What kind of memory is needed to win infinitary Muller games? *

Erich Grädel
Łukasz Kaiser

Mathematische Grundlagen der Informatik
RWTH Aachen University
52056 Aachen, Germany
{graedel,kaiser}@informatik.rwth-aachen.de

Abstract

In an influential paper entitled *"How much memory is needed to win infinite games"*, Dziembowski, Jurdziński, and Walukiewicz have shown that there are Muller games of size $O(n)$ whose winning strategies require memory of size at least $n!$. This shows that the LAR-memory, based on the latest appearance records introduced by Gurevich and Harrington, is optimal for solving Muller games. We review these results and reexamine the situation for the case of infinitary Muller games, i.e. Muller games with infinitely many priorities. We introduce a new, infinite, memory structure, based on finite appearance records (FAR) and investigate classes of Muller games that can be solved with FAR-memory.

1 Introduction

We study two-player games of infinite duration that are played on finite or infinite game graphs. Such a game is *determined* if, from each position, one of the two players has a winning strategy. On the basis of the axiom of choice it is not difficult to prove that there exist nondetermined games. The classical theory of infinite games in descriptive set theory links determinacy of games with topological properties of the winning conditions. Usually the format of Gale-Stewart games is used where the two players strictly alternate, and in each move a player selects an element of $\{0, 1\}$; thus the outcome of a play is an infinite string $\pi \in \{0, 1\}^\omega$. Gale-Stewart games can be viewed as graph

*This research has been partially supported by the European Community Research Training Network "Games and Automata for Synthesis and Validation" (GAMES)

game, for instance on the infinite binary tree, or on a bipartite graph with four nodes. Zermelo [20] proved already in 1913 that if in each play of a game, the winner is determined already after a finite number of moves, then one of the two players has a winning strategy. In topological terms the winning sets in such a game are clopen (open and closed). By a celebrated theorem due to Martin [16] every game where the winning condition is given by a Borel set is determined.

For game theory that relates to computer science, determinacy is just a first step in the analysis of a game. Rather than in the mere existence of winning strategies, one is interested in effective constructions of reasonably simple winning strategies. An aspect of crucial importance for the complexity of a strategy is its dependency on the history of the play.

In general, strategies may be very complicated functions that can depend on the entire history of the play. However, in many cases, simple strategies suffice. Of particular interest are *positional strategies* for which the next move depends only the current position, and not at all on previous history. That is, a player moving according to a positional strategy f will at a position v always perform the same move $v \to f(v)$ no matter how often and by what path position v has been reached. A game is *positionally determined*, if from each position, one of the two players has a positional winning strategy. Another important case are *finite-memory strategies* for which the dependency on the history can be calculated on the basis of a finite set of memory states and which can thus be implemented by a finite automaton.

Positional determinacy and determinacy via finite-memory strategies have been extensively studied for games whose winning conditions are defined in terms of a mapping that assigns to each position a *priority* from a finite set C. Specifically, in *Muller games* the winner of a play is determined by the set of those priorities that have been seen infinitely often. It has been proved by Gurevich and Harrington [12] that Muller games are determined via finite memory strategies that are based on a data structure called *latest appearance records* (LAR). Intuitively a latest appearance record is a list of priorities in the order in which they have last occurred in the play. Thus, on n priorities, an LAR-memory has $n!$ memory states. Dziembowski, Jurdziński, and Walukiewicz [6] have shown that LAR-strategies are essentially optimal for Muller games.

Theorem 1.1. There exists a sequence $(\mathcal{G}_n)_{n\in\omega}$ of Muller games such that the game graph of $\mathcal{G}_n$ is of size $O(n)$ and every winning strategy for $\mathcal{G}_n$ requires a memory of size at least $n!$

In particular, Muller games need not be positionally determined, not even for solitaire games (where only one player moves). An important special case of Muller games are *parity games*. These are games with a priority labeling Ω assigning to each position v a priority $\Omega(v) \in \{0, \ldots, d\}$, for some $d \in \mathbb{N}$, and with parity winning condition: Player 0 wins a play π if the least priority occurring infinitely often in π is even. Parity games are of importance for several reasons.

(1) Many classes of games arising in practical applications admit reductions to parity games (over larger game graphs). This is the case for games modeling reactive systems, with winning conditions specified in some temporal logic or in monadic second-order logic over infinite paths (S1S), for Muller games, but also for games with partial information apeearing in the synthesis of distributed controllers.

(2) Parity games arise as the model checking games for fixed point logics such as the modal μ-calculus or LFP, the extension of first-order logic by least and greatest fixed points [8, 10]. In particular the model checking problem for the modal μ-calculus can be solved in polynomial time if, and only if, winning regions for parity games can be decided in polynomial time.

(3) Parity games are positionally determined [7, 17]. This is a game theoretical result of fundamental importance and with great algorithmic relevance.

To establish positional determinacy or finite-memory determinacy is a fundamental step in the analysis of an infinite game, and is also crucial for the algorithmic construction of winning strategies. In the case of parity games with finitely many priorities the positional determinacy immediately implies that winning regions can be decided in NP $\cap$ Co-NP; with a little more effort it follows that the problem is in fact in UP $\cap$ Co-UP [13]. Further, although it is not known yet whether parity games can be solved in polynomial time, all known approaches towards an efficient algorithmic solution make use of positional determinacy. The same is true for the efficient algorithms that we have for specific classes of parity games, including parity games with a bounded number of priorities [14], games where even and odd cycles do not intersect, solitaire games and nested solitaire games [2], and parity games of bounded tree width [18], bounded entanglement [3], or bounded DAG-width [1, 19].

For several reasons it is interesting to generalise the theory of infinite games to the case of infinitely many priorities. Besides the theoretical interest, winning conditions depending on infinitely many priorities arise naturally in several contexts. In pushdown games, stack height and stack contents are natural parameters that may take infinitely many values. In [5], Cachat, Duparc, and Thomas study pushdown games with an infinity condition on stack contents, and Bouquet, Serre, and Walukiewicz [4] consider more general winning conditions for pushdown games, combining a parity condition on the states of the underlying pushdown automaton with an unboudedness condition on stack heights. Similarly, Gimbert [9] considers games of bounded degree where the parity winning condition is combined with the requirement that an infinite portion of the game graph is visited.

A systematic study of positional determinacy of games with infinitely many priorities has been initiated in [11]. It has been shown that there are interesting cases where positional determinacy is a consequence of the winning condition only, holding for all game graphs. Most notably this is the case for the parity condition on ω. Moreover a complete classification of the infinitary Muller conditions with this property has been established in [11] and it has been shown that all of them are equivalent to a parity condition.

Whereas the proof for the positional determinacy of parity games with priorities in ω is somewhat involved, it is quite easy to construct games with infinitary Muller winning conditions whose winning strategies require infinite memory. For instance there are very simple max-parity games (where the maximal priority seen infinitely often determines the winner) with this property (see Section 4). Nevertheless, the required (infinite) memory structures are often quite simple. In some cases it is enough to store just the maximal priority seen so far. In other cases a tuple (of fixed length) of previously seen priorities suffices to determine the next move of a winning strategy. This motivates the introduction of a new memory structure for winning strategy, that we call *finite appearance records* (FAR) which generalise the LARs used for finitary Muller games. We determine some classes of Muller games that can be reduced to parity games via FAR-memories. These include games where the wining condition is a downward cone, a singleton condition, a finite union of upwards cones or consists of finitely many winning sets only. Further the same property holds for all max-parity games where the difference between the priorities of any two consecutive positions is bounded.

Here is an outline of this paper. In Section 2 we present the technical definitions on games, winning strategies, memory structures and game reductions. In Section 3 we survey the case of Muller games with finitely many priorities and present proofs of two classical results of the field. First we show that Street-Rabin games are positionally determined for one player (which also implies that parity games are positionally determined for both players). Second, we describe the LAR-memory and show how Muller games can be reduced, via LAR-memory, to parity games. In Section 4 we briefly survey the results from [11] on parity games and Muller games with infinitely many priorities. In Section 5 we introduce finite appearance records and FAR-memory structures. Finally, in Section 6 we analyse some classes of Muller games that can be solved with FAR-memories.

2 Games, strategies, and memory structures

We study infinite two-player games with complete information, specified by a triple $\mathcal{G} = (G, \Omega, W)$ where $G = (V, V_0, V_1, E)$ is a game graph, equipped with a partioning $V = V_0 \cup V_1$ of the nodes into positions of Player 0 and positions of Player 1, where $\Omega : V \to C$ is a function that assigns to each position a *priority* (or colour) from a set C, and where W specifies a *winning condition.* The pair $(\mathcal{G}, \Omega)$ is called the *arena* of the game. In case $(v, w) \in E$ we call w a successor of v and we denote the set of all successors of v by vE. To avoid tedious case distinctions, we assume that every position has at least one successor. A *play* in $\mathcal{G}$ is an infinite path $v_0v_1 \ldots$ formed by the two players starting from a given initial position v_0. Whenever the current position v_i belongs to V_0, then Player 0 chooses a successor $v_{i+1} \in v_iE$, if $v_i \in V_1$, then $v_{i+1} \in v_iE$ is selected by Player 1. The *winning condition* describes which of the infinite plays $v_0v_1 \ldots$ are won by Player 0, in terms of the sequence $\Omega(v_0)\Omega(v_1)\ldots$ of priorities appearing in the play. Thus, a winning condition is given by a set $W \subseteq C^\omega$ of infinite sequences of priorities.

