Automata for MSO over Infinite Trees with Quantification over Borel Sets of Branches

Mikołaj Bojańczyk
Institute of Informatics, University of Warsaw, Poland

Antonio Casares
University of Kaiserslautern-Landau, Germany

Sven Manthe
University of Bonn, Germany

Paweł Parys
Institute of Informatics, University of Warsaw, Poland

Abstract

Rabin's Tree Theorem says that the MSO theory of the infinite binary tree 2^* is decidable. Shelah showed that MSO logic becomes undecidable if this tree is extended to $2^{\leq\omega}$, i.e. by allowing quantification over sets of infinite branches. A longstanding open problem is whether the decidability can be recovered in $2^{\leq\omega}$ by restricting set quantification to Borel sets. We make some progress in this direction, by identifying a suitable automaton model, and showing that most of the automata-theoretic approach to Rabin's Theorem can be extended to the new framework. The only missing part is a conjecture about finite-memory determinacy in certain games. This paper states and explores the conjecture. We prove it in some restricted cases, and give lower bounds on the memory required in those games.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases MSO logic, automata over infinite trees, games on graphs

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.21

Funding *Mikołaj Bojańczyk*: Author supported by the Polish National Science Centre (NCN) grant "Polynomial finite state computation" (2022/46/A/ST6/00072).
Antonio Casares: Partially supported by Deutsche Forschungsgemeinschaft (grant number 522843867) and European Research Council (grant number 101089343). Part of this work was done while Casares was at the University of Warsaw, Poland, supported by the Polish National Science Centre (NCN) grant "Polynomial finite state computation" (2022/46/A/ST6/00072).
Sven Manthe: Author supported by Hausdorff Center for Mathematics at the University of Bonn (DFG EXC 2047).
Paweł Parys: Author supported by the Polish National Science Centre (NCN) grant "Frontiers of automatic analysis of concurrent systems" (2021/41/B/ST6/00535).

1 Introduction

One of the landmark results of logic in computer science is Rabin's Tree Theorem [17], which states the decidability of the MSO theory of the infinite binary tree, which is the set 2^* equipped with the prefix relation. This is a powerful theory – for instance, it interprets the MSO theory of the rational numbers with the usual order [17, Thm. 2.1], which is thus decidable – and it is close to the border of undecidability. On the other side of this border we find the Cantor set 2^ω, equipped with the lexicographic order, and also the order of the reals. Both of these structures have an undecidable MSO theory (as shown by Shelah [18, Theorem 7], assuming the Continuum Hypothesis; this assumption was later removed in [9], i.e., undecidability can be proved in ZFC only). The undecidability proofs use sets that are complicated from a topological point of view, and Shelah conjectured [18, Conjectures 7A and 7B] (see also [7, Open Question, p. 503][19, 2.22]) that decidability can be recovered by restricting set quantification to range over Borel sets (we use the name *Borel* MSO for the resulting logic):

A. The Borel MSO theory of $2^{\leq\omega}$ equipped with the prefix and lexicographic orders is decidable.

B. The Borel MSO theory of $\mathbb{R}$ with the usual order is decidable.

Conjecture B is the weaker one, since an isomorphic copy of the real numbers can be defined using Borel MSO in $2^{\leq\omega}$. In 2024, Manthe announced a proof of the weaker conjecture [11]. A proof of both conjectures was announced by Džamonja in 2025 [5], but we could not confirm its correctness.

This paper is devoted to the stronger Conjecture A. One can think of this conjecture as a common generalisation of Rabin's Tree Theorem and Conjecture B. This is because the complete binary tree is recovered in $2^{\leq\omega}$ as the definable substructure consisting only of nodes in 2^*. Hence, any proof of Conjecture A would also need to subsume a proof of Rabin's Tree Theorem. In particular, this seems to rule out the proof of Manthe, which is based on linear orders and the compositionality of their MSO theories, and does not seem to be amenable to extensions that cover the tree structure. This paper proposes a programme to prove Conjecture A. Our main contributions can be summarised as follows:

1. We introduce a certain kind of infinite games, and conjecture that these games can be won with finite memory.
2. We prove that if the conjecture on finite memory is true, then Conjecture A is also true, that is, one can decide the Borel MSO theory of $2^{\leq\omega}$, and furthermore the logic has a corresponding automaton model.
3. We present a preliminary study of the conjecture on finite memory, with some partial positive results.

Let us describe these contributions in some more detail. Simply put, our programme is to copy the proof of Rabin's Tree Theorem, and extend it to take into account Borel sets of branches. We begin by briefly recalling how Rabin's Tree Theorem is proved, using a modern presentation that emphasises games, as proposed by Gurevich and Harrington [8]. In the proof, one shows that every MSO formula can be transformed into an equivalent automaton, and the latter can be tested for emptiness. Proving this requires to show that automata have the same closure properties as the logic, namely Boolean combinations and projection. Union and intersection are easily achieved using a product construction, and projection is also easy because the automaton model in question is nondeterministic. The difficult closure property is complementation. Unlike for ω-words, determinisation is not an option. To prove complementation, one describes rejection by the automaton in terms of a game of infinite duration, and then proves that this game can be won using simple strategies of finite memory. The finite memory can then be embedded into the state space of a nondeterministic automaton that recognises the complementary language.

As we describe in this paper, most of the proof discussed in the previous paragraph can be adapted to the more general setting of $2^{\leq\omega}$. In this more general setting, an input tree has type $2^{\leq\omega} \to \Sigma$, that is, it assigns labels to both nodes and branches, and a run of the automaton has type $2^{\leq\omega} \to Q$. In both cases, the maps are required to be Borel, since on the logical side we are targeting Borel MSO. The acceptance condition for the automaton is a variant of the Muller condition that takes into account the states assigned to the branches. Using this automaton model, we are able to implement all parts of the proof of Rabin's Tree Theorem, with one exception, namely finite-memory determinacy of the underlying games. This determinacy result is the content of our conjecture, and much of the paper is devoted to its study.

The conjecture. Let us now describe in more detail our conjecture on finite-memory strategies. Like most games studied in the context of automata, such as parity games or Muller games, our games are infinite duration games played by two players, called Adam and Eve, using directed graphs (called arenas), where each position is owned by one of the two players. This notion of game is standard, and will be recalled later in the paper. In this introduction, we focus on the distinguishing features of the games that arise within the context of Borel MSO, which are defined below. We call the games Muller-Borel, because the acceptance condition is a Boolean combination of Büchi and Borel conditions, and Boolean combinations of Büchi conditions are called Muller conditions.

▶ **Definition 1.** *A* Muller-Borel game *is a game satisfying the following two conditions:*

1. ***Tree-like arena.*** *The set of positions is $2^* \times Q$ for a finite set Q. Edges of the game can only move to child nodes, that is, if*

$$(v, q) \to (w, p)$$

is an edge in the game, then w is a child of v. Not all edges that move to children need to be available.

2. ***Muller-Borel condition.*** *The winning condition is a finite Boolean combination of two kinds of conditions:*
   (Büchi) *$q \in Q$ is seen infinitely often;*
   (Borel) *the branch belongs to a Borel set $B \subseteq 2^\omega$.*

Using Martin's Borel Determinacy Theorem [13, p. 371], one can show that the games are determined, that is, one of the players must have a winning strategy. However, our main interest is in the memory used by the winning strategy, which is beyond the scope of Martin's theorem. Our conjecture is that the games can be won with finite memory, as stated below. (The notion of finite memory is the standard one, and will be recalled later in the paper.)

▶ **Conjecture 2.** *For every Muller-Borel game, one of the players has a winning strategy that uses finite memory. Furthermore, an upper bound on the memory size can be computed based on the size of Q.*

Using the same framework as for Rabin's Tree Theorem, we prove in Theorem 15 that the above conjecture implies decidability of Borel MSO on $2^{\leq\omega}$, that is, Shelah's Conjecture A. The rest of this introduction is devoted to a preliminary discussion of the conjecture, with an extended discussion contained in the main body of the paper.

As indicated, the main examples of Muller-Borel games appear in the analysis of tree automata. One can recognize these games in the tree-like requirement (Definition 1, (1)), with Q being the set of states. For many kinds of games, such as Muller games or parity games, the tree-like shape of the arena is irrelevant in the analysis of the game. Hence, it is common to define those games just over arbitrary graphs [20, 6]. In our case, however, the tree-like shape is essential, since the Borel conditions in the game are defined in terms of the tree structure. Let us now explain how certain special cases of a Muller-Borel game correspond to important results on determinacy of games.

- Suppose that we only allow Büchi conditions in the game. In this case, we have a game with a Muller condition. Muller games are known to have finite-memory strategies, which was first proved by Büchi and Landweber for finite arenas [3, Theorem 1'], and then by Gurevich and Harrington for general arenas [8, Theorem 1]. Therefore, if we only allow Büchi conditions, then the conjecture is true.

