RWTH Aachen

Department of Computer Science
Technical Report

Solving Muller Games via Safety Games

Daniel Neider, Roman Rabinovich, and Martin Zimmermann

ISSN 0935–3232 · Aachener Informatik-Berichte · AIB-2011-14

RWTH Aachen · Department of Computer Science · July 2011

Solving Muller Games via Safety Games*

Daniel Neider[1], Roman Rabinovich[2], and Martin Zimmermann[1]

[1] Lehrstuhl für Informatik 7, RWTH Aachen University, Germany
{neider,zimmermann}@automata.rwth-aachen.de
[2] Mathematische Grundlagen der Informatik, RWTH Aachen University, Germany
rabinovich@logic.rwth-aachen.de

Abstract. We show how to 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 a winning strategy for one player.

1 Introduction

Infinite two-player games are a powerful tool in the automated verification and synthesis of non-terminating systems that have to interact with an antagonistic environment. There are also deep connections between infinite games and logical formalisms like fixed-point logics or automata on infinite objects. In such a game, two players move a token through a finite graph, thereby constructing a play which is an infinite path. The winner is determined by a winning condition, which partitions the set of infinite paths in a graph into those that are winning for Player 0 and those that are winning for Player 1. Typically, the winner of a play is only determined after infinitely many steps.

Nevertheless, in some cases it is possible to give a criterion to define a finite-duration variant of an infinite game. Such a criterion stops a play after a finite number of steps and then declares a winner based on the finite play constructed thus far. It is called sound if Player 0 has a winning strategy for the infinite-duration game if and only if Player 0 has one for the finite-duration game.

It is easy to see that there is a sound criterion for positionally determined games: the players move the token through the arena until a vertex is visited for the second time. An infinite play can then be obtained by assuming that the players continue to play the loop that they have constructed, and the winner of the finite play is declared to be the winner of this infinite continuation.

For parity games (say, min-parity), Bernet, Janin, and Walukiewicz [1] gave another sound criterion based on the following observation: let n_c be the number of vertices with priority c. If a play visits $n_c + 1$ vertices with odd priority c without visiting a smaller even priority in between, then the play has closed a loop which is losing for Player 0, assuming it is traversed from now on ad infinitum. However, no positional winning strategy can allow such a loop. Thus, Player 0 can prove that she has a winning strategy by allowing a play to visit an odd priority c at most n_c times without seeing a smaller even priority in between. This condition can be turned into a safety game whose solution allows to determine the winning regions of the parity game and a winning strategy for one of the players.

* 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*.

In games that are not positionally determined, the situation gets more interesting since a player might have to pick different successors when a vertex is visited several times. Therefore, the players have to play longer before the play can be stopped and analyzed. Previous work considers Muller games which are of the form $(\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$, where $\mathcal{A}$ is a finite arena and $(\mathcal{F}_0, \mathcal{F}_1)$ is a partition of the set of loops in the arena. Player i wins a play if the set of vertices visited infinitely often is in $\mathcal{F}_i$. Muller winning conditions are able to express all ω-regular winning conditions and subsume all other winning conditions that depend only on the infinity set of a play (e.g., Büchi, co-Büchi, parity, Rabin, or Streett conditions).

To give a sound criterion for Muller games, McNaughton [7] defined for every loop $F \in \mathcal{F}_0 \cup \mathcal{F}_1$ a scoring function Sc_F that keeps track of the number of times the set F was visited entirely (not necessarily in the same order) since the last visit of a vertex that is not in F. In an infinite play, the set of vertices seen infinitely often is the unique set F such that Sc_F tends to infinity after being reset to 0 only a finite number of times.

McNaughton proved the following criterion to be sound [7]: stop a play as soon as for some set F a score of $|F|! + 1$ is reached, and declare the winner to be the Player i such that $F \in \mathcal{F}_i$. However, it can take a large number of steps for a play to reach a score of $|F|! + 1$, as scores may increase slowly or be reset to 0. It can be shown that a play can be stopped by this criterion after at most $\prod_{j=1}^{|\mathcal{A}|}(j!+1)$ steps and there are examples in which it takes at least $\frac{1}{2}\prod_{j=1}^{|\mathcal{A}|}(j!+1)$ steps before the criterion declares a winner.

Also, a game reduction from Muller games to parity games provides another sound criterion. The reduction constructs a parity game of size $|\mathcal{A}| \cdot |\mathcal{A}|!$, and since parity games are positionally determined, a winner can be declared after the players have constructed a loop in the parity game. This gives a sound criterion that stops a play after at most $|\mathcal{A}| \cdot |\mathcal{A}|! + 1$ steps.

Both results were improved by showing that stopping a play after a score of 3 is reached for the first time is sound [2]. This criterion stops a play after at most $3^{|\mathcal{A}|}$ steps, and there are examples where this number of steps is necessary. The result is proven by constructing a winning strategy for Player i that bounds the opponent's scores by 2, provided the play starts in the winning region of Player i. Such a strategy ensures that Player i is the first to achieve a score of 3, as not all scores can be bounded. Thus, to determine the winner of a Muller game, it suffices to solve a finite reachability game in a tree of height $3^{|\mathcal{A}|}$.

However, this game only allows to determine the winner, but does not yield winning strategies, as each play ends after a bounded number of steps. We overcome this drawback by exploiting the existence of strategies that bound the losing player's scores. This implies that the winner of a Muller game can also be determined by solving a safety game. In this game, the scores of Player 1 are kept track of and Player 0 wins, if her opponent never reaches a score of 3. In this work, we analyze this safety game and show that one can turn the winning region of the player that has to bound the scores of her opponent into a finite-state winning strategy for her in the Muller game.

The size of the resulting safety game (and, thus, also the size of the finite-state winning strategy) is at most $(|\mathcal{A}|!)^3$. This is only polynomially larger than the parity game of size $|\mathcal{A}| \cdot |\mathcal{A}|!$ constructed in the game reduction mentioned above. Although our safety game is polynomially larger than the parity game, it is simpler and faster to solve than the latter.

The scores induce a partial order on the positions of the safety game. We also prove that it suffices to consider the maximal elements of this order to define a finite-state winning strategy for the player that tries to bound the scores of her opponent. This antichain approach is subject to further research that should estimate how much smaller this finite-state winning strategy can be.

We want to stress that our construction is not a proper game reduction, which would provide winning strategies no matter which player wins. Here, we only obtain a winning strategy for the player trying to avoid a score of 3. If the opponent is able to reach a score of 3, then the play stops immediately. Thus, not every play in the Muller game has a corresponding play in the safety game, as it is required in a game reduction. In fact, a game reduction from Muller games to safety or reachability games is impossible, as it would induce a continuous function mapping the winning plays of the Muller game to the winning plays of a safety or reachability game. Such a mapping cannot exist, since the set of winning plays of a Muller game is on a higher level of the Borel hierarchy than the set of winning plays of a safety or reachability game.

The remainder of this report is structured as follows: in Section 2 we introduce our notation, and in Section 3 we define the scoring functions for Muller games. Then, in Section 4 we show how to solve a Muller game (i.e., how to determine the winning regions and compute a winning strategy) by solving a safety game. In this context, we present an alternative way to compute a winning strategy based on antichains in Section 4.1 and discuss how to reduce the number of memory states needed to define a winning strategy in Section 4.2. Finally, Section 5 contains a brief conclusion.

2 Definitions

The power set of a set S is denoted by 2^S and $\mathbb{N}$ denotes the non-negative integers. The prefix relation on words is denoted by $\sqsubseteq$. Given a word $w = xy$, define $wy^{-1} = x$. For a non-empty word $w = w_1 \cdots w_n$, we define $\mathrm{Last}(w) = w_n$.

