Down the Borel Hierarchy: Solving Muller Games via Safety Games[*]

Daniel Neider
Lehrstuhl für Informatik 7
RWTH Aachen University, Germany
neider@automata.rwth-aachen.de

Roman Rabinovich
Mathematische Grundlagen der Informatik
RWTH Aachen University, Germany
rabinovich@logic.rwth-aachen.de

Martin Zimmermann
RWTH Aachen University, Germany & University of Warsaw, Poland
zimmermann@mimuw.edu.pl

We transform a Muller game with n vertices into a safety game with $(n!)^3$ vertices whose solution allows to determine the winning regions of the Muller game and to compute a finite-state winning strategy for one player. This yields a novel antichain-based memory structure and a natural notion of permissive strategies for Muller games. Moreover, we generalize our construction by presenting a new type of game reduction from infinite games to safety games and show its applicability to several other winning conditions.

1 Introduction

Muller games are a source of interesting and challenging questions in the theory of infinite games. They are expressive enough to describe all ω-regular properties. Also, all winning conditions that depend only on the set of vertices visited infinitely often can trivially be reduced to Muller games. Hence, they subsume Büchi, co-Büchi, parity, Rabin, and Streett conditions. Furthermore, Muller games are not positionally determined, i.e., both players need memory to implement their winning strategies. In this work, we consider three aspects of Muller games: solution algorithms, memory structures, and quality measures for strategies.

To date, there are two main approaches to solve Muller games: direct algorithms and reductions. Examples for the first approach are Zielonka's recursive polynomial space algorithm [22], which is based on earlier work by McNaughton [17], and Horn's polynomial time algorithm for explicit Muller games [12]. The second approach is to reduce a Muller game to a parity game using Zielonka trees [7] or latest appearance records (LAR) [11].

In general, the number of memory states needed to win a Muller game is prohibitively large [7]. Hence, a natural task is to reduce this number (if possible) and to find new memory structures which may implement small winning strategies in subclasses of Muller games.

As for the third aspect, to the best of our knowledge there is no previous work on quality measures for strategies in Muller games. This is in contrast to other winning conditions. Recently, much attention is being paid to not just synthesize some winning strategy, but to find an optimal one according to a certain quality measure, e.g., waiting times in request-response games [13] and their extensions [23], permissiveness in parity games [1, 3], and the use of weighted automata in quantitative synthesis [2, 5].

[*] This work was supported by the projects *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* and *Logic for Interaction (LINT)* of the *European Science Foundation*.

M. Faella, A. Murano (Eds.): Games, Automata, Logics
and Formal Verification (GandALF 2012)
EPTCS 96, 2012, pp. 169–182, doi:10.4204/EPTCS.96.13

Inspired by work of McNaughton [18], we present a framework to deal with all three issues. Our main contributions are a novel algorithm and a novel type of memory structure for Muller games. We also obtain a natural quality measure for strategies in Muller games and are able to extend the definition of permissiveness to Muller games.

While investigating the interest of Muller games for "casual living-room recreation" [18], McNaughton introduced scoring functions which describe the progress a player is making towards winning a play: consider a Muller game $(\mathscr{A}, \mathscr{F}_0, \mathscr{F}_1)$, where $\mathscr{A}$ is the arena and $(\mathscr{F}_0, \mathscr{F}_1)$ is a partition of the set of loops in $\mathscr{A}$ used to determine the winner. Then, the score of a set F of vertices measures how often F has been visited completely since the last visit of a vertex not in F. Player i wins a play if and only if there is an $F \in \mathscr{F}_i$ such that the score of F tends to infinity while being reset only finitely often (a reset occurs whenever a vertex outside F is visited).

McNaughton proved the existence of strategies for the winning player that bound her opponent's scores by $|\mathscr{A}|!$ [18], provided the play starts in her winning region. The characterization above implies that such a strategy is necessarily winning. The bound $|\mathscr{A}|!$ was subsequently improved to 2 (and shown to be tight) [9]. Since some score eventually reaches value 3, the winning regions of a Muller game can be determined by solving the reachability game in which a player wins if she is the first to reach a score of 3 [1]. However, it is cumbersome to obtain a winning strategy for the infinite-duration Muller game from a winning strategy for the finite-duration reachability game. The reason is that one has to carefully concatenate finite plays of the reachability game to an infinite play of the Muller game: reaching a score of 3 infinitely often does not prevent the opponent from visiting other vertices infinitely often.

The ability to bound the losing player's scores can be seen as a safety condition as well. This allows us to devise an algorithm to solve Muller games that computes both winning regions and a winning strategy for one player. However, we do not obtain a winning strategy for the other player. In general, it is impossible to reduce a Muller game to a safety game whose solution yields winning strategies for both players, since safety conditions are on a lower level of the Borel hierarchy than Muller conditions.

Given a Muller game, we construct a safety game in which the scores of Player 1 are tracked (up to score 3). Player 0 wins the safety game, if she can prevent Player 1 from ever reaching a score of 3. This allows to compute the winning region of the Muller game by solving a safety game. Furthermore, by exploiting the intrinsic structure of the safety game's arena we present an antichain-based memory structure for Muller games. Unlike the memory structures induced by Zielonka trees, which disregard the structure of the arena, and the ones induced by LARs, which disregard the structure of the winning condition $(\mathscr{F}_0, \mathscr{F}_1)$, our memory structure takes both directly into account: a simple arena or a simple winning condition should directly lead to a small memory. The other two structures only take one source of simplicity into account, the other one can only be exploited when the game is solved. Furthermore, our memory also implements the most general non-deterministic winning strategy among those that prevent the opponent from reaching a certain score in a Muller game. Thus, our framework allows to extend the notion of permissiveness from positionally determined games to games that require memory.

Our idea of turning a Muller game into a safety game can be generalized to other types of winning conditions as well. We define a weaker notion of reduction from infinite games to safety games which not only subsumes our construction but generalizes several constructions found in the literature. Based on work on small progress measures for parity games [14], Bernet, Janin, and Walukiewicz showed how to determine the winning regions in a parity game and a winning strategy for one player by reducing it to a safety game [1]. Furthermore, Schewe and Finkbeiner [20] as well as Filiot, Jin, and Raskin [10]

[1] This reachability game was the object of study in McNaughton's investigation of humanly playable games, which, for practical reasons, should end after a bounded number of steps.

used a translation from co-Büchi games to safety games in their work on bounded synthesis and LTL realizability, respectively. We present further examples and show that our reduction allows to determine the winning region and a winning strategy for one player by solving a safety game. Thus, all these games can be solved by a new type of reduction and an algorithm for safety games. Our approach simplifies the winning condition of the game – even down the Borel hierarchy. However, this is offset by an increase in the size of the arena. Nevertheless, in the case of Muller games, our arena is only cubically larger than the arena constructed in the reduction to parity games. Furthermore, a safety game can be solved in linear time, while the question whether there is a polynomial time algorithm for parity games is open.

2 Definitions

The power set of a set S is denoted by 2^S and $\mathbb{N}$ denotes the set of non-negative integers. The prefix relation on words is denoted by $\sqsubseteq$. For $\rho \in V^\omega$ and $L \subseteq V^\omega$ we define $\mathrm{Pref}(\rho) = \{w \in V^* \mid w \sqsubseteq \rho\}$ and $\mathrm{Pref}(L) = \bigcup_{\rho \in L} \mathrm{Pref}(\rho)$. For $w = w_1 \cdots w_n$, let $\mathrm{Last}(w) = w_n$.

