MSO on the Infinite Binary Tree: Choice and Order

Arnaud Carayol and Christof Löding

RWTH Aachen, Informatik 7, 52056 Aachen, Germany
{carayol,loeding}@i7.informatik.rwth-aachen.de

Abstract. We give a new proof showing that it is not possible to define in monadic second-order logic (MSO) a choice function on the infinite binary tree. This result was first obtained by Gurevich and Shelah using set theoretical arguments. Our proof is much simpler and only uses basic tools from automata theory. We discuss some applications of the result concerning unambiguous tree automata and definability of winning strategies in infinite games. In a second part we strengthen the result of the non-existence of an MSO-definable well-founded order on the infinite binary tree by showing that every infinite binary tree with a well-founded order has an undecidable MSO-theory.

1 Introduction

Our main purpose is to present a simple proof for the fact (first shown by Gurevich and Shelah in [GS83]) that on the infinite binary tree there is no choice function that can be defined in monadic second-order logic (MSO), i.e., in the extension of first-order logic by quantification over sets of elements. A choice function on the infinite binary tree is a mapping assigning to each nonempty set of nodes one element from this set, i.e., the function chooses for each set one of its elements. Such a function is MSO-definable if there is an MSO-formula with one free set variable X and one free element variable x such that for each nonempty set U of nodes there is exactly one element $u \in U$ such that the formula is satisfied if X is interpreted as U and x is interpreted as u.

The question of the existence of an MSO-definable choice function over the infinite binary tree can be seen as a special instance of the more general uniformization problem, which asks, given a relation that is defined by a formula with free variables, whether it is possible to define by another formula a function that is compatible with this relation. More precisely, given a formula $\phi(\bar{X}, \bar{Y})$ with vectors $\bar{Y}, \bar{X}$ of free variables, such that $\forall \bar{X} \exists \bar{Y} \phi(\bar{X}, \bar{Y})$ is satisfiable, uniformization asks for a formula $\phi^*(\bar{X}, \bar{Y})$ such that

1. ϕ^* implies ϕ (each interpretation of $\bar{Y}, \bar{X}$ making ϕ^* true also makes ϕ true),
2. and ϕ^* defines a function in the sense that for each interpretation of $\bar{X}$ there is exactly one interpretation of $\bar{Y}$ making ϕ^* true.

J. Duparc and T.A. Henzinger (Eds.): CSL 2007, LNCS 4646, pp. 161–176, 2007.

The question of the existence of a choice function is the uniformization problem for the formula $\phi(X,y) = X \neq \emptyset \rightarrow y \in X$.

The infinite line, i.e., the structure $(\omega, succ)$ of the naturals with the successor function is known to have the uniformization property for MSO [Sie75]. On the infinite binary tree MSO is known to be decidable [Rab69] but it does not have the uniformization property. This was conjectured in [Sie75] and proved in [GS83] where it is shown that there is no MSO-definable choice function on the infinite binary tree.

The reason for the present paper is that the proof in [GS83] uses complex set theoretical arguments, whereas it appears that the result can be obtained by much more basic techniques. We show that this is indeed true and present a proof that only relies on the equivalence of MSO and automata over infinite trees and otherwise only uses basic techniques from automata theory. Besides its simplicity, another advantage of the proof is that we provide a concrete family of sets (parameterized by natural numbers) such that each formula will fail to make a choice for those sets with the parameters chosen big enough. We use this fact when we discuss two applications of the result concerning unambiguous tree automata (as presented in [NW]) and the definability of strategies in infinite games.

The subject of MSO-definability of choice functions on trees has been studied in more depth in [LS96], where the authors consider more general trees not only the infinite binary tree. They show the following dichotomy: for a tree it is either not possible to define a choice function in MSO, or it is possible to define a well-ordering on the domain of the tree.

We strengthen this result by showing that extending the infinite binary tree by any well-ordering leads to a structure with undecidable MSO-theory. As a consequence we obtain that each structure in which we can MSO-define a well-ordering and MSO-interpret the infinite binary tree must have an undecidable MSO-theory.

The article is structured as follows. In the next section we introduce some notations and basic terminology. In Section 3 we give the proof that there is no MSO-definable choice function on the infinite binary tree and discuss applications of the result. In Section 4 the undecidability of the MSO-theory of the infinite binary tree augmented by any well-ordering is shown.

2 Preliminaries

Words. For a finite alphabet Σ, we write Σ^* for the set of all words over Σ. The length of a word $u \in \Sigma^*$ is denoted by $|u|$ and ε is the empty word. For all words $u, v \in \Sigma^*$, u is a *prefix* of v (written $u \sqsubseteq v$) if there exists $w \in \Sigma^*$ such that $v = uw$. If $w \in \Sigma^+$ then u is a *strict prefix* of v (written $u \sqsubset v$). The *greatest common prefix* of two words u and v (written $u \wedge v$) is the longest word which is a prefix of u and v.

Relational structures. A *signature* is a ranked set of symbol $\mathcal{S}$, where for all $R \in \mathcal{S}$, $|R|$ denotes the arity (which is ≥ 1) of the symbol R. A *relational* *structure* $\mathcal{R}$ over the *signature* $\mathcal{S}$ is given by a tuple $(U, (R^{\mathcal{R}})_{R \in \mathcal{S}})$ where U is the *universe* of $\mathcal{R}$ and where for all $R \in \mathcal{S}$, $R^{\mathcal{R}}$ (which is also called the interpretation of R in $\mathcal{R}$) is a subset of $U^{|R|}$. When $\mathcal{R}$ is clear from the context, we will simply write R instead of $R^{\mathcal{R}}$.

Monadic second-order logic. We adopt the definition of *monadic second-order logic (MSO)* over relational structures with the standard syntax and semantics (see e.g. [EF95] for a detailed presentation). We write $\varphi(X_1, \ldots, X_n, y_1, \ldots, y_m)$ to denote that the free variables of the formula φ are among $X_1, \ldots, X_n$ (monadic second-order) and $y_1, \ldots, y_m$ (first-order) respectively. A formula without free variables is called a *sentence.*

For a relational structure $\mathcal{R}$ and a sentence φ, we write $\mathcal{R} \models \varphi$ if $\mathcal{R}$ *satisfies* the formula φ. The *MSO-theory* of $\mathcal{R}$ is the set of sentences satisfied by $\mathcal{R}$. For every formula $\varphi(X_1, \ldots, X_n, y_1, \ldots, y_m)$, all subsets $U_1, \ldots, U_n$ of the universe of $\mathcal{R}$ and all elements $v_1, \ldots, v_m$ of the universe of $\mathcal{R}$, we write $\mathcal{R} \models \varphi[U_1, \ldots, U_n, v_1, \ldots, v_m]$ to express that φ holds in $\mathcal{R}$ when X_i is interpreted as U_i for all $i \in [1, n]$ and y_j is interpreted as v_j for all $j \in [1, m]$.