- Suppose that there is only one state in the game. In this case, each game position can be reached in only one way (since the positions form a tree), and therefore all strategies are memoryless. Thus, from Martin's Borel Determinacy Theorem we can conclude that the conjecture holds in this special case, even without any memory. Note that the full power of Martin's theorem is needed for this special case.

Later on in the paper, we prove the conjecture for other special cases of the game, including the special case where only the Borel condition is allowed (which extends the situation from the second item above, in which the Büchi condition trivializes).

Our conjecture talks about finite-memory strategies. A natural question would be about memoryless strategies. For example, in automata theory, finite-memory determinacy of Muller games is commonly proved by first reducing to a parity condition, and then proving that parity games can be won using memoryless strategies. The only need for memory arises from the reduction from Muller to parity, which is typically accomplished using a variant of McNaughton's "latest appearance record" construction. For our conjecture, we are unable to present such a reduction, that is, we are not aware of a variant of our conjecture which would talk about memoryless strategies. Using priorities does not seem to help: memory is needed even if we convert the game into a parity-style format, in which the Büchi part of the acceptance condition is replaced by a priority function. In fact, even in the presence of only three priorities, the amount of needed memory grows unboundedly as a function of the number of states (see Proposition 28). On the positive side, we show that for two priorities, only two states of memory are needed, regardless of the number of states in Q (see Proposition 24). Other lower and upper bounds are contained in Propositions 23, 26, and 27.

The conjecture seems rather strong. In particular, as we discuss in Section 4.3, the conjecture implies that if we restrict the semantics of Borel MSO to use only Boolean combinations of F_σ sets (i.e. countable unions of closed sets), then the same formulae are true. In other words, Borel MSO can essentially only talk about Boolean combinations of F_σ sets, and it cannot access more complicated Borel sets. Since one could expect Borel MSO to be more expressive, this could be seen as evidence against the conjecture. However, the recent proof of Manthe is consistent with our conjecture, since it implies that in the weaker setting of 2^ω one can also restrict quantification to Boolean combinations of F_σ sets, and the same formulae are true.

2 Borel MSO

In this section, we describe the logics in question. We begin with the usual notion of monadic second-order logic (MSO). In this logic, there are two kinds of variables: lowercase variables x, y, z for elements and uppercase variables X, Y, Z for sets. Formulas are built using the following constructors:

$$\underbrace{\exists x \ \forall x}_{\substack{\text{quantification}\\ \text{over elements}}} \quad \underbrace{\exists X \ \forall X}_{\substack{\text{quantification}\\ \text{over sets}}} \quad \underbrace{\vee \ \wedge \ \neg}_{\substack{\text{Boolean}\\ \text{combinations}}} \quad \underbrace{R(x_1, \ldots, x_n)}_{\substack{\text{relations from}\\ \text{the vocabulary}}} \quad \underbrace{x \in X}_{\substack{\text{membership}\\ \text{relation}}}.$$

The semantics of the logic are defined in the usual way (see e.g. [21] for a more in-depth discussion). When we talk about the MSO theory of a structure, we mean the set of MSO sentences that are true in this structure. Rabin's Tree Theorem [17] states the decidability of the MSO theory of the infinite binary tree, which is viewed as a structure where the underlying set is 2^*, and there are two binary relations: the prefix order, and the lexicographic order.

The logic that is studied in this paper is *Borel* MSO. This logic has the same syntax as MSO, except that the semantics are changed so that set quantification is restricted to range only over Borel sets. Recall that the family of Borel sets in a topological space is the least family of subsets that contains the open sets, and which is closed under complementation as well as under taking countable unions and intersections. Therefore, in order to speak of Borel MSO, the underlying set in a structure must be equipped with a topology. The case which interests us is where the underlying set is $2^{\leq\omega}$, that is, strings of bits that have length at most ω. The finite strings are called *nodes* and the infinite ones are called *branches*. The topology is the one which arises taking as base of open sets the sets of the form

$$U_w = \{x \in 2^{\leq\omega} \mid w \text{ is a prefix of } x\}, \quad \text{where } w \in 2^*.$$

That is, the *open sets* of this topology are the (arbitrary) unions of U_w, for $w \in 2^*$.

When we speak of $2^{\leq\omega}$ as a structure, we assume that it is equipped with the prefix and lexicographic orders, just as in Rabin's Tree Theorem. The lexicographic order gives us the tree structure on nodes, while the prefix relation allows us to check if some node belongs to a branch.

This paper is devoted to Conjecture A from the introduction, which says that the Borel MSO theory of $2^{\leq\omega}$ is decidable, meaning that there is an algorithm that inputs a sentence of the logic, and answers whether the sentence is true in the structure. Note that the Borel restriction is relevant only for branches. This is because every set of nodes is Borel, and thus a subset of $2^{\leq\omega}$ is Borel if and only if its restriction to branches is a Borel set.

▶ **Example 3** (Limit points)**.** In this example, we give an MSO sentence distinguishing the structures 2^* and $2^{\leq\omega}$. It suffices to consider the sentence:

$$\forall X.\ X \text{ is a chain for the prefix order } \to \exists y.\ \forall x \in X.\ x \text{ is a prefix of } y.$$

This does not hold in 2^*, taking X to be all nodes on an arbitrary branch. However, it holds in $2^{\leq\omega}$, as the branch itself is an element of the structure, so it provides an upper bound to X. (Note the similarities with the sentence "every bounded sequence admits a limit point" distinguishing $\mathbb{Q}$ and $\mathbb{R}$.)

▶ **Example 4** (Determinacy)**.** In this example, we illustrate the difference between MSO and Borel MSO over $2^{\leq\omega}$. A Gale-Stewart game is a game where two players, call them Adam and Eve, pick bits from $2 = \{0, 1\}$ in alternation. After ω rounds, an infinite play in 2^ω arises. The winner of the game is given by a winning condition $W \subseteq 2^\omega$, which tells us which plays are winning for Eve. A strategy for player Eve can be described as a set of nodes X, which indicates the chosen successors used by the strategy. Similarly, a strategy for Adam can be described a set of nodes Y. Therefore, one can express determinacy of such games using a formula:

$$\forall W \subseteq 2^\omega.\ \vee \begin{cases} \exists X \subseteq 2^*.\ \forall Y \subseteq 2^*.\ \varphi(W, X, Y) \\ \exists Y \subseteq 2^*.\ \forall X \subseteq 2^*.\ \neg\varphi(W, X, Y), \end{cases}$$

where $\varphi(W, X, Y)$ says that the unique play consistent with strategies X and Y belongs to the set W. If we think of this formula as a formula of Borel MSO, then it is true, since only Borel winning conditions are considered, and Martin's Borel Determinacy Theorem says that such games are determined. If we think of this formula as a formula of (unrestricted) MSO, then it is false, since there exist games with (non-Borel) winning conditions that are not determined, assuming the Axiom of Choice.

3 Nondeterministic automata over Borel trees

The idea behind the automata method is to find an automaton model that recognises exactly the same languages as those that can be defined in Borel MSO. These languages consist of labelled trees. The intuitive idea is that the labels describe a valuation of free variables for a formula that has free variables. Let us begin by describing the appropriate notion of trees.

▶ **Definition 5** (Borel tree). *A* Borel tree *over a finite alphabet* Σ *is defined to be a function*

$$t \colon 2^{\leq\omega} \to \Sigma$$

such that for every letter $a \in \Sigma$, the inverse image is a Borel set. We denote the set of Borel trees over the alphabet Σ by $\mathsf{T}\Sigma$.

Since the topology is only important on the infinite branches, the Borel condition is only important for those, and the labels of the nodes are unconstrained.

As mentioned earlier in this section, we consider languages of Borel trees that can be defined in logic, or recognised by automata. Let us first explain how a language of Borel trees can be defined in logic. We say that a language $L \subseteq \mathsf{T}\Sigma$ for $\Sigma = \{a_1, \ldots, a_n\}$ is definable in Borel MSO if there is a formula

$$\varphi(X_1, \ldots, X_n)$$

of Borel MSO such that a tree $t \in \mathsf{T}\Sigma$ belongs to L if and only if the formula is true under the valuation that maps each variable X_i to the set of nodes and branches that are labelled by the letter a_i. Our goal in this section is to define an automaton model which recognises exactly the languages definable in Borel MSO. (This goal is achieved conditionally on a certain conjecture about complementation of automata.)

Automata over Borel trees. We now describe the automaton model that is used to recognise languages of Borel trees. This model is a variant of nondeterministic Muller tree automata.

▶ **Definition 6** (Nondeterministic automaton). *A* nondeterministic Muller automaton on Borel trees *is given by:*

- *a finite set of states Q;*
- *a finite alphabet Σ;*
- *a subset $I \subseteq Q$ of initial states;*
- *two transition relations:*

$$\Delta_* \subseteq Q \times \Sigma \times Q \times Q, \qquad \Delta_\omega \subseteq \mathsf{P}Q \times \Sigma \times Q.$$

An input Borel tree $t \in \mathsf{T}\Sigma$ is accepted by the automaton if there exists an accepting run, which is defined to be a Borel tree $\rho \in \mathsf{T}Q$ such that the following three conditions hold:

1. the root is labelled by an initial state;
2. for every node $x \in 2^*$, the transition relation Δ_* contains the tuple consisting of: the state in x, the label of x in t, the state in the left child of x, the state in the right child of x;
3. for every branch $b \in 2^\omega$, the transition relation Δ_ω contains the tuple consisting of: the set of states that occur infinitely often along b, the label of b in t, the state in b.