An arena $\mathscr{A} = (V, V_0, V_1, E)$ consists of a finite, directed graph (V, E) without terminal vertices, $V_0 \subseteq V$ and $V_1 = V \setminus V_0$, where V_i denotes the positions of Player i. We require every vertex to have an outgoing edge to avoid the nuisance of dealing with finite plays. The size $|\mathscr{A}|$ of $\mathscr{A}$ is the cardinality of V. A loop $C \subseteq V$ in $\mathscr{A}$ is a strongly connected subset of V, i.e., for every $v, v' \in C$ there is a path from v to v' that only visits vertices in C. A play in $\mathscr{A}$ starting in $v \in V$ is an infinite sequence $\rho = \rho_0\rho_1\rho_2\dots$ such that $\rho_0 = v$ and $(\rho_n, \rho_{n+1}) \in E$ for all $n \in \mathbb{N}$. The occurrence set $\mathrm{Occ}(\rho)$ and infinity set $\mathrm{Inf}(\rho)$ of ρ are given by $\mathrm{Occ}(\rho) = \{v \in V \mid \exists n \in \mathbb{N} \text{ such that } \rho_n = v\}$ and $\mathrm{Inf}(\rho) = \{v \in V \mid \exists^\omega n \in \mathbb{N} \text{ such that } \rho_n = v\}$. We also use the occurrence set of a finite play infix w, which is defined in the same way. The infinity set of a play is always a loop in the arena. A game $\mathscr{G} = (\mathscr{A}, \mathrm{Win})$ consists of an arena $\mathscr{A}$ and a set $\mathrm{Win} \subseteq V^\omega$ of winning plays for Player 0. The set of winning plays for Player 1 is $V^\omega \setminus \mathrm{Win}$.

A strategy for Player i is a mapping $\sigma\colon V^*V_i \to V$ such that $(v, \sigma(wv)) \in E$ for all $wv \in V^*V_i$. We say that σ is positional if $\sigma(wv) = \sigma(v)$ for every $wv \in V^*V_i$. A play $\rho_0\rho_1\rho_2\dots$ is consistent with σ if $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for every n with $\rho_n \in V_i$. For $v \in V$ and a strategy σ for one of the players, we define the behavior of σ from v by $\mathrm{Beh}(v, \sigma) = \{\rho \in V^\omega \mid \rho \text{ play starting in } v \text{ that is consistent with } \sigma\}$. A strategy σ for Player i is a winning strategy from a set of vertices $W \subseteq V$ if every $\rho \in \mathrm{Beh}(v, \sigma)$ for $v \in W$ is won by Player i. The winning region $W_i(\mathscr{G})$ of Player i in $\mathscr{G}$ contains all vertices from which Player i has a winning strategy. We always have $W_0(\mathscr{G}) \cap W_1(\mathscr{G}) = \emptyset$ and $\mathscr{G}$ is determined if $W_0(\mathscr{G}) \cup W_1(\mathscr{G}) = V$. A winning strategy for Player i is uniform, if it is winning from all $v \in W_i(\mathscr{G})$.

A memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ for an arena (V, V_0, V_1, E) consists of a finite set M of memory states, an initialization function $\mathrm{Init}\colon V \to M$, and an update function $\mathrm{Upd}\colon M \times V \to M$. The update function can be extended to $\mathrm{Upd}^*\colon V^+ \to M$ in the usual way: $\mathrm{Upd}^*(\rho_0) = \mathrm{Init}(\rho_0)$ and $\mathrm{Upd}^*(\rho_0 \dots \rho_n\rho_{n+1}) = \mathrm{Upd}(\mathrm{Upd}^*(\rho_0 \dots \rho_n), \rho_{n+1})$. A next-move function (for Player i) $\mathrm{Nxt}\colon V_i \times M \to V$ has to satisfy $(v, \mathrm{Nxt}(v, m)) \in E$ for all $v \in V_i$ and all $m \in M$. It induces a strategy σ for Player i with memory $\mathfrak{M}$ via $\sigma(\rho_0 \dots \rho_n) = \mathrm{Nxt}(\rho_n, \mathrm{Upd}^*(\rho_0 \dots \rho_n))$. A strategy is called finite-state if it can be implemented with a memory structure. The size of $\mathfrak{M}$ (and, slightly abusive, σ) is $|M|$.

We consider two types of games defined by specifying Win implicitly. A safety game is a tuple $\mathscr{G} = (\mathscr{A}, F)$ with $F \subseteq V$ and $\mathrm{Win} = \{\rho \in V^\omega \mid \mathrm{Occ}(\rho) \subseteq F\}$. A Muller game is a tuple $\mathscr{G} = (\mathscr{A}, \mathscr{F}_0, \mathscr{F}_1)$ where $\mathscr{F}_0$ is a set of loops of $\mathscr{A}$, $\mathscr{F}_1$ contains the loops which are not in $\mathscr{F}_0$, and $\mathrm{Win} = \{\rho \in V^\omega \mid \mathrm{Inf}(\rho) \in \mathscr{F}_0\}$, i.e., ρ is winning for Player i if and only if $\mathrm{Inf}(\rho) \in \mathscr{F}_i$. Safety games are determined with uniform positional strategies and Muller games are determined with uniform finite-state strategies of size $|\mathscr{A}|!$ [11].

3 Scoring Functions for Muller Games

In this section, we introduce scores and accumulators for Muller games. These concepts describe the progress of a player throughout a play. Intuitively, for each set $F \subseteq V$, the score of F of a play prefix w measures how often F has been visited completely since the last visit of a vertex that is not in F or since the beginning of w. The accumulator of the set F measures the progress made towards the next score increase: $\mathrm{Acc}_F(w)$ contains the vertices of F seen since the last increase of the score of F or the last visit of a vertex $v \notin F$, depending on which occurred later. For a more detailed treatment we refer to [9, 18].

Definition 1. *Let $w \in V^+$, $v \in V$, and $F \subseteq V$.*

- *Define* $\mathrm{Sc}_{\{v\}}(v) = 1$ *and* $\mathrm{Acc}_{\{v\}}(v) = \emptyset$, *and for* $F \neq \{v\}$ *define* $\mathrm{Sc}_F(v) = 0$ *and* $\mathrm{Acc}_F(v) = F \cap \{v\}$.
- *If $v \notin F$, then* $\mathrm{Sc}_F(wv) = 0$ *and* $\mathrm{Acc}_F(wv) = \emptyset$.
- *If $v \in F$ and* $\mathrm{Acc}_F(w) = F \setminus \{v\}$, *then* $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1$ *and* $\mathrm{Acc}_F(wv) = \emptyset$.
- *If $v \in F$ and* $\mathrm{Acc}_F(w) \neq F \setminus \{v\}$, *then* $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w)$ *and* $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w) \cup \{v\}$.

Also, for $\mathscr{F} \subseteq 2^V$ define $\mathrm{MaxSc}_{\mathscr{F}} \colon V^+ \cup V^\omega \to \mathbb{N} \cup \{\infty\}$ *by* $\mathrm{MaxSc}_{\mathscr{F}}(\rho) = \max_{F \in \mathscr{F}} \max_{w \sqsubseteq \rho} \mathrm{Sc}_F(w)$.

Example 1. *Let $V = \{0,1,2\}$ and $F = \{0,1\}$. We have* $\mathrm{Sc}_F(10012100) = 1$ *and* $\mathrm{Acc}_F(10012100) = \{0\}$, *but* $\mathrm{MaxSc}_{\{F\}}(10012100) = 2$, *due to the prefix* 1001. *The score for F is reset to* 0 *by the occurrence of* 2, *i.e.,* $\mathrm{Sc}_F(10012) = 0$ *and* $\mathrm{Acc}_F(10012) = \emptyset$.

If w is a play prefix with $\mathrm{Sc}_F(w) \geq 2$, then F is a loop of the arena. In an infinite play ρ, $\mathrm{Inf}(\rho)$ is the unique set F such that Sc_F tends to infinity while being reset to 0 only finitely often. Hence, $\mathrm{MaxSc}_{\mathscr{F}_{1-i}}(\rho) < \infty$ implies $\mathrm{Inf}(\rho) \in \mathscr{F}_i$. Also, we always have $\mathrm{Acc}_F(w) \subsetneq F$.

Next, we give a score-based preorder and an induced equivalence relation on play prefixes.