Infinite binary labelled trees. An (infinite binary) tree labeled by a finite alphabet Σ is a mapping $t : \{0, 1\}^* \to \Sigma$. We denote by T_Σ the set of all trees labeled by Σ. For a set $U \subseteq \{0, 1\}^*$, we write $t(U) \in T_{\{0,1\}}$ for the characteristic tree of U, i.e., the tree which labels all nodes in U with 1 and all the other nodes with 0. This notation is extended to the case of several sets. The characteristic tree of $U_1, \ldots, U_n \subseteq \{0, 1\}^*$ is the tree labeled by $\{0, 1\}^n$ written $t(U_1, \ldots, U_n)$ and defined for all $u \in \{0, 1\}^*$ by $t(U_1, \ldots, U_n)(u) := (b_1, \ldots, b_n)$ where for all $i \in [1, n]$, $b_i = 1$ if $u \in U_i$ and $b_i = 0$ otherwise.

To every tree t labeled by $\Sigma = \{a_1, \ldots, a_n\}$, we associate a canonical structure over the signature $\{E_0, E_1, P_{a_1}, \ldots, P_{a_n}\}$ where E_0 and E_1 are binary symbols and the P_{a_i} are predicates. The universe of this structure is $\{0, 1\}^*$. The symbols E_0 and E_1 are respectively interpreted as $\{(w, w0) \mid w \in \{0, 1\}^*\}$ and $\{(w, w1) \mid w \in \{0, 1\}^*\}$. Finally for all $i \in [1, n]$, P_{a_i} is interpreted as $\{u \in \Sigma^* \mid t(u) = a_i\}$. In the following, we will not distinguish between a tree and its canonical relational structure.

In particular, for a formula $\varphi(X_1, \ldots, X_n)$ and sets $U_1, \ldots, U_n \subseteq \{0, 1\}^*$, we write $t(U_1, \ldots, U_n) \models \varphi$ to indicate that the infinite binary tree satisfies φ when X_i is interpreted by U_i.

3 Choice

As described in the introduction, an MSO-definable choice function is given by an MSO-formula $\phi(X, x)$ such that

$$\forall X \exists x.\ X \neq \emptyset \to (x \in X \wedge \phi(X, x) \wedge \forall y.\ \phi(X, y) \to x = y)$$

is true over the infinite binary tree. This section is mainly devoted to the proof of the following theorem of Gurevich and Shelah.

Theorem 1 ([GS83]). *There is no MSO-definable choice function on the infinite binary tree.*

The technical formulation of the result we prove is given in Theorem 3, where we concretely provide counter examples for which a given formula cannot choose a unique element. As a machinery for the proof we use tree automata, which are easier to manipulate (at least for our purpose) than formulas. In the following we give some basic definitions. More details on automata for infinite trees can be found in [Tho97].

In this section we take the view of labeled trees as mappings from $\{0,1\}^*$ to the label alphabet (which usually is $\{0,1\}^n$ for some number n).

A parity automaton on Σ-labeled trees is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \Omega)$ with a finite set Q of states, initial state $q_0 \in Q$, transition relation $\Delta \subseteq Q \times \Sigma \times Q \times Q$, and a priority function $\Omega : Q \to \mathbb{N}$. A run of $\mathcal{A}$ on a tree $t \in T_\Sigma$ from a state $q \in Q$ is a tree $\rho \in T_Q$ such that $\rho(\varepsilon) = q$, and for each $u \in \{0,1\}^*$ we have $(\rho(u), t(u), \rho(u0), \rho(u1)) \in \Delta$. We say that ρ is accepting if on each path the minimal priority appearing infinitely often is even. If we only speak of a run of $\mathcal{A}$ without specifying the state at the root, we implicitly refer to a run from q_0.

We extend this model to automata that do not only accept or reject trees but also mark some of the nodes with special marking states. A marking parity automaton (MPA) is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, \Omega, P)$ with an additional set of marking states $P \subseteq Q$. A run ρ of such an automaton defines a set $U_\rho \subseteq \{0,1\}^*$ as the set of those nodes that are labeled by a marking state, i.e., $U_\rho = \rho^{-1}(P)$.

Given a set $U \subseteq \{0,1\}^*$ and an MPA $\mathcal{A}$ on $\{0,1\}$-labeled trees, we define

$$T(\mathcal{A}, U) = \{U_\rho \mid \rho \text{ is an accepting run of } \mathcal{A} \text{ on } t(U)\}.$$

That is, the set of all sets that are marked by $\mathcal{A}$ in an accepting run on $t(U)$.

Theorem 2 ([Rab69]). *For each MSO-formula $\phi(X, Y)$ there is an MPA $\mathcal{A}_\phi$ such that $T(\mathcal{A}_\phi, U) = \{U' \subseteq \{0,1\}^* \mid t(U, U') \models \phi\}$ for each $U \subseteq \{0,1\}^*$.*

For a set $U \subseteq \{0,1\}^*$ we say that $\mathcal{A}$ marks an element u of U if $u \in U$ and there is an accepting run ρ of $\mathcal{A}$ on $t(U)$ such that $U_\rho = \{u\}$. Note that with other runs $\mathcal{A}$ might mark other elements or sets of elements.

For two trees t, t' we say that they are $\mathcal{A}$-equivalent, written as $t \equiv_\mathcal{A} t'$, if for each state q of $\mathcal{A}$ there is an accepting run from q on t iff there is an accepting run from q on t'. Intuitively, this means that $\mathcal{A}$ cannot distinguish the two trees.

3.1 Undefinability of Choice Functions

We now define a family $(U_{M,N})_{M,N}$ of sets such that for each MPA we can find M and N such that this MPA cannot mark a unique element of $U_{M,N}$. To achieve this, we "hide" the elements from the set very deep in the tree such that MPAs up to a certain size are not able to uniquely choose an element that they can mark.

For $M, N \in \mathbb{N}$ the set $U_{M,N} \subseteq \{0,1\}^*$ is defined by the following regular expression $U_{M,N} = \{0,1\}^*(0^N 0^* 1)^M \{0,1\}^*$. Let $t_{M,N} = t(U_{M,N})$. This tree can be obtained by unfolding the finite graph $G_{M,N}$ depicted in Figure 1 from x_M. In this picture, the dashed arrows represent 1-labeled edges leading back to the node x_M. The chains of 0-edges between x_{k+1} and x_k have length N. All nodes in this graph are labeled 0 except x_0, which is labeled by 1.