In traditional studies of infinite games it is usually assumed that the set C of priorities is finite, although the game graph itself (i.e., the set of positions) may well be infinite. This permits, for instance, to specify winning condition by formulae from a logic on infinite paths, such as LTL (linear time temporal logic), FO (first-order logic), or MSO (monadic second-order logic) over a vocabulary that uses the linear order $<$ and monadic predicates P_c for each priority $c \in C$.

A *(deterministic) strategy* for Player σ in a game $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ is a partial function $f : V^*V_\sigma \to V$ that maps initial segments $v_0v_1 \ldots v_m$ of plays ending in a position $v_m \in V_\sigma$ to a successor $f(v_0 \ldots v_m) \in v_mE$. A play $v_0v_1 \cdots \in V^\omega$ is *consistent with* f, if Player σ always moves according to f, i.e., if $v_{m+1} = f(v_0 \ldots v_m)$ for every m with $v_m \in V_\sigma$. We say that such a strategy f is winning from position v_0, if every play that starts at

v_0 and that is consistent with f, is won by Player σ. The *winning region* of Player σ, denoted W_σ, is the set of positions from which Player σ has a winning strategy.

A game $\mathcal{G}$ is *determined* if $W_0 \cup W_1 = V$, i.e., if from each position one of the two players has a winning strategy. In general, winning strategies can be very complicated. It is of interest to determine which games admit simple strategies, in particular *finite memory strategies* and *positional strategies*. While positional strategies only depend on the current position, not on the history of the play, finite memory strategies have access to bounded amount of information on the past. Finite memory strategies can be defined as strategies that are realisable by finite automata. However, we will also need to consider strategies that require infinite memory. We therefore introduce a general notion of a memory structure and of a strategy with memory, generalising the finite memory strategies studied for instance in [6].

Definition 2.1. A *memory structure* for a game $\mathcal{G}$ with positions in V is a triple $\mathfrak{M} = (M, \text{update}, \text{init})$, where M is a set of *memory states*, update : $M \times V \to M$ is a *memory update function* and init : $V \to M$ is a *memory initialisation function*. The *size* of the memory is the cardinality of the set M. A *strategy with memory* $\mathfrak{M}$ for Player σ is given by a next-move function $F : V_\sigma \times M \to V$ such that $F(v, m) \in vE$ for all $v \in V_\sigma, m \in M$. If a play, from starting position v_0, has gone through positions $v_0 v_1 \dots v_n$ the memory state is $m(v_0 \dots v_n)$, defined inductively by $m(v_0) = \text{init}(v_0)$, and $m(v_0 \dots v_i v_{i+1}) = \text{update}(m(v_0 \dots v_i), v_{i+1})$. In case $v_n \in V_\sigma$, the next move from $v_1 \dots v_n$, according to the strategy, leads to $F(v_n, m(v_0 \dots, v_n))$. In case $|M| = 1$, the strategy is positional; it can be described by a function $F : V_\sigma \to V$.

We will say that a game is determined via memory $\mathfrak{M}$ if it is determined and both players have winning strategies with memory $\mathfrak{M}$ on their winning regions. A game is *positionally determined* if it is determined via positional winning strategies.

Given a game graph $G = (V, V_0, V_1, E)$ and a memory structure $\mathfrak{M} = (M, \text{update}, \text{init})$ we obtain a new game graph $G \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E_{\text{update}})$ where

$$E_{\text{update}} = \{(v, m)(v', m') : (v, v') \in E \text{ and } m' = \text{update}(m, v')\}.$$

Obviously, every play $(v_0, m_0)(v_1, m_1) \dots$ in $G \times \mathfrak{M}$ has a unique projection to the play $v_0 v_1 \dots$ in G. Conversely, every play $v_0, v_1, \dots$ in G has a unique extension to a play $(v_0, m_0)(v_1, m_1)\dots$ in $G \times \mathfrak{M}$ with $m_0 = \text{init}(v_0)$ and $m_{i+1} = \text{update}(m_i, v_{i+1})$.

Consider two games $\mathcal{G} = (G, \Omega, W)$ and $\mathcal{G}' = (G', \Omega', W')$. We say that $\mathcal{G}$ *reduces via memory* $\mathfrak{M}$ *to* $\mathcal{G}'$, (in short $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$) if $G' = G \times \mathfrak{M}$ and every play in $\mathcal{G}'$ is won by the same player as the projected play in $\mathcal{G}$.

Given a memory structure $\mathfrak{M}$ for G and a memory structure $\mathfrak{M}'$ for $G \times \mathfrak{M}$ we obtain a memory structure $\mathfrak{M}^* = \mathfrak{M} \times \mathfrak{M}'$ for G. The set of memory locations is $M \times M'$ and we have memory initialization $\text{init}^*(v) = (\text{init}(v), \text{init}'(v, \text{init}(v))$ and the update function

$$\text{update}^*((m, m'), v) := (\text{update}(m, v), \text{update}'(m', (v, \text{update}(m, v))).$$

Proposition 2.2. Suppose that a game $\mathcal{G}$ reduces to $\mathcal{G}'$ via memory $\mathfrak{M}$ and that Player σ has a winning strategy for $\mathcal{G}'$ with memory $\mathfrak{M}'$ from $(v_0, \text{init}(v_0)))$. Then Player σ has a winning strategy for $\mathcal{G}$ with memory $\mathfrak{M} \times \mathfrak{M}'$ from position v_0.

Proof. Given a strategy $F' : (V_\sigma \times M) \times M' \to (V \times M)$ for Player σ on $\mathcal{G}'$ we have to construct a strategy $F : (V_\sigma \times (M \times M')) \to V \times (M \times M')$.

For any pair $(v, m) \in V_\sigma \times M$ we have that $F'(v, m) = (w, \text{update}(m, w))$ where $w \in vE$. We now put $F(v, mm') = w$. If a play in $\mathcal{G}$ that is consistent with F proceeds from position v, with current memory location (m, m'), to a new position w, then the memory is updated to (n, n') with $n = \text{update}(m, w)$ and $n' = \text{update}'(m', (w, n))$. In the extended play in $\mathcal{G}'$ we have an associated move from position (v, m) to (w, n) with memory update from m' to n'. Thus, every play in $\mathcal{G}$ from initial position v_0 that is consistent with F is the projection of a play in $\mathcal{G}'$ from $(v_0, \text{init}(v_0))$ that is consistent with F'. Therefore, if F' is a winning strategy from $(v_0, \text{init}(v_0))$, then F is a winning strategy from v_0. Q.E.D.

Corollary 2.3. Every game that reduces via memory $\mathfrak{M}$ to a positionally determined game, is determined via memory $\mathfrak{M}$.

Obviously, memory reductions between games compose. If $\mathcal{G}$ reduces to $\mathcal{G}'$ with memory $\mathfrak{M} = (M, \text{update}, \text{init})$ and $\mathcal{G}'$ reduces to $\mathcal{G}''$ with memory $\mathfrak{M}' = (M', \text{init}', \text{update}')$ then $\mathcal{G}$ reduces to $\mathcal{G}''$ with memory $(M \times M', \text{init}'', \text{update}'')$ with $\text{init}''(v) = (\text{init}(v), \text{init}'(v, \text{init}(v)))$ and

$$\text{update}((m, m'), v) = (\text{update}(m, v), \text{update}'(m', (v, \text{update}(m, v))).$$

3 Games with finitely many priorities

In this section we consider Muller games, Street-Rabin games, and parity games with finitely many priorities.

3.1 Muller games and Street-Rabin games

Definition 3.1. A *Muller winning condition* over a finite set C of priorities is written in the form $(\mathcal{F}_0, \mathcal{F}_1)$ where $\mathcal{F}_0 \subseteq \mathcal{P}(C)$ and $\mathcal{F}_1 = \mathcal{P}(C) - \mathcal{F}_0$. A play π in a game with Muller winning condition $(\mathcal{F}_0, \mathcal{F}_1)$ is won by Player σ if, and only if, $\mathrm{Inf}(\pi)$, the set of priorities occurring infinitely in π, belongs to $\mathcal{F}_\sigma$.

The *Zielonka tree* for a Muller condition $(\mathcal{F}_0, \mathcal{F}_1)$ over C is a tree $Z(\mathcal{F}_0, \mathcal{F}_1)$ whose nodes are labelled with pairs (X, σ) such that $X \in \mathcal{F}_\sigma$. We define $Z(\mathcal{F}_0, \mathcal{F}_1)$ inductively as follows. Let $C \in \mathcal{F}_\sigma$ and $C_0, \ldots, C_{k-1}$ be the maximal sets in $\{X \subseteq C : X \in \mathcal{F}_{1-\sigma}\}$. Then $Z(\mathcal{F}_0, \mathcal{F}_1)$ consists of a root, labeled by (C, σ), to which we attach as subtrees the Zielonka trees $Z(\mathcal{F}_0 \cap \mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$, for $i = 0, \ldots, k-1$.

Besides parity games there are other important special cases of Muller games. Of special relevance are games with Rabin and Street conditions because these are positionally determined for one player [15].

Definition 3.2. A *Streett-Rabin condition* is a Muller condition $(\mathcal{F}_0, \mathcal{F}_1)$ such that $\mathcal{F}_0$ is closed under union.

In the Zielonka tree for a Streett-Rabin condition, the nodes labeled $(X, 1)$ have only one successor. We remark that in the literature, Streett and Rabin conditions are often defined in a different manner, based on a collection $\{(E_i, F_i) : i = 1, \ldots k\}$ of pairs of sets. However, it is not difficult to see that the definitions are equivalent [21]. Further, it is also easy to show that if both $\mathcal{F}_0$ and $\mathcal{F}_1$ are closed under union, then $(\mathcal{F}_0, \mathcal{F}_1)$ is equivalent to a parity condition. The Zielonka tree for a parity condition is just a finite path.