Definition 2. *Let $\mathscr{F} \subseteq 2^V$ and $w, w' \in V^+$.*

1. *w is $\mathscr{F}$-smaller than w', denoted by $w \leq_{\mathscr{F}} w'$, if* $\mathrm{Last}(w) = \mathrm{Last}(w')$ *and for all $F \in \mathscr{F}$:*
   - $\mathrm{Sc}_F(w) < \mathrm{Sc}_F(w')$, *or*
   - $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ *and* $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$.
2. *w and w' are $\mathscr{F}$-equivalent, denoted by $w =_{\mathscr{F}} w'$, if $w \leq_{\mathscr{F}} w'$ and $w' \leq_{\mathscr{F}} w$.*

The condition $w =_{\mathscr{F}} w'$ is equivalent to $\mathrm{Last}(w) = \mathrm{Last}(w')$ and for every $F \in \mathscr{F}$ the equalities $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ and $\mathrm{Acc}_F(w) = \mathrm{Acc}_F(w')$ hold. Thus, $=_{\mathscr{F}}$ is an equivalence relation. Both $\leq_{\mathscr{F}}$ and $=_{\mathscr{F}}$ are preserved under concatenation, i.e., $=_{\mathscr{F}}$ is a congruence.

Lemma 1. *Let $\mathscr{F} \subseteq 2^V$ and $w, w' \in V^+$.*

1. *If $w \leq_{\mathscr{F}} w'$, then $wu \leq_{\mathscr{F}} w'u$ for all $u \in V^*$.*
2. *If $w =_{\mathscr{F}} w'$, then $wu =_{\mathscr{F}} w'u$ for all $u \in V^*$.*

4 Solving Muller Games by Solving Safety Games

In this section, we show how to solve a Muller game by solving a safety game. Our approach is based on the existence of winning strategies for Muller games that bound the losing player's scores by 2.

Lemma 2 ([9])**.** *In every Muller game $\mathscr{G} = (\mathscr{A}, \mathscr{F}_0, \mathscr{F}_1)$ Player i has a winning strategy σ from $W_i(\mathscr{G})$ such that* $\mathrm{MaxSc}_{\mathscr{F}_{1-i}}(\rho) \leq 2$ *for every play $\rho \in$* $\mathrm{Beh}(v, \sigma)$ *with $v \in W_i(\mathscr{G})$.*

The following example shows that the bound 2 is tight.

Example 2. *Consider the Muller game $\mathscr{G} = (\mathscr{A}, \mathscr{F}_0, \mathscr{F}_1)$, where $\mathscr{A}$ is depicted in Figure 1, $\mathscr{F}_0 = \{\{0\},\{2\},\{0,1,2\}\}$ and $\mathscr{F}_1 = \{\{0,1\},\{1,2\}\}$. By alternatingly moving from 1 to 0 and to 2, Player 0 wins from every vertex, i.e., we have $W_0(\mathscr{G}) = \{0,1,2\}$, and she bounds Player 1's scores by 2. However, he is able to achieve a score of two: consider a play starting at 1 and suppose (w.l.o.g.) that Player 0 moves to vertex 0. Then, Player 1 uses the self-loop once before moving back to 1, thereby reaching a score of 2 for the loop $\{0,1\} \in \mathscr{F}_1$.*

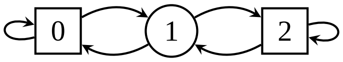

Figure 1: The arena $\mathscr{A}$ for Example 2

A simple consequence of Lemma 2 is that a vertex v is in Player 0's winning region of the Muller game $\mathscr{G}$ if and only if she can prevent her opponent from ever reaching a score of 3 for a set in $\mathscr{F}_1$. This is a safety condition which only talks about small scores of one player. To determine the winner of $\mathscr{G}$, we construct an arena which keeps track of the scores of Player 1 up to threshold 3. The winning condition F of the safety game requires Player 0 to prevent a score of 3 for her opponent.

Theorem 1. *Let $\mathscr{G}$ be a Muller game with vertex set V. One can effectively construct a safety game $\mathscr{G}_S$ with vertex set V^S and a mapping $f\colon V \to V^S$ with the following properties:*

1. *For every $v \in V$: $v \in W_i(\mathscr{G})$ if and only if $f(v) \in W_i(\mathscr{G}_S)$.*
2. *Player 0 has a finite-state winning strategy from $W_0(\mathscr{G})$ with memory $M \subseteq W_0(\mathscr{G}_S)$.*
3. *$|V^S| \leq \left(\sum_{k=1}^{|V|} \binom{|V|}{k} \cdot k! \cdot 2^k \cdot k!\right) + 1 \leq (|V|!)^3$.*

Note that the first statement speaks about both players while the second one only speaks about Player 0. This is due to the fact that the safety game keeps track of Player 1's scores only, which allows Player 0 to prove that she can prevent him from reaching a score of 3. But as soon as a score of 3 is reached, the play is stopped. To obtain a winning strategy for Player 1, we have to swap the roles of the players and construct a safety game which keeps track of the scores of Player 0. Alternatively, we could construct the arena which keeps track of both player's scores. However, that would require to define two safety games in this arena: one in which Player 0 has to avoid a score of 3 for Player 1 and vice versa. This arena is larger than the ones in which only the scores of one player are tracked (but still smaller than $(|V|!)^3$). It is well-known that it is impossible to reduce a Muller game to a single safety game and thereby obtain winning strategies for both players. We come back to this in Section 6.

We begin the proof of Theorem 1 by defining the safety game $\mathscr{G}_S$. Let $\mathscr{G} = (\mathscr{A}, \mathscr{F}_0, \mathscr{F}_1)$ with arena $\mathscr{A} = (V, V_0, V_1, E)$. We define

$$\mathrm{Plays}_{<3} = \{w \mid w \text{ play prefix in } \mathscr{G} \text{ and } \mathrm{MaxSc}_{\mathscr{F}_1}(w) < 3\}$$

to be the set of play prefixes in $\mathscr{G}$ in which the scores of Player 1 are at most 2 and we define

$$\mathrm{Plays}_{=3} = \{w_0 \cdots w_{n+1} \mid w_0 \cdots w_{n+1} \text{ play prefix in } \mathscr{G},\ \mathrm{MaxSc}_{\mathscr{F}_1}(w_0 \cdots w_n) \leq 2, \text{ and } \mathrm{MaxSc}_{\mathscr{F}_1}(w_0 \cdots w_n w_{n+1}) = 3\ \}$$

to be the set of play prefixes in which Player 1 just reached a score of 3. Furthermore, let $\mathrm{Plays}_{\leq 3} = \mathrm{Plays}_{<3} \cup \mathrm{Plays}_{=3}$. Note that these definitions ignore the scores of Player 0. The arena of the safety game we are about to define is the $=_{\mathscr{F}_1}$-quotient of the unraveling of $\mathscr{A}$ up to the positions where Player 1 reaches a score of 3 for the first time (if he does at all).

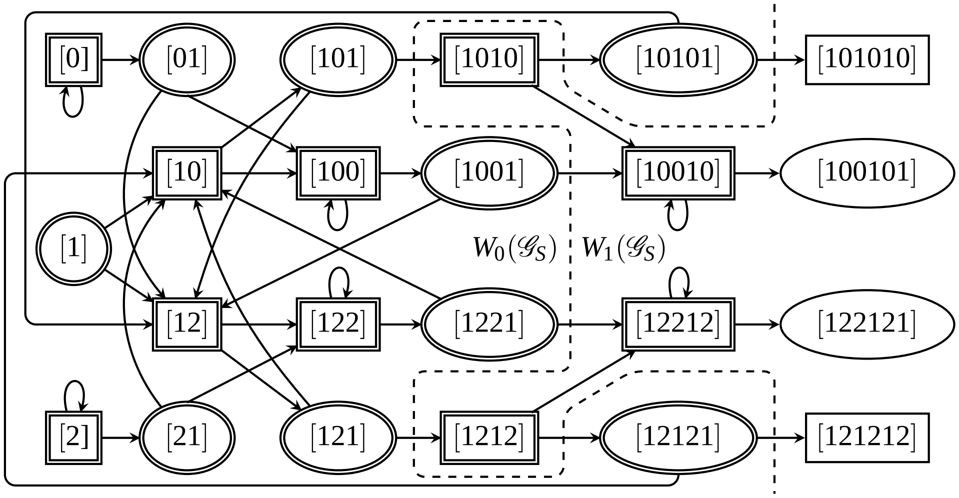

Figure 2: The safety game $\mathscr{G}_S$ for $\mathscr{G}$ from Example 2 (vertices drawn with double lines are in F); the dashed line separates the winning regions.