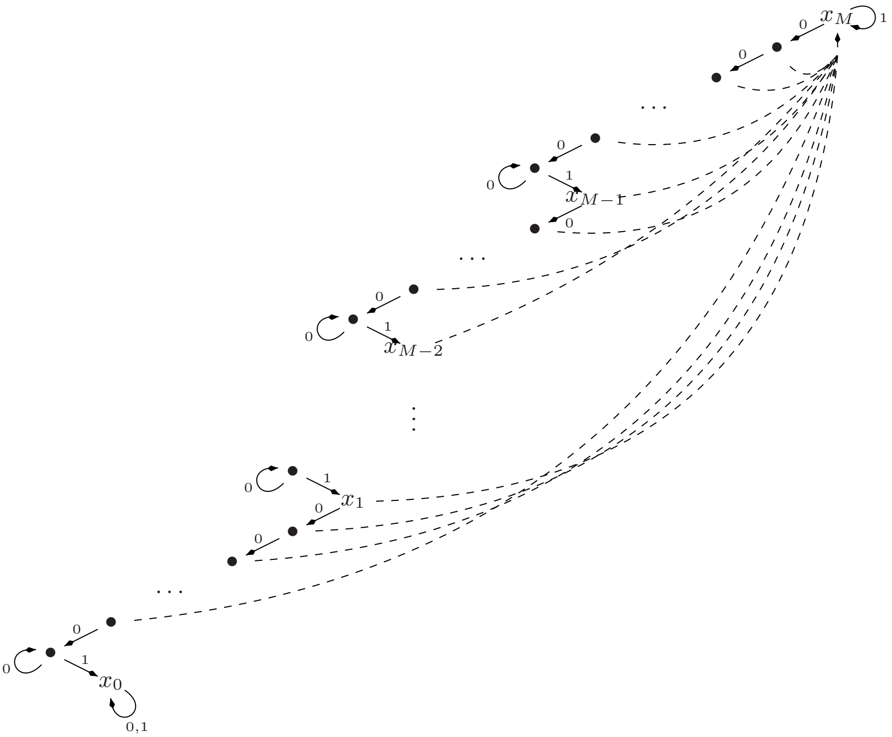

Fig. 1. A representation of the regular tree $t_{M,N}$ by the graph $G_{M,N}$

It is easy to verify that x_0 is reachable from x_M by exactly those paths whose sequence of edge labels is in the set $U_{M,N}$. So $G_{M,N}$ can be viewed as the minimal DFA accepting the language $U_{M,N}$ where x_0 is the only final state. Let $t_{k,M,N}$ denote the tree that we obtain by unfolding the graph $G_{M,N}$ from the node x_k.

We now fix an MPA $\mathcal{A} = (Q, \{0,1\}, q_0, \Delta, \Omega, P)$ on $\{0,1\}$-labeled trees and take $M = 2^{|Q|} + 1$ and $N = |Q| + 1$. For these fixed parameters we simplify the notation by letting $t_k = t_{k,M,N}$. In particular, $t_M = t_{M,M,N} = t_{M,N}$.

We say that a subtree (of some tree t) that is isomorphic to t_k for some k is of type t_k.

Our aim is to trick the automaton $\mathcal{A}$ to show that it cannot choose a unique element from the set $U_{M,N}$. This is done by modifying a run that marks an element u of $U_{M,N}$ such that we obtain another run marking something different.

To understand the general idea, consider the path between x_{k+1} and x_k for some k. If we take a 1-edge before having reached the end of the 0-chain, i.e., if we take a dashed edge in the picture, then we reach a subtree of type t_M. But if we walk to the end of the 0-chain and then move to the right using a 1-edge, then we arrive at a subtree of type t_k. If we show that there is $\ell < M$ such that t_M and t_ℓ are $\mathcal{A}$-equivalent, then this means that $\mathcal{A}$ has no means to identify when it enters the part where taking a 1-edge leads to subtree of type t_ℓ. We then exploit this fact by pumping the run on this part of the tree such that we obtain another run marking something different.

Lemma 1. *There exists $\ell < M$ such that $t_M \equiv_{\mathcal{A}} t_\ell$.*

Proof. We consider for each tree $t \in T_{\{0,1\}}$ the function $f_t : Q \to \{a, r\}$ with $f_t(q) = a$ if there is an accepting run from q on t, and $f_t(q) = r$ otherwise. By definition, two trees t, t' are $\mathcal{A}$-equivalent if $f_t = f_{t'}$. There are at most $2^{|Q|}$ different such functions. By the choice of M there are $1 \leq k_1 < k_2 \leq M$ such that $t_{k_1} \equiv_{\mathcal{A}} t_{k_2}$. Let $k = k_2 - k_1$ and $\ell = M - k$.

We show that we can obtain t_ℓ from t_M by substituting some of the subtrees of type t_{k_2} in t_M by subtrees of type t_{k_1}. As we have seen that t_{k_1} and t_{k_2} are $\mathcal{A}$-equivalent, this suffices to show that t_M and t_ℓ are also $\mathcal{A}$-equivalent.

We know that t_M is obtained by unraveling the graph $G_{M,N}$ (Figure 1) from x_M. One way to obtain t_ℓ is the following. We take a second copy of $G_{M,N}$ and denote in this copy the vertices corresponding to $x_0, \ldots, x_M$ by $x'_0, \ldots, x'_M$. Now we redirect the edge leading to x_{k_2} in the first copy to point to x'_{k_1} in the second copy. It is easy to verify that unravelling this new graph from x_M (in the first copy) yields the tree t_ℓ. And furthermore, this shows that t_ℓ can be obtained by replacing some subtrees of type t_{k_2} in t_M by subtrees of type t_{k_1}. Hence, $t_M \equiv_{\mathcal{A}} t_\ell$. □

The following lemma states that it is impossible for $\mathcal{A}$ to distinguish a unique element of $U_{M,N}$, i.e., it is not possible that $T(\mathcal{A}, U_{M,N}) = \{\{u\}\}$ for some $u \in U_{M,N}$.

Lemma 2. *If $\mathcal{A}$ marks an element of $U_{M,N}$, then $|T(\mathcal{A}, U_{M,N})| > 1$*

Proof. Assume that that there is an accepting run ρ of $\mathcal{A}$ on $t_M = t_{M,N}$ such that $U_\rho = \{u\}$ with $u \in U_{M,N}$. From ρ we construct another accepting run marking a different set of nodes.

For $0 \leq k \leq M$ let u_k denote the maximal prefix of u such that the subtree at u_k is of type t_k. Let ℓ be as in Lemma 1. For $i \geq 0$ we let $v_i = u_{\ell+1}0^i$ and $v'_i = v_i1$. Note that $v_0 = u_{\ell+1}$ and that for $0 \leq i < N$ the subtrees at v'_i are of type t_M, and for $i \geq N$ the subtrees at v'_i are of type t_ℓ.

From Lemma 1 we know that $t_\ell \equiv_{\mathcal{A}} t_M$. Hence, for each accepting run ρ_q of $\mathcal{A}$ from q on t_M we can pick an accepting run ρ'_q of $\mathcal{A}$ from q on t_ℓ.

By the choice of N there are $0 \leq j < j' < N$ such that $\rho(v_j) = \rho(v_{j'})$. For the moment, consider only the transitions taken in ρ on the sequence $v_0, v_1, \ldots$, i.e., on the infinite branch to the left starting from v_0. We now simply repeat the part of the run between v_j and $v_{j'}$ once. The effect is that some of the states that were at a node v'_i for $i < N$ are pushed to nodes v'_i for $i \geq N$, i.e., the states are moved from subtrees of type t_M to subtrees of type t_ℓ. But for those states q we can simply plug the runs ρ'_q that we have chosen above.

