \ldots 1^{n_k+1} 0$ whose associated path is twice 1-labelled form a set that that is closed under prefixes (i.e., there is no prefix whose associated path is only once 1-labelled), and that each path through $r_e(1^+0)^\omega$ eventually hits a node outside the coded tree, i.e., a node whose associated path is only once 1-labelled. All these conditions can be expressed by a monadic sentence φ_e. Hence we have $e \in \mathrm{FPT}$ iff $(\mathcal{S}_2, P) \models \varphi_e$, as desired. □

4 Some Background on Types and Tree Automata

For the comparison between the weak and the strong monadic theory of structures $(\mathcal{S}_2, P)$, we need some preparations concerning "types" (i.e., finite theory fragments) and concerning tree automata. For a more detailed treatment, the reader can consult [11] or [27].

For the analysis of weak monadic logic over structures $(\mathcal{S}_2, P)$, it is convenient to use a syntax in which only second-order variables $X, Y, Z, \ldots$ are present. As atomic formulas we use $X \subseteq Y$, $\mathrm{Sing}(X)$ ("X is a singleton"), $S_i(X, Y)$ for $i = 0, 1$ ("X, Y are singletons, and the element of X has the element of Y as the i-th successor"), and $X \subseteq P$. Formulas are built up from atomic formulas by means of Boolean connectives and the (weak monadic) quantifiers $\exists, \forall$. It is clear that this relational language is equivalent in expressive power to the original one with first-order and weak monadic second-order quantifiers and the (functional) signature with symbols for the functions $\cdot 0$ and $\cdot 1$.

As in the previous section, we identify a structure $(\mathcal{S}_2, P)$ with a $\{0,1\}$-labelled tree t_P, i.e. with a mapping $t_P : \{0,1\}^* \to \{0,1\}$. Conversely, each $\{0,1\}$-labelled infinite binary tree t induces a structure $(\mathcal{S}_2, P_t)$; we freely use this correspondence and mean by "tree" always a $\{0,1\}$-labelled infinite tree. The set of all these trees is denoted by $T_{\{0,1\}}$. A tree t is *regular* if it has only finitely many non-isomorphic subtrees (or equivalently, if a finite Moore automaton generates t by producing the label $t(u)$ after processing the input word u). It is well-known that a regular tree is definable in the weak monadic language over $\mathcal{S}_2$; so its (weak and strong) monadic theory is decidable.

Let $m > 1$. Two trees s, t are *m-equivalent* (short: $s \equiv_m t$) if they satisfy the same weak monadic sentences (of the relational signature just introduced) of quantifier depth $\leq m$. There are finitely many equivalence classes, called *m-types*. Each m-type τ is definable by a weak monadic sentence φ_τ which again is of quantifier depth m. As finite representations of an m-type τ we use such a sentence φ_τ defining it.

In the sequel we shall work with natural compositions of trees and corresponding compositions of m-types. First we consider the combination of two trees via a 0-labelled or 1-labelled root: For two trees s, t let $0 \cdot \langle s, t \rangle$, respectively $1 \cdot \langle s, t \rangle$, be the tree with a 0-, respectively 1-labelled root and s, t as its left and right subtree. Next, we consider the composition of a given infinite sequence $t_0, t_1, t_2, \ldots$ of trees or of a sequence $(s_0, t_0), (s_1, t_1), \ldots$ of pairs of trees. In the first case we attach the trees $t_0, t_1, \ldots$ along the 0-labelled right-hand branch of the binary tree: We insert the tree t_i at the node 1^i0; i.e., the root of t_0 is node 0, the root of t_1 is 10, etc., and – as mentioned – the right-hand branch 1^ω is labelled 0. The resulting tree we denote as $[t_0, t_1, \ldots]$. In the second case we consider the two sons of the nodes $0, 10, 110$ etc. and insert s_i at the left son of 1^i0 and t_i at the right son of 1^i0. The nodes 1^i and 1^i0 are all labelled 0. We denote the tree obtained in this way as $[(s_0, t_0), (s_1, t_1), \ldots]$.