▶ **Lemma 7.** *Emptiness is decidable for nondeterministic Muller automata on Borel trees.*

Proof. We reduce the problem to emptiness for nondeterministic Muller automata on the usual kind of infinite trees, which do not have labels or states on branches. For the latter kind of automata, emptiness is decidable.

Consider a nondeterministic Muller automaton on Borel trees. The Muller automaton on Borel trees is nonempty if and only if there is a labelling $\rho : 2^* \to Q$ which satisfies the three conditions (1), (2), (3) in the definition of an accepting run, except that condition (3) is changed so that for each branch $b \in 2^\omega$, the set of states that occur infinitely often along b belongs to

$$\mathcal{F} = \{R \subseteq Q \mid (R, a, q) \in \Delta_\omega \text{ for some } a \in \Sigma \text{ and } q \in Q\}\,.$$

We define a Muller tree automaton on infinite trees in the usual sense, which has $\mathcal{F}$ as its acceptance condition. This new automaton is nonempty if and only if the original Borel tree automaton is nonempty. ◀

We note that atomic MSO formulas (prefix and lexicographic orders and membership relation) can be encoded into Muller automata on Borel trees just as they can be translated into classical Muller automata. The only subtlety in such encoding, which already arises in the classical setting of 2^*, is the handling of free variables: a tree with an evaluation of its free variables can be encoded using an extended alphabet.

We now establish some straightforward closure properties of our automata, namely union, intersection, and projection. The difficult part is complementation. We are unable to prove complementation, but the rest of this paper is devoted to a discussion of this problem, and some possible approaches towards its solution.

Let us begin with union and intersection.

▶ **Lemma 8.** *Languages recognised by nondeterministic Muller automata on Borel trees are closed under union and intersection.*

Proof. Standard product construction. This construction works because the Borel condition for trees is closed under pairing. This means that if we take two trees over alphabets Σ_1 and Σ_2 and pair them into a single tree over the alphabet $\Sigma_1 \times \Sigma_2$, then the new tree is a Borel tree if and only if both of the original trees were Borel trees. ◀

We now turn to projection, which corresponds to existential set quantification in logic. Since our automaton model is nondeterministic, it is also closed under taking projections, understood as follows. Consider a function

$$f\colon \Sigma \to \Gamma$$

from one finite alphabet to another. This function can be extended to a function on Borel trees

$$\mathsf{T}f\colon \mathsf{T}\Sigma \to \mathsf{T}\Gamma$$

in the obvious way, by applying f to the label of every node. We can apply this function to some language $L \subseteq \mathsf{T}\Sigma$, obtaining a new language $f(L) \subseteq \mathsf{T}\Gamma$, which is called the *projection* of L along f.

▶ **Lemma 9.** *Languages recognised by nondeterministic Muller automata on Borel trees are closed under taking projections.*

Proof. This is the same proof as usual: take the automaton before the projection, and apply the projection function to each input letter. This proof works because if we take the projection of a Borel tree, then the new tree is also Borel. (It is important to avoid confusion here: individual Borel trees are closed under projection, but Borel languages of Borel trees are not.) ◀

As in the proofs of Büchi or Rabin, the essential step is complementation. We do not know how to prove this, and leave it as a conjecture. In the next section, we present a conjecture on finite-memory strategies in games, which implies that languages recognised by nondeterministic Muller automata on Borel trees are effectively closed under complementation, that is, for every automaton one can compute an automaton for its complement language. This in turn implies our main conjecture about decidability of Borel MSO (see Theorem 15).

4 Complementation of automata and games

When proving closure under complementation for nondeterministic automata on infinite trees, games and their determinacy play a central role. We follow that same path. Let us begin by recalling the kind of games that are used in our proofs.

4.1 Games

The games that we consider have infinite duration, and are played by two players Adam and Eve on a directed graph. We now recall the definition of such games, and the basic concepts such as strategies.

Here, a directed graph is a set of vertices V together with a binary relation E on the vertices. We typically write $v \to w$ to indicate that there is an edge from v to w. Paths, both finite and infinite, are defined in the usual way.

▶ **Definition 10** (Game). *A* game *is given by:*
1. *a directed graph* (V, E)*, called the* arena*;*
2. *an ownership function of type* $V \to \{\text{Adam}, \text{Eve}\}$*;*
3. *an initial position* $v_0 \in V$*;*
4. *a winning condition* W*, which assigns a winner in* $\{\text{Adam}, \text{Eve}\}$ *to every infinite path starting in* v_0*.*

In the context of a game, vertices of the underlying graph are called *positions*. We assume that in the arena, every position has at least one outgoing edge; this can be easily achieved by possibly adding two losing sink positions with self-loops, one for each player.

We now briefly recall the usual notions of strategies and winning strategies for games. A *strategy* for player i is a function that inputs a finite path in the arena from the initial position to some position owned by i, and outputs an edge that extends this path. In the following, we refer to such finite paths as *histories*. Once both players fix their strategies, a unique infinite path arises in the game, which we call a *play*, and the winning condition tells us which of the two players wins. Globally, a player wins if they have a winning strategy, that is, a strategy that wins against every strategy of the opponent. A game is *determined* if one of the players has a winning strategy. Under the Axiom of Choice, not every game is determined, but Martin's Borel Determinacy Theorem says that the game is determined if the winning condition is a Borel subset of all paths, according to the usual topology (which, in particular, is the case for games arising in this paper).

We are particularly interested in *finite-memory strategies*, which are defined as follows. Essentially, these are strategies that are produced by finite state transducers reading the edges in the game. More formally, a strategy for Eve (for Adam, the concept is defined symmetrically) with memory M consists of some initial memory state m_0 and two functions

$$\underbrace{\delta\colon M \times E \to M}_{\text{memory update}} \quad \text{and} \quad \underbrace{\sigma\colon M \times \overbrace{V_{\text{Eve}}}^{\text{positions owned by Eve}} \to E}_{\text{Eve's choice}}.$$

The first function is used to *update the memory* along a play (as in a deterministic automaton) and the second function shows how Eve chooses edges in the game from her positions. A *finite-memory strategy* is one where the memory set M is finite. A *memoryless strategy* is the special case of a strategy with finite memory in which the set M has one element only. A game has *finite-memory determinacy* if one of the two players can win using a strategy with finite memory. It was shown by Gurevich and Harrington [8, Theorem 1] that games with a Muller winning condition (which topologically speaking is a finite Boolean combination of F_σ sets) have finite-memory determinacy.

The two determinacy results discussed above are incomparable. Indeed, Martin's Borel Determinacy Theorem has a weaker assumption (the winning condition is Borel), but a weaker conclusion (the winning strategy might use infinite memory). Both results play an important in our setting, since our games are a hybrid of Borel and Muller games.

4.2 Complementation

To complement nondeterministic automata, we use a game based approach, following Gurevich and Harrington [8]. In this approach, for a nondeterministic automaton $\mathcal{A}$ and an input tree t, we define an *acceptance game* as follows. In the game, Eve tries to show that the tree is accepted, by constructing an accepting run, and Adam tries to dispute Eve's claim, by showing that her run breaks the infinitary part Δ_ω of the transition relation. The positions of the game are pairs in

$$\underbrace{(Q + \Delta_*)}_{\substack{\text{disjoint union of} \\ \text{states and transitions}}} \times 2^*.$$

The initial position consists of the initial state and the root of the tree. (One can assume without loss of generality that the automaton has one initial state.) In a position (q, v) that uses a state $q \in Q$, Eve chooses a transition $(q, a, q_0, q_1) \in \Delta_*$, having source state q. In the resulting position, Adam picks a child $i \in 2$ and the game proceeds to the position $(q_i, v \cdot i)$. The winning condition in the game is defined as follows. Suppose that $R \subseteq Q$ is the set of states that are visited infinitely often in the play, and $b \in 2^\omega$ is the branch that is traced by this play in the underlying tree. Then Eve wins the play if and only if the branch b belongs to the set

$$\{b \in 2^\omega \mid \text{there is a state } q \text{ such that } (R, t(b), q) \in \Delta_\omega\}. \tag{1}$$

The above set is Borel, because the labelling $t(b)$ is Borel. It follows that the acceptance game is almost a special case of a Muller-Borel game, with the only minor issue being that the arena is not strictly speaking tree shaped, since the play can stay in the same node for two consecutive positions. This minor issue will be resolved shortly (see Claim 13).