In a Streett-Rabin game, Player 1 has a positional wining strategy on his winning region. On the other hand, Player 0 can win, on his winning region, via a finite memory strategy, and the size of the memory can be directly read of from the Zielonka tree. We present an elementary proof of this result. The exposition is inspired by [6]. In the proof we use the notion of an attractor.

Definition 3.3. Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be an arena and let $X, Y \subseteq V$, such that X induces a subarena of $\mathcal{G}$ (i.e., every position in X has a successor in X). The *attractor* of Player σ of Y in X is the set $\mathrm{Attr}_\sigma^X(Y)$ of those positions

$v \in X$ from which Player σ has a strategy in $\mathcal{G}$ to force the play into Y. More formally $\mathrm{Attr}_\sigma^X(Y) = \bigcup_\alpha Z^\alpha$ where

$$\begin{aligned} Z^0 &= X \cap Y, \\ Z^{\alpha+1} &= Z^\alpha \cup \{v \in V_\sigma \cap X : vE \cap Z^\alpha \neq \emptyset\} \cup \{v \in V_{1-\sigma} \cap X : vE \subseteq Z^\alpha\} \\ Z^\lambda &= \bigcup_{\alpha<\lambda} Z^\alpha \quad \text{for limit ordinals } \lambda \end{aligned}$$

On $\mathrm{Attr}_\sigma^X(Y)$, Player σ has a *positional attractor strategy* to bring the play into Y. Moreover $X \setminus \mathrm{Attr}_\sigma^X(Y)$ is again a subarena.

Theorem 3.4. Let $\mathcal{G} = (V, V_0, V_1, E, \Omega)$ be game with Streett-Rabin winning condition $(\mathcal{F}_0, \mathcal{F}_1)$. Then $\mathcal{G}$ is determined, i.e. $V = W_0 \cup W_1$, with a finite memory winning strategy of Player 0 on W_0, and a positional winning strategy of Player 1 on W_1. The size of the memory required by the winning strategy for Player 0 is bounded by the number of leaves of the Zielonka tree for $(\mathcal{F}_0, \mathcal{F}_1)$.

Proof. We proceed by induction on the number of priorities in C or, equivalently, the depth of the Zielonka tree $Z(\mathcal{F}_0, \mathcal{F}_1)$. Let ℓ be number of leaves of $Z(\mathcal{F}_0, \mathcal{F}_1)$. We distinguish two cases.

First, we assume that $C \in \mathcal{F}_1$. Let

$$X_0 := \{v : \text{Player 0 has a winning strategy with memory of size} \leq \ell \text{ from } v\},$$

and $X_1 = V \setminus X_0$. It suffices to prove that Player 1 has a positional winning strategy on X_1. To construct this strategy, we combine three positional strategies of Player 1, a trap strategy, an attractor strategy, and a winning strategy on a subgame with fewer priorities.

We observe that X_1 is a trap for Player 0; this means that Player 1 has a positional trap-strategy t on X_1 to enforce that the play stays within X_1.

Since $\mathcal{F}_0$ is closed under union, there is a unique maximal subset $C' \subseteq C$ with $C' \in \mathcal{F}_0$. Let $Y := X_1 \cap \Omega^{-1}(C \setminus C')$ and let $Z = \mathrm{Attr}_1^{X_1}(Y) \setminus Y$. Observe that Player 1 has a positional attractor strategy a, by which he forces from any position $z \in Z$ that the play reaches Y.

Finally, let $V' = X_1 \setminus (Y \cup Z)$ and let $\mathcal{G}'$ be the subgame of $\mathcal{G}$ induced by V', with winning condition $(\mathcal{F}_0 \cap \mathcal{P}(C'), \mathcal{F}_1 \cap \mathcal{P}(C'))$. Since this game has fewer priorities, the induction hypothesis applies, i.e. $V' = W_0' \cup W_1'$, Player 0 has a winning strategy with memory $\leq \ell$ on W_0' and Player 1 has a positional winning strategy g' on W_1'. However, $W_0' = \emptyset$; otherwise we could combine the strategies of Player 0 to obtain a winning strategy with memory $\leq \ell$ on $X_0 \cup W_0' \supsetneq X_0$ contradicting the definition of X_0. Hence $W_1' = V'$.

We can now define a positional strategy g for Player 1 on X_1 by

$$g(x) = \begin{cases} g'(x) & \text{if } x \in V' \\ a(x) & \text{if } x \in Z \\ t(x) & \text{if } x \in Y \end{cases}$$

Consider any play π that starts at a position $v \in X_1$ and is consistent with g. Obviously π stays within X_1. If it hits $Y \cup Z$ only finitely often, then from some point onward, it stays within V_1 and coincides with a play consistent with g'. It is therefore won by Player 1. Otherwise π hits $Y \cup Z$, and hence also Y, infinitely often. Thus, $\mathrm{Inf}(\pi) \cap (C \setminus C') \neq \emptyset$ and therefore $\mathrm{Inf}(\pi) \in \mathcal{F}_1$.

We now consider the second case, $C \in \mathcal{F}_0$. There exist maximal subsets $C_0, \ldots, C_{k-1} \subseteq C$ with $C_i \in \mathcal{F}_1$. Observe that for every set $D \subseteq C$, we have that if $D \cap (C \setminus C_i) \neq \emptyset$ for all $i < k$, then $D \in \mathcal{F}_0$. Let

$$X_1 := \{v : \text{Player 1 has a positional winning strategy from } v\},$$

and $X_0 = V \setminus X_1$. We claim that Player 0 has a finite memory winning strategy of size $\leq \ell$ on X_0. To construct this strategy, we proceed in a similar way as above, for each of the sets $C \setminus C_i$. We will obtain strategies $f_0, \ldots, f_{k-1}$ for Player 0, such that f_i has finite memory M_i, and we will use these strategies to build a winning strategy f on X_0 with memory $M_0 \cup \cdots \cup M_{k-1}$.

For $i = 0, \ldots, k-1$, let $Y_i = X_0 \cap \Omega^{-1}(C \setminus C_i)$ let $Z_i = \mathrm{Attr}_0^{X_0}(Y_i) \setminus Y_i$, and let a_i be a positional attractor strategy, by which Player 0 can force a play from any position in Z_i to Y_i. Further, let $U_i = X_0 \setminus (Y_i \cup Z_i)$ and let $\mathcal{G}_i$ be the subgame of $\mathcal{G}$ induced by U_i, with winning condition $(\mathcal{F}_0 \cap \mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$. The winning region of Player 1 in $\mathcal{G}_i$ is empty; indeed, if Player 1 could win $\mathcal{G}_i$ from v, then, by induction hypothesis, he could win with a positional winning strategy. By combining this strategy with the positional winning strategy of Player 1 on X_1, this would imply that $v \in X_1$; but $v \in U_i \subseteq V \setminus X_1$.

Hence, by induction hypothesis, Player 0 has a winning strategy f_i with finite memory M_i on U_i. Let $(f_i + a_i)$ be the combination of f_i with the attractor strategy a_i. From any position $v \in U_i \cup Z_i$ this strategy ensures that the play either remains inside U_i and is winning for Player 1, or it eventually reaches a position in Y_i.

We now combine the finite-memory strategies $(f_0 + a_0), \dots, (f_{k-1} + a_{k-1})$ to a winning strategy f on X_0, which ensures that either the play ultimately remains within one of the regions U_i and coincides with a play according to f_i, or that it cycles infinitely often through all the regions $Y_0, \dots, Y_{k-1}$.

At positions in $\bigcap_{i<k} Y_i$, Player 0 just plays with a (positional) trap strategy ensuring that the play remains in X_0. At the first position $v \notin \bigcap_{i<k} Y_i$, Player 0 takes the minimal i such that $v \notin Y_i$, i.e. $v \in U_i \cup Z_i$, and uses the strategy $(f_i + a_i)$ until a position in $w \in Y_i$ is reached. At this point, Player 0 switches from i to $j = i + \ell \pmod k$ for the minimal ℓ such that $w \notin Y_j$. Hence $w \in U_j \cup Z_j$; Player 0 now plays with strategy $(f_j + a_j)$ until a position in Y_j is reached. There Player 0 again switches to the appropriate next strategy, and so on.

Assuming that $M_i \cap M_j = \emptyset$ for $i \neq j$ it is not difficult to see that f can be implemented with memory $M = M_0 \cup \dots \cup M_{k-1}$. We leave a formal definition of f to the reader.

It remains to prove that f is winning on X_0. Let π be a play that starts in X_0 and is consistent with f. If π eventually remains inside some U_i then it coincides, from some point onwards, with a play that is consistent with f_i, and therefore won by Player 0. Otherwise it hits each of the sets $Y_0, \dots, Y_{k-1}$ infinitely often. But this means that $\mathrm{Inf}(\pi) \cap (C \setminus C_i) \neq \emptyset$ for all $i \leq k$; as observed above this implies that $\mathrm{Inf}(\pi) \in \mathcal{F}_0$.

Note that, by induction hypothesis, the size of the memory M_i is bounded by the number of leaves of the Zielonka subtrees $Z(\mathcal{F}_0 \cap \mathcal{P}(C_i), \mathcal{F}_1 \cap \mathcal{P}(C_i))$. Consequently the size of M is bounded by the number of leaves of $Z(\mathcal{F}_0, \mathcal{F}_1)$.

Q.E.D.