More formally, we define the new run ρ' of $\mathcal{A}$ on t_M as follows. On the part that is not in the subtree below v_0 the run ρ' corresponds to ρ. In the subtree at v_0 we make the following definitions, where $h = j' - j$.

- For $i < j'$ let $\rho'(v_i) = \rho(v_i)$ and $\rho'(v'_i) = \rho(v'_i)$.
- For $i \geq j'$ let $\rho'(v_i) = \rho(v_{i-h})$ and $\rho'(v'_i) = \rho(v'_{i-h})$.
- For the subtrees at v'_i for $i < j'$ we take the subrun of ρ at v'_i.
- For the subtrees at v'_i for $j' \leq i < N$ or $i \geq N + h$ we take the subrun of ρ at v'_{i-h}. This is justified because in these cases $\rho'(v'_i) = \rho(v'_{i-h})$ and the subtrees at v'_i and v'_{i-h} are of the same type (both of type t_M or both of type t_ℓ).
- For the subtrees at v'_i for $N \leq i < N+h$ we take the runs ρ'_{q_i} for $q_i = \rho'(v'_i)$. This is justified as follows. From $q_i = \rho'(v'_i)$ and the definition of ρ' we know that $\rho(v'_{i-h}) = q_i$. Hence, there is an accepting run of $\mathcal{A}$ from q_i on t_M. Thus, ρ'_{q_i} as chosen above is an accepting run of $\mathcal{A}$ from q_i on t_ℓ.

This run ρ' is accepting. Furthermore, the state marking u in the run ρ has been moved to another subtree: There are $n \geq N$ and $w \in \{0,1\}^*$ such that $u = u_{\ell+1}0^n w$. In ρ' the state marking u is at $u' = u_{\ell+1}0^{n+h}w$. Hence, we have constructed an accepting run marking a set different from $\{u\}$. □

Of course, the statement is also true if we increase the value of M or N, e.g., if we let $N = M = 2^{|Q|+1}$. Thus, combining Theorem 2 and Lemma 2 we obtain the following.

Theorem 3. *Let $\phi^*(X, x)$ by an MSO-formula. There exists $n \in \mathbb{N}$ such that for each $u \in U_{n,n}$ with $t(U_{n,n}, u) \models \phi^*$ there is $u' \neq u$ with $t(U_{n,n}, u') \models \phi^*$.*

A direct consequence is the theorem of Gurevich and Shelah. The advantage of our proof is that we obtain a rather simple family of counter examples (the sets $U_{M,N}$).

An easy reduction allows us to extend the non-existence of an MSO-definable choice function to the case where we allow a finite number of fixed predicates as parameters. This result has already been shown in [LS98] in an even more general context, but again relying on the methods employed in [GS83].

Corollary 1. *Let $P_1, \ldots, P_n \subseteq \{0,1\}^*$ be arbitrary predicates. There is no MSO-formula $\phi^*(X_1, \ldots, X_n, X, x)$ such that for each nonempty set U there is exactly one $u \in U$ with $t(P_1, \ldots, P_n, U, u) \models \phi^*$.*

Proof. Assume that there are $P_1, \ldots, P_n \subseteq \{0,1\}^*$ and $\phi^*(X_1, \ldots, X_n, X, x)$ such that for each set U there is exactly one $u \in U$ with $t(P_1, \ldots, P_n, U, u) \models \phi^*$. Then the formula

$$\exists X_1, \ldots, X_n \forall X \exists x \; \phi^*(X_1, \ldots, X_n, X, x) \land \forall y \; \phi^*(X_1, \ldots, X_n, X, y) \rightarrow x = y$$

is satisfiable. Hence, by the Rabin Basis Theorem (cf. [Tho97]), there are regular predicates $P_1, \ldots, P_n$ such that

$$t(P_1, \ldots, P_n) \models \forall X \exists x \; \phi^*(X_1, \ldots, X_n, X, x) \wedge \forall y \; \phi^*(X_1, \ldots, X_n, X, y) \rightarrow x = y.$$

Regular predicates are MSO-definable, so let $\psi_1(X_1), \ldots, \psi_n(X_n)$ be formulas defining $P_1, \ldots, P_n$, respectively. Then the formula $\phi'(X, x)$ defined as

$$\exists X_1, \ldots, X_n \; \phi^*(X_1, \ldots, X_n, X, y) \wedge \bigwedge_{i=1}^{n} \psi_i(X_i)$$

describes a choice function, contradicting Theorem 3. □

We point out here that this method only relies on the fact that the property of being a choice function is MSO-definable. So this way of reducing the case with parameters to the parameter free case can be applied whenever the properties of the object under consideration are MSO-definable (at the end of the last section we briefly mention another application of this technique).

3.2 Applications of the Result and Its Proof

We now discuss a few applications of the results presented so far in this section. One immediate application concerns the non-definability of well-founded orders over the infinite binary tree. We skip this subject here because it is treated in detail in the next section.

The first application is about unambiguous tree automata. It is well known that parity automata on infinite trees cannot be determinized. A weaker requirement than determinism is unambiguity. An automaton is called unambiguous if for each object that it accepts there is exactly one accepting run. For example, it is known that all regular languages of infinite words can be accepted by an unambiguous Büchi automaton [CM03] (and deterministic Büchi automata do not suffice to accept all regular ω-languages).

In an unpublished note [NW] Niwiński and Walukiewicz have shown that not every parity tree automaton is equivalent to an unambiguous one.

Theorem 4 ([NW]). *There is no unambiguous parity automaton accepting exactly those $\{0, 1\}$-labeled trees in which at least one node is labeled 1.*

The underlying idea is that the set of 1-labeled nodes represents the set from which an element has to be chosen. Now we assume that $\mathcal{A}$ is a parity automaton accepting the language used in Theorem 4. Using a game that is similar to the emptiness game for tree automata (cf. [Tho97]) one can show that each accepting run on a tree allows us to pick a unique 1-labeled node from the accepted tree in an MSO-definable way. If $\mathcal{A}$ is unambiguous, then this means that for each accepted tree there is a unique run and hence we can build an MSO-formula picking exactly one 1-labeled node for each accepted tree. This yields an MSO-definable choice function.

In fact, the proof in [NW] yields a more general result: Assume that $\mathcal{A}$ is a parity tree automaton (over the alphabet $\{0,1\}$) that does not accept the tree that is completely labeled by 0, i.e., the tree $t(\emptyset)$. Then there is a formula $\phi_{\mathcal{A}}(X,x)$ such that for each $U \subseteq \{0,1\}^*$ for which there is a unique accepting run of $\mathcal{A}$ on $t(U)$, there is a unique element $u \in U$ such that $t(U,u) \models \phi_{\mathcal{A}}$.

We can use this fact to show the following result.