An arena $\mathcal{A} = (V, V_0, V_1, E)$ consists of a finite, directed graph (V, E) without terminal vertices and a partition $\{V_0, V_1\}$ of V denoting the positions of Player 0 (drawn as circles or rectangles with rounded corners) and Player 1 (drawn as squares or rectangles). We require every vertex to have an outgoing edge to avoid the nuisance of dealing with finite plays. The size $|\mathcal{A}|$ of $\mathcal{A}$ is the cardinality of V. A loop $C \subseteq V$ in $\mathcal{A}$ 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 safety game $\mathcal{G} = (\mathcal{A}, F)$ consists of an arena $\mathcal{A}$ and a set $F \subseteq V$. A Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ consists of an arena $\mathcal{A}$ and a partition $\{\mathcal{F}_0, \mathcal{F}_1\}$ of the set of loops in $\mathcal{A}$.

A play in $\mathcal{A}$ starting in $v \in V$ is an infinite sequence $\rho = \rho_0\rho_1\rho_2 \ldots$ 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 w, which is defined straightforwardly. The infinity set of a play is always a loop in the arena.

A play ρ is winning for Player 0 in a safety game if $\mathrm{Occ}(\rho) \subseteq F$, and it is winning for Player 0 in a Muller game if $\mathrm{Inf}(\rho) \in \mathcal{F}_0$. A play in any game is winning for Player 1 if it is not winning for Player 0, i.e., ρ leaves F in case of a safety game or $\mathrm{Inf}(\rho) \in \mathcal{F}_1$ in case of a Muller game.

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 $\rho_n \in V_i$. A strategy σ for Player i is a winning strategy from a set of vertices $W \subseteq V$ if every play that starts in $v \in W$ and is consistent with σ is won by Player i. The winning region $W_i(\mathcal{G})$ of Player i in a game $\mathcal{G}$ contains all vertices of the game's arena from which Player i has a winning strategy. A game is determined if $\{W_0(\mathcal{G}), W_1(\mathcal{G})\}$ forms a partition of V.

A memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd}, \mathrm{Nxt})$ for Player i in (V, V_0, V_1, E) consists of a finite set of memory states M, a memory initialization function $\mathrm{Init}\colon V \to M$, a memory update function $\mathrm{Upd}\colon M \times V \to M$, and a next-move function $\mathrm{Nxt}\colon V_i \times M \to V$, which has to satisfy $(v, \mathrm{Nxt}(v, m)) \in E$ for every v and every m. Upd can be extended to finite plays by defining $\mathrm{Upd}^*(v) = \mathrm{Init}(v)$ and $\mathrm{Upd}^*(wv) = \mathrm{Upd}(\mathrm{Upd}^*(w), v)$. The memory structure induces a strategy $\sigma_{\mathfrak{M}}$ for Player i via $\sigma_{\mathfrak{M}}(wv) = \mathrm{Nxt}(v, \mathrm{Upd}^*(wv))$. The size of $\mathfrak{M}$ (and, slightly abusive, of $\sigma_{\mathfrak{M}}$) is $|M|$. We say that a strategy is finite-state if it can be implemented using a memory structure.

An arena $\mathcal{A}$ and a memory structure $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$ without next-move function induce the expanded arena $\mathcal{A} \times \mathfrak{M} = (V \times M, V_0 \times M, V_1 \times M, E')$ where $((s, m), (s', m')) \in E'$ if and only if $(s, s') \in E$ and $\mathrm{Upd}(m, s') = m'$. For every play $\rho = \rho_0\rho_1\rho_2\dots$ in $\mathcal{A}$ define the extended play $\rho' = (\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\dots$ in $\mathcal{A} \times \mathfrak{M}$ by $m_0 = \mathrm{Init}(\rho_0)$ and $m_{n+1} = \mathrm{Upd}(m_n, \rho_{n+1})$.

A game $\mathcal{G}$ with arena $\mathcal{A}$ is reducible to a game $\mathcal{G}'$ with arena $\mathcal{A}'$ via $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$, written $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$, if $\mathcal{A}' = \mathcal{A} \times \mathfrak{M}$ and every play ρ in $\mathcal{G}$ is won by the player who wins the extended play ρ' in $\mathcal{G}'$.

Lemma 1. *Let $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd})$. If $\mathcal{G} \leq_{\mathfrak{M}} \mathcal{G}'$ and Player i has a positional winning strategy σ for $\mathcal{G}'$, then she also has a finite-state winning strategy induced by a memory structure $(M, \mathrm{Init}, \mathrm{Upd}, \mathrm{Nxt})$ for $\mathcal{G}$, where* Nxt *is a suitable next-move function induced by σ.*

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 $\mathrm{win_S} \subseteq V'^\omega$ of winning plays of a safety game. Hence, in general, there exists no continuous (in the Cantor topology) function $f\colon V^\omega \to V'^\omega$ such that $\rho \in \mathrm{win_M}$ if and only if $f(\rho) \in \mathrm{win_S}$ (e.g., see [5]). Since the mapping from a play in $\mathcal{A}$ to a play in $\mathcal{A} \times \mathfrak{M}$ is continuous, one obtains the following impossibility result.

Remark 1. In general, Muller games can not be reduced to safety games.

Let $\mathcal{A} = (V, V_0, V_1, E)$ be an arena. The attractor for Player i of a set $F \subseteq V$ in $\mathcal{A}$ is $\mathrm{Attr}_i^{\mathcal{A}}(F) = \bigcup_{n=0}^{|V|} A_n$ where $A_0 = F$ and

$$A_{n+1} = A_n \cup \{v \in V_i \mid \exists v' \in A_n \text{ such that } (v, v') \in E\} \\ \cup \{v \in V_{1-i} \mid \forall v' \in V \text{ with } (v, v') \in E : v' \in A_n\} \ .$$

A set $X \subseteq V$ is a trap for Player i if all outgoing edges of the vertices in $V_i \cap X$ lead to X and at least one successor of every vertex in $V_{1-i} \cap X$ is in X, i.e., Player $1-i$ has a positional strategy to keep a play in X once it has entered the trap. The following statement summarizes well-known facts about safety games.

Lemma 2. *Let $\mathcal{A}$ be an arena with vertex set V and $F \subseteq V$.*

1. *Player i has a positional strategy to bring the play from every $v \in \mathrm{Attr}_i^{\mathcal{A}}(F)$ into F.*
2. *The set $V \setminus \mathrm{Attr}_i^{\mathcal{A}}(F)$ is a trap for Player i in $\mathcal{A}$.*

A strategy as in the first statement is called attractor strategy. The previous lemma directly implies that $W_1(\mathcal{G}) = \mathrm{Attr}_1^{\mathcal{A}}(V \setminus F)$ and $W_0(\mathcal{G}) = V \setminus W_1(\mathcal{G})$ are the winning regions in the safety game $\mathcal{G} = (\mathcal{A}, F)$. Thus, safety games are determined with positional strategies.

Theorem 1 ([4]). *Muller games are determined with finite-state strategies of size $|\mathcal{A}| \cdot |\mathcal{A}|!$.*

3 Scoring Functions for Muller Games

We begin with some definitions and facts about scoring functions for Muller games. A more detailed treatment can be found in [2, 7].

Let V be a set of vertices. For every $F \subseteq V$ we define $\mathrm{Sc}_F \colon V^+ \to \mathbb{N}$ by

$$\mathrm{Sc}_F(w) = \max\{k \in \mathbb{N} \mid \exists x_1, \ldots, x_k \in V^+ \text{ such that } \mathrm{Occ}(x_i) = F \text{ for all } i \text{ and } x_1 \cdots x_k \text{ is a suffix of } w\} \enspace .$$

The score of F of a play 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. Note that if w is a play with $\mathrm{Sc}_F(w) \geq 2$, then F is a loop of the arena.

Next, we define the accumulator of a set F, which measures the progress made towards the next score increase. For every $F \subseteq V$, we define $\mathrm{Acc}_F \colon V^+ \to 2^F$ by $\mathrm{Acc}_F(w) = \mathrm{Occ}(x)$, where x is the longest suffix of w such that $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(wy^{-1})$ for every suffix y of x, and $\mathrm{Occ}(x) \subseteq F$. Intuitively, $\mathrm{Acc}_F(w)$ contains the vertices of F seen since the last increase or the last reset of the score of F, depending on which occurred later. Hence, the accumulator of a set F is always a strict subset of F.