Formally, we define $\mathscr{G}_S = ((V^S, V_0^S, V_1^S, E^S), F)$ where

- $V^S = \text{Plays}_{\leq 3}/_{=_{\mathscr{F}_1}}$,
- $V_i^S = \{[w]_{=_{\mathscr{F}_1}} \mid [w]_{=_{\mathscr{F}_1}} \in V^S \text{ and } \text{Last}(w) \in V_i\}$ for $i \in \{0,1\}$,
- $([w]_{=_{\mathscr{F}_1}}, [wv]_{=_{\mathscr{F}_1}}) \in E^S$ for all $w \in \text{Plays}_{<3}$ and all v with $(\text{Last}(w), v) \in E$ [2], and
- $F = \text{Plays}_{<3}/_{=_{\mathscr{F}_1}}$.

The definitions of V_0^S and V_1^S are independent of representatives, as $w =_{\mathscr{F}_1} w'$ implies $\text{Last}(w) = \text{Last}(w')$, we have $V^S = V_0^S \cup V_1^S$ due to $V = V_0 \cup V_1$, and F is well-defined, since every equivalence class in $\text{Plays}_{<3}/_{=_{\mathscr{F}_1}}$ is also one in $\text{Plays}_{\leq 3}/_{=_{\mathscr{F}_1}}$. Finally, let $f(v) = [v]_{=_{\mathscr{F}_1}}$ for every $v \in V$.

Remark 1. *If $([w]_{=_{\mathscr{F}_1}}, [w']_{=_{\mathscr{F}_1}}) \in E^S$, then we have $(\text{Last}(w), \text{Last}(w')) \in E$.*

For the sake of readability, we denote $=_{\mathscr{F}_1}$-equivalence classes by $[w]$ from now on. All definitions and statements below are independent of representatives and we refrain from mentioning it from now on.

Example 3. *To illustrate these definitions, Figure 2 depicts the safety game $\mathscr{G}_S$ for the Muller game $\mathscr{G}$ from Example 2. One can verify easily that the vertices $[v]$ for $v \in V$ are in the winning region of Player 0. This corresponds to the fact that Player 0's winning region in the Muller game contains every vertex.*

The proof of Theorem 1 is split into several lemmata. Due to determinacy of both games, it suffices to consider only one Player (we pick $i = 0$) to prove Theorem 1.1.

To win the safety game, we simulate a winning strategy for the Muller game that bounds Player 1's scores by 2, which suffices to avoid the vertices in $V^S \setminus F$, which encode that a score of three is reached.

Lemma 3. *For every $v \in V$: if $v \in W_0(\mathscr{G})$ then $[v] \in W_0(\mathscr{G}_S)$.*

[2] Hence, every vertex in $\text{Plays}_{=3}$ is terminal, contrary to our requirements on an arena. However, every play visiting these vertices is losing for Player 0 no matter how it is continued. To simplify the following proofs, we refrain from defining outgoing edges for these vertices.

For the other direction of Theorem 1.1 we show that a subset of $W_0(\mathscr{G}_S)$ can be turned into a memory structure for Player 0 in the Muller game that induces a winning strategy. We use the $=_{\mathscr{F}_1}$-equivalence class of w as memory state to keep track of Player 1's scores in $\mathscr{G}$. But instead of using all equivalence classes in the winning region of Player 0, it suffices to consider the maximal ones with respect to $\leq_{\mathscr{F}_1}$ that are reachable via a fixed positional winning strategy for her in the safety game. Formally, we have to lift $\leq_{\mathscr{F}_1}$ to equivalence classes by defining $[w] \leq_{\mathscr{F}_1} [w']$ if and only if $w \leq_{\mathscr{F}_1} w'$.

The following proof is similar to the reductions from co-Büchi [10, 16, 20] and parity games [1] to safety games, but for the more general case of Muller games. We come back to the similarities to the latter reduction when we want to determine permissive strategies in the next section.

Lemma 4. *For all $v \in V$: if $[v] \in W_0(\mathscr{G}_S)$ then $v \in W_0(\mathscr{G})$.*

Proof. Let σ be a uniform positional winning strategy for Player 0 in $\mathscr{G}_S$ and let $R \subseteq V^S$ be the set of vertices which are reachable from $W_0(\mathscr{G}_S) \cap \{[v] \mid v \in V\}$ by plays consistent with σ. Every $[w] \in R \cap V_0^S$ has exactly one successor in R (which is of the form $[wv]$ for some $v \in V$) and dually, every successor of $[w] \in R \cap V_1^S$ (which are exactly the classes $[wv]$ with $(\mathrm{Last}(w), v) \in E$) is in R. Now, let $R_{\max}$ be the set of $\leq_{\mathscr{F}_1}$-maximal elements of R. Applying the facts about successors of vertices in R stated above, we obtain the following remark.

Remark 2. *Let $R_{\max}$ be defined as above.*

1. *For every $[w] \in R_{\max} \cap V_0^S$, there is a $v \in V$ with $(\mathrm{Last}(w), v) \in E$ and there is a $[w'] \in R_{\max}$ such that $[wv] \leq_{\mathscr{F}_1} [w']$.*
2. *For every $[w] \in R_{\max} \cap V_1^S$ and each of its successors $[wv]$, there is a $[w'] \in R_{\max}$ such that $[wv] \leq_{\mathscr{F}_1} [w']$.*

Thus, instead of updating the memory from $[w]$ to $[wv]$ (and thereby keeping track of the exact scores) when processing a vertex v, we can directly update it to a maximal element that is $\mathscr{F}_1$-larger than $[wv]$ (and thereby over-approximate the exact scores). Formally, we define $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ by $M = R_{\max} \cup \{\bot\}$ [3],

$$\mathrm{Init}(v) = \begin{cases} [w] & \text{if } [v] \in W_0(\mathscr{G}_S) \text{ and there exists } [w] \in R_{\max} \text{ with } [v] \leq_{\mathscr{F}_1} [w], \\ \bot & \text{otherwise,} \end{cases}$$

$$\mathrm{Upd}([w], v) = \begin{cases} [w'] & \text{if there is some } [w'] \in R_{\max} \text{ such that } [wv] \leq_{\mathscr{F}_1} [w'], \\ \bot & \text{otherwise.} \end{cases}$$

This implies $[w] \leq_{\mathscr{F}_1} \mathrm{Upd}^*(w)$ for every $w \in V^+$ with $\mathrm{Upd}^*(w) \neq \bot$. Thus, $\mathrm{Last}(w) = \mathrm{Last}(w')$, where $[w'] = \mathrm{Upd}^*(w)$. Using Remark 2, we define the next-move function

$$\mathrm{Nxt}(v, [w]) = \begin{cases} v' & \text{if } \mathrm{Last}(w) = v,\ (v, v') \in E, \text{ and there exists } [w'] \in R_{\max} \text{ such that } [wv'] \leq_{\mathscr{F}_1} [w'], \\ v'' & \text{otherwise (where } v'' \text{ is some vertex with } (v, v'') \in E), \end{cases}$$

and $\mathrm{Nxt}(v, \bot) = v''$ for some v'' with $(v, v'') \in E$. The second case in the case distinction above is just to match the formal definition of a next-move function; it is never invoked due to $\mathrm{Last}(w) = \mathrm{Last}(w')$ for $\mathrm{Upd}^*(w) = [w']$ or $\mathrm{Upd}^*(w) = \bot$.

[3] We use the memory state $\bot$ to simplify our proof. It is not reachable via plays that are consistent with the implemented strategy and can therefore be eliminated.

It remains to show that the strategy σ implemented by $\mathfrak{M}$ and Nxt is a winning strategy for Player 0 from $W = \{v \mid [v] \in W_0(\mathscr{G}_S)\}$. An inductive application of Remark 2 shows that every play w that starts in W and is consistent with σ satisfies $\mathrm{Upd}^*(w) \neq \perp$. This bounds the scores of Player 1 by 2, as we have $[w] \leq_{\mathscr{F}_1} \mathrm{Upd}^*(w) \in R_{\max} \subseteq \mathrm{Plays}_{<3}$ for every such play. Hence, σ is indeed a winning strategy for Player 0 from W. □

Using Lemma 3 and the construction in the proof of Lemma 4 proves Theorem 1.2.