Of course it also follows from this Theorem that parity games are positionally determined.

3.2 Latest appearance records and reductions for Muller games

The classical example of a game reduction with finite memory is the reduction of Muller games to parity games via latest appearance records. Intuitively, a latest appearance record (LAR) is a list of priorities ordered by their latest occurrence. More formally, for a finite set C of priorities, $\mathrm{LAR}(C)$ is the set of sequences $c_1 \dots c_k \natural c_{k+1} \dots c_\ell$ of elements from $C \cup \{\natural\}$ in which each priority $c \in C$ occurs at most once, and $\natural$ occurs precisely once. At a position v, the LAR $c_1 \dots c_k \natural c_{k+1} \dots c_\ell$ is updated by moving the priority $\Omega(v)$ to the end, and moving $\natural$ to the previous position of $\Omega(v)$ in the sequence. For instance, at a position with priority c_2, the LAR $c_1 c_2 c_3 \natural c_4 c_5$ is updated to $c_1 \natural c_3 c_4 c_5 c_2$.

(If $\Omega(v)$ did not occur in the LAR, we simply append $\Omega(v)$ at the end). Thus, the LAR-memory for an arena with priority labeling $\Omega : V \to C$ is the triple $(LAR(C), \mathrm{init}, \mathrm{update})$ with $\mathrm{init}(v) = \natural\Omega(v)$ and

$$\mathrm{update}(c_1 \dots c_k \natural c_{k+1} \dots c_\ell, v) = c_1 \dots c_k \natural c_{k+1} \dots c_\ell \Omega(v)$$

in case $\Omega(v) \notin \{c_1 \dots c_\ell\}$, and

$$\mathrm{update}(c_1 \dots c_k \natural c_{k+1} \dots c_\ell, v) = c_1 \dots c_{m-1} \natural c_{m+1} \dots c_\ell c_m$$

if $\Omega(v) = c_m$.

The *hit-set* of a LAR $c_1 \dots c_k \natural c_{k+1} \dots c_\ell$ is the set $\{c_{k+1} \dots c_\ell\}$ of priorities occuring after the symbol $\natural$. Note that if in a play $\pi = v_0 v_1 \dots$, the LAR at position v_n is $c_1 \dots c_k \natural c_{k+1} \dots c_\ell$ then $\Omega(v_n) = c_\ell$ and the hit-set $\{c_{k+1} \dots c_\ell\}$ is the set of priorities that have been seen since the latest previous occurrence of c_ℓ in the play.

Lemma 3.5. Let π be a play of a Muller game $\mathcal{G}$, and let $\mathrm{Inf}(\pi)$ be the set of priorities occurring infinitely often in π. On π the hit-set of the latest appearance record is, from some point onwards, always a subset of $\mathrm{Inf}(\pi)$ and infinitely often coincides with $\mathrm{Inf}(\pi)$.

Proof. For each play $\pi = v_0 v_1 v_2 \dots$ there is a position v_m such that $\Omega(v_n) \in \mathrm{Inf}(\pi)$ for all $n \geq m$. Since no priority outside $\mathrm{Inf}(\pi)$ is seen anymore after position v_m, the hit-set will from that point onwards always be contained in $\mathrm{Inf}(\pi)$, and the LAR will always have the form $c_1 \dots c_{j-1} c_j \dots c_k \natural c_{k+1} \dots c_\ell$ where $c_1, \dots c_{j-1}$ remain fixed and $\{c_j, \dots, c_k, c_{k+1}, \dots c_\ell\} = \mathrm{Inf}(\pi)$. Since all priorities in $\mathrm{Inf}(\pi)$ are seen again and again, it happens infinitely often that, among these, the one occuring leftmost in the LAR is hit. At such positions, the LAR is updated to $c_1, \dots, c_{j-1} \natural c_{j+1} \dots c_\ell c_j$ and the hit-set then coincides with $\mathrm{Inf}(\pi)$. Q.E.D.

Theorem 3.6. Every Muller game with finitely many priorities reduces via LAR memory to a parity game.

Proof. Let $\mathcal{G}$ be a Muller game with game graph G, priority labelling $\Omega : V \to C$ and winning condition $(\mathcal{F}_0, \mathcal{F}_1)$. We have to prove that $\mathcal{G} \leq_{\mathrm{LAR}} \mathcal{G}'$ for a parity game $\mathcal{G}'$ with game graph $G \times \mathrm{LAR}(C)$ and an appropriate priority labeling Ω' on $V \times \mathrm{LAR}(C)$ which is defined as follows.

$$\Omega'(v, c_1 c_2 \dots c_k \natural c_{k+1} \dots c_\ell) = \begin{cases} 2k & \text{if } \{c_{k+1}, \dots, c_\ell\} \in \mathcal{F}_0, \\ 2k+1 & \text{if } \{c_{k+1}, \dots, c_\ell\} \in \mathcal{F}_1. \end{cases}$$

Let $\pi = v_0v_1v_2\dots$ be a play on $\mathcal{G}$ and fix a number m such that, for all numbers $n \geq m$ and $\Omega(v_n) \in \mathrm{Inf}(\pi)$, the LAR at position v_n has the form $c_1 \dots c_j c_{j+1} \dots c_k \natural c_{k+1} \dots c_\ell$ where $\mathrm{Inf}(\pi) = \{c_{j+1}, \dots c_\ell\}$ and the prefix $c_1 \dots c_j$ remains fixed. In the extended play $\pi' = (v_0r_0)(v_1, r_1)\dots$ all nodes (v_n, r_n) for $n \geq$ will therefore have a priority $2k+\rho$ with $k \geq j$ and $\rho \in \{0, 1\}$. Assume that the play π is won by Player σ, i.e., $\mathrm{Inf}(\pi) \in \mathcal{F}_\sigma$. Since infinitely often the hit-set of the LAR coincides with $\mathrm{Inf}(\pi)$, the minimal priority seen infinitely often on the extended play is $2j + \sigma$. Thus the extended play in the parity game $\mathcal{G}'$ is won by the same player as the original play in the Muller game $\mathcal{G}$. Q.E.D.

4 Games with infinitely many priorities

The definition of Muller games (Definition 3.1) directly generalises to countable sets C of priorities[1]. However, a representation of a Muller condition by a Zielonka tree is not always possible, since we may have sets $D \in \mathcal{F}_\sigma$ that have subsets in $\mathcal{F}_{1-\sigma}$ but no maximal ones. Further, it turns out that the condition that $\mathcal{F}_0$ and $\mathcal{F}_1$ are both closed under finite unions is no longer sufficient for positional determinacy. To see this let us discuss the possible generalisations of parity games to the case of priority assigments $\Omega : V \to \omega$. For parity games with finitely many priorities it is of course purely a matter of taste whether we let the winner be determined by the least priority seen infinitely often or by the greatest one. Here this is no longer the case. Based on priority assignments $\Omega : V \to \omega$ we consider the following classes of games.

Infinity games are games where Player 0 wins those infinite plays in which no priority at all appears infinitely often, i.e.

$$\mathcal{F}_0 = \{\emptyset\}$$
$$\mathcal{F}_1 = \mathcal{P}(\omega) \setminus \{\emptyset\}$$

Parity games are games where Player 0 wins the plays in which the least priority seen infinitely often is even, or where no priority appears infinitely often. Thus,

$$\mathcal{F}_0 = \{X \subseteq \omega : \min(X) \text{ is even}\} \cup \{\emptyset\}$$
$$\mathcal{F}_1 = \{X \subseteq \omega : \min(X) \text{ is odd}\}$$

[1]With minor modifications, it can also be generalised to uncountable sets C. See [11] for a discussion of this.

Max-parity games are games where Player 0 wins if the maximal priority occurring infinitely often is even, or does not exist, i.e.

$$\mathcal{F}_0 = \{X \subseteq \omega : \text{ if } X \text{ is finite and non-empty, then } \max(X) \text{ is even}\}$$
$$\mathcal{F}_1 = \{X \subseteq \omega : X \text{ is finite, non-empty, and } \max(X) \text{ is odd}\}$$

It is easy to see that infinity games are a special case of parity games (via a simple reassignment of priorities). Further we note that for both parity games and max-parity games, $\mathcal{F}_0$ and $\mathcal{F}_1$ are closed under finite unions. Nevertheless the conditions behave quite differently. The parity condition has a very simple Zielonka tree, namely just a Zielonka path

$$\omega \longrightarrow \omega \setminus \{0\} \longrightarrow \omega \setminus \{0,1\} \longrightarrow \omega \setminus \{0,1,2\} \longrightarrow \cdots$$

whereas there is no Zielonka tree for the max-parity condition since $\omega \in \mathcal{F}_0$ has no maximal subset in $\mathcal{F}_1$ (and $\mathcal{F}_1$ is not closed under unions of chains). This is in fact related to a much more important difference concerning the memory needed for winning strategies.

Proposition 4.1. Max-parity games with infinitely many priorities in general do not admit finite memory winning strategies.

Proof. Consider the max-parity game with positions $V_0 = \{0\}$ and $V_1 = \{2n+1 : n \in \mathbb{N}\}$ (where the name of a position is also its priority), such that Player 0 can move from 0 to any position $2n+1$ and Player 1 can move back from $2n+1$ to 0. Clearly Player 0 has a winning strategy from each position but no winning stategy with finite memory. Q.E.D.

On the other hand it has been shown in [11] that infinity games and parity games with priorities in ω do admit positional winning strategies for both players on all game graphs. In fact, parity games over ω turn out to be the only Muller games with this property.