Let us remark that the scores and accumulators of a play can be defined (and computed) inductively as well.

Remark 2 (cf. [7]). Let $w \in V^+$, $v \in V$, and $\emptyset \neq F \subseteq V$.

1. We have $\mathrm{Sc}_{\{v\}}(v) = 1$ and $\mathrm{Acc}_{\{v\}}(v) = \emptyset$, and for every $F \neq \{v\}$: $\mathrm{Sc}_F(v) = 0$ and $\mathrm{Acc}_F(v) = F \cap \{v\}$.
2. Let $v \notin F$. Then we have $\mathrm{Sc}_F(wv) = 0$ and $\mathrm{Acc}_F(wv) = \emptyset$.
3. Let $v \in F$. If $\mathrm{Acc}_F(w) = F \setminus \{v\}$, then we have $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1$ and $\mathrm{Acc}_F(wv) = \emptyset$.
4. Let $v \in F$. If $\mathrm{Acc}_F(w) \neq F \setminus \{v\}$, then we have $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w)$ and $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w) \cup \{v\}$.

Finally, for every $\mathcal{F} \subseteq 2^V$, we define $\mathrm{MaxSc}_\mathcal{F}\colon V^+ \cup V^\omega \to \mathbb{N} \cup \{\infty\}$ by

$$\mathrm{MaxSc}_\mathcal{F}(\rho) = \max_{F \in \mathcal{F}} \max_{w \sqsubseteq \rho} \mathrm{Sc}_F(w) \enspace .$$

Example 1. Consider the Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ where $\mathcal{A}$ is depicted in Figure 1, $\mathcal{F}_0 = \{\{0\}, \{2\}, \{0, 1, 2\}\}$ and $\mathcal{F}_1 = \{\{0, 1\}, \{1, 2\}\}$. By alternatingly moving from 1 to 0 and to 2, Player 0 wins from every vertex, i.e., we have $W_0(\mathcal{G}) = \{0, 1, 2\}$.

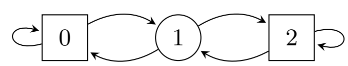

Fig. 1. The arena $\mathcal{A}$.

To illustrate the definitions, consider the play $w = 12210122$ and the set $F = \{1, 2\}$. We have that $\mathrm{Sc}_F(w) = 1$, because 122 is the longest suffix of w that is contained in F, and the entire set $\{1, 2\}$ is seen once during this suffix. We have $\mathrm{Acc}_F(w) = \{2\}$, because only vertex 2 has been seen since the score of F increased to 1. On the other hand, we have $\mathrm{MaxSc}_{\{F\}}(w) = 2$ because the prefix $w' = 1221$ of w has $\mathrm{Sc}_F(w') = 2$. By visiting the vertex 0 the score of F is reset to 0, e.g., we have $\mathrm{Sc}_F(12210) = 0$.

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. This implies that every play ρ of a Muller game satisfying $\mathrm{MaxSc}_{\mathcal{F}_{1-i}}(\rho) < \infty$ is winning for Player i.

We continue by giving a score-based preorder and an induced score-based equivalence relation on finite plays in a Muller game.

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

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

Note that the condition $w =_\mathcal{F} w'$ is equivalent to $\mathrm{Last}(w) = \mathrm{Last}(w')$ and for every $F \in \mathcal{F}$ the equalities $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ and $\mathrm{Acc}_F(w) = \mathrm{Acc}_F(w')$ hold. Thus, $=_\mathcal{F}$ is an equivalence relation.

We conclude this section by showing that $\leq_\mathcal{F}$ (and therefore also $=_\mathcal{F}$) is preserved under concatenation, i.e., $=_\mathcal{F}$ is a congruence.

Lemma 3. *If $w \leq_\mathcal{F} w'$, then $wu \leq_\mathcal{F} w'u$ for all $u \in V^*$.*

Proof. It suffices to show $w \leq_\mathcal{F} w'$ implies $wv \leq_\mathcal{F} w'v$ for all $v \in V$. So, let $F \in \mathcal{F}$: if $v \notin F$, then we have $\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w'v) = 0$ and $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w'v) = \emptyset$.

Now, suppose we have $v \in F$. First, consider the case $\mathrm{Sc}_F(w) < \mathrm{Sc}_F(w')$: then, either the score of F does not increase in wv and we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) < \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v)$$

or the score increases in wv and we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1 \leq \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v)$$

and $\mathrm{Acc}_F(wv) = \emptyset$, due to the score increase. This proves our claim.

Now, consider the case $\mathrm{Sc}_F(w) = \mathrm{Sc}_F(w')$ and $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$. If $\mathrm{Acc}_F(w) = F \setminus \{v\}$, then also $\mathrm{Acc}_F(w') = F \setminus \{v\}$, as the accumulator for F can never be F. In this situation, we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) + 1 = \mathrm{Sc}_F(w') + 1 = \mathrm{Sc}_F(w'v)$$

and $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w'v) = \emptyset$. Otherwise, we have

$$\mathrm{Sc}_F(wv) = \mathrm{Sc}_F(w) = \mathrm{Sc}_F(w') \leq \mathrm{Sc}_F(w'v) \ .$$

If $\mathrm{Sc}_F(w') < \mathrm{Sc}_F(w'v)$, then we are done. So, consider the case $\mathrm{Sc}_F(w') = \mathrm{Sc}_F(w'v)$: we have $\mathrm{Acc}_F(wv) = \mathrm{Acc}_F(w) \cup \{v\} \subseteq \mathrm{Acc}_F(w') \cup \{v\} = \mathrm{Acc}_F(w'v)$, due to $\mathrm{Acc}_F(w) \subseteq \mathrm{Acc}_F(w')$. □

Corollary 1. *If $w =_{\mathcal{F}} w'$, then $wu =_{\mathcal{F}} w'u$ 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 following theorem, which shows the existence of winning strategies for Muller games that bound the opponent's scores by 2.

Theorem 2 ([2]). *In every Muller game $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$, Player i has a winning strategy σ from $W_i(\mathcal{G})$ such that $\mathrm{MaxSc}_{\mathcal{F}_{1-i}}(\rho) \leq 2$ for every play ρ that is consistent with σ and begins in $W_i(\mathcal{G})$.*

Going back to the Muller game $\mathcal{G}$ of Example 1, it is clear that Player 0 has no winning strategy from the vertex $1 \in W_0(\mathcal{G})$ that bounds Player 1's scores by 1, since the prefix 1001 or the prefix 1221 is consistent with every strategy for Player 0 from vertex 1. Hence, the bound 2 above is optimal.

A simple consequence of Theorem 2 is that a vertex v is in Player 0's winning region of the Muller game $\mathcal{G}$ if and only if she can prevent her opponent from ever reaching a score of 3 for a set in $\mathcal{F}_1$. This is a safety condition which only talks about small scores of one player. To determine the winner of $\mathcal{G}$, 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 3 (Main theorem). *Let $\mathcal{G}$ be a Muller game with vertex set V. One can effectively construct a safety game $\mathcal{G}_S$ with vertex set V^S and a mapping $f \colon V \to V^S$ with the following properties:*

1. *For every $v \in V$: $v \in W_i(\mathcal{G})$ if and only if $f(v) \in W_i(\mathcal{G}_S)$.*
2. *Player 0 has a finite-state winning strategy from $W_0(\mathcal{G})$ with memory states $W_0(\mathcal{G}_S)$.*
3. *$|V^S| \leq (|V|!)^3$.*

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, one has to swap the roles of the players and construct a safety game which keeps track of the scores of Player 0. Alternatively, one could construct an arena which keeps track of the scores of both players. But then, one has 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 (but still smaller than $(|V|!)^3$) than the ones in which only the scores of one player are tracked. Due to Remark 1, it is impossible to reduce a Muller game to a single safety game and thereby obtain a winning strategy for both players.