Corollary 1. *Player* 0 *has a finite-state winning strategy from* $W_0(\mathscr{G})$ *whose memory states form an* $\leq_{\mathscr{F}_1}$*-antichain in* $W_0(\mathscr{G}_S)$*.*

To finish the proof of Theorem 1, we determine the size of $\mathscr{G}_S$ to prove the third statement. To this end, we use the concept of a *latest appearance record* (LAR) [11, 17]. Note that we do not need a hit position for our purposes. A word $\ell \in V^+$ is an LAR if every vertex $v \in V$ appears at most once in ℓ. Next, we map each $w \in V^+$ to a unique LAR, denoted by $\mathrm{LAR}(w)$, as follows: $\mathrm{LAR}(v) = v$ for every $v \in V$ and for $w \in V^+$ and $v \in V$ we define $\mathrm{LAR}(wv) = \mathrm{LAR}(w)v$ if $v \notin \mathrm{Occ}(w)$ and $\mathrm{LAR}(wv) = p_1 p_2 v$ if $\mathrm{LAR}(w) = p_1 v p_2$. A simple induction shows that $\mathrm{LAR}(w)$ is indeed an LAR, which also ensures that the decomposition of w in the second case of the inductive definition is unique. We continue by showing that $\mathrm{LAR}(w)$ determines all but $|\mathrm{LAR}(w)|$ many of w's scores and accumulators.

Lemma 5. *Let* $w \in V^+$ *and* $\mathrm{LAR}(w) = v_k v_{k-1} \cdots v_1$*.*

1. $w = x_k v_k x_{k-1} v_{k-1} \cdots x_2 v_2 x_1 v_1$ *for some* $x_i \in V^*$ *with* $\mathrm{Occ}(x_i) \subseteq \{v_1, \dots, v_i\}$ *for every i.*
2. $\mathrm{Sc}_F(w) > 0$ *if and only if* $F = \{v_1, \dots, v_i\}$ *for some i.*
3. *If* $\mathrm{Sc}_F(w) = 0$*, then* $\mathrm{Acc}_F(w) = \{v_1, \dots, v_i\}$ *for the maximal i such that* $\{v_1, \dots, v_i\} \subseteq F$ *and* $\mathrm{Acc}_F(w) = \emptyset$ *if no such i exists.*
4. *Let* $\mathrm{Sc}_F(w) > 0$ *and* $F = \{v_1, \dots, v_i\}$*. Then,* $\mathrm{Acc}_F(w) \in \{\emptyset\} \cup \{\{v_1, \dots, v_j\} \mid j < i\}$*.*

This characterization allows us to bound the size of $\mathscr{G}_S$ and to prove Theorem 1.3.

Lemma 6. *We have* $|V^S| \leq \left(\sum_{k=1}^{|V|} \binom{|V|}{k} \cdot k! \cdot 2^k \cdot k!\right) + 1 \leq (|V|!)^3$*.*

Proof. In every safety game, we can merge the vertices in $V \setminus F$ to a single vertex without changing $W_0(\mathscr{G})$. Since $[v] \in F$, we also retain the equivalence $v \in W_i(\mathscr{G}) \Leftrightarrow [v] \in W_i(\mathscr{G}_S)$.

Hence, it remains to bound the index of $\mathrm{Plays}_{<3}/_{=_{\mathscr{F}_1}}$. Lemma 5 shows that a play prefix $w \in V^+$ has $|\mathrm{LAR}(w)|$ many sets with non-zero score. Furthermore, the accumulator of the sets with score 0 is determined by $\mathrm{LAR}(w)$. Now, consider a play $w \in \mathrm{Plays}_{<3}$ and a set $F \in \mathscr{F}_1$ with non-zero score. We have $\mathrm{Sc}_F(w) \in \{1, 2\}$ and there are exactly $|F|$ possible values for $\mathrm{Acc}_F(w)$ due to Lemma 5.4. Finally, $\mathrm{LAR}(w) = \mathrm{LAR}(w')$ implies $\mathrm{Last}(w) = \mathrm{Last}(w')$. Hence, the index of $\mathrm{Plays}_{<3}/_{=_{\mathscr{F}_1}}$ is bounded by the number of LARs, which is $\sum_{k=1}^{n} \binom{n}{k} \cdot k!$, times the number of possible score and accumulator combinations for each LAR ℓ of length k, which is bounded by $2^k \cdot k!$. □

In the proof of Theorem 1.2, we used the maximal elements of Player 0's winning region of the safety game that are reachable via a fixed winning strategy. It is the choice of this strategy that determines the size of our memory structure. However, finding a winning strategy that visits at most $k \in \mathbb{N}$ vertices in an arena (from a fixed initial vertex) for a given k is NP-complete. This can be shown by a reduction from the vertex cover problem (compare, e.g., [8] where a more general result is shown). Moreover, it is not even clear that a small strategy also yields few maximal elements.

In general, a player cannot prevent her opponent from reaching a score of 2, but there are arenas in which she can do so. By first constructing the subgame $\mathscr{G}_S'$ up to threshold 2 (which is smaller than $\mathscr{G}_S$), we can possibly determine a subset of Player 0's winning region faster and obtain a (potentially) smaller finite-state winning strategy for this subset. But Example 2 shows that this approach is not complete.

5 Permissive Strategies for Muller Games

Bernet et al. introduced the concept of permissive strategies for parity games[4] [1], a non-deterministic winning strategy that subsumes the behavior of every positional (non-deterministic) winning strategy. To compute such a strategy, they reduce a parity game to a safety game. The main observation underlying their reduction is the following: if we denote the number of vertices of priority c by n_c, then a (non-deterministic) positional winning strategy for Player 0 does not allow a play in which an odd priority c is visited $n_c + 1$ times without visiting a smaller priority in between. This property can be formulated using scoring functions for parity games as well. The scoring function for a priority c counts the occurrences of c since the last occurrence of a smaller priority (such an occurrence resets the score for c to 0). Hence, our work on Muller games can be seen as a generalization of Bernet et al.'s work. While the bound n_c on the scores in a parity game is straightforward, the bound 2 for Muller games is far from obvious.

Since both constructions are very similar, it is natural to ask whether we can use the concept of permissive strategies for Muller games. In parity games, we ask for a non-deterministic strategy that subsumes the behavior of every *positional* strategy. As positional strategies do not suffice to win Muller games, we have to give a new definition of permissiveness for such games. In other words, we need to specify the strategies whose behaviors a permissive strategies for a Muller game should subsume. One way to do this is to fix a sufficiently large bound M and to require that a permissive strategy for a Muller games subsumes the behavior of every finite-state winning strategy of size at most M (this was already proposed by Bernet et al. for parity games [1]).

However, we prefer to take a different approach. By closely inspecting the reduction of parity to safety games, it becomes apparent that the induced strategy does not only subsume the behavior of every positional winning strategy, but rather the behavior of every strategy that prevents the opponent from reaching a score of $n_c + 1$ for some odd priority c (in terms of scoring functions for parity games). It is this formulation that we extend to Muller games: a (non-deterministic) winning strategy for a Muller game is permissive, if it subsumes the behavior of every (non-deterministic) winning strategy that prevents the losing player from reaching a score of 3. We formalize this notion in the following and show how to compute such strategies from the safety game constructed in the previous section.

A multi-strategy for Player i in an arena (V, V_0, V_1, E) is a mapping $\sigma \colon V^*V_i \to 2^V \setminus \{\emptyset\}$ such that $v' \in \sigma(wv)$ implies $(v, v') \in E$. A play ρ is consistent with σ if $\rho_{n+1} \in \sigma(\rho_0 \cdots \rho_n)$ for every n such that $\rho_n \in V_i$. We still denote the plays starting in a vertex v that are consistent with a multi-strategy σ by $\mathrm{Beh}_{\mathscr{A}}(v, \sigma)$ and define $\mathrm{Beh}_{\mathscr{A}}(W, \sigma) = \bigcup_{v \in W} \mathrm{Beh}_{\mathscr{A}}(v, \sigma)$ for every subset $W \subseteq V$. A multi-strategy σ is winning for Player 0 from a set of vertices W in a game $(\mathscr{A}, \mathrm{Win})$ if $\mathrm{Beh}_{\mathscr{A}}(W, \sigma) \subseteq \mathrm{Win}$, and a multi-strategy τ is winning for Player 1 from W, if $\mathrm{Beh}_{\mathscr{A}}(W, \sigma) \subseteq V^\omega \setminus \mathrm{Win}$. It is clear that the winning regions of a game do not change when we allow multi-strategies instead of standard strategies.