▶ **Lemma 11.** *A tree is accepted by a nondeterministic automaton if and only if Eve has a winning strategy for the corresponding acceptance game.*

Proof. From an accepting run, we easily construct a winning strategy for Eve which follows this run, and vice versa. The only issue that requires some attention is that when we convert a winning strategy into a run, we need to ensure that the states in the branches are defined in a Borel way. To this end, consider a winning strategy σ for Eve, and for a branch $b \in 2^\omega$ choose $q \in Q$ to be the least state (in some chosen linear order on the states) which witnesses that b belongs to the set from (1), in the unique play of the game that is consistent with σ and that traces branch b. The map $b \mapsto q$ defined this way is Borel, since it is defined in terms of a Muller condition on the states visited infinitely often in the branch, and of the label $t(b)$ of that branch. ◀

The acceptance game is designed so that a tree is rejected if and only if player Adam has a winning strategy in the acceptance game. As we show in the following lemma, this can be expressed by a nondeterministic automaton, provided that the winning strategy has finite memory, as ensured by Conjecture 2.

▶ **Lemma 12.** *Assume Conjecture 2. Then languages recognised by nondeterministic Muller automata on Borel trees are closed under complementation, and the complement automaton can be effectively computed.*

Proof. Fix a nondeterministic Muller automaton on Borel trees $\mathcal{A}$. We want to compute an automaton, of the same kind, for its complement language. This automaton will guess a finite-memory winning strategy for Adam in the corresponding acceptance game. Such a strategy exists for trees in the complement language by Conjecture 2, as shown in the following claim.

▷ Claim 13. Assume Conjecture 2. Based on the state space Q of $\mathcal{A}$ one can compute a memory bound m such that for every input tree, the corresponding acceptance game can be won using a strategy with at most m memory states.

Proof. This is essentially the content of the conjecture, with the minor issue that the games from the conjecture require that one always moves to a child, while the acceptance game stays for two positions in the same node: after choosing a state, and then after choosing a transition. This can be easily resolved by considering a new game, in which we always move to the left child after selecting a transition, and the Borel part is suitably updated to take into account these artificial moves (i.e., one ignores every second direction in the branch). ◁

Let $M = \{1, \dots, m\}$ be the memory that is needed according to the above claim. By the claim it follows that an input tree t is rejected by the automaton $\mathcal{A}$ if and only if there exists a winning strategy of player Adam in the acceptance game that uses memory M. Such a strategy can be viewed as a function that assigns two pieces of information to each node $v \in 2^*$ in the input tree, namely the functions

$$\underbrace{\delta_v \colon M \times (\Delta_* + \Delta_* \times 2) \to M}_{\text{memory update}} \quad \text{and} \quad \underbrace{\sigma_v \colon M \times \Delta_* \to 2}_{\text{Adam's choice}}.$$

A nondeterministic automaton can guess such a strategy, since a function as above can be thought of as a tree over alphabet

$$\Gamma \stackrel{\text{def}}{=} (M \times (\Delta_* + \Delta_* \times 2) \to M) \times (M \times \Delta_* \to 2) + \{\omega\},$$

where the label ω is used for branches. Below, we show that once an input tree and a strategy for Adam are given in this representation, then a nondeterministic automaton can check if the strategy is winning for Adam. In other words, the following language, which we call the *strategy language*, is recognised by a nondeterministic automaton (in fact, a deterministic one would be enough):

$$\left\{ t \in \mathsf{T}(\Sigma \times \Gamma) \mid \begin{array}{l} \text{the } \Gamma \text{ part represents a winning strategy for} \\ \text{Adam in the acceptance game for the } \Sigma \text{ part} \end{array} \right\}.$$

This completes the complementation proof, since by projecting the strategy language to $\mathsf{T}\Sigma$, we get the complement of the language accepted by $\mathcal{A}$.

To define an automaton for the strategy language, we observe that it only checks a certain ω-regular properties on all branches. This is because a tree belongs to the strategy language if and only if: for every branch $b \in 2^\omega$, (*) if we take any play of the acceptance condition that is consistent with Adam's strategy and which stays on the branch b, then this play is winning for Adam. To determine whether a play is winning for Adam, we need access to the label of the branch b in the input tree, since this label is used in the winning condition. Once we know this label, condition (*) becomes an ω-regular property of the branch, since it can be defined in MSO over ω-words. Therefore, to show that the strategy language is recognised by an automaton, we can use the following claim, where the alphabet Σ is the alphabet $\Sigma \times \Gamma$ of the strategy language.

▷ Claim 14. Let Σ be a finite alphabet, and suppose that for each letter $a \in \Sigma$ we have an associated ω-regular language L_a over the alphabet $\Sigma \times 2$. Then we can construct a nondeterministic (even deterministic) Muller automaton over Borel trees that accepts a tree $t \in \mathsf{T}\Sigma$ if and only if for every branch $b \in 2^\omega$ with label a, the language L_a contains the ω-word given by the sequence

$$(\text{label of } t \text{ in } i\text{-th node on } b, \text{direction in } i\text{-th step of } b)_{i=0}^{\omega}.$$

Proof. For each language L_a take a deterministic Muller automaton on ω-words that recognises it, and run these automata in parallel (using a product construction) along each branch. The infinitary part of the transition relation of the resulting tree automaton, having access to the label $t(b)$ of the branch and to the states of the automata that occur along this branch, checks whether the automaton for $L_{t(b)}$ accepts. ◁

This concludes the proof of Lemma 12. ◀

Let us now put together all results proved above to show that Conjecture 2 implies decidability of the Borel MSO theory of $2^{\leq\omega}$, as stated in Shelah's Conjecture A.

▶ **Theorem 15.** *Assume Conjecture 2. Then nondeterministic Muller automata on Borel trees define the same languages as Borel* MSO*, and translations in both directions are effective. Furthermore, the Borel* MSO *theory of* $2^{\leq\omega}$ *is decidable.*

Proof. To go from an automaton to a formula of Borel MSO, one simply uses the logic to express the existence of an accepting run, in the usual way. To go in the opposite direction, we realize particular constructors of logical formulae using the closure properties that have been established above: intersections and unions (Lemma 8), projections (Lemma 9) and complementation (Lemma 12). The conjecture is used only for complementation. Finally, to decide the Borel MSO theory, we convert a sentence of the logic into an equivalent automaton (over a single-letter alphabet), and then use Lemma 7 to check whether it is nonempty. ◀

4.3 Boolean combinations of $\mathbf{F}_\sigma$ sets

In this section, we discuss a consequence of our main conjecture, which is that Borel MSO is equivalent to a logic where quantification is much more restricted, namely to Boolean combinations of F_σ sets.

Recall that F_σ sets are countable unions of closed sets; another notation for the same family of sets is level $\mathbf{\Sigma}^0_2$ of the Borel hierarchy. In the topological space 2^ω, these are exactly the sets that can be described using a co-Büchi condition on the visited nodes, that is, as

$$X = \{b \in 2^\omega \mid b \text{ visits nodes from } F \text{ finitely often}\}$$

for some set of nodes $F \subseteq 2^*$; see [10, Corollary 3.2]. If we take a Boolean combination of such sets, we arrive at a Muller condition, that is, we have a finite family of sets $F \subseteq 2^*$, and whether a branch belongs to X depends on which sets F are visited infinitely often.

One could consider a variant of Borel MSO, which has the same syntax, but for which the semantics are changed so that set quantification is restricted to Boolean combinations of F_σ sets. In his proof of Conjecture B, Manthe showed that over the real line, this new logic gives rise to the same theory, that is, restricting quantification to Boolean combination of F_σ sets does not change the truth value of sentences [11, Theorem 1.1]. The following theorem shows that, assuming our conjecture, the same result would hold for the structure $2^{\leq\omega}$.

▶ **Theorem 16.** *Assume Conjecture 2. Borel* MSO *has the same true sentences as the restriction of* MSO *where set quantifiers range over Boolean combinations of F_σ sets.*

Proof. Let us define $\mathsf{S}\Sigma \subseteq \mathsf{T}\Sigma$ to be the set of Borel trees where the labelling is restricted to Boolean combinations of F_σ sets, that is, the inverse image of each label $a \in \Sigma$ is a Boolean combination of F_σ sets. We prove that for every formula φ, which describes a property of trees labelled by an alphabet Σ, we have

$$\underbrace{t \models_{\mathsf{T}} \varphi}_{\text{Borel semantics}} \quad \Leftrightarrow \quad \underbrace{t \models_{\mathsf{S}} \varphi}_{\text{Bool}(F_\sigma)\text{ semantics}} \qquad \text{for every tree } t \in \mathsf{S}\Sigma. \tag{$\star$}$$

The proof is a straightforward induction, and the only interesting case is that of a quantifier, say an existential one. To prove this step, we convert the quantified formula into an automaton using Theorem 15, and then we apply the following claim to show that the quantified set can always be chosen so that it is a Boolean combination of F_σ sets.

▷ Claim 17. Let $\mathcal{A}$ be a nondeterministic Muller automaton over Borel trees with input alphabet Γ, and let $f \colon \Gamma \to \Sigma$ be some projection function. For every tree $t \in \mathsf{S}\Sigma$, the following conditions are equivalent:

1. $\mathcal{A}$ accepts some $s \in \mathsf{S}\Gamma$ that projects to t;
2. $\mathcal{A}$ accepts some $s \in \mathsf{T}\Gamma$ that projects to t.