We begin the proof of Theorem 3 by defining the safety game $\mathcal{G}_S$. Then, we prove two lemmata that imply the three statements above. Let $\mathcal{G} = (\mathcal{A}, \mathcal{F}_0, \mathcal{F}_1)$ with $\mathcal{A} = (V, V_0, V_1, E)$. We define

$$\mathrm{Plays}_{\leq 2} = \{w \mid w \text{ finite play of } \mathcal{G} \text{ and } \mathrm{MaxSc}_{\mathcal{F}_1}(w) \leq 2\}$$

to be the set of finite plays of the Muller game in which the scores of Player 1 are at most 2 and we define

$$\begin{aligned}\mathrm{Plays}_{=3} = \{w_0 \cdots w_n w_{n+1} \mid\ & w_0 \cdots w_n w_{n+1} \text{ finite play of } \mathcal{G},\\ & \mathrm{MaxSc}_{\mathcal{F}_1}(w_0 \cdots w_n) \leq 2, \text{ and } \mathrm{MaxSc}_{\mathcal{F}_1}(w_0 \cdots w_n w_{n+1}) = 3\ \}\end{aligned}$$

to be the set of finite plays in which Player 1 just reached a score of 3. Furthermore, let $\mathrm{Plays}_{\leq 3} = \mathrm{Plays}_{\leq 2} \cup \mathrm{Plays}_{=3}$.

The arena of the safety game we are about to define is the unraveling of $\mathcal{A}$ (modulo $=_{\mathcal{F}_1}$) up to the positions where Player 1 reaches a score of 3 for the first time (if he does at all).

We define $\mathcal{G}_S = ((V^S, V_0^S, V_1^S, E^S), F)$ where

- $V^S = \mathrm{Plays}_{\leq 3} /_{=_{\mathcal{F}_1}}$,
- $V_0^S = \{[w]_{=_{\mathcal{F}_1}} \mid [w] \in V^S \text{ and } \mathrm{Last}(w) \in V_0\}$,
- $V_1^S = \{[w]_{=_{\mathcal{F}_1}} \mid [w] \in V^S \text{ and } \mathrm{Last}(w) \in V_1\}$,
- $([w]_{=_{\mathcal{F}_1}}, [wv]_{=_{\mathcal{F}_1}}) \in E^S$ for all $w \in \mathrm{Plays}_{\leq 2}$ and all v with $(\mathrm{Last}(w), v) \in E$ [3],
- $F = \mathrm{Plays}_{\leq 2} /_{=_{\mathcal{F}_1}}$.

The definitions of V_0^S and V_1^S are independent of representatives, as $w =_{\mathcal{F}_1} w'$ implies $\mathrm{Last}(w) = \mathrm{Last}(w')$, and we have $V^S = V_0^S \cup V_1^S$ due to $V = V_0 \cup V_1$. Furthermore, every equivalence class in $\mathrm{Plays}_{\leq 2} /_{=_{\mathcal{F}_1}}$ is also an equivalence class in $\mathrm{Plays}_{\leq 3} /_{=_{\mathcal{F}_1}}$, i.e., F is well-defined.

Remark 3. If $([w]_{=_{\mathcal{F}_1}}, [w']_{=_{\mathcal{F}_1}}) \in E^S$, then $(\mathrm{Last}(w), \mathrm{Last}(w')) \in E$.

Example 2. The safety game $\mathcal{G}_S$ for the Muller game $\mathcal{G}$ of Example 1 is depicted in Figure 2. One can verify easily that the vertices $[v]$ for $v \in V$ are in the winning region of Player 0.

[3] Hence, every vertex in $\mathrm{Plays}_{=3}$ 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.

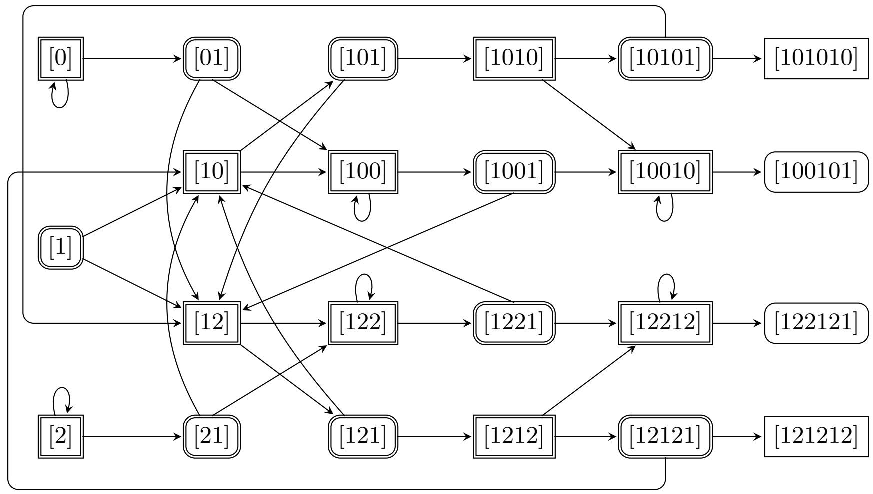

Fig. 2. The safety game $\mathcal{G}_S$ for $\mathcal{G}$ of Example 1. Vertices in F are drawn with double lines.

We are now able to prove the first statement of Theorem 3. For the sake of readability, we drop the subscripts and denote the $\mathcal{F}_1$-equivalence class of w from now on by $[w]$. Also, all definitions and statements below are independent of representatives, mostly since $w =_{\mathcal{F}_1} w'$ implies $\text{Last}(w) = \text{Last}(w')$. Hence, we refrain from mentioning independence of representatives from now on.

Lemma 4. *For every $v_0 \in V$: $v_0 \in W_i(\mathcal{G})$ if and only if $[v_0] \in W_i(\mathcal{G}_S)$.*

To show the direction from left to right, we turn a winning strategy that bounds Player 1's scores by 2 into a winning strategy for the safety game. For the other direction, we use the winning region of Player 0 in the safety game as a memory structure to implement a winning strategy in the Muller game. This is possible, since the winning region is a trap for Player 1. Both directions are straightforward, but slightly technical, as we have to deal with equivalence classes of plays.

Proof. Due to determinacy of both games, it suffices to consider $i = 0$.

We begin with the direction from left to right. Theorem 2 guarantees the existence of a strategy σ for Player 0 that bounds the scores of her opponent by 2 in every consistent play that starts in $W_0(\mathcal{G})$. We turn this strategy into a winning strategy for her in $\mathcal{G}_S$ from $\{[v_0] \mid v_0 \in W_0(\mathcal{G})\}$.

We track a play in $\mathcal{G}_S$ and a play in $\mathcal{G}$ simultaneously and translate moves of Player 1 in $\mathcal{G}_S$ to the play in $\mathcal{G}$ and moves of Player 0 in $\mathcal{G}$ to moves in $\mathcal{G}_S$. Formally, we construct a function g mapping finite plays in $\mathcal{G}_S$ starting in a vertex $[v_0]$ for some $v_0 \in V$ to finite plays in $\mathcal{G}$ starting in v_0. Then, we use the image of this function to turn σ into a strategy for $\mathcal{G}_S$. Let $g([v_0]) = v_0$ for every $v_0 \in V$ and

$$g([w_0] \cdots [w_n][w_{n+1}]) = g([w_0] \cdots [w_n]) \cdot \text{Last}(w_{n+1}) \ . \tag{1}$$

We have $\text{Last}(g([w_0] \cdots [w_n])) = \text{Last}(w_n)$ and applying Remark 3 repeatedly shows that $g([w_0] \cdots [w_n])$ is indeed a play in $\mathcal{G}$. Also, the image of a play $[w_0] \cdots [w_n]$ has the same scores and accumulators as w_n:

Lemma 5. $g([w_0]\cdots[w_n]) \in [w_n]$.