Theorem 4.2 (Grädel, Walukiewicz)**.** Let $(\mathcal{F}_0, \mathcal{F}_1)$ be a Muller winning condition over a countable set C of priorities. Then the following are equivalent.

(i) Every game with winning condition $(\mathcal{F}_0, \mathcal{F}_1)$ is positionally determined.

(ii) Both $\mathcal{F}_0$ and $\mathcal{F}_1$ are closed under finite unions, unions of chains, and non-empty intersections of chains.

(iii) The Zielonka tree of $(\mathcal{F}_0, \mathcal{F}_1)$ exists, and is a path of co-finite sets (and possibly the empty set at the end).

(iv) $(\mathcal{F}_0, \mathcal{F}_1)$ reduces to a parity condition over $n \leq \omega$ priorities.

5 Finite Appearance Records

Although over an infinite set of priorities one can easily define Muller games that do not admit finite memory strategies, these games are often solvable by strategies with very simple infinite memory structures. For instance, for the max-parity game described in the proof of Proposition 4.1, it suffices for Player 0 to store the maximal priority seen so far, in order to determine the next move in her winning strategy. One can readily come up with other games where the memory required by a winning strategies is essentially a finite collection of previously seen priorities.

This motivates the definition of an infinite memory structure that we call *finite appearance records* (FAR) which generalises the LAR-memory for finitely coloured games. In a FAR we store tuples of previously encountered priorities or some other symbols from a finite set. Additionally the update function in the appearance record is restricted, so that new values of the memory can be equal only to the values stored before or to the currently seen priority.

Definition 5.1. A d-dimensional FAR-memory for a game $\mathcal{G}$ with priorities in C is a memory structure $(M, \text{update}, \text{init})$ for $\mathcal{G}$ with $M = (C \cup N)^d$ for some finite set N such that whenever

$$\text{update}(m_1, \ldots, m_d, v) = (m'_1, \ldots, m'_d)$$

then $m'_i \in \{m_1, \ldots, m_d\} \cup N \cup \{\Omega(v)\}$.

Note that an LAR-memory over a finite set C is a special case of an FAR-memory, with $d = |C| + 1$ and $N = \{\natural, B\}$, where B is a blank symbol used to pad latest appearance records in which some priorities are missing. Here the dimension of the FAR depends on the size of C. Hence, the question arises whether there is a fixed dimension d and a fixed additional set N such that every finitely coloured Muller game reduces to a parity game via d-dimensional FAR-memory. From Theorem 1.1 it follows that his is not the case. Indeed, since $n!$ grows faster than n^d for any constant d, we infer that for any dimension d there is a Muller game $\mathcal{G}_d$ that can not be reduced to a parity game via d-dimensional FAR-memory.

From this we obtain the following conclusion.

Proposition 5.2. There exists an infinitely coloured Muller game $\mathcal{G}$ that does not reduce to a parity game with any FAR-memory.

Proof. Take $\mathcal{G}$ to be the disjoint sum of the games $\mathcal{G}_d$, assuming that all these games have disjoint sets of priorities. Suppose that $\mathcal{G}$ reduces to a parity game via some FAR-memory of dimension d. Since game extensions preserve connectivity it follows that the extension of the connected component $\mathcal{G}_d$ of $\mathcal{G}$ will also be a parity game. But this contradicts the fact that $\mathcal{G}_d$ does not reduce to a parity game via d-dimensional FAR-memory. Q.E.D.

6 FAR-reductions for Muller games

In this section we consider some cases of Muller games with priorities in ω that admit FAR-reductions to positionally determined games.

To illustrate the idea consider any *downwards cone* $\mathcal{F}_0 = \{X : X \subseteq A\}$ for a fixed set $A \subseteq \omega$. Again it is easy to see that such games may require infinite-memory strategies. To reduce such a game to a parity game $\mathcal{G}'$ it suffices to store the maximal priority m seen so far, and to define priorities in $\mathcal{G}'$ by

$$\Omega'(v, m) = \begin{cases} 2m + 2 & \text{if } \Omega(v) \in A \\ 2\Omega(v) + 1 & \text{otherwise.} \end{cases}$$

If $\mathrm{Inf}(\pi) \subseteq A$ then Player 0 wins π' since no odd priority is seen infinitely often in π'. If there is some $a \in \mathrm{Inf}(\pi) \setminus A$, then $2a + 1$ occurs infinitely often in π', and since $a \leq m$ from some point onwards, no smaller even priority can have this property, so Player 1 wins π'.

Hence any Muller game such that $\mathcal{F}_0$ (or $\mathcal{F}_1$) is a downwards cone is determined via one-dimensional FAR-memory.

6.1 Visiting sequences and singleton Muller conditions

Our next example for winning conditions that are amenable for an approach via FAR-reductions are Muller games where the winning condition of Player 0 is a singleton, i.e., $\mathcal{F}_0 = \{A\}$, $\mathcal{F}_1 = \mathcal{P}(\omega) \setminus \{A\}$.

We first observe that such games may require infinite memory.

Theorem 6.1. For any $A \neq \emptyset$, there exists a (solitaire) Muller game with $\mathcal{F}_0 = \{A\}$ whose winning strategies all require infinite memory.

Proof. If $A = \{a_1, a_2, \dots\}$ is infinite, take the game with set of positions $V = V_0 = A$ (where the name of a positions indicates also its priority), and moves (a_1, a_n) and (a_n, a_1) for all $n \geq 2$. If $A = \{a_1, \dots, a_n\}$ is finite, let $\omega \setminus A = \{b_1, b_2, \dots\}$ we consider instead the game with $V = V_0 = A \cup (\omega \setminus A)$, and set of moves

$$E = \{(a_i, a_{i+1}) : 1 \leq i < n\} \cup \{(a_n, b) : b \in (\omega \setminus A)\} \cup \{(b, a_1) : b \in (\omega \setminus A)\}.$$

In both cases, Player 0 wins, but requires infinite memory to do so. Q.E.D.

We will prove that singleton Muller games can be reduced via FAR-memory to parity games with priorities in ω which, as shown in [11], are positionally determined. The FAR-memory that we use for this reduction is based on a particular order in which the elements of the winning sets have to be seen infinitely often, which is specified by a visiting sequence.

Definition 6.2. Let $A = \{a_1 < a_2 < \dots\}$ be an infinite subset of ω. For each $n \in \omega$, let $p(a_n) := a_1 a_2 \dots a_n$ be the *prefix* of a_n. The *visiting sequence* of A is the concatenation of the prefixes of all elements of A

$$\mathrm{visit}(A) = p(a_1)p(a_2)p(a_3)\dots$$

For a finite set $\{a_1 < a_2 < \dots < a_n\} \subseteq \omega$ we define $\mathrm{visit}(A) = p(a_n)^\omega$.

Let $\mathcal{G}$ be a Muller game over ω.

Lemma 6.3. For any play $\pi = v_1 v_2 \dots$ of $\mathcal{G}$ the set $\mathrm{Inf}(\pi)$ is the unique set A with the following two properties:

(1) There exists a sequence of indices $i_1 < i_2 < \dots$ such that $\Omega(v_{i_1})\Omega(v_{i_2})\dots$ forms the visiting sequence of A.

(2) If $\Omega(v_k) \in \omega \setminus A$ then there is only a finite number of indices $i > k$ such that $\Omega(v_i) \in \{0, \dots, \Omega(v_k)\} \cap \omega \setminus A$.

Proof. First we notice that $A = \mathrm{Inf}(\pi)$ indeed fulfils these two properties. The visiting sequence can be chosen from the play as all elements of $\mathrm{Inf}(\pi)$ appear infinitely often. Since all elements of $\omega \setminus \mathrm{Inf}(\pi)$ occur only finitely often in the play, the second property must also hold.

Conversely, if a set A satisfies property (1), then all elements of A appear infinitely often in π, so $A \subseteq \mathrm{Inf}(\pi)$. If there were an element $a \in \mathrm{Inf}(\pi) \setminus A$, then for any k with $\Omega(v_k) = a$, there were infinitely many indices $i > k$, with $\Omega(v_i) = a$ which contradicts property (2). Thus if A satisfies properties (1) and (2), then $A = \mathrm{Inf}(\pi)$. Q.E.D.

Let $A \subseteq \omega$ be infinite. Any initial segment of the visiting sequence of A can be written in the form $p(a_1)p(a_2)\dots p(a_i)a_1 a_2 \dots a_j$ where $1 \le j \le i+1$. It can be represented by a pair (p, c) where $c = a_j$ indicates the position of the last letter in the current prefix $p(a_{i+1})$, and $p = a_i$ indicates the last previously compeleted prefix (or ε if we are at the first element). For instance, the initial segment $a_1\, a_1a_2\, a_1a_2a_3\, a_1a_2a_3$ of the visiting sequence of A is encoded by (a_3, a_3), the initial segment a_1 is encoded by (ε, a_1), and the empty initial segment by $(\varepsilon, \varepsilon)$. We write $\mathrm{visit}_n(A)$ for the initial segment of length A of $\mathrm{visit}(A)$.

Given a (finite or infinite) winning set A, we want to use a three-dimensional FAR-memory to check whether $\mathrm{Inf}(\pi) = A$. For infinite A, the memory state after an initial segment of a play is a triple (p, c, q) where (p, c) encode the initial segment of the visiting sequence of A that has been seen so far, and q is the maximal priority that has occurred.