Proof. The tree before the projection can be seen as an accepting run, and therefore in order to prove the claim it is enough to show that if an automaton $\mathcal{A}$ accepts a tree in $\mathsf{S}\Sigma$, then this can be done by an accepting run in $\mathsf{S}Q$. This follows from the acceptance game, since a run that arises from a finite-memory strategy in the acceptance game is necessarily in $\mathsf{S}Q$. ◁

This concludes the proof of Theorem 16. ◀

We can recast the above theorem, or more precisely the slightly stronger statement of ($\star$), as follows. We can view Borel MSO as the first-order theory of the structure where the universe consists of Borel subsets of $2^{\leq\omega}$. Then ($\star$) tells us that if we further restrict the universe to Boolean combinations of F_σ sets, then we get an elementary substructure. As we have remarked in the introduction, this type of situation does arise in the weaker setting of Borel MSO on 2^ω, and therefore it is plausible that it could also occur for $2^{\leq\omega}$.

5 Prioritized Borel games

As we have explained in the previous section, the crucial point of our approach is finite-memory determinacy for acceptance games that arise from automata complementation. The rest of this paper is devoted to a study of these games, and evidence for the conjecture that they are determined with finite memory.

5.1 Definition and finite-memory conjecture

Before continuing, we clean up the definition of the games, by removing certain artifacts that arise from automata and which are not important to understand the games. The cleaned up version is presented below; one of its advantages is that it uses priorities (similar to the parity condition) instead of a Muller condition. Also, we switch from vertex labels to edge labels, which will make it more convenient to identify relevant special cases.

▶ **Definition 18** (Prioritized Borel game). *A* prioritized Borel game *is defined in the same way as a Muller-Borel game, except that the winning condition in Item 2 of Definition 1 is replaced by:*

2. ***Prioritized Borel condition.*** *The winning condition is described using:*
   a. *a priority function, which assigns a* priority *from* $\{1, ..., k\}$ *to every edge in the game, for some* $k \in \mathbb{N}$*; and*
   b. *a sequence of Borel sets*

   $$W_1, \dots, W_k \subseteq 2^\omega.$$

   The winner of an infinite play is obtained as follows: take i to be the maximal priority that occurs infinitely often on the edges of the play. Then, Eve wins if the branch of the play belongs to W_i, and Adam wins otherwise.

Note that the tree-like structure influences the information that is explicitly available for the players during a play. Namely, the sequence of produced nodes of the tree is encoded in the arena, while the sequence of states or priorities is not; players need memory to keep track of this latter information.

In the remainder of this paper we discuss the following conjecture.

▶ **Conjecture 19.** *For every prioritized Borel game, one of the players has a winning strategy that uses finite memory. Furthermore, an upper bound on the memory size can be computed based on the number of* states *(i.e., elements of Q) and number of priorities.*

We note that prioritized Borel games are determined, that is, there is a "*winner*", as they are a special case of games with Borel winning conditions. The interesting part of the conjecture is therefore the one concerning memory.

It is easy to see that this conjecture is equivalent to Conjecture 2, as we can translate a Boolean combination of Büchi conditions into a prioritized condition using, e.g., a "latest appearance record" construction.

▶ Remark 20. If every prioritized Borel game can be won with finite memory, then for a fixed number of states and priorities there is a uniform bound on the memory needed. That is, for all integers n and k, there is a constant $c_{n,k}$ such that in any prioritized Borel game with $|Q| \leq n$ and using k priorities, one of the players has a winning strategy that uses at most $c_{n,k}$ memory states.

Indeed, if this was not the case, there would be a sequence of games where the winner (that we can assume to be Eve by symmetry) requires unbounded memory. We could combine these games into a single one: Adam can stay in the leftmost branch and move into the i-th game at the i-th node. If he stays in the leftmost branch forever, he loses.

▶ **Example 21** (One state and priority)**.** To get a feeling for the conjecture, let us discuss the special case where there is one state and one priority. In this case, the finite-memory assumption trivialises, so the above conjecture is just a restatement of Borel determinacy. Indeed, in this case there is no difference between general strategies and finite-memory strategies (even memoryless strategies), since the current position determines the entire history.

Next, we show an example of a prioritized Borel game where the winner requires memory.

▶ **Example 22** (Copying game)**.** In Figure 1 we show a prioritized Borel game with only one state per tree node, and using two priorities. In this game, Adam chooses to produce a priority 1 or 2, and then Eve chooses to go left or right in the tree. We let the Eve-winning Borel subsets be W_1 = branches that eventually only take the left child, and W_2 = branches that contain a right child infinitely often. That is, Eve wins if she goes right infinitely often if and only if Adam produces 2 infinitely often.

Eve can win this game: each time that Adam takes a 2-edge, she can go to the right child in the tree. However, she needs memory to implement this strategy. In fact, she cannot win with a memoryless strategy, as such a strategy determines one branch, and Adam can produce only priorities 1 or 2 depending on whether right children occur infinitely often in the branch.

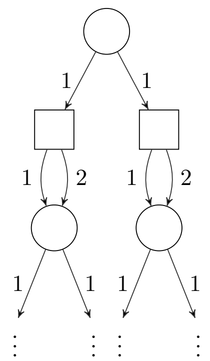

Figure 1 The prioritized Borel game from Example 22. Eve nodes are represented as circles, and Adam nodes as squares.

5.2 Upper bounds: Finite-memory results

The rest of this paper is dedicated to a discussion of Conjecture 19. In this section, we give some positive evidence, by proving some special cases of the conjecture. In Section 5.3, we exhibit some challenges in proving the conjecture, in particular we show that unbounded memory is necessary (Proposition 28). Some further special cases can be found in Manthe's PhD thesis [12, Chapter 3.4].

5.2.1 One priority

We start discussing the case of prioritized Borel games with just one priority. Although this case is relatively simple, it illustrates well our general strategy for tackling Conjecture 19. In this case, there is only one Borel winning set W_1 and the only condition for Eve to win is that the final branch lies in this set.

▶ **Proposition 23.** *In every prioritized Borel game with one priority, the winner has a memoryless strategy.*

Proof. Assume that Eve wins a given game with one priority using a strategy σ. Let $W_\sigma \subseteq 2^\omega$ be the subset of branches that can be produced when Eve follows the strategy σ.

Observe that W_σ is the set of branches that do not contain nodes from the set

$$N = \{w \in 2^* \mid w \text{ is unattainable using } \sigma\}.$$

Also, $W_\sigma \subseteq W_1$, as the strategy is winning.

We improve σ to a memoryless strategy: in every position of Eve reached by many possible histories, we arbitrarily choose one of them, and follow the decision made by σ after this history. Observe that the new strategy allows to reach only positions reachable by σ; in particular no nodes from N are visited, hence every branch consistent with the new strategy belongs to $W_\sigma \subseteq W_1$. ◀

Note that the set W_σ in the proof above is topologically closed: it is the complement an open set. Thus, in a sense, we achieved our goal in this proof by replacing the set W_1, of arbitrary complexity in the Borel hierarchy, by a topologically simple one: a closed set, for which memoryless strategies suffice.

More generally, Conjecture 19 holds for Borel subsets of "low" complexity. If all winning sets $W_1, \ldots, W_k$ are Boolean combination of at most m F_σ sets, then the winner has a strategy that uses memory bounded by a constant that only depends on k and m (this follows from the classical finite-memory determinacy result of Büchi-Landweber [3] generalized to infinite arenas [8]).

We conjecture that in every prioritized Borel game, we can replace the winning sets $W_1, \ldots, W_k$ by smaller subsets that are Boolean combination of F_σ sets of bounded complexity (depending on the number of states and priorities) while preserving the winner. This implies Conjecture 19. In fact, both conjectures are equivalent: given a finite-memory strategy, we can replace the winning sets by the sets that are consistent with the strategy (just as in the proof of Proposition 23 above). These sets are of bounded topological complexity by [2, Corollary 3.1]. (Formally, the statement of the corollary is not sufficient, but we also need to obtain a bound on the complexity of the Boolean combination. This follows from its proof. See also [12, Theorem 3.5.3].)

5.2.2 Two priorities

We now prove Conjecture 19 in the case of two priorities. Example 22 uses two priorities, so we know that memoryless strategies do not suffice in this case.

▶ **Proposition 24.** *In every prioritized Borel game with two priorities, the winner has a strategy using two states of memory.*

Proof. Assume that Eve wins a given prioritized Borel game with two priorities using a strategy σ. We improve this strategy to a new one that uses only two states of memory. As in the previous proof, at each position reached by multiple histories consistent with σ, we choose some of them (based only on the memory state), and we follow the decision of σ made after this history.