Proof. By induction over $[w_0]\cdots[w_n]$. The induction start is immediate due to $g([v_0]) = v_0 \in [v_0]$. So, consider a play $[w_0]\cdots[w_n][w_{n+1}]$. Since there is an edge from $[w_n]$ to $[w_{n+1}]$, we have $w_n \cdot \mathrm{Last}(w_{n+1}) \in [w_{n+1}]$. By induction hypothesis, we have $g([w_0]\cdots[w_n]) =_{\mathcal{F}_1} w_n$ and applying Corollary 1, we obtain

$$\begin{aligned}&g([w_0]\cdots[w_n][w_{n+1}])\\=&g([w_0]\cdots[w_n]) \cdot \mathrm{Last}(w_{n+1}) =_{\mathcal{F}_1} w_n \cdot \mathrm{Last}(w_{n+1}) =_{\mathcal{F}_1} w_{n+1} \ ,\end{aligned}$$

i.e., $g([w_0]\cdots[w_n][w_{n+1}]) \in [w_{n+1}]$. □

Now, we define the strategy σ_S for Player 0 from $\{[v_0] \mid v_0 \in W_0(\mathcal{G})\}$ in $\mathcal{G}_S$ by

$$\sigma_S([w_0]\cdots[w_n]) = [w_n \cdot \sigma(g([w_0]\cdots[w_n]))] \ ,$$

i.e., σ_S translates the play in $\mathcal{G}_S$ into a play in $\mathcal{G}$ and then uses the successor v prescribed by σ to determine the next equivalence class to move to by appending v to the current class. We show next that this is always a legal move, provided the play up to the current position is consistent with σ_S.

We show inductively that if $[w_0]\cdots[w_n]$ starting in some vertex $[v_0]$ for some $v_0 \in V$ is consistent with σ_S, then $g([w_0]\cdots[w_n])$ is consistent with σ and $\sigma_S([w_0]\cdots[w_n])$ describes a legal move in $\mathcal{G}_S$. This also implies that σ_S is a winning strategy from $\{[v_0] \mid v_0 \in W_0(\mathcal{G})\}$: assume a play $[w_0]\cdots[w_n]$ starting in $[v_0] \in \{[v_0] \mid v_0 \in W_0(\mathcal{G})\}$ consistent with σ_S leaves F by reaching $\mathrm{Plays}_{=3}$. This implies $\mathrm{MaxSc}_{\mathcal{F}_1}(g([w_0]\cdots[w_n])) = 3$ since we have $g([w_0]\cdots[w_n]) \in [w_n]$. Thus, $g([w_0]\cdots[w_n])$ starting in $v_0 \in W_0(\mathcal{G})$, being consistent with σ, and reaching a score of 3 contradicts the fact that σ prevents Player 1 from ever reaching a score of 3. Hence, σ_S is a winning strategy for Player 0 in $\mathcal{G}_S$ from $\{[v_0] \mid v_0 \in W_0(\mathcal{G})\}$.

Since the first statement is clear for the induction start, we only discuss the second one in detail: if $[v_0] \in V_0^S$, then also $v_0 \in V_0$ and we have

$$\sigma_S([v_0]) = [v_0 \cdot \sigma(g([v_0]))] = [v_0 \cdot \sigma(v_0)] \ .$$

Thus, $(v_0, \sigma(v_0)) \in E$ and since $[v_0], [v_0 \cdot \sigma(v_0)] \in \mathrm{Plays}_{\leq 2}$, we conclude also $([v_0], [v_0 \cdot \sigma(v_0)]) \in E^S$, i.e., σ_S indeed prescribes a legal move.

For the induction step, consider a play $[w_0]\cdots[w_{n-1}][w_n]$ that is consistent with σ_S and remember that we have

$$\mathrm{Last}(g([w_0]\cdots[w_{n-1}])) = \mathrm{Last}(w_{n-1}) \ . \tag{2}$$

By induction hypothesis, we can assume that $g([w_0]\cdots[w_{n-1}])$ is consistent with σ, hence, it only remains to consider the transition from $\mathrm{Last}(w_{n-1})$ to $\mathrm{Last}(w_n)$.

If $[w_{n-1}] \in V_0^S$, then also $\mathrm{Last}(g([w_0]\cdots[w_{n-1}])) \in V_0$ due to (2), and we have

$$[w_n] = \sigma_S([w_0]\cdots[w_{n-1}]) = [w_{n-1} \cdot \sigma(g([w_0]\cdots[w_{n-1}]))] \ .$$

Thus, we have $\mathrm{Last}(w_n) = \sigma(g([w_0]\cdots[w_{n-1}]))$. Applying this, the induction hypothesis, and (1) shows that

$$g([w_0]\cdots[w_{n-1}][w_n]) = g([w_0]\cdots[w_{n-1}]) \cdot \mathrm{Last}(w_n)$$

is indeed consistent with σ.

Now, consider the second statement. By definition, we have

$$\sigma_S([w_0]\cdots[w_n]) = [w_n \cdot \sigma(g([w_0]\cdots[w_n]))] \ ,$$

which implies that there is an edge between $\mathrm{Last}(w_n) = \mathrm{Last}(g([w_0]\cdots[w_n]))$ and $\sigma(g([w_0]\cdots[w_n]))$. Furthermore, $g([w_0]\cdots[w_n])$ is consistent with σ by induction hypothesis, and hence $g([w_0]\cdots[w_n])\cdot\sigma(g([w_0]\cdots[w_n]))$ as well. Since σ bounds the scores of Player 1 by 2, both of these finite plays are in $\mathrm{Plays}_{\leq 2}$ and therefore we can conclude that there is an edge in $\mathcal{G}_S$ between $[w_n]$ and $[w_n \cdot \mathrm{Last}(g([w_0]\cdots[w_n]))]$, which shows that σ_S indeed prescribes a legal move.

On the other hand, if $[w_{n-1}] \in V_1^S$, then $\mathrm{Last}(g([w_0]\cdots[w_{n-1}])) \in V_1$ due to (2), and we have $([w_{n-1}],[w_n]) \in E^S$. Hence, $(\mathrm{Last}(w_{n-1}), \mathrm{Last}(w_n)) \in E$ due to Remark 3. Since $\mathrm{Last}(g([w_0]\cdots[w_{n-1}])) = \mathrm{Last}(w_{n-1}) \in V_1$ and

$$g([w_0]\cdots[w_{n-1}][w_n]) = g([w_0]\cdots[w_{n-1}]) \cdot \mathrm{Last}(w_n) \ ,$$

$g([w_0]\cdots[w_{n-1}][w_n])$ is indeed consistent with σ.

For the other direction of Lemma 4, we show that $W_0(\mathcal{G}_S)$ can be turned into a memory structure for Player 0 in the Muller game that induces a winning strategy.

Example 3. Consider the winning region $W_0(\mathcal{G})$ in the safety game $\mathcal{G}_S$ of Example 2 as depicted in Figure 3 (for the sake of readability, we omit two vertices that are not reachable from a vertex $[v]$ for some $v \in V$). We obtain a finite-state winning strategy by using the equivalence class $[w]$ as memory state for a finite play w. Since the safety game is the unraveling of the original arena and its winning region is a trap for Player 1, Player 0 can always prolong a play in the Muller game such that the finite play prefixes w satisfy $[w] \in W_0(\mathcal{G}_S)$ no matter which successors Player 1 picks. This strategy also bounds the scores of Player 1 by two. Hence, it is winning for Player 0.

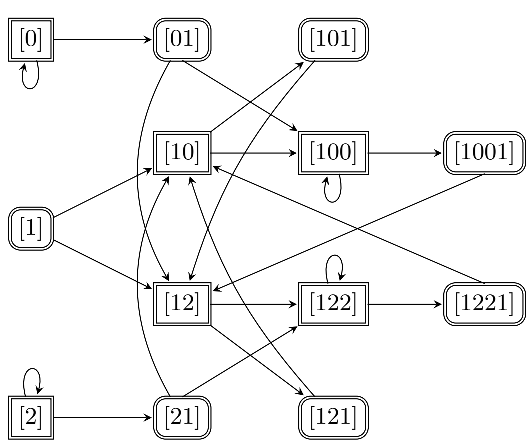

Fig. 3. The winning region $W_0(\mathcal{G}_S)$ of the safety game $\mathcal{G}_S$ of Example 2.