To define finite-state multi-strategies we have to allow a next-move function to return more than one vertex, i.e., we have $\mathrm{Nxt} \colon V_i \times M \to 2^V \setminus \{\emptyset\}$ such that $v' \in \mathrm{Nxt}(v, m)$ implies $(v, v') \in E$. A memory structure $\mathfrak{M}$ and Nxt implement a multi-strategy σ via $\sigma(wv) = \mathrm{Nxt}(v, \mathrm{Upd}^*(wv))$.

Definition 3. *A multi-strategy σ' for a Muller game $\mathscr{G}$ is permissive, if*

1. *σ' is a winning strategy from every vertex in $W_0(\mathscr{G})$, and*
2. $\mathrm{Beh}_{\mathscr{A}}(v, \sigma) \subseteq \mathrm{Beh}_{\mathscr{A}}(v, \sigma')$ *for every multi-strategy σ and every vertex v with* $\mathrm{MaxSc}_{\mathscr{F}_1}(\rho) \leq 2$ *for every $\rho \in \mathrm{Beh}_{\mathscr{A}}(v, \sigma)$.*

[4] A parity game $(\mathscr{A}, \Omega)$ consists of an arena $\mathscr{A}$ and a priority function $\Omega \colon V \to \mathbb{N}$. A play ρ is winning for Player 0 if the minimal priority that is seen infinitely often during the play is even.

The original definition for parity games replaces the second condition by the following requirement: $\mathrm{Beh}_{\mathscr{A}}(v,\sigma) \subseteq \mathrm{Beh}_{\mathscr{A}}(v,\sigma')$ for every positional multi-strategy σ and every v from which σ is winning.

Example 4. *Once again consider the Muller game of Example 2. Starting at vertex* 1*, moving to* 0 *is consistent with a winning strategy for Player* 0 *that bounds Player* 1*'s scores by* 2*. Similarly, moving to* 2 *is also consistent with a winning strategy for Player* 0 *that bounds Player* 1*'s scores. Hence, we have* $\sigma'(1) = \{0,2\}$ *for every permissive strategy* σ'*. Now consider the play prefix* 10*. Here it is Player* 1*'s turn and he can use the self-loop either infinitely often (which yields a play that is winning for Player* 0*) or only finitely often (say n times) before moving back to vertex* 1*. In this situation, i.e., with play prefix* $10^{n+1}1$*, a strategy that bounds Player* 1*'s scores by* 2 *has to move to vertex* 2*. Hence, we must have* $\sigma'(10^{n+1}1) \supseteq \{2\}$*. However, it is possible that we also have* $1 \in \sigma'(10^{n+1}1)$*, since a permissive strategy may allow more plays than the ones of strategies that bound Player* 1*'s scores by* 2*. However, at some point,* σ' *has to disallow the move back to vertex* 0*, otherwise it would allow a play that is losing for her.*

Using the safety game $\mathscr{G}_S$ defined in the previous section, we are able to show that Player 0 always has a finite-state permissive strategy and how to compute one.

Theorem 2. *Let* $\mathscr{G}$ *be a Muller game and* $\mathscr{G}_S$ *the corresponding safety game as above. Then, Player* 0 *has a finite-state permissive strategy for* $\mathscr{G}$ *with memory states* $W_0(\mathscr{G}_S)$*.*

The proof is very similar to the one for Theorem 1.2 (cf. the construction in the proof of Lemma 4), but we have to use all vertices in $W_0(\mathscr{G}_S)$ as memory states to implement a permissive strategy, only using the maximal ones (restricted to those reachable by some fixed winning strategy for the safety games) does not suffice. Furthermore, the next-move function does not return one successor that guarantees a memory update to a state from $W_0(\mathscr{G}_S)$, but it returns all such states.

Proof. We define $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ where $M = W_0(\mathscr{G}_S) \cup \{\perp\}$ [5],

$$\mathrm{Init}(v) = \begin{cases} [v] & \text{if } [v] \in W_0(\mathscr{G}_S), \\ \perp & \text{otherwise,} \end{cases}$$

$$\mathrm{Upd}([w],v) = \begin{cases} [wv] & \text{if } [wv] \in W_0(\mathscr{G}_S), \\ \perp & \text{otherwise.} \end{cases}$$

Hence, we have $\mathrm{Upd}^*(w) = [w] \in W_0(\mathscr{G}_S)$ as long as every prefix x of w satisfies $[x] \in W_0(\mathscr{G}_S)$, and $\mathrm{Upd}^*(w) = \perp$ otherwise. We define Nxt by $\mathrm{Nxt}(v,\perp) = \{v'\}$ for some successor v' of v and

$$\mathrm{Nxt}(v,[w]) = \begin{cases} \{v' \mid [wv'] \in W_0(\mathscr{G}_S)\} & \text{if } [w] \in W_0(\mathscr{G}_S) \text{ and } \mathrm{Last}(w) = v, \\ \{v''\} & \text{otherwise, where } v'' \text{ is some successor of } v. \end{cases}$$

Since every vertex in $W_0(\mathscr{G}_S) \cap V_0^S$ has at least one successor in $W_0(\mathscr{G}_S)$, the next-move function always returns non-empty set of successors of v in $\mathscr{G}$.

It remains to show that the strategy σ' implemented by $\mathfrak{M}$ and Nxt is permissive. We begin by showing that σ is winning from every vertex $v \in W_(\mathscr{G})$: due to Lemma 3, we have $[v] \in W_0(\mathscr{G}_S)$. Hence, the memory is initialized with $[v] \in W_0(\mathscr{G}_S)$. A simple induction shows $\mathrm{Upd}^*(w) = [w] \in W_0(\mathscr{G}_S)$ for every play prefix that starts in $[v]$ is consistent with σ'. This bounds Player 1's scores by 2. Hence, σ' is indeed winning from v.

[5] Again, we use the memory state $\perp$ to simplify our proof. It is not reachable via plays that are consistent with the strategy implemented by $\mathfrak{M}$ and can therefore eliminated and its incoming transitions can be redefined arbitrarily.

Finally, consider a multi-strategy σ and a vertex v such that $\mathrm{MaxSc}_{\mathscr{F}_1}(\rho) \leq 2$ for every play $\rho \in \mathrm{Beh}_{\mathscr{A}}(v,\sigma)$. We have to show that every play $\rho \in \mathrm{Beh}_{\mathscr{A}}(v,\sigma)$ is consistent with σ'. Since σ is winning from v (as it bounds Player 1's scores), we have $v \in W_0(\mathscr{G})$. Now, assume ρ is not consistent with σ' and let $\rho_0 \cdots \rho_n \rho_{n+1}$ be the shortest prefix such that $\rho_{n+1} \notin \sigma'(\rho_0 \cdots \rho_n)$. Then, we have $[\rho_0 \cdots \rho_n \rho_{n+1}] \notin W_0(\mathscr{G}_S)$. Hence, Player 1 has a strategy to enforce a visit to $V^S \setminus F$ in $\mathscr{G}_S$ starting in $[\rho_0 \cdots \rho_n \rho_{n+1}]$. Player 1 can mimic this strategy in $\mathscr{G}$ to enforce a score of 3 against every strategy of Player 0 when starting with the play prefix $\rho_0 \cdots \rho_n \rho_{n+1}$. Since this prefix is consistent with σ, which we have assumed to bound Player 1's scores by 2, we have derived the desired contradiction. □

6 Safety Reductions for Infinite Games

It is well-known that classical game reductions are not able to reduce Muller games to safety games, since they induce continuous functions mapping (winning) plays of the original game to (winning) plays of the reduced game. The existence of such functions is tied to topological properties of the sets of winning plays in both games. However, we transformed a Muller game to a safety game which allowed us to determine the winning regions and a winning strategy for one player. This is possible, since our reduction does not induce a continuous function: a play is stopped as soon as Player 1 reaches a score of 3, but it can (in general) be extended to be winning for Player 0.