Compared to Proposition 23, we additionally have to ensure that we produce priority 2 as often as σ did: either finitely or infinitely often. We have two seemingly contradictory goals here: tend to produce priority 2 when σ did so, and tend not to produce priority 2 more often than σ did. Intuitively, we use the memory bit to remember which goal we are currently realizing, and we switch between them from time to time. Namely, whenever our play produced priority 2, we are temporarily happy with the first goal, and we switch to the second goal. But when all histories consistent with σ produced priority 2, then we can (and should) also produce it, so we switch to the first goal.

More formally, we identify some nodes (i.e., elements of 2^*, not positions) as *witnesses*. They are defined inductively as follows. The root is a witness. A non-root node $v \in 2^*$ is a witness if all plays consistent with σ necessarily use priority 2 between the previous witness node and v.

The memory in the new strategy records whether priority 2 has been produced since the last witness, which takes two memory states: "no" and "yes". In particular, if we are in a witness node, then no priorities have been produced since the last witness, and hence the memory says "no". Then, for the n-th position of a play, based on this memory we define a history h_n as follows:

- Suppose that the memory says "no", that is, priority 2 has not been seen since the last witness. As h_n we take any history that ends in the current position, is consistent with σ, and has not seen priority 2 since the last witness.
- Suppose that the memory says "yes", that is, priority 2 has been seen since the last witness. Define the *profile* of a history to be the sequence of priorities that were seen in the history. Among all histories that reach the current position consistently with σ, as h_n we choose one that has a maximal profile lexicographically.

If it is Eve's move, she does whatever σ would have done after h_n. It it is Adam's move, all his choices are valid continuations of h_n, consistent with σ.

In order to prove that the new strategy is winning for Eve, consider some play π that is consistent with this strategy, and consider three cases.

(1) Suppose that π visits infinitely many witnesses. We conclude that Eve wins by showing that that priority 2 is seen infinitely often along π, and that the branch of π belongs to W_2.

To show that priority 2 occurs infinitely often in π, we prove that it occurs between every two consecutive witness nodes (we cannot simply use the definition of witness nodes here, since π is not necessarily consistent with σ). Consider the position of π just before the second witness. If π did not produce priority 2 yet, then also the history h_n defined in this position did not produce priority 2 since the last witness. The move of π from this position extends h_n in a way consistent with σ, it thus produces priority 2 by the definition of a witness.

To show that the branch belongs to W_2, note that there are arbitrarily long histories (namely h_n) with the property: the history is consistent with σ and stays on the same branch as π. By Kőnig's lemma, some infinite play also has this property. Since this play is consistent with σ, the definition of witnesses tells us that it sees priority 2 between every two witnesses. Thus, the branch must belong to W_2, since σ is winning for Eve.

(2) Suppose now that π visits finitely many witnesses, and priority 2 is seen finitely often. Let $w \in 2^*$ be the last witness in the play. Because there are no more witnesses after w in the branch, we can use the definition of witnesses to conclude that there are arbitrarily long histories (namely h_n) with the property: the history is consistent with σ, stays on the same branch as π, and does not see priority 2 after node w. Again by Kőnig's lemma, there is some infinite play with this property. Since this play is winning for Eve, it follows that the branch is in W_1, and hence π is also winning for Eve.

(3) The last remaining case is when the play visits finitely many witnesses, but sees priority 2 infinitely often. In particular, priority 2 occurs after the last witness, and after this occurrence, each chosen history h_n is defined by maximising profiles. The history h_n extended by the next move of π is one of the histories taken into account while defining h_{n+1}; we may thus only switch to a history with a larger profile. This monotonicity ensures that the profiles of h_n converge to some sequence $\alpha \in \{1,2\}^\omega$. Moreover, whenever π contains priority 2, then the profile of h_{n+1} either contains 2 on the last position, or is strictly greater than h_n on earlier positions. This strict improvement occurs infinitely often, implying that the limit profile α has infinitely many occurrences of priority 2.

Now, the convergence means that there are arbitrarily long histories (appropriate prefixes of h_n) with the property: the history is consistent with σ, stays on the same branch as π, and its profile is a prefix of α. Another application of Kőnig's lemma gives us an infinite play with this property. Since this play is winning for Eve, it follows that the branch is in W_2, and thus π is winning for Eve as well. ◀

▶ Remark 25. We discuss how this proof relates to the topological simplification approach described in Section 5.2.1. For the sake of simplicity, we restrict to games with a single state. The case of more states requires the general argument explained at the end of Section 5.2.1.

The definition of a witness yields a coloring of the tree with two colors ("witness" or "not a witness"). We can replace W_1 by the set of branches that are compatible with σ and contain finitely many witnesses, which is an F_σ set contained in W_1.

The corresponding subset of W_2 is slightly more complex to define; it can be described as the intersection of branches compatible with σ with the union of two sets, corresponding to the first two cases examined in the above proof. The first one is the set of branches that contain infinitely many witnesses (a G_δ set, i.e., the complement of an F_σ set). The second is the set of branches in which the limit α of the profiles (which is well-defined by the single-state assumption) contains infinitely many twos. This is also a G_δ set.

This provides two topologically simple subsets, and the proof above describes a winning strategy for Eve in the obtained game. For games with more states, Boolean combinations of unboundedly many F_σ sets may be needed.

In the next section, we show that in games with 3 priorities there is no constant bound on the size of the memory required, independent of the number of states (see Proposition 28). We have not succeeded in proving Conjecture 19 for games with 3 priorities.

5.2.3 Nested branch conditions

We say that a sequence of subsets $W_1, \ldots, W_k \subseteq 2^\omega$ is *alternatingly nested* if for every $1 \leq p \leq k$,

$$W_p \subseteq \bigcap_{i<p} W_i \text{ for odd } p, \quad \text{and} \quad W_p \supseteq \bigcup_{i<p} W_i \text{ for even } p.$$

Note that the usual parity condition can be seen as a special case of an alternatingly nested condition: we take $W_p = \emptyset$ (i.e., Eve loses) for odd p and $W_p = 2^\omega$ (i.e., Eve wins) for even p. The next proposition generalizes memoryless determinacy of parity games to prioritized Borel games with winning conditions of this form.

In the classical setting, Muller games can be reduced to parity games, using the latest appearance record [8]. In the setting of this paper, we are not aware of any way of ensuring that the winning condition in a prioritized Borel game becomes alternatingly nested. In particular, as already explained, the latest appearance record applied to Muller-Borel games gives us prioritized Borel games, but the resulting winning condition needs not to be alternatingly nested.

▶ **Proposition 26.** *If the Borel sets of a prioritized Borel game are alternatingly nested, then one of the players can win using a memoryless strategy.*

Proof. Assume that Eve wins the game using a strategy σ. Once again, we improve it to a memoryless strategy by choosing one history leading to a position, and performing the decision σ made after this history.

While proving Proposition 24, it was neither better to have infinitely many nor finitely many occurrences of priority 2; we thus had to use a memory state to alternate between two modes. For alternatingly nested conditions, it is always good to produce a priority p infinitely often if p is even, and finitely often if p is odd. Thus, we should tend to produce even priorities when σ can do so, and tend not to produce odd priorities more often than σ did. To this end, we follow the worst possible history, defined next.

First, we generalize the notion of a profile from the previous proof to the case of more priorities. Namely, for a priority p, the *p-profile* of a history h is the sequence in $\{0,1\}^\omega$ having 1 at indices for which priority p was produced by h (we pad the sequence with zeroes after the end of the history). To compare two histories h_1, h_2, we find the largest p for which their p-profiles α_1, α_2 differ; then h_1 is *worse* than h_2 if

- p is odd, and α_1 is lexicographically larger than α_2 (intuitively: h_1 produces priority p more often), or
- p is even, and the last 1 in α_1 is earlier than in α_2 (intuitively: h_1 produces priority p less often), or
- p is even, the last 1 in α_1 and α_2 is at the same position, and α_1 is lexicographically larger than α_2 (just to break ties).

Then, among all histories that reach the current position consistently with σ, as h_n we choose the worst possible history (choosing arbitrarily among histories having the same p-profiles for all p). If it is Eve's move, she does whatever σ would have done after h_n.

In order to prove that the new strategy is winning for Eve, consider some play π that is consistent with this strategy. For every p (by induction, starting from the highest priorities), we prove that either

1. for all $j > p$, priority j is seen finitely often in π and, from some moment on, all histories h_n have the same j-profile, or
2. π is winning for Eve.

This is enough, since (1) cannot hold for $p = 1$.

In order to prove the above, fix some p. By the induction hypothesis (or, in the base case, because there are no larger priorities), we know that priorities greater than p are seen only finitely often in π and, from some moment on, all histories h_n have the same j-profile α_j, for every $j > p$. Note that α_j, being a profile of finite histories, may have 1 only at finitely many positions.

Suppose first that p is even and priority p occurs infinitely often in π. To show that the branch belongs to W_p, note that there are arbitrarily long histories (namely h_n) with the property: the history is consistent with σ, stays on the same branch as π, and for every $j > p$ its j-profile is α_j. By Kőnig's lemma, some infinite play also has this property. It is a play won by Eve, where priorities greater than p occur finitely often, which implies that the considered branch belongs to W_i for some $i \leq p$. We conclude that Eve wins on π, thanks to the inclusion $W_i \subseteq W_p$.