Definition 6.4. For any infinite set $A \subseteq \omega$, we define a three-dimensional FAR-memory $\mathrm{FAR}(A) = (M, \mathrm{init}, \mathrm{update})$ with $M = \{(p, c, q) : p, c \in \omega \cup \{\varepsilon\}, q \in \omega\}$. The initialisation function is defined by

$$\mathrm{init}(v) = \begin{cases} (\varepsilon, \Omega(v), \Omega(v)) & \text{if } \Omega(v) = a_1 \\ (\varepsilon, \varepsilon, \Omega(v)) & \text{if } \Omega(v) \neq a_1 \end{cases}$$

The update function is defined by

$$\mathrm{update}(p, c, q, v) := (p', c', q'),$$

where $q' = \max(q, \Omega(v))$, and where either (p, c) and respectively (p', c') encode, for some n, the initial segments $\mathrm{visit}_n(A)$ and $\mathrm{visit}_{n+1}(A)$ of the visiting sequence of A such that $\mathrm{visit}_{n+1}(A) = \mathrm{visit}_n(A)\Omega(v)$, or otherwise, $(p', c') = (p, c)$.

For a more formal description, let

$$\mathrm{up}(p, c, v) = \begin{cases} 2 & \text{if, for some } i,\ p = a_i, c = a_{i+1}, \Omega(v) = a_1 \\ 1 & \text{if, for some } j \leq i,\ p = a_i, c = a_j, \Omega(v) = a_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

(where, to simplify notation, we identify ε with a_0). Note that $\mathrm{up}(p, c, v) = 2$ if, at node v, the visiting sequence is updated with an a_1 (i.e. a prefix $p(a_i)$ has been completed and a new one is started), that $\mathrm{up}(p, c, v) = 1$ if the visiting sequence is updated by another value, and that $\mathrm{up}(p, c, v) = 0$ if no update of the visiting sequence happens at v. Then we can define $\mathrm{update}(p, c, q, v) :=$

(p', c', q') by

$$\begin{aligned} (p', c') &= \begin{cases} (c, \Omega(v)) & \text{if } \mathrm{up}(p, c, v) = 2 \\ (p, \Omega(v)) & \text{if } \mathrm{up}(p, c, v) = 1 \\ (p, c) & \text{if } \mathrm{up}(p, c, v) = 0 \end{cases} \\ q' &= \max(q, \Omega(v)) \end{aligned}$$

For finite $A = \{a_1 < a_2 < \cdots < a_n\}$ this has to be modified since once cannot really encode the part of the visiting sequence that one has seen with priorities in A. In this case the value (p, c, q) is so that c is the last element of the visiting sequence, q is the maximal priority that has occurred so far, and p is the maximal priority that had occured up to the last time when, in the visiting sequence of A, a prefix $p(a_n)$ had been completed and c had been updated from a_n to a_1. Thus we set

$$\mathrm{up}(p, c, v) = \begin{cases} 2 & \text{if } c = a_n, \Omega(v) = a_1 \\ 1 & \text{if, for some } i < n, c = a_i, \Omega(v) = a_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

and $\mathrm{update}(p, c, q, v) := (p', c', q')$ with

$$\begin{aligned} (p', c') &= \begin{cases} (q, \Omega(v)) & \text{if } \mathrm{up}(p, c, v) = 2 \\ (p, \Omega(v)) & \text{if } \mathrm{up}(p, c, v) = 1 \\ (p, c) & \text{if } \mathrm{up}(p, c, v) = 0 \end{cases} \\ q' &= \max(q, \Omega(v)). \end{aligned}$$

Theorem 6.5. Any singleton Muller game with $\mathcal{F}_0 = \{A\}$ can be reduced, via memory $\mathrm{FAR}(A)$, to a parity game.

Proof. The given Muller game $\mathcal{G}$ with arena (G, Ω) and Muller condition such that $\mathcal{F}_0 = \{A\}$ is reduced via memory $\mathrm{FAR}(A)$ to a parity game $\mathcal{G}'$ with priority function $\Omega' : V \times \mathrm{FAR}(A) \to \omega$ defined as follows.

$$\Omega'(v, p, c, q) = \begin{cases} 2p + 2 & \text{if } \Omega(v) \in A, \mathrm{up}(p, c, v) \in \{1, 2\} \\ 2p + 3 & \text{if } \Omega(v) \in A, \mathrm{up}(p, c, v) = 0 \\ \min(2p + 3, 2\Omega(v) + 1) & \text{if } \Omega(v) \notin A \end{cases}$$

We have to prove that any play $\pi = v_0 v_1 v_2 \ldots$ of $\mathcal{G}$ is won by the same player as the extended play $\pi' = (v_0, p_0, c_0, q_0)(v_1, p_1, c_1, q_1)(v_2, p_2, c_2, q_2) \ldots$ of $\mathcal{G}'$.

We first assume that $\mathrm{Inf}(\pi) = A$ and prove that either no priority at all occurs infinitely often in π' or the minimal such is even. If A is infinite, then the sequence of the values p_n diverges and therefore no priority will be seen infinitely often in π'. If A is finite then it may be the case that the sequence $(p_n)_{n\in\omega}$ converges, i.e., $p_n = p$ from some point onwards. But since the visiting sequence will be updated again and again this means that infinitely often the priority $2p + 2$ occurs in π', and the only other priority that may occur infinitely often is $2p + 3$. Hence Player 0 wins π'.

For the converse, we assume that Player 1 wins π. We distinguish several cases. If there exist some $a \in A \setminus \mathrm{Inf}(\pi)$ then from some point onwards, the visiting sequence cannot be updated anymore, so the sequence $(p_n)_{n\in\omega}$ stabilises at some value p. Then the minimal priority seen infinitely often is either $2p + 3$, or $2\Omega(v) + 1$ for some $\Omega(v) \in \omega \setminus A$ and Player 1 also wins π'. If no such element a exists, then $A \subsetneq \mathrm{Inf}(\pi)$ and there is a minimal element $b \in \mathrm{Inf}(\pi) \setminus A$. If the sequence $(p_n)_{n\in\omega}$ diverges (which is always the case for infinite winning sets A) then the minimal priority seen infinitely often in π' is $2b + 1$. If A is finite then the sequence p_n may stabilise at some value p which coincides with the largest priority ever occurring in π. Hence $b \leq p$ and therefore $2b+1 < 2p+2$, so the minimal priority seen infinitely often in π' is $2b + 1$. Again Player 1 wins the associated play in the parity game. Q.E.D.

Corollary 6.6. Singleton Muller games are determined with FAR memory.

6.2 Finite unions of upwards cones

Visiting sequences can also be used for the case where $\mathcal{F}_0$ is a finite union of upwards cones, i.e.

$$\mathcal{F}_0 = \bigcup_{i=1}^{k} \{X : A_i \subseteq X \subseteq \omega\}$$

for some finite collection of sets $A_1, \ldots, A_k$.

The FAR-memory stores the pairs (p_i, c_i) encoding the visiting sequences of $A_1, \ldots, A_k$. All that has to checked is whether $A_i \subseteq \mathrm{Inf}(\pi)$ for some i, which is the case if, and only if, one of the visiting sequences is updated infinitely often. Thus we can define priorities by

$$\Omega'(v, p_1, c_1, \ldots, p_k, c_k) = \begin{cases} 0 & \text{if } \mathrm{up}(p_i, c_i, v) = 2 \text{ for some } i \\ 1 & \text{otherwise.} \end{cases}$$

Theorem 6.7. Any Muller game such that $\mathcal{F}_\sigma$ is a finite union of upwards cones is determined via FAR-memory.

6.3 Muller conditions with finitely many winning sets

We now consider the case of Muller games whose winning conditions are defined by a finite collection of (possibly infinite) sets, $\mathcal{F}_0 = \{A_1, \ldots, A_k\}$. To extend the idea presented above to this case we are going to use the memory $\mathrm{FAR}(A_i)$ for each set A_i and additionally we have to remember when the set A_i is *active*, as is described below. The property of being active is stored in a value $a_i \in \{0, 1, 2\}$.

Definition 6.8. For any finite collection $\{A_1, \ldots, A_k\}$ of sets $A_i \subseteq \omega$, we define a $4k$-dimensional FAR-memory $\mathrm{FAR}(A_1, \ldots, A_k) = (M, \mathrm{init}, \mathrm{update})$. We denote the FAR-memory of A_i by $\mathrm{FAR}(A_i) = (M_i, \mathrm{init}_i, \mathrm{update}_i)$. Then $M = M_1 \times M_2 \times \ldots \times M_k \times \{0, 1, 2\}^k$. The initialisation function is defined by

$$\mathrm{init}(v) = (\mathrm{init}_1(v), \ldots, \mathrm{init}_k(v), \overline{0}).$$

The update function is defined by

$$\begin{aligned}&\mathrm{update}(m_1, \ldots, m_k, a_1, \ldots, a_k, v) = \\ &\qquad (\mathrm{update}_1(m_1, v), \ldots, \mathrm{update}_k(m_k, v), a'_1, \ldots, a'_k),\end{aligned}$$

where a'_i is the new activation value for sequence i defined by

$$a'_i = \begin{cases} 0 & \text{if } v \notin A_i \text{ and for some } j \leq k \ \mathrm{up}_j(m_j, v) > 0 \\ \min(2, a_i + 1) & \text{if } \mathrm{up}_i(m_i, v) = 2 \\ a_i & \text{otherwise.} \end{cases}$$

Theorem 6.9. Any Muller game with $\mathcal{F}_0 = \{A_1, \ldots, A_k\}$ can be reduced, via memory $\mathrm{FAR}(A_1, \ldots, A_k)$, to a parity game.