A simple Ehrenfeucht-Fraïssé type argument now shows the following lemma:

Lemma 1. *Let $m > 1$.*
(a) *The m-types σ of s and τ of t determine the m-types of $0 \cdot \langle s, t\rangle$ and $1 \cdot \langle s, t\rangle$ and these types are computable from σ, τ.*
(b) *If $t_i \equiv_m t'_i$ for $i > 0$ then $[t_1, t_2, \ldots] \equiv_m [t'_1, t'_2, \ldots]$. Similarly, if $s_i \equiv_m s'_i$ and $t_i \equiv_m t'_i$, then $[(s_0, t_0), (s_1, t_1), \ldots] \equiv_m [(s'_0, t'_0), (s'_1, t'_1), \ldots]$.*
(c) *If the sequence $\tau_0, \tau_1, \ldots$ of m-types of $t_0, t_1, \ldots$ is ultimately perdiodic, say of the form $\tau_0 \ldots \tau_{k-1}(\tau_k \ldots \tau_{\ell-1})^\omega$, then the m-type of $[t_0, t_1, \ldots]$ is determined by the types $\tau_1, \ldots, \tau_{\ell-1}$ and computable from them.*

Next we turn to prerequisites from tree automata theory, mainly using the concept of Büchi tree automaton (see e.g. [27] for details) and a fundamental example due to Rabin [20] which shows their expressive weakness in comparison with Rabin tree automata. Rabin presented a tree language T_0 which is definable in monadic logic (or by a Rabin tree automaton) but which is *not* recognizable by a Büchi tree automaton. It is a variant of the language of finite-path trees:

$$T_0 = \{t \in T_{\{0,1\}} \mid \text{on each path of } t \text{ there are only finitely many letters } 1\} \ .$$

We have to recall the construction of Rabin since we exploit it below. For $n \geq 0$ define the tree t_n inductively as follows:

1. t_0 has a 1-labelled root and is otherwise labelled 0.
2. t_{n+1} has a 1-labelled root, otherwise a 0-labelled right-hand branch 1^ω, a 0-labelled left subtree, and a copy of t_n inserted at each node in 1^+0.

So

$$t_n(u) = 1 \ \text{ iff } \ (u = \varepsilon \ \text{ or } u \in 1^+0 + (1^+01^+0) + \ldots + (1^+0)^n) \ .$$

Let us verify that the m-type of t_n determines the m-type of t_{n+1} (and that the latter can be computed from the former): By Lemma 1 (c) we can compute the m-type of the right-hand subtree of the root of t_{n+1} from the m-type of t_n (note that the copies of t_n give a constant and hence periodic sequence of m-types). The left-hand subtree of the root of t_{n+1} is labelled 0; we can compute its m-type (since it is regular). Now Lemma 1 (a) yields the claim. So there is a map F over the finite domain of m-types that produces the m-type of t_{n+1} from the m-type of t_n. Starting with the m-type τ_0 of t_0, we obtain with the values $F^{(i)}(\tau_0)$ an ultimately periodic sequence. We summarize:

Lemma 2. *The m-types of the trees $t_0, t_1, t_2, \ldots$ form a computable ultimately periodic sequence $\tau_0 \ldots \tau_{k-1}(\tau_k \ldots \tau_{\ell-1})^\omega$.*

Clearly, each tree t_n belongs to T_0. We use the following lemma shown in [20] (see also [27]):

Lemma 3. *For each Büchi tree automaton $\mathcal{A}$ with $< n$ states accepting t_n one can construct a regular tree $t'_n \notin T_0$ which is again accepted by $\mathcal{A}$.*