Next, suppose that p is even, but priority p occurs finitely often in π. Restrict our attention to positions after the moment where all j-profiles of h_n for $j > p$ have stabilized, and after all occurrences of priority p. We observe that the history h_n extended by the next move of π is one of the histories taken into account while defining h_{n+1}. Since this move does not produce priority p, the history h_{n+1} has either the same p-profile as h_n, or a worse one. The order used to define a worse history for even p is well-founded, meaning that we can change to a worse p-profile only a finite number of times. Afterwards, the p-profile of successive histories h_n stays the same; we thus obtain (1).

We now switch to the case of odd p. As soon as the j-profiles for $j > p$ remain the same, we have the same property as above: the history h_{n+1} has either the same p-profile as h_n, or a worse one. It follows that the p-profiles converge monotonically to some limit sequence α_p. Moreover, whenever priority p occurs in π, the p-profile of h_{n+1} is strictly worse than h_n (due to the history obtained by appending the move of π producing priority p to h_n).

If the limit profile α_p has 1 only on finitely many positions, then such a strict improvement could happen only finitely many times: the prefix of α_p containing all the ones becomes equal to prefixes of p-profiles of histories from some moment on, and the remaining suffix of zeroes is minimal lexicographically. This means that priority p is seen finitely often in π and, from some moment on, the p-profile of all further histories h_n is α_p; we obtain (1).

The opposite case is that the limit profile α_p contains infinitely many ones. Then, the convergence gives us arbitrarily long histories (appropriate prefixes of h_n) with the property: the history is consistent with σ, stays on the same branch as π, and for every $j \geq p$ its j-profile is α_j. By Kőnig's lemma, some infinite play also has this property. It is a play won by Eve, where priorities greater than p occur finitely often, but priority p occurs infinitely often; in consequence the branch belongs to W_p. The largest priority occurring infinitely often on π is some $i \leq p$; we conclude that Eve wins on π, thanks to the inclusion $W_i \subseteq W_p$. ◀

5.3 Lower bounds

This section presents some lower bounds on the memory required in prioritized Borel games. Namely, we show that the required memory size necessarily depends on both the number of priorities and the number of states of the game. We hope that this will help guide future approaches to solving Conjecture 19.

For a cleaner presentation of the lower bounds, we use prioritized Borel games where the branching of the tree may be greater than 2, but still finite and bounded. Formally, we let the set of positions be a subset of $C^* \times Q$, where C is a finite set of children. These games can be simulated by prioritized Borel games over the binary tree, in a way that adds neither priorities nor states: it suffices to introduce $\lceil \log |C| \rceil - 1$ intermediate levels between each two nodes of the tree of a given C-branching game.

5.3.1 Many priorities, one state

We generalize Example 22 to games with k priorities, for any k, and show that in these games Eve requires k states of memory to win.

▶ **Proposition 27.** *For every k, there is a prioritized Borel game with a single state and k priorities where Eve requires k states of memory to win.*

Proof. Consider the (k-branching) prioritized Borel game using k priorities depicted in Figure 2. As previously, we use circles to represent nodes owned by Eve, and squares to represent those owned by Adam. In this game, Adam picks a priority in $\{1, \dots, k\}$; then, Eve chooses to go to one of k possible children. Eve wins if the highest priority that occurs infinitely often coincides with the highest child number that is taken infinitely often. Formally, W_i is the set of branches where the i-th child is taken infinitely often, but for all $j > i$, the j-th child is taken only finitely often.

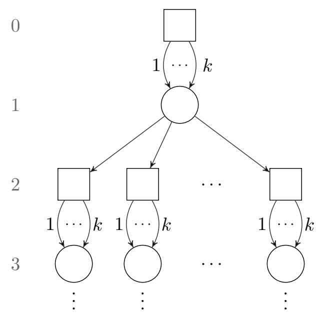

Figure 2 The prioritized Borel game from Proposition 27.

Eve can win this game by copying Adam's move (go to the i-th child if Adam picks priority i). This strategy uses k memory states, which record the most recent priority played by Adam.

We claim that this is optimal. Assume by contradiction that Eve has a winning strategy using $k-1$ memory states. Intuitively, in every step she fails to distinguish between two priorities; we will use this fact to build two plays on the same branch that produce a different maximal priority infinitely often. For each level n of the tree, we define by recursion:

- a position v_n of the game (controlled by Adam when n is even, and by Eve when n is odd),
- a memory state m_n, and
- two priorities, i_n, i'_n (only for odd n).

We let v_0 be the root of the tree and m_0 the initial memory state. Assume that these elements have been defined up to level n, for n even (v_n controlled by Adam). As there are only $k-1$ memory states, there are two priorities, i_{n+1} and i'_{n+1}, such that the memory update from state m_n when reading these two priorities goes to the same memory state: we let m_{n+1} be this state. We let v_{n+1} be the only successor of v_n and v_{n+2} be the child of v_{n+1} picked by the strategy in the memory state m_{n+1}. Finally, let m_{n+2} be the memory state obtained by updating m_{n+1} over this last transition.

This shows that, when Eve plays according to this strategy, Adam can decide to produce a play in the branch $v_0v_1v_2\dots$, while producing any sequence of priorities in $\{i_0, i'_0\}\{i_1, i'_1\}\dots$. Taking the sequences of the maximal and minimal elements of $\{i_n, i'_n\}$, respectively, we obtain two sequences with different largest priority occurring infinitely often, concluding the proof. ◀

5.3.2 Three priorities, many states

In Proposition 24, we showed that for games using two priorities, we can obtain a memory bound (two states) which is independent of the number of states of the game. This is no longer true for three priorities.

▶ **Proposition 28.** *For every s, there is a prioritized Borel game using three priorities where Eve requires $s+1$ memory states to win.*

We define a prioritized Borel game G_s where Eve requires $s+1$ memory states to win. We begin with an intuitive turn-based description of the game, which we later formalize as a prioritized Borel game with three priorities. In the game, the players construct a branch in a tree with branching $2 \cdot s$. In order to avoid confusion with priorities, which are numbers, we use $C = \{c_1, \dots, c_{2s}\}$ for the child numbers in the tree. The players repeat the following two steps infinitely often:

1. Adam picks an even number $2i \in \{1, \dots, 2s\}$ and appends a finite sequence from $\{c_1, \dots, c_{2i}\}^+$ to the current branch.
2. Eve appends either c_1 or c_{2i} to the current branch.

In order to win, Eve needs to append c_1 infinitely often, and to ensure that the branch satisfies the parity condition, that is, the largest number appearing infinitely often is even.

Eve can win the game by alternating her choices (c_1 or c_{2i}) separately for each i. That is, for each $i \in \{1, \dots, s\}$, she remembers which choice she made the last time i was selected by Adam, and she uses the opposite choice. This strategy requires 2^s memory states. We will show below that, after the game is suitably formalised in our model, at least $s+1$ memory states are needed.

At every point, Eve has direct access to the finite prefix of the branch that has been produced up to that point. However, importantly, she does not have direct access to information about which players produced particular elements of the sequence, in particular when did Eve append c_1. She needs memory to recall this information. In a crucial way, Adam can add arbitrarily long finite sequences of nodes in his turns, which he can use to confuse Eve about the history of the game.

We now show that this game can be formalised using prioritized Borel games. In our formalisation, we use three priorities $\{1, 2, 3\}$ to enforce that Adam plays finite sequences in each step, and that Eve appends c_1 infinitely often. Also, Adam's turn (1) will be modified so that Adam starts with $i = 1$, and then he is allowed to gradually increase it during his turn. This will have no effect on the game, but makes the representation easier. Finally, we will constrain Adam so that he must end his turn by picking c_{2i-1}. This constraint will only make it harder to prove a lower bound on Eve's memory.

Here is the formal definition of the prioritized Borel game. Each tree node has $2s$ states: s of them controlled by Adam, $q_1, \dots, q_s$; and s of them controlled by Eve, $p_1, \dots, p_s$. When in a state q_i, Adam picks a number $j \geq i$ (which corresponds to the option of increasing i), emits priority 1, and either

- goes to state q_j in any child from $\{c_1, \dots, c_{2j}\}$, or
- goes to state p_j in the child c_{2j-1}.

In the first case, Adam controls the new state, which means that he intends to continue choosing children, while in the second case he hands control over to Eve. When in a state p_i, Eve can choose to either produce priority 2 and go to the child c_{2i}, or to produce priority 3 and go to the first child c_1. In both cases, we go to the state q_1.

The winning condition is as follows. The set W_1 is full (if eventually only priority 1 is produced, Eve wins). The set W_2 is empty (Adam always wins if 2 is produced infinitely often and 3 finitely often). If 3 is produced infinitely often, Eve wins if and only if the highest child that occurs infinitely often is even. As we have explained earlier, Eve can win this game with finite memory. However, the amount of memory must grow with s, as the following lemma shows.