Theorem 5. *There is a regular language $T \subseteq T_{\{0,1\}}$ and a tree $t \in T$ such that there is no parity automaton accepting T that has a unique accepting run for t.*

Proof. Consider the language T of trees with the property that each subtree rooted at a node of the form 1^*0 contains a node labeled 1. The tree t is defined to have all nodes of the form 1^* labeled 0, and as the subtree rooted at the nodes 1^n0 we plug the trees $t_{n,n}$. As each $t_{n,n}$ contains a node labeled 1, we have $t \in T$.

Assume that there is parity automaton $\mathcal{A}$ accepting T that has a unique run on t. Let q be a state of $\mathcal{A}$ that occurs at infinitely many nodes of the form 1^*0 in this run. Let $\mathcal{A}'$ be the automaton $\mathcal{A}$ with initial state q. As $\mathcal{A}$ accepts T it is clear that $\mathcal{A}'$ does not accept the tree $t(\emptyset)$. Furthermore, as the run of $\mathcal{A}$ on t is unique, there are infinitely many n such that $\mathcal{A}'$ has a unique run on $t_{n,n}$. In combination with the result from [NW] that we discussed above, this gives a contradiction to Theorem 3. □

Another application of Theorem 1 concerns the definability of winning strategies in infinite games. In the following we show that there exist game trees that do not admit the definition of winning strategies in MSO.

A game tree is a tuple $\Gamma = (U_1, U_2, \Omega, W)$ where $U_1, U_2 \subseteq \{0,1\}^*$ form a partition of $\{0,1\}^*$, $\Omega : \{0,1\}^* \to \{0,\dots,n\}$ maps the nodes to a finite set of natural numbers, and $W \subseteq \{0,\dots,n\}^\omega$ is the winning condition. A play of Γ starts in ε. If the play is currently in $u \in \{0,1\}^*$, then Player 1 or Player 2, depending on whether $u \in U_1$ or $u \in U_2$, chooses $b \in \{0,1\}$ and the next game position is ub. In the limit, such a play forms an infinite word in $\{0,1\}^\omega$. This infinite word corresponds to an infinite sequence over $\{0,\dots,n\}$ by applying Ω to each prefix. If this sequence is in W, then Player 1 wins, and otherwise Player 2 wins. We identify a play with the corresponding infinite word in $\{0,1\}^\omega$.

A strategy for Player i is a function $f_i : U_i \to \{0,1\}$ and a play γ is played according to f_i if for each prefix $u \in U_i$ of γ, Player 1 uses the strategy to determine the next move, i.e., if $uf_i(u)$ is a prefix of γ. A strategy f_i is winning for Player i if each play γ that is played according to f_i is winning for Player i.

If $W \subseteq \{0,\dots,n\}^\omega$ is a regular ω-language (i.e. MSO-definable), then for each game tree $\Gamma = (U_1, U_2, \Omega, W)$ one of the players has a winning strategy ([BL69]). If we represent the mapping Ω by sets $\Omega_0, \dots, \Omega_n \subseteq \{0,1\}^*$, each Ω_i corresponding to the set of nodes mapped to i by Ω, the set of all trees $t(U_1, U_2, \Omega_0, \dots, \Omega_n)$ such that Player 1 has a winning strategy in $\Gamma = (U_1, U_2, \Omega, W)$ is MSO-definable.

This raises the question whether it is also possible to define winning strategies in MSO. Note that we can represent a strategy for Player 1 by a subset of nodes that contains ε, for each $u \in U_1$ one successor $u0$ or $u1$, and for each $u \in U_2$ both successors $u0$ and $u1$. The plays according to this strategy are exactly the infinite paths contained in this set. If the winning condition is MSO-definable, then the set of all winning strategies for Player 1 is also MSO-definable, i.e., there is a formula $\phi(X_1, X_2, Y_0, \ldots, Y_n, X)$ such that $t(U_1, U_2, \Omega_0, \ldots, \Omega_n, U) \models \phi$ iff U is a winning strategy for Player 1 (for the fixed winning condition W).

We now look at the case where we want to select a single strategy, i.e., we are interested in a formula that defines exactly one strategy. We show in the following that there is even a fixed game tree on which no formula can define a single winning strategy (this tree is similar to the one used in the proof of Theorem 5).

Theorem 6. *There is a game tree $\Gamma = (U_1, U_2, \Omega, W)$ with an MSO-definable winning condition W such that Player 1 has a winning strategy for Γ but there is no MSO-definable winning strategy for Player 1.*

Proof. Consider the following $\{0, 1, 2\}$-labeled tree t such that for each $n \in \mathbb{N}$ the subtree at the node 1^n0 is isomorphic to $t_{n,n}$, and all nodes of the form 1^n are labeled by 2. The labeling of t defines the mapping Ω, i.e., $\Omega(u) = t(u)$. We let $U_2 = \{1^n \mid n \in \mathbb{N}\}$ and $U_1 = \{0, 1\}^* \setminus U_2$. The winning condition W contains all infinite words over $\{0, 1, 2\}$ that do not contain 0 or that contain a 1.

Intuitively, Player 2 can move along the right branch of the game tree. If he continues like this forever, then he loses because only nodes labeled 2 are visited during the play. Otherwise, he moves to the left at some position, that is, to the root of a subtree $t_{n,n}$. Now Player 1 chooses all the following moves and wins if a node labeled 1 is reached eventually. As each subtree $t_{n,n}$ contains a node labeled 1 it is obvious that Player 1 has a winning strategy.

Assume that there is a winning strategy on t that is MSO-definable by a formula $\phi(X)$ (as the game tree is fixed we omit the other free variables). This formula is equivalent to a parity automaton $\mathcal{A}$ that accepts exactly one strategy labeling of t, corresponding to the winning strategy defined by ϕ.

Using $\mathcal{A}$ we can construct a formula $\phi^*(X, x)$ that chooses exactly one $u \in U_{n,n}$ for each $n \in \mathbb{N}$, contradicting Theorem 3. For this we fix an arbitrary order on the states of $\mathcal{A}$. For each n there is at least one state q of $\mathcal{A}$ such that $\mathcal{A}$ accepts exactly one winning strategy on the subtree $t_{n,n}$ of t (namely the state assumed at the root of the subtree $t_{n,n}$ in an accepting run for the unique winning strategy on Γ that is accepted by $\mathcal{A}$). The formula ϕ^* picks the smallest state q with this property and then chooses the element of $U_{n,n}$ that is described by the unique winning strategy accepted by $\mathcal{A}$ from q on $t_{n,n}$. It is not difficult to verify that this is indeed possible in MSO. □

One should note here that the tree constructed in the proof of Theorem 6 (and also the one from Theorem 5) is not too complicated: it belongs to the Caucal hierarchy[1] ([Cau02]). This means that it can be obtained from a regular tree by a finite number of applications of MSO-interpretations and unfoldings, or

[1] In particular, as all graphs in the Caucal hierarchy have a decidable MSO-theory, the tree constructed in the proof of Theorem 6 also has a decidable MSO-theory.

equivalently, it is the transition graph of a higher-order pushdown automaton ([CW03]).