Let us sketch the proof. Assume that the Büchi tree automaton $\mathcal{A}$ with $< n$ states and the set F of final states accepts t_n. Then one can construct a regular run ϱ of $\mathcal{A}$ on t_n (since t_n is regular and accepted). We define a path in ϱ as follows: Pick a node $u_1 = 1^{k_1}$ on the right-hand branch where $\varrho(1^{k_1}) \in F$. Pick a node $u_2 = 1^{k_1}01^{k_2}$ on the right-hand branch starting in $1^{k_1}0$ where again $\varrho(u_2) \in F$, and so on until such a node $u_n = 1^{k_1}01^{k_2} \ldots 1^{k_n}$ with $\varrho(u_n) \in F$ is chosen. These nodes exist since on each path of ϱ infinitely many visits of F occur. Now $t_n(u_i0) = 1$ for $i = 1, \ldots, n$ by definition of t_n. Since $\mathcal{A}$ has $< n$ states, there are u_i, u_j with $i < j$ such that $\varrho(u_i) = \varrho(u_j)$; observe that between these nodes a 1-labelled node of t_n occurs (for example at u_i0). Repeating the t_n-segment determined by the path segment from u_i (included) to u_j (excluded) indefinitely, we obtain a regular tree t'_n which is accepted by $\mathcal{A}$ and which has a path with infinitely many labels 1.

A set of trees definable in weak monadic logic is easily seen to be recognized by a Büchi tree automaton. So the lemma also shows that T_0 is not definable in weak monadic logic.

5 Comparing Weak and Strong Monadic Logic

The aim of this section is to show the following:

Theorem 2. *There is a predicate $P \subseteq \{0,1\}^*$ such that $\mathrm{WMT}(\mathcal{S}_2, P)$ is deciadable and $\mathrm{MT}(\mathcal{S}_2, P)$ undecidable.*

We shall start with a tree t_ω which for each given quantifier depth m is m-equivalent to an effectively constructible regular tree. This gives us the decidability of the weak monadic theory of t_ω. Then we modify t_ω first to a tree s_ω and then to a tree t'_ω such that for each quantifier depth m the trees t_ω, s_ω, and t'_ω cannot be distinguished by m-types from some computable level onwards (which ensures that the weak monadic theory of t'_ω is also decidable). However, t'_ω will be constructed such that in the full monadic theory an undecidability proof as for Proposition 1 can be carried through.

Proof of Theorem 2: Define, using the trees t_i of the previous section,

$$t_\omega := [(t_0, t_0), (t_1, t_1), (t_2, t_2), \ldots] \ .$$

By Lemma 2, for each given m the tree t_ω is m-equivalent to an effectively constructible regular tree; just take a fixed representative for each m-type τ that appears in the ultimately periodic sequence of m-types of the trees $0 \cdot \langle t_0, t_0 \rangle, 0 \cdot \langle t_1, t_1 \rangle, \ldots$ (use Lemma 2 and Lemma 1 (a)). Hence the weak monadic theory of t_ω is decidable.

As a next step we now construct a tree s_ω from t_ω. First we pick, for each m-type τ ($m = 1, 2, \ldots$), a Büchi tree automaton $\mathcal{A}_\tau$ that defines τ. Let n_τ be the number of states of $\mathcal{A}_\tau$. Define

$$N_m := \max\{n_\tau \mid \tau \text{ is } m\text{-type}\} \ +1 \ .$$

These numbers N_m will be called *special* below.

Consider t_{N_m}; denote by τ its m-type. Then $\mathcal{A}_\tau$ accepts t_{N_m}. The number of states of $\mathcal{A}_\tau$ is $< N_m$. By Lemma 3, we can construct a tree $t'_{N_m} \notin T_0$ that is again accepted by $\mathcal{A}_\tau$; so its m-type is τ. We conclude

$$(*) \qquad t_{N_m} \equiv_m t'_{N_m} \text{ and also } t_{N_i} \equiv_m t'_{N_i} \ \text{ for } i > m \ .$$

Now let s_ω be obtained from $t_\omega = [(t_0, t_0), (t_1, t_1), (t_2, t_2), \ldots]$ by replacing, for each $m > 1$, the pair (t_{N_m}, t_{N_m}) of subtrees by (t'_{N_m}, t_{N_m}). By Lemma 1 (b), for each m, the subtree of s_ω with root 1^{N_m} is m-equivalent to the corresponding subtree of t_ω (note that $t_{N_i} \equiv_m t'_{N_i}$ for $i \geq m$). Hence also s_ω is m-equivalent to an effectively constructible regular tree, and thus its weak monadic theory is decidable.

We now focus on the "special numbers" N_m (including N_0 which is set to 0). A tree node 1^n is called special if n is special. It is worth noting that the set of special tree nodes 1^n of s_ω is definable in monadic logic: A node 1^n is special iff the subtree with root 1^n00 does not belong to the tree language T_0, which in turn is definable in (strong!) monadic logic.