To simplify the proof, we add one more memory state $\perp$, denoting that a score of 3 was reached. As long as Player 0 sticks to the induced strategy, this memory state will not be reached. Hence, $\perp$ can be eliminated and its incoming transitions can be redefined arbitrarily.

Define $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd}, \mathrm{Nxt})$ by $M = W_0(\mathcal{G}_S) \cup \{\perp\}$,

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

and

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

Then, for every $w \in V^+$ with $\mathrm{Upd}^*(w) \neq \perp$ we have $\mathrm{Upd}^*(w) = [w]$. Furthermore, since M is the winning region of a safety game, every $[w] \in M \cap V_0^S$ has a successor $[wv']$ for some $v' \in V$ which is in $W_0(\mathcal{G}_S)$ as well. Remark 3 yields $(\mathrm{Last}(w), v') \in E$. Using this, we define the next-move function by

$$\mathrm{Nxt}(v, [w]) = \begin{cases} v' & \text{if } \mathrm{Last}(w) = v \text{ and } [wv'] \text{ as above,} \\ v'' & \text{otherwise, where } v'' \text{ is some vertex with } (v, v'') \in E, \end{cases}$$

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

Let $W = \{v \mid [v] \in W_0(\mathcal{G}_S)\}$. It remains to show that $\sigma_{\mathfrak{M}}$ is a winning strategy for Player 0 from W. A simple induction shows that every play w that starts in W and is consistent with $\sigma_{\mathfrak{M}}$ satisfies $\mathrm{Upd}^*(w) \neq \perp$, since the next-move function always prescribes a successor such that the memory is updated to a state in $W_0(\mathcal{G}_S)$. Similarly, Player 1 can only pick successors in $\mathcal{G}$ such that the memory is updated to a state in $W_0(\mathcal{G}_S)$, since the winning region of the safety game (which is the unraveling of the original game modulo $=_{\mathcal{F}_1}$) is a trap for him. Since we have $\mathrm{Upd}^*(w) = [w] \in \mathrm{Plays}_{\leq 2}$ for every play that starts in W and is consistent with σ, the scores of Player 1 are bounded by 2. Hence, $\sigma_{\mathfrak{M}}$ is indeed a winning strategy for Player 0 from W. □

The second direction of the proof above also proves the second statement of Theorem 3.

Corollary 2. *Player* 0 *has a finite-state winning strategy for* $W_0(\mathcal{G})$ *with memory states* $W_0(\mathcal{G}_S)$.

To finish the proof of Theorem 3, we determine the size of $\mathcal{G}_S$ to prove the third statement. To this end, we use the concept of a *latest appearance record* (LAR) [4, 6]. 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

$$\mathrm{LAR}(wv) = \begin{cases} \mathrm{LAR}(w)v & \text{if } v \notin \mathrm{Occ}(w), \\ p_1 p_2 v & \text{if } \mathrm{LAR}(w) = p_1 v p_2. \end{cases}$$

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 the LAR of a play w determines all but $|\operatorname{Occ}(w)|$ many of w's scores and accumulators.

Lemma 6. *Let $w \in V^+$ and $\mathrm{LAR}(w) = v_k v_{k-1} \cdots v_1$.*

1. *w can be decomposed into $x_k v_k x_{k-1} v_{k-1} \cdots v_2 x_1 v_1$ for some $x_i \in V^*$ with $\operatorname{Occ}(x_i) \subseteq \{v_1, \ldots, v_i\}$ for every i.*
2. *$\mathrm{Sc}_F(w) > 0$ if and only if $F = \{v_1, \ldots, v_i\}$ for some i.*
3. *If $\mathrm{Sc}_F(w) = 0$, then $\mathrm{Acc}_F(w) = \{v_1, \ldots, v_i\}$ for the maximal i such that $\{v_1, \ldots, v_i\} \subseteq F$.*
4. *Let $\mathrm{Sc}_F(w) > 0$ and $F = \{v_1, \cdots, v_i\}$. Then, $\mathrm{Acc}_F(w) \in \{\emptyset\} \cup \{\{v_1, \ldots, v_j\} \mid j < i\}$.*

Proof. 1.) By induction over $|w|$. If $|w| = 1$, then the claim follows immediately from $w = \mathrm{LAR}(w)$. Now, let $|wv| > 1$. If $v \notin \operatorname{Occ}(w)$, then $\mathrm{LAR}(wv) = \mathrm{LAR}(w)v$ and the claim follows by induction hypothesis.

Now, suppose $\mathrm{LAR}(w) = p_1 v p_2$ with $p_1 = v_k \cdots v_{i+1}$ and $p_2 = v_{i-1} \cdots v_1$, and hence $v_i = v$. By induction hypothesis, there exists a decomposition $w = x_k v_k x_{k-1} v_{k-1} \cdots v_2 x_1 v_1$ for some $x_i \in V^*$ such that $\operatorname{Occ}(x_i) \subseteq \{v_1, \ldots, v_i\}$ for every i. Furthermore, we have $\mathrm{LAR}(wv) = p_1 p_2 v = v'_k \cdots v'_1$ where $v'_1 = v_i$, $v'_j = v_{j-1}$ for every j in the range $1 < j \leq i$, and $v'_j = v_i$ for every j in the range $i < j \leq k$. Now, define $x'_1 = \varepsilon$, $x'_j = x_{j-1}$ for every j in the range $1 < j < i$, $x'_i = x_i v_i x_{i-1}$, and $x'_j = x_j$ for every j in the range $i < j \leq k$. It is easy to verify, that the decomposition $wv = x'_k v'_k x'_{k-1} v'_{k-1} \cdots v'_2 x'_1 v'_1$ has the desired properties.

2.) We have $\mathrm{Sc}_F(w) > 0$ if and only if there exists a suffix x of w with $\operatorname{Occ}(x) = F$. Due to the decomposition characterization, having a suffix x with $\operatorname{Occ}(x) = F$ is equivalent to $F = \{v_1, \ldots, v_i\}$ for some i.

3.) By definition, we have $\mathrm{Acc}_F(w) = \operatorname{Occ}(x)$ where x is the longest suffix of w such that the score of F does not change throughout x and $\operatorname{Occ}(x) \subseteq F$. Consider the decomposition characterization of w as above. We have $\{v_1, \ldots, v_i\} \subseteq \mathrm{Acc}_F(w)$, since $x_i v_i \cdots v_1 v_1$ is a suffix of w satisfying $\operatorname{Occ}(x) \subseteq F$. Furthermore, since $v_{i+1} \notin F$ by the maximality of i, this is the longest such suffix and we have indeed $\mathrm{Acc}_F(w) = \{v_1, \ldots, v_i\}$.

4.) The lastest increase of $\mathrm{Sc}_F(w)$ occurs after (or at) the last visit of v_i, since $\operatorname{Occ}(v_i x_{i-1} \cdots x_1 v_1) = F$. Hence, $\mathrm{Acc}_F(w)$ is the occurrence set of a suffix of $x_{i-1} \cdots x_1 v_1$ and the decomposition characterization yields the result. □

The previous characterization allows us to bound the size of $\mathcal{G}_S$.

Lemma 7. *We have $|V^S| \leq \left(\sum_{k=1}^{n} \binom{n}{k} \cdot k! \cdot 2^k \cdot k!\right) + 1 \leq (n!)^3$, where $n = |V|$.*

Proof. We can merge all vertices in $V \setminus F$ to a single vertex while retaining the equivalence $v \in W_i(\mathcal{G}) \Leftrightarrow [v] \in W_i(\mathcal{G}_S)$ (since $f(v) \in F$ for every $v \in V$) and without changing the winning region of Player 0 (since $W_0(\mathcal{G}_S) \subseteq F$).