Proof. The given Muller game $\mathcal{G}$ with arena (G, Ω) and Muller condition such that $\mathcal{F}_0 = \{A_1, \ldots, A_k\}$ is reduced to a parity game $\mathcal{G}'$ with priority function Ω' defined as follows.

$$\Omega'(v,\overline{m},\overline{a}) = \begin{cases} \max_{\{i\,:\,\Omega(v)\in A_i \wedge a_i=2\}}(2kp_i+2r_i+2) & \text{if exists } j \text{ such that} \\ & \Omega(v)\in A_j, a_j=2, \\ & \mathrm{up}_j(m_j,v)\in\{1,2\} \\ \max_{\{i\,:\,\Omega(v)\in A_i \wedge a_i=2\}}(2kp_i+2r_i+3) & \text{if exists } j \text{ such that} \\ & \Omega(v)\in A_j, a_j=2, \\ & \text{and } \mathrm{up}_j(m_j,v)=0 \\ & \text{for all such } j \\ \min(2k\max(p_1\dots p_k)+3, 2\Omega(v)+1) & \text{otherwise} \end{cases}$$

where p_i is the first component of the i-th memory $m_i = (p_i, c_i, q_i)$ and for each $A_i \in \mathcal{F}_0$ we have $r_i = |\{A_j \in \mathcal{F}_0 : A_i \subseteq A_j\}|$.

We have to prove that any play $\pi = v_0v_1v_2\dots$ of $\mathcal{G}$ is won by the same player as the extended play

$$\pi' = (v_0, m_{10}, \dots, m_{k0}, a_{10}, \dots a_{k0})(v_1, m_{11}, \dots, m_{k1}, a_{11}, \dots a_{k1})\dots.$$

For a given play π of $\mathcal{G}$, we divide the sets $A_1, \dots, A_k \in \mathcal{F}_0$ into three classes.

The good: A_i is a good set if A_i is active (i.e., $a_i = 2$) only finitely often in π.

The bad: A_i is a bad set, if $A_i \subseteq \mathrm{Inf}(\pi)$ and A_i is not a good set.

The ugly: A_i is an ugly set if there is a priority $c \in A_i \setminus \mathrm{Inf}(\pi)$ and A_i is not a good set.

Lemma 6.10. If A_i is bad and A_j is ugly, then $A_i \subseteq A_j$.

Proof. Assume that there is a $b \in A_i \setminus A_j$. Since $A_i \subseteq \mathrm{Inf}(\pi)$ the visiting sequence for A_i is updated infinitely often, hence infinitely often with b, and whenever this happens then a_j is reset to 0. By definition there is a $c \in A_j$ that is seen only finitely many times in π. Therefore $a_j = 0$ from some point onwards. But this contradicts the assumption that A_j is not good. Q.E.D.

We first assume that $\mathrm{Inf}(\pi) = A_i$ and prove that either no priority at all occurs infinitely often in π' or the minimal such priority is even.

Since from some point on there is no priority $d \notin A_i$ that occurs infinitely often, then for all sets A_j that are not subsets of A_i the visiting sequence will not be updated any more, and so the sequence $(p_{jn})_{n\in\omega}$ stabilises at some value p_j. Since the visiting sequence of A_i is updated infinitely often, we get that from some point on $a_i = 2$. Hence A_i is a bad set. We can now argue as in the proof of Theorem 6.5: if infinitely many priorities appear in π, then the sequence $(p_{in})_{n\in\omega}$ diverges and no priority at all will be seen infinitely often in π'. It remains to consider the case where only finitely many priorities occur in π. Then the sequence $(p_{in})_{n\in\omega}$ stabilises at some value p, which is the maximal priority appearing in π. For any $A_j \subsetneq A_i$, the sequence $(p_{jn})_{n\in\omega}$ will then also stabilise at the same value p, and $r_j > r_i$. It follows that some priority of form $2kp + 2r_\ell + 2$ occurs infinitely often in π', where $r_\ell \geq r_i$.

Suppose now that some smaller odd priority occurs infinitely often in π'. Then it would have to be of the form $2kp+2r_j+3$ with $r_j < r_\ell$ such that $a_j = 2$ infinitely often. However, only finitely many priorities appear in π. Hence if there are infinitely many positions v such that $\Omega(v) \in A_j$ and $a_j = 2$, then from some point onwards all these positions v satisfy that $\Omega(v) \in A_j \cap A_i$ and $a_i = 2$. On infinitely many such positions an update happens, and therefore, also the priority $2kp+2r_j+2$ appears infinitely often. Hence Player 0 wins π'.

For the converse, we now assume that Player 1 wins π.

Lemma 6.11. Suppose that some even priority $2kq+2r+2$ is seen infinitely often in π'. Then q is the maximal priority that occurs in π and $r = r_\ell$ for some bad set A_ℓ.

Proof. If there are infinitely many occurrences of $2kq + 2r + 2$ in π', then q is the maximal priority that occurs in π and some A_i is updated infinitely often (i.e. $A_i \subseteq \mathrm{Inf}(\pi)$) and active infinitely often. Obviously A_i is bad and $r \geq r_i$. If $r \neq r_\ell$ for all bad set A_ℓ, then $r = r_j$ for some other A_j that is active infinitely often. Thus A_j has to be ugly. But then by Lemma 6.10 $A_i \subseteq A_j$ and thus $r_i > r_j = r$. But $r \geq r_i$. Q.E.D.

Let $r = \min\{r_\ell : A_\ell \text{ is bad}\}$. To show that Player 1 wins π' it suffices to prove that there is an odd priority occurring infinitely often in π' which, in case there exists a bound q on all priorities appearing in π, is smaller than $2kq + 2r + 2$.

Notice that for any ugly set A_i, the sequence $(p_{in})_{n\in\omega}$ stabilises at some value p_i. Let $p = \max\{p_i : A_i \text{ is ugly}\}$.

We distinguish two cases. First we assume that there exists some priority

$$b \in \mathrm{Inf}(\pi) \setminus \bigcup\{A_i : A_i \text{ is bad}\}.$$

Fix n_0 such that, for all $n > n_0$, $p_{in} = p_i$ for all ugly sets A_i and $a_{in} \neq 2$ for all good sets A_i. Since $b \in \mathrm{Inf}(\pi)$ there exist infinitely many v_n with $n > n_0$ and $\Omega(v_n) = b$. For such v_n we have $\Omega'(v_n, \overline{m}_n, \overline{a}_n) = 2kp + 2r_i + 3$ if there is a set A_i (which has to be ugly) such that $b \in A_i$ and $a_i = 2$.

Otherwise $\Omega'(v_n, \overline{m}_n, \overline{a}_n)$ is odd and $\leq 2b + 1$. Since A_i is ugly and A_ℓ is bad it follows that $A_\ell \subseteq A_i$. Thus, $r_i < r$. Further $p \leq q$. It follows that there exists some odd priority $s \leq \max(2kp + 2r_i + 3, 2b + 1) < 2kq + r + 2$ that appears in π' infinitely often.

Now we consider the other case: every $b \in \mathrm{Inf}(\pi)$ is contained in some bad set $A_{i(b)}$. Let $A_1, \ldots, A_\ell$ be the bad sets. Without loss of generality, we assume that A_1 is a maximal bad set, i.e., $A_1 \not\subseteq A_i$ for $i = 2, \ldots, \ell$. Since A_1 is a strict subset of $\mathrm{Inf}(\pi)$, we can fix a priority $d \in \mathrm{Inf}(\pi) \setminus A_1$; since any priority $d \in \mathrm{Inf}(\pi)$ is contained in some bad set, we can assume that $d \in A_2$. Further, by the maximality of A_1, we can fix priorities $e_2, \ldots, e_\ell$ where $e_i \in A_1 \setminus A_i$.

We consider a suffix of π that starts at a position where

- all sequences $(p_{in})_{n \in \omega}$ that stabilise at some value p_i have already reached that value;
- all good sets A_i have become inactive for good (i.e. $a_i \neq 2$),
- in the visiting sequence for A_1 the prefixes $p(e_1), \ldots p(e_\ell)$ have already been completed.

Note that A_1 is updated infinitely often, and between any two consecutive points in this suffix at which $\mathrm{up}_1 = 2$ all priorities $e_2, \ldots, e_\ell$ are seen at least once. Since the priority d appears infinitely often in π and A_2 is updated infinitely often, we are going to see infinitely many points v_{n_0} in the considered suffix of π for which $\Omega(v_{n_0}) = d$ and $a_1 = 0$ (since a_1 is reset with an update of A_2). Since a_1 increases to 2 infinitely often, there are infinitely many tuples $n_0 < n_1 < n_2$ such that $a_1 = i$ at all positions v_n with $n_i \leq n < n_{i+1}$ and $a_1 = 2$ at v_{n_2}.

By definition $\mathrm{up}_1 = 2$ at v_{n_1} and v_{n_2} and there cannot be any updates on priority d between v_{n_1} and v_{n_2}, as then a_1 would be reset to 0. By our choice of the considered suffix of π, there are updates on all $e_2, \ldots, e_\ell$ between v_{n_1} and v_{n_2}. Therefore, for any bad set A_j that contains d, we have that $a_j < 2$ between position v_{n_2} and the first position v_n with $\Omega(v_n) = d$ that comes after v_{n_2}. This is the case because between v_{n_1} and v_{n_2} the value a_j was reset to 0 by the update of the visiting sequence for A_1 by priority e_j, and since then it has not increased by more than 1 since there was no update on priority d.