Now, copying Proposition 1, we code a non-recursive, recursively enumerable set Q with enumeration $j_0, j_1, \ldots$ on the domain S of special numbers. We introduce a *marker* on a special number N_i when in the proof of Proposition 1 the value 1 was chosen for i. So the number N_0 is marked, the next j_0 special numbers are unmarked, the special number N_{j_0+1} is marked, the next j_1 special numbers are unmarked, and so on. For each *marked* N_i we modify the entry (t'_{N_i}, t_{N_i}) of s_ω to (t'_{N_i}, t'_{N_i}), thus obtaining the desired tree t'_ω, or in other words, the desired predicate P over $\mathcal{S}_2$. Again we call a node 1^n marked if n is marked.

We finish the proof by verifying that the weak monadic theory of t'_ω is decidable and that the strong monadic theory of t'_ω is undecidable.

For the first claim, one observes, using $(*)$, that exactly as for the tree s_ω, also t'_ω is m-equivalent to an effectively constructible regular tree, for each given $m > 0$. For the second claim we use the following equivalence, regarding the considered non-recursive set Q: $n \in Q$ iff there are two marked and special nodes $1^k, 1^{k'}$ in t'_ω such that there are exactly n special nodes between them, all of them unmarked. Clearly this condition is expressible in monadic logic. Hence Q is 1-reducible to the monadic theory of t'_ω $(=: (\mathcal{S}_2, P))$. □

6 Conclusion

The study of the monadic theory of structures $(\mathcal{S}_2, P)$ with monadic predicate P seems far from finished. Let us list three open problems.

1. More examples of predicates P shold be found such that $\mathrm{MT}(\mathcal{S}_2, P)$ is decidable. The "contraction method" of Elgot and Rabin [9] has been transferred to the binary tree by Montanari and Puppis [18], but it seems that not many interesting predicates are (as yet) manageable by this approach. For example, consider predicates induced by the binary representations (or inverse binary representations) of numbers of interesting sets $S \subseteq \mathbb{N}$. For the powers of 2, the corresponding predicate P with the nodes $0, 10, 100, \ldots$ is a definable set, whence the monadic theory of $(\mathcal{S}_2, P)$ is decidable. What about the corresponding predicate for the set of squares?
2. The lack of a deterministic automaton model over trees (capturing monadic logic over the binary tree) may be considered as the deeper reason for the result of Sect. 3 that the theory $\mathrm{MT}(\mathcal{S}_2, P)$ can be non-arithmetical for recursive P. However, this leaves open the question whether an undecidability proof for such a theory can be done via a reduction of true first-order arithmetic. A partial (negative) answer follows from work of Gurevich and Shelah [15] on the uniformization problem for monadic logic over $\mathcal{S}_2$; it is shown there that no well-ordering of $\mathcal{S}_2$ exists that is definable in monadic logic. A more recent treatment, also covering definability in structures $(\mathcal{S}_2, P)$ with monadic P, is given in [5,6]: The structure $(\mathbb{N}, <)$ (and hence $(\mathbb{N}, +, \cdot)$) is not monadic second-order interpretable in a structure $(\mathcal{S}_2, P)$ (even with non-recursive P) when the universe $\mathbb{N}$ is represented by the full domain of the binary tree.
3. A natural question, already raised at the end of Rabin's paper [20] (and attributed there to H. Gaifman), is concerned with decidability of weak definability: Can one decide for a monadic formula $\varphi(X_1, \ldots, X_n)$ interpreted over $\mathcal{S}_2$ whether it is equivalent to a formula of weak monadic logic?

7 Acknowledgment

Many thanks are due to Nachum Dershowitz for his patience and help and to Christof Löding and Alex Rabinovich for their comments.

References

1. D. Berwanger, A. Blumensath, The monadic theory of tree-like structures, in: E. Grädel et al. (eds.) *Automata, Logics, and Infinite Games*, Springer LNCS 2500 (2002), pp. 285-302.
2. J.R. Büchi, On a decision method in restricted second-order arithmetic, in: E. Nagel et al., *Logic, Methodology, and Philosophy of Science: Proceedings of the 1960 International Congress*, Stanford Univ. Press 1962, pp. 1-11.