In this section, we briefly discuss the reason why Muller games can not be reduced to safety games in the classical sense, and then we present a novel type of game reduction that allows us to reduce many games known from the literature to safety games. The advantage of this safety reduction is that the reduced game is always a safety game. Hence, we can determine the winning regions of various games from different levels of the Borel hierarchy using the same technique. However, we only obtain a winning strategy for one player, and to give such a reduction, we need to have some information on the type of winning strategies a player has in such a game. Let us begin by discussing classical game reductions.

An arena $\mathscr{A} = (V, V_0, V_1, E)$ and a memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ for $\mathscr{A}$ induce the expanded arena $\mathscr{A} \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E')$ where $((s,m),(v',m')) \in E'$ if and only if $(v,v') \in E$ and $\mathrm{Upd}(m,v') = m'$. Every play ρ in $\mathscr{A}$ has a unique extended play $\rho' = (\rho_0,m_0)(\rho_1,m_1)(\rho_2,m_2)\ldots$ in $\mathscr{A} \times \mathfrak{M}$ defined by $m_0 = \mathrm{Init}(\rho_0)$ and $m_{n+1} = \mathrm{Upd}(m_n, \rho_{n+1})$, i.e., $m_n = \mathrm{Upd}^*(\rho_0 \cdots \rho_n)$. A game $\mathscr{G} = (\mathscr{A}, \mathrm{Win})$ is reducible to $\mathscr{G}' = (\mathscr{A}', \mathrm{Win}')$ via $\mathfrak{M}$, written $\mathscr{G} \leq_{\mathfrak{M}} \mathscr{G}'$, if $\mathscr{A}' = \mathscr{A} \times \mathfrak{M}$ and every play ρ in $\mathscr{G}$ is won by the player who wins the extended play ρ' in $\mathscr{G}'$, i.e., $\rho \in \mathrm{Win}$ if and only if $\rho' \in \mathrm{Win}'$.

Lemma 7. *Let $\mathscr{G}$ be a game with vertex set V and $W \subseteq V$. If $\mathscr{G} \leq_{\mathfrak{M}} \mathscr{G}'$ and Player i has a positional winning strategy for $\mathscr{G}'$ from $\{(v, \mathrm{Init}(v)) \mid v \in W\}$, then she also has a finite-state winning strategy with memory $\mathfrak{M}$ for $\mathscr{G}$ from W.*

The set $\mathrm{Win}_M \subseteq V^\omega$ of winning plays of a Muller game is in general on a higher level of the Borel hierarchy than the set $\mathrm{Win}_S \subseteq U^\omega$ of winning plays of a safety game. Hence, in general, there exists no continuous (in the Cantor topology) function $f \colon V^\omega \to U^\omega$ such that $\rho \in \mathrm{Win}_M$ if and only if $f(\rho) \in \mathrm{Win}_S$ (see, e.g., [15]). Since the mapping from a play in $\mathscr{A}$ to its extended play in $\mathscr{A} \times \mathfrak{M}$ is continuous, we obtain the following negative result (which holds for other pairs of games as well).

Corollary 2. *In general, Muller games cannot be reduced to safety games.*

To overcome this, we present a novel type of game reduction which encompasses the construction presented in Section 4 and is applicable to many other infinite games.

Definition 4. *A game $\mathcal{G} = (\mathcal{A}, \mathrm{Win})$ with vertex set V is (finite-state) safety reducible, if there is a regular language $L \subseteq V^*$ of finite words such that:*

- *For every play $\rho \in V^\omega$: if $\mathrm{Pref}(\rho) \subseteq L$, then $\rho \in \mathrm{Win}$.*
- *If $v \in W_0(\mathcal{G})$, then Player 0 has a strategy σ such that $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L$.*

Note that a strategy σ satisfying $\mathrm{Pref}(\mathrm{Beh}(v, \sigma)) \subseteq L$ is winning for Player 0 from v. Many games appearing in the literature on infinite games are safety reducible, although these reductions do neither yield fast solution algorithms nor optimal memory structures.

- In a Büchi game $\mathcal{G}$, Player 0 has a positional winning strategy such that every consistent play visits a vertex in F at least every $k = |V \setminus F|$ steps. Hence, $\mathcal{G}$ is safety reducible with $L = \mathrm{Pref}(((V \setminus F)^{\leq k} \cdot F)^\omega)$.
- In a co-Büchi game $\mathcal{G}$, Player 0 has a positional winning strategy such that every consistent play stays in F after visiting each vertex in $V \setminus F$ at most once. Hence, $\mathcal{G}$ is safety reducible with $L = \mathrm{Pref}(\{w \cdot F^\omega \mid \text{ each } v \in V \setminus F \text{ appears at most once in } w\})$.
- In a request-response game $\mathcal{G}$, Player 0 has a finite-state winning strategy such that in every consistent play every request is answered within $k = |V| \cdot r \cdot 2^{r+1}$ steps, where r is the number of request-response pairs [21]. Hence, $\mathcal{G}$ is safety reducible to the language of prefixes of plays in which every request is answered within k steps.
- In a parity game, Player 0 has a positional winning strategy such that every consistent play does not visit $n_c + 1$ vertices with an odd priority c without visiting a smaller even priority in between, where n_c is the number of vertices with priority c. Hence, $\mathcal{G}$ is safety reducible to the language of prefixes of plays satisfying this condition for every odd priority c.
- Lemma 2 shows that a Muller game is safety reducible to the language of prefixes of plays that never reach a score of 3 for Player 1.

Let $\mathcal{A} = (V, V_0, V_1, E)$ be an arena and let $\mathfrak{A} = (Q, V, q_0, \delta, F)$ be a deterministic finite automaton recognizing a language over V. We define the arena $\mathcal{A} \times \mathfrak{A} = (V \times Q, V_0 \times Q, V_1 \times Q, E \circ \delta)$ where $((v,q),(v',q')) \in E \circ \delta$ if and only $(v, v') \in E$ and $\delta(q, v') = q'$.

Theorem 3. *Let $\mathcal{G}$ be a game with vertex set V that is safety reducible with language $L(\mathfrak{A})$ for some DFA $\mathfrak{A} = (Q, V, q_0, \delta, F)$. Define the safety game $\mathcal{G}' = (\mathcal{A} \times \mathfrak{A}, V \times F)$.*

1. *$v \in W_0(\mathcal{G})$ if and only if $(v, \delta(q_0, v)) \in W_0(\mathcal{G}')$.*
2. *Player 0 has a finite-state winning strategy from $W_0(\mathcal{G})$ with memory states Q.*

It is easy to show that every game in which Player 0 has a finite-state winning strategy is safety-reducible to the prefixes of plays consistent with this strategy, which is a regular language. However, for this construction, we need a finite-state winning strategy, i.e., there is no need for a safety reduction.

If $\mathcal{G}$ is determined, then Theorem 3.1 is equivalent to $v \in W_i(\mathcal{G})$ if and only if $(v, \delta(q_0, v)) \in W_i(\mathcal{G}')$. Hence, all games discussed above can be solved by solving safety games. We conclude by mentioning that safety reducibility of parity games was used implicitly to construct an algorithm for parity games [14] and to compute permissive strategies for parity games [1]. Furthermore, the safety reducibility of co-Büchi games is used implicitly in work on bounded synthesis [20] and LTL realizability [10], so-called “Safraless” constructions [16] which the do not rely on determinization of automata on infinite words.

Furthermore, the new notion of reduction allows to generalize permissiveness to all games discussed in Example 3: if a game is safety reducible to L, then we can construct a multi-strategy that allows every play ρ in which Player 1 cannot leave L starting from any prefix of ρ. Thereby, we obtain what one could call L-permissive strategies. For example, this allows to construct the most general non-deterministic winning strategy in a request-response game that guarantees a fixed bound on the waiting times.