Let us now consider the new priority at v_n. Since all bad sets A_j containing d are inactive, we have the same situation as in the first case: $\Omega'(v_n, \overline{m}_n, \overline{a}_n) = 2kp + 2r_i + 3$ if there is a set A_i (which has to be ugly) such that $d \in A_i$ and $a_i = 2$. Otherwise $\Omega'(v_n, \overline{m}_n, \overline{a}_n)$ is odd and $\leq 2d + 1$. Since A_i is ugly and A_ℓ is bad it follows that $A_\ell \subseteq A_i$ and thus $r_i < r_\ell = r$. Further $p \leq q$.

There are infinitely many such positions v_n. Thus there must exist some odd priority $s \leq \max(2kp + 2r_i + 3, 2d + 1) < 2kq + r + 2$ that appears in π' infinitely often. Q.E.D.

Of course, the same arguments apply to the case where $\mathcal{F}_1$ is finite.

Corollary 6.12. Let $(\mathcal{F}_0, \mathcal{F}_1)$ be a Muller winning condition such that either $\mathcal{F}_0$ or $\mathcal{F}_1$ is finite. Then every Muller game with this winning condition is determined via FAR memory.

6.4 Max parity games with bounded moves

We say that a an arena (G, Ω) has *bounded moves* if there is a natural number d such that $|\Omega(v) - \Omega(w)| \leq d$ for all edges (v, w) of G.

We have shown in Proposition 4.1 that, in general, winning strategies for max-parity games require infinite memory, but we do not know whether max-parity games are determined via FAR-memory.

For max-parity games with bounded moves, it is still the case that winning strategies may require infinite memory, but now we can prove determinacy via FAR-memory.

Proposition 6.13. There exist max-parity games with bounded moves whose winning strategies require infinite memory.

Proof. Consider a (solitaire) max-parity game with a single node v_0 of priority 0 from which Player 0 has, for every odd number $2n + 1$, the option to go through a cycle C_n consisting of nodes with priorities $2, 4, \ldots, 2n, 2n + 1, 2n, 2n - 2, \ldots, 4, 2$ and back to the node v_0. All these cycles intersect only at v_0. Clearly Player 0 has a winning strategy, namely to go successively through cycles $C_1, C_2, \ldots$ with the result that there is no maximal priority occurring infinitely often. However, if Player 0 moves according to a finite-memory strategy then only finitely many cycles will be visited and there is a maximal n such that the cycle C_n will be visited infinitely often. Thus the maximal priority seen infinitely often will be $2n + 1$ and Player 0 loses. Q.E.D.

Lemma 6.14. Let π be a play of a max-parity game $\mathcal{G}$ with bounded moves such that infinitely many different priorities occur in π. Then $\max(\mathrm{Inf}(\pi))$ does not exist, so π is won by Player 0.

Proof. Assume that moves of $\mathcal{G}$ are bounded by d and $\mathrm{Inf}(\pi) \neq \emptyset$ and let q be any priority occurring infinitely often on π. Since infinitely many different priorities occur on π it must happen infinitely often that from a position with priority q the play eventually reaches a priority larger than $q+d$. Since moves are bounded by d, this means that on the way the play has to go through at least one of the priorities $q+1, \ldots, q+d$. Hence at least one of these priorities also occurs infinitely often, so q cannot be maximal in $\mathrm{Inf}(\pi)$. Q.E.D.

Theorem 6.15. Every max-parity game with bounded moves can be reduced via a one-dimensional FAR-memory to a parity game. Hence max-parity games are determined via strategies with one-dimensional FAR-memory.

Proof. The FAR-memory simply stores the maximal priority m that has been seen so far. To reduce a max-parity game $\mathcal{G}$ with bounded moves, via this memory, to a parity game $\mathcal{G}'$ we define the priorities of $\mathcal{G}'$ by

$$\Omega'(v, m) = 2m - \Omega(v).$$

Let π be a play of $\mathcal{G}$ and let π' be the extended play in $\mathcal{G}'$. We distinguish two cases. First, we assume that on π the sequence of values for m is unbounded. This means that infinitely many different priorities occur on π, so by Lemma 6.14, Player 0 wins π. But since $m \leq \Omega'(v, m)$ and m never stabilises there is no priority that occurs infinitely often on π, so π' is also won by Player 0.

In the second case there exists a suffix of π on which m remains fixed on the maximal priority of π. In that case $\mathrm{Inf}(\pi)$ is a non-empty subset of $\{0, \ldots, m\}$ and $\mathrm{Inf}(\pi')$ is a non-empty subset of $\{m, \ldots, 2m\}$. Further, $\Omega'(v, m)$ is even if, and only if $\Omega(v)$ is even, and $\Omega'(v_1, m) < \Omega'(v_2, m)$ if, and only if, $\Omega(v_1) > \Omega(v_2)$. Thus, $\min(\mathrm{Inf}(\pi'))$ is even if, and only if, $\max(\mathrm{Inf}(\pi))$ is even. Hence π is won by the same Player as π'. Q.E.D.

7 Conclusion

We have introduced a new memory structure for strategies in infinite games, called FAR-memory, which is approproiate for games with infinitely many priorites and which generalises the LAR-memory for finitary Muller games. We have shown that there are a number of infinitary Muller winning conditions with the following two properties.

(1) There exist Muller games with these winning conditions all whose winning strategies require infinite memory.

(2) All Muller games with such winning conditions can be reduced via FAR-memory to parity games. Therefore all these games are determined via FAR-memory.

The class of these Muller conditions includes:

- Downward cones,
- Singleton conditions,
- Finite unions of upwards cones,
- Winning conditions with finitely many winning sets.

Further we have shown that the same property holds for max-parity games with bounded moves. It is open whether arbitrary max-parity games are determined via FAR-memory. It would also be desirable to obtain a complete classification of the infinitary Muller conditions with this property.

References

[1] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer, *Dag-width and parity games*, in STACS 2006, Proceedings of the 23rd Symposium on Theoretical Aspects of Computer Science, vol. 3884 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 424–436.

[2] D. Berwanger and E. Grädel, *Fixed-point logics and solitaire games*, Theory of Computing Systems, 37 (2004), pp. 675–694.

[3] D. Berwanger and E. Grädel, *Entanglement - A measure for the complexity of directed graphs with applications to logic and games*, in Proceedings of LPAR 2004, Montevideo, Uruguay, vol. 3452 of LNCS, Springer-Verlag, 2005, pp. 209–223.

[4] A. Bouquet, O. Serre, and I. Walukiewicz, *Pushdown games with unboundedness and regular conditions*, in Proceedings of FSTTCS'03, vol. 2914 of Lexture Notes in Computer Science, 2003, pp. 88–99.

[5] T. Cachat, J. Duparc, and W. Thomas, *Solving pushdown games with a Σ_3 winning cndition*, in Computer Science Logic, CSL 2002, Lecture Notes in Computer Science Nr. 2471, Springer-Verlag, 2002, pp. 322–336.

[6] S. Dziembowski, M. Jurdziński, and I. Walukiewicz, *How much memory is needed to win infinite games?*, in Proceedings of 12th Annual IEEE Symposium on Logic in Computer Science (LICS 97), 1997, pp. 99–110.

[7] A. Emerson and C. Jutla, *Tree automata, mu-calculus and determinacy*, in Proc. 32nd IEEE Symp. on Foundations of Computer Science, 1991, pp. 368–377.

[8] A. Emerson, C. Jutla, and P. Sistla, *On model checking for the μ-calculus and its fragments*, Theoretical Computer Science, 258 (2001), pp. 491–522.

[9] H. Gimbert, *Parity and exploration games on infinite graphs*, in Proceedings of CSL 2004, vol. 3210 of Lecture Notes in Computer Science, Springer, 2004, pp. 56–70.

[10] E. Grädel, *Finite model theory and descriptive complexity*, in Finite Model Theory and Its Applications, Springer-Verlag, 2007, pp. 125–230.

[11] E. Grädel and I. Walukiewicz, *Positional determinacy of games with infinitely many priorities*, Logical Methods in Computer Science, vol. 2 (4:6), 2006.

[12] Y. Gurevich and L. Harrington, *Trees, automata and games*, in Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC ’82, 1982, pp. 60–65.

[13] M. Jurdziński, *Deciding the winner in parity games is in UP $\cap$ Co-UP*, Information Processing Letters, 68 (1998), pp. 119–124.

[14] M. Jurdziński, *Small progress measures for solving parity games*, in STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science, Proceedings, vol. 1770 of Lecture Notes in Computer Science, Springer, 2000, pp. 290–301.

[15] N. Klarlund, *Progress measures, immediate determinacy, and a subset construction for tree automata*, Annals of Pure and Applied Logic, 69 (1994), pp. 243–268.

[16] D. Martin, *Borel determinacy*, Annals of Mathematics, 102 (1975), pp. 336–371.

[17] A. Mostowski, *Games with forbidden positions*, Tech. Rep. Tech. Report 78, University of Gdansk, 1991.

[18] J. Obdrzalek, *Fast mu-calculus model checking when tree-width is bounded*, in Proceedings of CAV 2003, vol. 2752 of LNCS, Springer, 2003, pp. 80–92.

[19] J. Obdrzalek, *DAG-width - connectivity measure for directed graphs*, in SODA 2006, Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, 2006, pp. 814–821.

[20] E. Zermelo, *über eine Anwendung der Mengenlehre auf die Theorie des Schachpiels*, in Proc. 5th Internat. Congr. Mathematicians, vol. 2, Cambridge, 1913, pp. 501–504.

[21] W. Zielonka, *Infinite games on finitely coloured graphs with applications to automata on infinite trees*, Theoretical Computer Science, 200 (1998), pp. 135–183.