3. J.R. Büchi, L.H. Landweber, Definability in the monadic second-order theory of successor, *J. Symb. Logic* 34 (1969), 166-170.
4. P. T. Bateman, C. G. Jockusch, and A. R. Woods, Decidability and undecidability of theories with a predicate for the primes, *J. Symb. Logic* 58 (1993), 672-687.
5. A. Carayol, C. Löding, MSO on the infinite binary tree: Choice and order, in *Proc. CSL 2007*, Springer LNCS 4646 (2007), 161-176.
6. A. Carayol, C. Löding, D. Niwiński, I. Walukiewicz, Choice functions and well-orderings over the infinite binary tree, *Central Europ. J. of Math.* (to appear).
7. K. Compton, J.E. Pin, W. Thomas (eds.), *Automata Theory: Infinite Computations*, Dagstuhl Seminar Report 9202 (1992).
8. O. Carton, W. Thomas, The monadic theory of morphic infinite words and generalizations, *Information and Computation* 176 (2002), 51-76.
9. C.C. Elgot, M.O. Rabin, Decidability and undecidability of extensions of second (first) order theory of (generalized) successor, *J. Symb. Logic* 31 (1966), 169-181.
10. E. Grädel, W. Thomas, Th. Wilke (Eds.), *Automata, Logics, and Infinite Games*, Springer LNCS 2500 (2002).
11. Y. Gurevich, Monadic theories, in: J. Barwise, S. Feferman (Eds.), *Model-Theoretic Logics*, Springer-Verlag, Berlin 1985, pp. 479-506.
12. Y. Gurevich, L. Harrington, Trees, automata, and games, in *Proc. 14th STOC* (1982), 60-65.
13. Y. Gurevich, Modest theory of short chains, *J. Symb. Logic* 44 (1979), 481-490.
14. Y. Gurevich, S. Shelah, Modest theory of short chains II, *J. Symb. Logic* 44 (1979), 491-502.
15. Y. Gurevich, S. Shelah, Rabin's uniformization problem, *J. Symb. Logic* 48 (1983), 1105-1119.
16. L. Löwenheim, Über Möglichkeiten im Relativkalkül, *Math. Ann.* 76 (1915), 447-470.
17. R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Inf. Contr.* 9 (1966), 521-530.
18. A. Montanari, G. Puppis, A contraction method to decide MSO theories of deterministic trees, in: *Proc. 22nd IEEE Symposium on Logic in Computer Science (LICS)*, pp. 141-150.
19. M.O. Rabin, Decidability of second-order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* 141 (1969), 1-35.
20. M.O. Rabin, Weakly definable relations and special automata, in: Y. Bar-Hillel (Ed.) *Math. Logic and Foundations of Set Theory*, North-Holland, Amsterdam 1970, pp. 1-23.
21. H. Rogers, *The Theory of Recursive Functions and Effective Computability*, McGraw-Hill, New York 1967.
22. A. Rabinovich, W. Thomas, Decidable theories of the ordering of natural numbers with unary predicates, *Proc. CSL 2006*, Springer LNCS 4207 (2006), 562-574.
23. A. Semenov, Decidability of monadic theories, in: *Proc. MFCS 1984*, Springer LNCS 176 (1984), 162-175.
24. S. Shelah, The monadic theory of order, *Ann. Math.* 102 (1975), 379-419.
25. W. Thomas, The theory of successor with an extra predicate, *Math. Ann.* 237 (1978), 121-132.
26. W. Thomas, On the bounded monadic theory of well-ordered structures, *J. Symb. Logic* 45 (1980), 334-338.

27. W. Thomas, Languages, automata and logic, in: G. Rozenberg, A. Salomaa (Eds.), *Handbook of Formal Language Theory*, Vol. 3, Springer-Verlag, New York 1997.
28. Moshe Y. Vardi, Logic and Automata: A match made in heaven, in: *Proc. ICALP 2003*, LNCS 2719 (2003), 64-65.