7 Conclusion

We have shown how to translate a Muller game into a safety game to determine both winning regions and a finite-state winning strategy for one player. Then, we generalized this construction to a new type of reduction from infinite games to safety games with the same properties. We exhibited several implicit applications of this reduction in the literature as well as several new ones. Our reduction from Muller games to safety games is implemented in the tool `GAVS+`[6] [6]. In the future, we want to compare the performance of our solution to classical reductions to parity games as well as to direct algorithms.

Our construction is based on the notion of scoring functions for Muller games. Considering the maximal score the opponent can achieve against a strategy leads to a hierarchy of all finite-state strategies for a given game. Previous work has shown that the third level of this hierarchy is always non-empty, and there are games in which the second level is empty. Currently, there is no non-trivial characterization of the games whose first or second level of the hierarchy is non-empty, respectively.

The quality of a strategy can be measured by its level in the hierarchy. We conjecture that there is always a finite-state winning strategy of minimal size in the least non-empty level of this hierarchy, i.e., there is no tradeoff between size and quality of a strategy. This tradeoff may arise in many other games for which a quality measure is defined. Also, a positive resolution of the conjecture would decrease the search space for a smallest finite-state strategy.

We used scores to construct a novel antichain-based memory structure for Muller games. The antichain is induced by a winning strategy for the safety game. It is open how the choice of such a strategy influences the size of the memory structure and how heuristic approaches to computing winning strategies that only visit a small part of the arena [19] influence the performance of our reduction.

Finally, there is a tight connection between permissive strategies, progress measure algorithms, and safety reductions for parity games: the progress measure algorithm due to Jurdziński [14] and the reduction from parity games to safety game due to Bernet et al. [1] to compute permissive strategies are essentially the same. Whether the safety reducibility of Muller games can be turned into a progress measure algorithm is subject to ongoing research.

Acknowledgments The authors want to thank Wolfgang Thomas for bringing McNaughton's work to their attention, Wladimir Fridman for fruitful discussions, and Chih-Hong Cheng for his implementation of the algorithm.

References

[1] Julien Bernet, David Janin & Igor Walukiewicz (2002): *Permissive strategies: from parity games to safety games*. *ITA* 36(3), pp. 261–275, doi:`10.1051/ita:2002013`.

[2] Roderick Bloem, Krishnendu Chatterjee, Thomas A. Henzinger & Barbara Jobstmann (2009): *Better Quality in Synthesis through Quantitative Objectives*. In Ahmed Bouajjani & Oded Maler, editors: *CAV*, *LNCS* 5643, Springer, pp. 140–156, doi:`10.1007/978-3-642-02658-4_14`.

[3] Patricia Bouyer, Nicolas Markey, Jörg Olschewski & Michael Ummels (2011): *Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited*. In Bultan & Hsiung [4], pp. 135–149, doi:`10.1007/978-3-642-24372-1_11`.

[4] Tevfik Bultan & Pao-Ann Hsiung, editors (2011): *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*. *Lecture Notes in Computer Science* 6996, Springer, doi:`10.1007/978-3-642-24372-1`.

[6] See `http://www6.in.tum.de/~chengch/gavs/` for details and to download the tool.

[5] Pavol Černý, Krishnendu Chatterjee, Thomas A. Henzinger, Arjun Radhakrishna & Rohit Singh (2011): *Quantitative Synthesis for Concurrent Programs*. In Ganesh Gopalakrishnan & Shaz Qadeer, editors: *CAV*, *LNCS* 6806, Springer, pp. 243–259, doi:10.1007/978-3-642-22110-1_20.

[6] Chih-Hong Cheng, Alois Knoll, Michael Luttenberger & Christian Buckl (2011): *GAVS+: An Open Platform for the Research of Algorithmic Game Solving*. In Parosh Aziz Abdulla & K. Rustan M. Leino, editors: *TACAS*, *LNCS* 6605, Springer, pp. 258–261, doi:10.1007/978-3-642-19835-9_22.

[7] Stefan Dziembowski, Marcin Jurdziński & Igor Walukiewicz (1997): *How Much Memory is Needed to Win Infinite Games?* In: *LICS*, IEEE Computer Society, pp. 99–110, doi:10.1109/LICS.1997.614939.

[8] Ruediger Ehlers (2011): *Small witnesses, accepting lassos and winning strategies in omega-automata and games*. *CoRR* abs/1108.0315. Available at http://arxiv.org/abs/1108.0315.

[9] John Fearnley & Martin Zimmermann (2012): *Playing Muller Games in a Hurry*. *Int. J. Found. Comput. Sci.* 23(3), pp. 649–668, doi:10.1142/S0129054112400321.

[10] Emmanuel Filiot, Naiyong Jin & Jean-François Raskin (2011): *Antichains and compositional algorithms for LTL synthesis*. *Formal Methods in System Design* 39(3), pp. 261–296, doi:10.1007/s10703-011-0115-3.

[11] Yuri Gurevich & Leo Harrington (1982): *Trees, Automata, and Games*. In: *STOC*, ACM, pp. 60–65.

[12] Florian Horn (2008): *Explicit Muller Games are PTIME*. In Ramesh Hariharan, Madhavan Mukund & V. Vinay, editors: *FSTTCS*, *LIPIcs* 2, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 235–243, doi:10.4230/LIPIcs.FSTTCS.2008.1756.

[13] Florian Horn, Wolfgang Thomas & Nico Wallmeier (2008): *Optimal Strategy Synthesis in Request-Response Games*. In Sung Deok Cha, Jin-Young Choi, Moonzoo Kim, Insup Ldoi & Mahesh Viswanathan, editors: *ATVA*, *LNCS* 5311, Springer, pp. 361–373, doi:10.1007/978-3-540-88387-6_31.

[14] Marcin Jurdziński (2000): *Small Progress Measures for Solving Parity Games*. In Horst Reichel & Sophie Tison, editors: *STACS*, *LNCS* 1770, Springer, pp. 290–301, doi:10.1007/3-540-46541-3_24.

[15] Alexander Kechris (1995): *Classical Descriptive Set Theory*. *Graduate Texts in Mathematics* 156, Springer.

[16] Orna Kupferman & Moshe Y. Vardi (2005): *Safraless Decision Procedures*. In: *FOCS*, IEEE Computer Society, pp. 531–542, doi:10.1109/SFCS.2005.66.

[17] Robert McNaughton (1993): *Infinite Games Played on Finite Graphs*. *Ann. Pure Appl. Logic* 65(2), pp. 149–184, doi:10.1016/0168-0072(93)90036-D.

[18] Robert McNaughton (2000): *Playing Infinite Games in Finite Time*. In Arto Salomaa, Derick Wood & Sheng Yu, editors: *A Half-Century of Automata Theory*, World Scientific, pp. 73–91.

[19] Daniel Neider (2011): *Small Strategies for Safety Games*. In Bultan & Hsiung [4], pp. 306–320, doi:10.1007/978-3-642-24372-1_22.

[20] Sven Schewe & Bernd Finkbeiner (2007): *Bounded Synthesis*. In Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino & Yoshio Okamura, editors: *ATVA*, *LNCS* 4762, Springer, pp. 474–488, doi:10.1007/978-3-540-75596-8_33.

[21] Nico Wallmeier, Patrick Hütten & Wolfgang Thomas (2003): *Symbolic Synthesis of Finite-State Controllers for Request-Response Specifications*. In Oscar H. Ibarra & Zhe Dang, editors: *CIAA*, *LNCS* 2759, Springer, pp. 11–22, doi:10.1007/3-540-45089-0_3.

[22] Wiesław Zielonka (1998): *Infinite Games on Finitely Coloured Graphs with Applications to Automata on Infinite Trees*. *Theor. Comput. Sci.* 200(1-2), pp. 135–183, doi:10.1016/S0304-3975(98)00009-7.

[23] Martin Zimmermann (2009): *Time-Optimal Winning Strategies for Poset Games*. In Sebastian Maneth, editor: *CIAA*, *LNCS* 5642, Springer, pp. 217–226, doi:10.1007/978-3-642-02979-0_25.