4 Order

A direct consequence of Theorem 1 is that there exists no MSO-definable well-founded order on the nodes of the infinite binary tree. In fact from an MSO-formula $\varphi_{\leq}(x,y)$ defining a well-founded order on the nodes of the infinite binary tree, a choice function $\varphi_{\text{choice}}(x,X) := x \in X \wedge \forall y, y \in X \rightarrow x \leq y$ is easily defined by taking the smallest element of the set. In this section, we prove the following stronger result.

Theorem 7. *The MSO-theory of the full-binary tree together with any well-founded order is undecidable.*

As the infinite binary tree has a decidable MSO-theory [Rab69], the existence of an MSO-definable well-order would contradict Theorem 7. In the particular case of t_{llex}, the infinite binary tree with length-lexicographic order (formally defined below), this result is well-known [BG00]. We show that t_{llex} can be MSO-interpreted in the infinite binary tree with any well-founded order.

Theorem 8. *There exists an MSO-interpretation $\mathcal{I}$ such that for every well-ordered infinite binary tree t, $\mathcal{I}(t)$ is isomorphic to t_{llex}*

As MSO-interpretations preserve the decidability of MSO, Theorem 7 follows from the undecidability of the MSO-theory of t_{llex}. The rest of this section is dedicated to the proof of Theorem 8.

4.1 Well-Ordered Trees

We consider structures over the binary signature $\mathcal{S} = \{E_0, E_1, \leq\}$. We say that an $\mathcal{S}$-structure t is a well-ordered (infinite binary) tree if it is isomorphic to a well-ordered tree with universe $\{0,1\}^*$ where E_0 and E_1 are respectively interpreted as $\{(u, u0) \mid u \in \{0,1\}^*\}$ and $\{(u, u1) \mid u \in \{0,1\}^*\}$, and $\leq$ is interpreted as a well-founded order on $\{0,1\}^*$. Such an $\mathcal{S}$-structure will be referred to as a canonical well-ordered tree. Up to isomorphism, a well-ordered tree is entirely characterized by the well-founded order on the set of words over $\{0,1\}$.

For example, consider the length-lexicographic order $\leq_{\text{llex}}$ defined by: $u \leq_{\text{llex}} v \Leftrightarrow |u| < |v|$ or $(|u| = |v| \,\text{and}\, u \leq_{\text{lex}} v)$ where $\leq_{\text{lex}}$ refers to the standard lexicographic order. This order is well-founded and we write t_{llex} the canonical well-ordered tree associated to $\leq_{\text{llex}}$. The key property of t_{llex} is that it is MSO-definable (up to isomorphism) in the class of well-ordered trees[2].

Proposition 1. *There exists an MSO-formula φ_{llex} such that for every well-ordered tree t, $t \models \varphi_{llex} \Leftrightarrow t \cong t_{llex}$.*

[2] The class of well-ordered trees is itself MSO-definable in the class of $\mathcal{S}$-structures.

Proof. Consider the MSO-formula φ_{llex} over $\mathcal{S}$ expressing that: the root ε is the smallest element for $\leq$, for all nodes $u \in \{0,1\}^*$ and $v \in 1^*$, the successor of the node $u0v$ is the smallest element for the prefix order of the set $u10^* \setminus \{\text{Succ}(u0v') \mid v' \sqsubset v\}$ where $\text{Succ}(u)$ the successor of u for the order $\leq$ and for all nodes $u \in 1^*$, the successor of u is the smallest element for the prefix order of the set $0^+ \setminus \{\text{Succ}(u') \mid u' \sqsubset u\}$.

It is easy to see that t_{llex} satisfies the formula φ_{llex}. It remains to show that for every well-ordered tree t, if t satisfies φ_{llex} then t is isomorphic to t_{llex}.

Let t be a well-founded tree satisfying φ_{llex}. We can assume w.l.o.g that t is a canonical well-ordered tree. It is therefore enough to show that $t = t_{\text{llex}}$.

For all nodes $u \in \{0,1\}$, we write $\text{Succ}_{\text{llex}}(u)$ for the successor of u for the order $\leq_{\text{llex}}$. As by condition 1 the root ε of t is the minimal element of $\leq$, it is enough to establish that for all $u \in \{0,1\}^*$, $\text{Succ}(u) = \text{Succ}_{\text{llex}}(u)$.

Assume by contradiction that this property is not satisfied. Let u_0 be the smallest node u for the order $\leq_{\text{llex}}$ such that $\text{Succ}(u) \neq \text{Succ}_{\text{llex}}(u)$ We distinguish two cases depending whether u_0 contains an occurrence of 0 or not.

If u_0 contains an occurrence of 0 then u_0 can be uniquely written as $u0v$ with $u \in \{0,1\}^*$ and $v \in 1^*$. The successor of u_0 in the order $\leq_{\text{llex}}$ is $\text{Succ}_{\text{llex}}(u_0) = u10^{|v|}$. By condition 2 of the definition of φ_{llex}, $\text{Succ}(u_0)$ is the smallest element for the prefix order of the set $u10^* \setminus \{\text{Succ}(u0v') \mid v' \sqsubset v\}$.

By minimality (for the order $\leq_{\text{llex}}$) of u_0, we have for all $v' \sqsubset v$ that $\text{Succ}(u0v') = \text{Succ}_{\text{llex}}(u0v') = u10^{|v'|}$. Therefore, $\text{Succ}(u_0)$ is the minimal element for the prefix order of the set $u10^* \setminus \{u10^{|v'|} \mid v' \sqsubset v\}$. This implies that $\text{Succ}(u_0) = u10^{|v|} = \text{Succ}_{\text{llex}}(u_0)$ which contradicts the definition of u_0.

If u_0 does not contain an occurrence of 0 then $u_0 \in 1^*$. The successor of u_0 for the order $\leq_{\text{llex}}$ is $0^{|u_0|+1}$. By condition 3 of the definition of φ_{llex}, $\text{Succ}(u_0)$ is the smallest element for the prefix order of the set $0^+ \setminus \{\text{Succ}(u) \mid u \sqsubset u_0\}$.

By minimality (for the order $\leq_{\text{llex}}$) of u_0, we have for all $u \sqsubset u_0$ that $\text{Succ}(u) = \text{Succ}_{\text{llex}}(u) = 0^{|u|+1}$. Therefore, $\text{Succ}(u_0)$ is the minimal element for the prefix order of the set $0^+ \setminus \{0^{|u|+1} \mid u \sqsubset u_0\}$. This implies that $\text{Succ}(u_0) = 0^{|u_0|+1} = \text{Succ}_{\text{llex}}(u_0)$ which contradicts the choice of u_0. □

4.2 Interpreting t_{llex}

We now define the notion of induced well-ordered tree. Consider a canonical well-ordered tree t and a set $U \subseteq \{0,1\}^*$ of nodes which is closed under greatest common prefix (i.e. $u \in U \wedge v \in V \to u \wedge v \in U$) and such that for all $u \in U$, $u0\{0,1\}^* \cap U \neq \emptyset$ and $u1\{0,1\}^* \cap U \neq \emptyset$. The well-ordered tree $t|_U$ induced by U in t has universe U and its signature is interpreted as $E_i^{t|_U} = \{(u,v) \in U^2 \mid v \text{ is the smallest element for } \sqsubseteq \text{ of } ui\{0,1\}^* \cap U\}$ for $i \in \{0,1\}$ and $\leq^{t|_U} = \{(u,v) \in U^2 \mid u \leq^t v\}$. It is easy to check that $t|_U$ is a well-ordered tree.