Hence, it remains to bound the number of equivalence classes in $\mathrm{Plays}_{\leq 2} /_{=_{\mathcal{F}_1}}$. Lemma 6 shows that a finite play $w \in V^+$ has $|\mathrm{LAR}(w)|$ many sets with non-zero score. Furthermore, the accumulator of the sets with score zero is determined by $\mathrm{LAR}(w)$. Now, consider a play $w \in \mathrm{Plays}_{\leq 2}$ and a set $F \in \mathcal{F}_1$ with non-zero score.

We have $\text{Sc}_F(w) \in \{1, 2\}$ and there are exactly $|F|$ possible values for $\text{Acc}_F(w)$ due to Lemma 6.4, which bounds the number of occurrence sets of suffixes of w. Finally, two finite plays having the same LAR also have the same last vertex.

Hence, the number of equivalence classes is bounded by the number of LARs, which is at most $\sum_{k=1}^{n} \binom{n}{k} \cdot k!$, times the number of possible score and accumulator combinations for an LAR of length k, which is at most $2^k \cdot k!$. □

We conclude this section by mentioning that if one is not interested in computing the complete winning regions of the Muller game, but only wants to determine which player has a winning strategy from a given vertex v, then it suffices to construct only the part of $\mathcal{G}_S$ that is reachable from $[v]$.

Also note that while a player in general can not prevent her opponent from reaching a score of 2, there are arenas in which she can do so. By first constructing the safety game $\mathcal{G}'_S$ up to threshold 2, which is smaller than the one for threshold 3, one can possibly determine a subset of Player 0's winning region faster and obtain a (potentially) smaller finite-state winning strategy for this subset: we have $W_0(\mathcal{G}'_S) \subseteq W_0(\mathcal{G}_S)$. However, if Player 0 cannot prevent her opponent from reaching a score of 2 when starting in v, then this does not imply that Player 1 wins the Muller game from v as well. In this case, one has to solve the safety game with threshold 3 to determine the winner of the Muller game from this vertex.

4.1 Antichain-based Winning Strategies for Muller Games

Using an antichain construction one can construct a smaller finite-state winning strategy for Player 0: instead of considering all equivalence classes in the winning region of Player 0, we only consider the maximal ones with respect to $\leq_{\mathcal{F}_1}$ which are reachable via a fixed positional winning strategy for her in the safety game. To this end, we lift $\leq_{\mathcal{F}_1}$ to equivalence classes by defining $[w] \leq_{\mathcal{F}_1} [w']$ if and only if $w \leq_{\mathcal{F}_1} w'$.

Let σ be a positional winning strategy for Player 0 in $\mathcal{G}_S$ and let R be the set of vertices in V^S which are reachable from $\{[v] \mid [v] \in W_0(\mathcal{G}_S)\}$ 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 $[wv]$ for $v \in V$) is in R.

Now, let $R_{\max}$ be the $\leq_{\mathcal{F}_1}$-maximal elements of R. Applying the facts about successors of vertices in R stated above, we obtain the following remark.

Remark 4. 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 $(\text{Last}(w), v) \in E$ and there is a $[w'] \in R_{\max}$ such that $[wv] \leq [w']$.
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 $[wv] \leq_{\mathcal{F}_1} [w']$.

Thus, instead of updating the memory from $[w]$ to $[wv]$ when processing a vertex v, we can directly update it to a maximal element that is $\mathcal{F}_1$-larger than $[wv]$. Intuitively, instead of keeping track of the exact scores, we store a maximal element that over-approximates the exact values.

We define $\mathfrak{M} = (M, \mathrm{Init}, \mathrm{Upd}, \mathrm{Nxt})$ by $M = R_{\max} \cup \{\bot\}$ [4],

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

and

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

Then, for every $w \in V^+$ with $\mathrm{Upd}^*(w) \neq \bot$ we have $[w] \leq_{\mathcal{F}_1} \mathrm{Upd}^*(w)$, which implies $\mathrm{Last}(w) = \mathrm{Last}(w')$, where $[w'] = \mathrm{Upd}^*(w)$.

Using Remark 4.1, we define the next-move function by

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

and $\mathrm{Nxt}(v, \bot) = v''$ for some v'' with $(v, v'') \in E$. Again, the second case in the definition above is just to match the formal definition of a next-move function. It will never be invoked due to $[w] \leq_{\mathcal{F}_1} \mathrm{Upd}^*(w)$ or $\mathrm{Upd}^*(w) = \bot$.

Analogously to the construction in the previous section, it remains to show that $\sigma_{\mathfrak{M}}$ is a winning strategy for Player 0 from $W = \{v \mid [v] \in W_0(\mathcal{G}_S)\}$. An inductive application of Remark 4 shows that every play w that starts in W and is consistent with $\sigma_{\mathfrak{M}}$ satisfies $\mathrm{Upd}^*(w) \neq \bot$. This bounds the scores of Player 1 by 2, as we have $[w] \leq_{\mathcal{F}_1} \mathrm{Upd}^*(w) \in R_{\max} \subseteq \mathrm{Plays}_{\leq 2}$ for every such play. Hence, $\sigma_{\mathfrak{M}}$ is indeed a winning strategy for Player 0 from W.

4.2 Reducing the Number of Memory States

In the proof of Theorem 3 we used the whole winning region $W_0(\mathcal{G}_S)$ of the safety game as memory structure for a winning strategy of the Muller game. However, when defining the next-move function, we may have to choose between several vertices v' with $v' \in W_0(\mathcal{G}_S)$. Depending on this choice, parts of the memory structure may never be reached (as long as Player 0 sticks to the strategy) and, therefore, can be omitted. Hence, it is possible to reduce the number of memory states necessary to realize a winning strategy by defining the next-move function wisely. The same idea applies to antichain-based winning strategies where the fixed strategy σ for the safety game $\mathcal{G}_S$ determines the size of the set R of reachable vertices and, hence, of the number of maximal elements.

Unfortunately, it is not clear how to efficiently find a small solution, i.e., a Nxt function or strategy σ that induces a small (or even minimal) reachable part of $\mathcal{G}_S$. One straightforward heuristic is to compute a "closed" initial part of the safety game by starting in some initial vertex and considering all successors of Player 1 vertices but only one successor of Player 0 vertices. The choice of the successor in a Player 0 vertex can be made using some order on the successors, or by simply picking an arbitrary one. Another way is to use the automata learning-based approach described in [8].

[4] Again, 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.

5 Conclusion

We have presented a new algorithm to determine the winning regions of a Muller game and to determine a winning strategy for one of the players by solving a safety game. The safety game is polynomially larger than the parity game obtained in a reduction, but it is faster to solve than the latter.

The scores induce a hierarchy of *all* finite-state winning strategies, since each one of them prevents the opponent from reaching a score that is larger than a certain threshold. We suggest to use the highest score the opponent can achieve against a given strategy as quality measure for the strategy. In ongoing research we investigate whether one can minimize the size of a finite-state strategy and the scores it allows simultaneously.

Furthermore, it is easy to see that the solution of the safety game actually yields a non-deterministic strategy which only disallows those moves that would allow the opponent to reach a score of 3 (e.g., see the vertices [1], [01], and [21] in Figure 3). In this sense, our work extends the results of Bernet, Janin, and Walukiewicz [1] on permissive strategies for parity games to Muller games. In upcoming work, we show that for every fixed k there is a unique most general non-deterministic winning strategy that subsumes all strategies preventing the opponent from reaching a score of k.

Acknowledgments We want to thank Wladimir Fridman for many helpful discussions.