▶ **Lemma 29.** *Eve cannot win the game G_s using s memory states.*

Proof. We prove the lower bound by induction on s. We use a slightly stronger induction hypothesis: Eve cannot win G_s with a strategy that uses s reachable memory states in positions owned by her (that is, positions of the form (w, p_i)).

For the base case, $s = 1$, a strategy using 1 memory state in Eve positions is just a memoryless strategy; we show that no such strategy suffices to win G_1. Fix a memoryless strategy, that is, in each node $w \in C^*$ fix Eve's choice of producing priority 3 or going to child c_2 from p_1. For every node $v \in C^*$, there is some descendant w in the leftmost branch below v such that Eve goes to c_2 from (w, p_1), as otherwise Adam can produce an infinite play in this branch. But then, Adam can let Eve play only in these nodes, so priority 3 will never be produced. Therefore, the strategy is not winning.

For $s \geq 2$ assume, towards a contradiction, that Eve has a winning strategy σ in G_s using s memory states in her positions. We say that Eve *plays* 3 *conditionally* from a position (w, p_i), if for some memory state the strategy σ chooses to play priority 3 from that position. We let $C_{s-1} = \{c_1, \dots, c_{2s-1}\}$. We say that game position (w, p_i) is a C_{s-1}-*descendant* of a node w_0 if $w \in w_0 C^*_{s-1} c_{2i-1}$, that is, w is the child c_{2i-1} of its parent, it is below w_0, and the child c_{2s} does not occur after w_0.

▷ Claim 30. There exists a node such that from every of its C_{s-1}-descendants Eve plays 3 conditionally.

Proof. Assume that this is not the case. Then, we can build by induction a play consistent with σ in which priority 3 is never produced: In each step, Adam chooses a C_{s-1}-descendant of the current node where Eve does not play 3 conditionally. He can go to that position, and let her play, producing priority 2 and going to child c_{2i} for some i (which she does regardless of the memory state). Repeating this strategy, Adam produces an infinite path where priority 3 is never visited, so it is losing for Eve, a contradiction. ◁

Since at the beginning of the play Adam may go to any node of the tree, we may assume that the initial node given in the previous claim is the root of the tree. We fix a function

$$f\colon C^*_{s-1} \times \{p_1, \dots, p_s\} \to M$$

that associates to each position (w, p_i) some memory state where the strategy σ chooses to produce priority 3. (This function is well-defined by the previous claim.)

The induction hypothesis is used to obtain the next claim.

▷ Claim 31. Let $w_0 \in C^*_{s-1}$. There is a C_{s-1}-descendant (w, p_i) of w_0 and a history consistent with σ going from (w_0, q_1) to (w, p_i) such that the memory state when in (w, p_i) is $f(w, p_i)$.

Proof. Suppose that the claim does not hold. Consider the restriction to the game G_{s-1} starting at the node w_0; this only eliminates some moves of Adam, so the strategy of Eve is winning also in this subgame. In each position (w, p_i) controlled by Eve, there is a memory state that is not reachable. Therefore, Eve wins G_{s-1} using only $s - 1$ memory states in her positions, contradicting the induction hypothesis. ◁

To conclude, we build recursively a play where Eve never chooses to pick c_{2i}; such a play is losing for her, as Adam always proceeds to child c_{2i-1} while going to p_i. At every step, it suffices that Adam plays accordingly to the play given by the previous claim. ◀

6 Generalizations

Our upper bounds are proved by taking some winning strategy with possibly infinite memory, and improving it so that it uses finite memory. In this sense, the Borel determinacy is used only to ensure that the former strategy exists. Therefore, one could consider a stronger variant of Conjecture 2, in which: (a) we no longer require the conditions $W \subseteq 2^\omega$ to be Borel, but we allow them to be arbitrary sets; and (b) we require that the game has a winner. This variant would be stronger than Conjecture 2, since (b) is automatic if we assume that the conditions $W \subseteq 2^\omega$ are Borel. The stronger variant of the conjecture might be useful for sets beyond Borel sets, which still have determinacy results, such as analytic sets. (We do not use uncountable branching, which can be an obstruction to such generalizations.) Another related observation is that we do not depend on the full axiom of choice, and thus we could consider the full axiom of determinacy and the unrestricted monadic theory of $2^{\leq\omega}$.

Yet another possible generalization involves the monadic theory of $2^{<\alpha}$ for various ordinals α. Rabin's Tree Theorem is the case of $\alpha = \omega$, and this paper is about $\alpha = \omega + 1$. Increasing α causes at least two kinds of additional difficulties. First, the behavior of automata for such a monadic theory along a single branch should be related to automata for words of length α, which become more intricate for greater α [4, 15, 16]. Second, our approaches require determinacy, which beyond $2^{<\omega\cdot 2}$ requires large cardinals even for Borel sets [1] and fails for games of length ω_1 [14, Theorem 7.3].

References

1 J. P. Aguilera. Long Borel games. *Israel J. Math.*, 243(1):273–314, 2021. doi:10.1007/s11856-021-2160-y.

2 Mikołaj Bojańczyk, Damian Niwiński, Alexander Rabinovich, Adam Radziwończyk-Syta, and Michał Skrzypczak. On the Borel complexity of MSO definable sets of branches. *Fund. Inform.*, 98(4):337–349, 2010. doi:10.3233/fi-2010-231.

3 J. Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969. doi:10.2307/1994916.

4 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. doi:10.1002/malq.19830290503.

5 Mirna Džamonja. MSO logic of the real order with the set quantifiers ranging over the Borel sets. *CoRR*, 2025. arXiv:2512.23003.

6 Nathanaël Fijalkow, C. Aiswarya, Guy Avni, Nathalie Bertrand, Patricia Bouyer, Romain Brenguier, Arnaud Carayol, Antonio Casares, John Fearnley, Paul Gastin, Hugo Gimbert, Thomas A. Henzinger, Florian Horn, Rasmus Ibsen-Jensen, Nicolas Markey, Benjamin Monmege, Petr Novotný, Pierre Ohlmann, Mickael Randour, Ocan Sankur, Sylvain Schmitz, Olivier Serre, Mateusz Skomra, Nathalie Sznajder, and Pierre Vandenhove. Games on graphs: From logic and automata to algorithms, 2026. Cambridge University Press. arXiv:2305.10546.

7 Yuri Gurevich. Monadic second-order theories. In J. Barwise and S. Feferman, editors, *Model-theoretic logics*, Perspectives in Mathematical Logic, pages 479–506. Springer-Verlag, New York, 1985.

8 Yuri Gurevich and Leo Harrington. Trees, automata and games. In *Proc. ACM Symp. on Theory of Computing*, pages 60–65. Assoc. Comput. Mach., 1982. doi:10.1145/800070.802177.

9 Yuri Gurevich and Saharon Shelah. Monadic theory of order and topology in ZFC. *Ann. Math. Logic*, 23(2-3):179–198, 1982. doi:10.1016/0003-4843(82)90004-3.

10 Lawrence H. Landweber. Decision problems for omega-automata. *Math. Syst. Theory*, 3(4):376–384, 1969. doi:10.1007/BF01691063.

11 Sven Manthe. The Borel monadic theory of order is decidable. *CoRR*, 2024. arXiv:2410.00887.
12 Sven Manthe. *Decidability in Borel Monadic Logic*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2026.
13 Donald A. Martin. Borel determinacy. *Ann. of Math. (2)*, 102(2):363–371, 1975. doi:10.2307/1971035.
14 Jan Mycielski. On the axiom of determinateness. *Fund. Math.*, 53:205–224, 1963/64. doi:10.4064/fm-53-2-205-224.
15 Itay Neeman. Finite state automata and monadic definability of singular cardinals. *J. Symbolic Logic*, 73(2):412–438, 2008. doi:10.2178/jsl/1208359052.
16 Itay Neeman. Monadic definability of ordinals. In *Computational prospects of infinity. Part II. Presented talks*, volume 15 of *Lect. Notes Ser. Inst. Math. Sci. Natl. Univ. Singap.*, pages 193–205. World Sci. Publ., Hackensack, NJ, 2008. doi:10.1142/9789812796554_0010.
17 Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. Soc.*, 141:1–35, 1969. doi:10.2307/1995086.
18 Saharon Shelah. The monadic theory of order. *Ann. of Math. (2)*, 102(3):379–419, 1975. doi:10.2307/1971037.
19 Saharon Shelah. On what I do not understand (and have something to say), model theory. *Math. Japon.*, 51(2):329–377, 2000.
20 Wolfgang Thomas. On the synthesis of strategies in infinite games. In Ernst W. Mayr and Claude Puech, editors, *STACS 95, 12th Annual Symposium on Theoretical Aspects of Computer Science, Munich, Germany, March 2-4, 1995, Proceedings*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995. doi:10.1007/3-540-59042-0_57.
21 Wolfgang Thomas. Languages, automata, and logic. In *Handbook of formal languages, Vol. 3*, pages 389–455. Springer, Berlin, 1997. doi:10.1007/978-3-642-59126-6_7.