Lemma 3. *For every MSO-formula φ over $\mathcal{S}$, there exists a formula $\varphi^*(X)$ such that for every canonical well-ordered tree t and set U, $t \models \varphi^*[U]$ if and only if the set U induces a well-ordered tree $t|_U$ on t and $t|_U \models \varphi$.*

Proof. Consider an MSO-formula φ over $\mathcal{S}$. Let $\varphi_{\text{ind}}(X)$ be an $\mathcal{S}$-formula expressing that X satisfies the conditions to induce a full binary tree and let $\varphi'(X)$ be the formula obtained from φ by relativizing the quantifications to X and by replacing $E_i(x,y)$ with $y \in xi\{0,1\}^* \cap X \wedge \forall z, z \in xi\{0,1\}^* \cap X \rightarrow y \sqsubseteq z$ for $i \in \{0,1\}$. It is easy to check that the formula $\varphi^*(X) := \varphi_{\text{ind}}(X) \wedge \varphi'(X)$ satisfies the property stated in the lemma. □

We first show that for every canonical well-ordered tree t, there exists a subset $U \subseteq \{0,1\}^*$ such that $t|_U$ is isomorphic to t_{llex}. To construct such a set we need the following technical definition. A node $u \in \{0,1\}^*$ of a canonical well-ordered tree t is *mixed* if for all $v \sqsupseteq u, v' \sqsupseteq u \in \{0,1\}^*$ there exists a $w \in \{0,1\}^*$ such that $v < v'w$.

Lemma 4. *For every canonical well-ordered tree t, there exists a mixed node.*

Proof. Let t be a canonical well-ordered tree. Assume by contradiction that t does not have any mixed nodes. We construct by induction two sequences of nodes $(u_i)_{i\in\mathbb{N}}$ and $(v_i)_{i\in\mathbb{N}}$ such that for all $i \geq 0$, $u_i > u_{i+1}$ and $u_i \geq v_i w$ for all $w \in \{0,1\}^*$.

As ε is not mixed there exist two nodes u and v such that $u \geq vw$ for all $w \in \{0,1\}^*$. We take $u_0 = u$ and to ensure that $u_0 > v_0 w$ for all $w \in \{0,1\}^*$, we pick as v_0 an element of $v\{0,1\}^* \setminus \{u' \mid u' \sqsubseteq u_0\}$.

Assume that both sequences are constructed up to rank $i \geq 0$, we define u_{i+1} and v_{i+1}. As v_i is not mixed, there exists two nodes $u \sqsupseteq v_i$ and $v \sqsupseteq v_i$ such that $u \geq vw$ for all $w \in \{0,1\}^*$. We take u_{i+1} equal to u and v_{i+1} an element of $v\{0,1\}^* \setminus \{u' \mid u' \sqsubseteq u_{i+1}\}$ thus ensuring that for all $w \in \{0,1\}^*$, $u_{i+1} > v_{i+1}w$. By induction hypothesis, we have that $u_i > u_{i+1}$.

The sequence $(u_i)_{i\in\mathbb{N}}$ is an infinite strictly decreasing sequence which contradicts the fact that $\leq$ is a well-founded order. □

Proposition 2. *For every canonical well-ordered tree t, there exists a set of nodes U inducing a well-ordered tree $t|_U$ isomorphic to t_{llex}.*

Proof. Let t be a canonical well-ordered tree. We construct a sequence of nodes $(u_w)_{w\in\{0,1\}^*}$ indexed by the set of words over $\{0,1\}^*$ such that: for all $w, w' \in \{0,1\}^*$, $w \leq_{\text{llex}} w'$ implies $u_w \leq u_{w'}$, and for all $w \in \{0,1\}^*$ and $i \in \{0,1\}$, $u_{wi} \in u_w i\{0,1\}^*$.

If we assume that this sequence has been constructed and we take $U := \{u_w \mid w \in \{0,1\}^*\}$, it is easy to check that U is closed by greatest common prefix and hence U induces a full binary tree on t. Furthermore the mapping from $\{0,1\}^*$ to U associating w to u_w is an isomorphism from t_{llex} to $t|_U$.

We now construct the sequence $(u_w)_{w\in\{0,1\}^*}$ by induction on the length-lexicographic order $\leq_{\text{llex}}$. By Lemma 4, the tree t has a mixed node. We take u_ε to be a mixed node of t. Assume that the sequence has been constructed up to $w_0 \in \{0,1\}^*$, we construct the element u_{w_1} where w_1 is the successor of w_0 for the length-lexicographic order. From the definition of $\leq_{\text{llex}}$, it follows that the element w_1 is equal to $w_2 i$ for some $i \in \{0,1\}$ and $w_2 \leq_{\text{llex}} w_0$. As u_ε is mixed, there exists a v such that $u_{w_0} < u_{w_2}iv$. We take $u_{w_1} = u_{w_2}iv$. By induction hypothesis, for all $w \leq_{\mathrm{llex}} w_1$, we have $u_w \leq u_{w_1}$. □

Note that Theorem 7 can already be derived from the above proposition. For every formula φ over $\mathcal{S}$, consider the formula $\varphi^*(X)$ obtained from φ by Lemma 3 and $\varphi^*_{\mathrm{llex}}(X)$ obtained from the formula φ_{llex} of Proposition 1. By Proposition 2, for every well-ordered tree t, $t \models \exists X,\ \varphi^*_{\mathrm{llex}}(X) \wedge \varphi^*(X)$ if and only if $t_{\mathrm{llex}} \models \varphi$. As the formula $\varphi^*(X)$ can be effectively constructed from the formula φ, it follows that the MSO-theory of t_{llex} is recursive in the MSO-theory of any well-ordered tree t.

We now strengthen the result of Proposition 2 by showing that in every well-ordered tree t there exists an MSO-definable set of nodes inducing a well-ordered tree isomorphic to t_{llex}.

Proposition 3. *For every canonical well-ordered tree, there exists an MSO-definable set of nodes U_0 inducing a well-ordered tree isomorphic t_{llex}.*

Proof. Consider the following MSO-formula $\psi(X)$ defined by:

$$\varphi^*_{\mathrm{llex}}(X) \ \wedge\ \forall x \in X, \forall Z, \quad (X_{<x} < Z \wedge \varphi^*_{\mathrm{llex}}(X_{<x} \cup Z)) \ \rightarrow\ x \leq \min Z$$

where $X_{<x} = \{x' \in X \mid x' < x\}$, $X < Y$ stands for $\forall x \in X, \forall y \in Y, x < y$ and $\min Z$ designates the smallest element of the set Z for the order $\leq$.