References

1. Julien Bernet, David Janin, and Igor Walukiewicz. Permissive strategies: from parity games to safety games. *ITA*, 36(3):261–275, 2002.
2. John Fearnley and Martin Zimmermann. Playing Muller games in a hurry. *Int. J. Found. Comput. Sci.* To appear. Journal version of [3].
3. John Fearnley and Martin Zimmermann. Playing Muller games in a hurry. In Angelo Montanari, Margherita Napoli, and Mimmo Parente, editors, *GANDALF*, volume 25 of *EPTCS*, pages 146–161, 2010. Conference version of [2].
4. Yuri Gurevich and Leo Harrington. Trees, automata, and games. In *STOC*, pages 60–65. ACM, 1982.
5. Alexander Kechris. *Classical Descriptive Set Theory*, volume 156 of *Graduate Texts in Mathematics*. Springer, 1995.
6. Robert McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl. Logic*, 65(2):149–184, 1993.
7. Robert McNaughton. Playing infinite games in finite time. In Arto Salomaa, Derick Wood, and Sheng Yu, editors, *A Half-Century of Automata Theory*, pages 73–91. World Scientific, 2000.
8. Daniel Neider. Small strategies for safety games. In *Proceedings of the Ninth International Symposium on Automated Technology for Verification and Analysis (ATVA 2011)*, LNCS. Springer, to appear.

Aachener Informatik-Berichte

This list contains all technical reports published during the past three years. A complete list of reports dating back to 1987 is available from http://aib.informatik.rwth-aachen.de/. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: `biblio@informatik.rwth-aachen.de`

2008-01 *	Fachgruppe Informatik: Jahresbericht 2007
2008-02	Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing
2008-03	Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, Harald Zankl: Maximal Termination
2008-04	Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler
2008-05	Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations
2008-06	Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs
2008-07	Alexander Nyßen, Horst Lichter: The MeDUSA Reference Manual, Second Edition
2008-08	George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs
2008-09	George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs
2008-10	George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time
2008-11	George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows
2008-12	Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages
2008-13	Beatriz Alarcón, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raúl Gutiérrez, Salvador Lucas, Peter Schneider-Kamp, René Thiemann: Improving Context-Sensitive Dependency Pairs
2008-14	Bastian Schlich: Model Checking of Software for Microcontrollers
2008-15	Joachim Kneis, Alexander Langer, Peter Rossmanith: A New Algorithm for Finding Trees with Many Leaves
2008-16	Hendrik vom Lehn, Elias Weingärtner and Klaus Wehrle: Comparing recent network simulators: A performance evaluation study
2008-17	Peter Schneider-Kamp: Static Termination Analysis for Prolog using Term Rewriting and SAT Solving
2008-18	Falk Salewski: Empirical Evaluations of Safety-Critical Embedded Systems
2008-19	Dirk Wilking: Empirical Studies for the Application of Agile Methods to Embedded Systems
2009-02	Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Quantitative Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications

2009-03 Alexander Nyßen: Model-Based Construction of Embedded Real-Time Software - A Methodology for Small Devices
2009-04 Daniel Klünder: Entwurf eingebetteter Software mit abstrakten Zustandsmaschinen und Business Object Notation
2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs
2009-06 George B. Mertzios, Ignasi Sau, Shmuel Zaks: The Recognition of Tolerance and Bounded Tolerance Graphs is NP-complete
2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I
2009-08 Joachim Kneis, Alexander Langer: Satellites and Mirrors for Solving Independent Set on Sparse Graphs
2009-09 Michael Nett: Implementation of an Automated Proof for an Algorithm Solving the Maximum Independent Set Problem
2009-10 Felix Reidl, Fernando Sánchez Villaamil: Automatic Verification of the Correctness of the Upper Bound of a Maximum Independent Set Algorithm
2009-11 Kyriaki Ioannidou, George B. Mertzios, Stavros D. Nikolopoulos: The Longest Path Problem is Polynomial on Interval Graphs
2009-12 Martin Neuhäußer, Lijun Zhang: Time-Bounded Reachability in Continuous-Time Markov Decision Processes
2009-13 Martin Zimmermann: Time-optimal Winning Strategies for Poset Games
2009-14 Ralf Huuck, Gerwin Klein, Bastian Schlich (eds.): Doctoral Symposium on Systems Software Verification (DS SSV'09)
2009-15 Joost-Pieter Katoen, Daniel Klink, Martin Neuhäußer: Compositional Abstraction for Stochastic Systems
2009-16 George B. Mertzios, Derek G. Corneil: Vertex Splitting and the Recognition of Trapezoid Graphs
2009-17 Carsten Kern: Learning Communicating and Nondeterministic Automata
2009-18 Paul Hänsch, Michaela Slaats, Wolfgang Thomas: Parametrized Regular Infinite Games and Higher-Order Pushdown Strategies
2010-02 Daniel Neider, Christof Löding: Learning Visibly One-Counter Automata in Polynomial Time
2010-03 Holger Krahn: MontiCore: Agile Entwicklung von domänenspezifischen Sprachen im Software-Engineering
2010-04 René Wörzberger: Management dynamischer Geschäftsprozesse auf Basis statischer Prozessmanagementsysteme
2010-05 Daniel Retkowitz: Softwareunterstützung für adaptive eHome-Systeme
2010-06 Taolue Chen, Tingting Han, Joost-Pieter Katoen, Alexandru Mereacre: Computing maximum reachability probabilities in Markovian timed automata
2010-07 George B. Mertzios: A New Intersection Model for Multitolerance Graphs, Hierarchy, and Efficient Algorithms
2010-08 Carsten Otto, Marc Brockschmidt, Christian von Essen, Jürgen Giesl: Automated Termination Analysis of Java Bytecode by Term Rewriting
2010-09 George B. Mertzios, Shmuel Zaks: The Structure of the Intersection of Tolerance and Cocomparability Graphs

2010-10 Peter Schneider-Kamp, Jürgen Giesl, Thomas Ströder, Alexander Serebrenik, René Thiemann: Automated Termination Analysis for Logic Programs with Cut

2010-11 Martin Zimmermann: Parametric LTL Games

2010-12 Thomas Ströder, Peter Schneider-Kamp, Jürgen Giesl: Dependency Triples for Improving Termination Analysis of Logic Programs with Cut

2010-13 Ashraf Armoush: Design Patterns for Safety-Critical Embedded Systems

2010-14 Michael Codish, Carsten Fuhs, Jürgen Giesl, Peter Schneider-Kamp: Lazy Abstraction for Size-Change Termination

2010-15 Marc Brockschmidt, Carsten Otto, Christian von Essen, Jürgen Giesl: Termination Graphs for Java Bytecode

2010-16 Christian Berger: Automating Acceptance Tests for Sensor- and Actuator-based Systems on the Example of Autonomous Vehicles

2010-17 Hans Grönniger: Systemmodell-basierte Definition objektbasierter Modellierungssprachen mit semantischen Variationspunkten

2010-18 Ibrahim Armaç: Personalisierte eHomes: Mobilität, Privatsphäre und Sicherheit

2010-19 Felix Reidl: Experimental Evaluation of an Independent Set Algorithm

2010-20 Wladimir Fridman, Christof Löding, Martin Zimmermann: Degrees of Lookahead in Context-free Infinite Games

2011-02 Marc Brockschmidt, Carsten Otto, Jürgen Giesl: Modular Termination Proofs of Recursive Java Bytecode Programs by Term Rewriting

2011-03 Lars Noschinski, Fabian Emmes, Jürgen Giesl: A Dependency Pair Framework for Innermost Complexity Analysis of Term Rewrite Systems

2011-04 Christina Jansen, Jonathan Heinen, Joost-Pieter Katoen, Thomas Noll: A Local Greibach Normal Form for Hyperedge Replacement Grammars

2011-11 Nils Jansen, Erika Ábrahám, Jens Katelaan, Ralf Wimmer, Joost-Pieter Katoen, Bernd Becker: Hierarchical Counterexamples for Discrete-Time Markov Chains

* These reports are only available as a printed version.

Please contact biblio@informatik.rwth-aachen.de to obtain copies.