Let t be a well-ordered tree. We claim that $t \models \exists^{=1} X, \psi(X)$ (where $\exists^{=1}$ stands for there exists a unique), which establishes the MSO-definability of a set of nodes U_0 inducing on t a well-ordered tree isomorphic to t_{llex}.

The first step is to show that $t \models \exists X, \psi(X)$. For this, we define a sequence of nodes $(u_w)_{w \in \{0,1\}^*}$ of t by induction on the order $\leq_{\mathrm{llex}}$. The node u_ε is the smallest element of $\{\min Z \mid Z \subseteq \{0,1\}^* \wedge t \models \varphi^*_{\mathrm{llex}}[Z]\}$. Proposition 2 guarantees that this set is not empty. Assume that the sequence has been constructed up to the element u_{w_0}. We define u_{w_1} where w_1 is the successor of w_0 for the order $\leq_{\mathrm{llex}}$. We take u_{w_1} as the smallest element for $\leq$ of the set $\{\min Z \mid w_0 < Z \wedge \varphi^*_{\mathrm{llex}}(\{u_w \mid w \leq w_0\} \cup Z\})\}$. This set is not empty by definition of u_{w_0}.

Consider the set $U_0 := \{u_w \mid w \in \{0,1\}^*\}$. It is straightforward to show that U_0 induces a well-ordered tree on t isomorphic to t_{llex}. It follows that $t \models \psi[U_0]$. Hence $t \models \exists X, \psi(X)$.

We now show that $t \models \exists^{=1} X, \psi(X)$. It is enough to show that for all $U_1 \subseteq \{0,1\}^*$, $t \models \psi[U_1]$ implies $U_1 = U_0$. For all $u \in U_0$ (resp. $u \in U_1$), we write $\mathrm{Succ}_0(u)$ (resp. $\mathrm{Succ}_1(u)$) the smallest element of U_0 (resp. of U_1) strictly greater than u.

By induction on w for the order $\leq_{\mathrm{llex}}$, it is easy to show that $u_w \in U_1$ for all $w \in \{0,1\}$ and hence $U_0 \subseteq U_1$. Assume by contradiction that $U_0 \subsetneq U_1$. Let u be the smallest element of $U_1 \setminus U_0$. It is easy to see that both sets have the same minimal element and therefore $u \neq \min U_1$. As U_1 induces a well-ordered tree isomorphic to t_{llex} and since $u \neq \min U_1$, there exists $v \in U_1$ such that $u = \mathrm{Succ}_1(v)$. By minimality of u, v belongs to U_0 and for all $v' \leq v$, $v' \in U_1$ implies $v' \in U_0$. Furthermore $\mathrm{Succ}_0(v)$ which belongs to U_0 is different from

$\mathrm{Succ}_1(v)$ which belongs to U_1. Remarking that U_1 is equal to $(U_1)_{<u} \cup (U_1)_{\geq u} = (U_0)_{\leq v} \cup (U_1)_{\geq u}$and using the fact that $t \models \psi[U_0]$, we obtain that $\mathrm{Succ}_0(v) \leq u$. Similarly U_0 is equal to $(U_0)_{<\mathrm{Succ}_0(v)} \cup (U_0)_{\geq \mathrm{Succ}_0(v)} = (U_1)_{\leq v} \cup (U_0)_{\geq \mathrm{Succ}_0(v)}$ and using the fact that $t \models \psi[U_1]$, we obtain that $u = \mathrm{Succ}_1(v) \leq \mathrm{Succ}_0(v)$. It follows that $u = \mathrm{Succ}_0(v)$ which brings the contradiction. □

Theorem 8 directly follows from Proposition 3. An immediate consequence of this result is that the infinite binary tree cannot be MSO-interpreted in $(\omega, succ)$, i.e., the natural numbers with successor. As a well-founded order can be defined in MSO on $(\omega, succ)$, one could interpret the full binary tree with a well-founded order. From Theorem 7, this structure has an undecidable MSO-theory which would contradict the fact that the MSO-theory of $(\omega, succ)$ is decidable [Büc62]. More generally, we obtain that the infinite binary tree cannot be MSO-interpreted in any structure having both a decidable MSO-theory and an MSO-definable well-founded order. Combining the result that the infinite binary tree cannot be interpreted in $(\omega, succ)$ with the same technique as in the proof of Corollary 1, we can also show that the infinite binary tree cannot be interpreted in $(\omega, succ, P_1, \ldots, P_n)$ for arbitrary fixed predicates P_i.

References

[BG00] Blumensath, A., Grädel, E.: Automatic structures. In: Proc. of LICS '00, pp. 51–62. IEEE Computer Society Press, Los Alamitos (2000)

[BL69] Büchi, J.R., Landweber, L.H.: Solving sequential conditions by finite-state strategies. Transactions of the AMS 138, 295–311 (1969)

[Büc62] Büchi, J.R.: On a decision method in restricted second order arithmetic. In: International Congress on Logic, Methodology and Philosophy of Science, pp. 1–11. Stanford University Press (1962)

[Cau02] Caucal, D.: On infinite terms having a decidable monadic theory. In: Diks, K., Rytter, W. (eds.) MFCS 2002. LNCS, vol. 2420, pp. 165–176. Springer, Heidelberg (2002)

[CM03] Carton, O., Michel, M.: Unambiguous büchi automata. Theor. Comput. Sci. 297(1–3), 37–81 (2003)

[CW03] Carayol, A., Wöhrle, S.: The Caucal hierarchy of infinite graphs. In: Pandya, P.K., Radhakrishnan, J. (eds.) FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science. LNCS, vol. 2914, pp. 112–123. Springer, Heidelberg (2003)

[EF95] Ebbinghaus, H.D., Flum, J.: Finite Model Theory. Springer, Heidelberg (1995)

[GS83] Gurevich, Y., Shelah, S.: Rabin's uniformization problem. J. Symb. Log. 48(4), 1105–1119 (1983)

[LS96] Lifsches, S., Shelah, S.: Uniformization, choice functions and well orders in the class of trees. J. Symb. Log. 61(4), 1206–1227 (1996)

[LS98] Lifsches, S., Shelah, S.: Uniformization and skolem functions in the class of trees. J. Symb. Log. 63(1), 103–127 (1998)

[NW] Niwiński, D., Walukiewicz, I.: Ambiguity problem for automata on infinite trees.(Unpublished note)

[Rab69] Rabin, M.O.: Decidability of second-order theories and automata on infinite trees. Transactions of the American Mathematical Society 141, 1–35 (1969)

[Sie75] Siefkes, D.: The recursive sets in certain monadic second order fragments of arithmetic. Arch. für mat. Logik und Grundlagenforschung 17, 71–80 (1975)

[Tho97] Thomas, W.: Languages, automata, and logic. In: Handbook of Formal Language Theory, vol. III, pp. 389–455 (1997)