Playing Safe

Thomas Colcombet[1], Nathanaël Fijalkow[1,2], and Florian Horn[1]

1 **LIAFA, Université Paris 7 - Denis Diderot, France**
2 **University of Warsaw, Poland**

Abstract

We consider two-player games over graphs and give tight bounds on the memory size of strategies ensuring safety conditions. More specifically, we show that the minimal number of memory states of a strategy ensuring a safety condition is given by the size of the maximal antichain of left quotients with respect to language inclusion. This result holds for all safety conditions without any regularity assumptions, and for all (finite or infinite) graphs of finite degree.

We give several applications of this general principle. In particular, we characterize the exact memory requirements for the opponent in generalized reachability games, and we prove the existence of positional strategies in games with counters.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Game Theory, Synthesis, Safety Specifications, Program Verification

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Graphs games provide a mathematical framework to automatically address many questions, for instance they are used to model reactive systems (we refer to [9] for a survey on the topic). We focus here on the Synthesis Problem to motivate the problem we consider, which is to characterize the amount of memory required in games with safety conditions.

The Synthesis Problem. The inputs of the Synthesis Problem are a *system* and a *specification*. The expected output is a *controller* for the system, that ensures the specification.

We describe here an approach to solve the Synthesis Problem through Game Theory. We model the system as a graph, whose vertices represent states and edges represent transitions. Its evolution consists in interactions between a controller and an environment, which is turned into a game on the graph between two players, Eve and Adam. If in a given state, the controller can choose the evolution of the system, then the corresponding vertex is controlled by Eve. If the system evolves in an uncertain way, we consider the worst-case scenario, where Adam controls those states.

A pebble is initially placed on the vertex representing the initial state of the system, then Eve and Adam move this pebble along the edges. The sequence built describes a run of the system: Eve tries to ensure that it satisfies the specification.

So, in order to synthesize a controller, we are interested in whether Eve can ensure this objective and what resources she needs. In particular, the most salient question is: what is the size of a *minimal* controller satisfying the specification? Since a controller is here given by a strategy for Eve, this is equivalent to the following question: what is the minimal amount of memory used by a winning strategy?

The following diagram shows the correspondence between the notions from the Synthesis

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–12

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Problem (left-hand side) and the game-theoretical notions (right-hand side).

$$\underbrace{\mathbb{S}}_{\text{system}}, \underbrace{\mathcal{C}}_{\text{controller}} \models \underbrace{\Phi}_{\text{specification}} \qquad \Longleftrightarrow \qquad \underbrace{\mathcal{A}}_{\text{arena (graph)}}, \underbrace{\sigma}_{\text{strategy}} \models \underbrace{W}_{\text{condition}}$$

Safety Specifications. Since we consider non-terminating sequences, a specification is given by a language of infinite words. Of special interest are the specifications asserting that "nothing bad" ever happens; such specifications are called *safety specifications*. A safety specification is induced by a (possibly infinite and non-regular) set of bad prefixes P, and the specification is met by a run if no prefixes belong to P.

Although quite simple, the safety specifications proved useful in both theory and practice, and are actively studied (we refer to [12] for a survey).

Our contribution. In this paper, we show the following general principle:

> For a safety winning condition W, the minimal number of memory states of a winning strategy is exactly the cardinal of the maximal antichain of left quotients of W.

We refer to Section 3 for the missing definitions. Note that this result holds for all safety winning conditions, without any regularity assumption. The only assumption made is that graphs have finite degree: we prove that this is necessary, by providing a counter example not satisfying this property. The Section 3 is devoted to the proof of this main result. We give several examples and applications in Section 4. For instance, it allows to characterize the memory requirements for the opponent in generalized reachability games, and to prove the existence of positional strategies in games with counters.

Related Works: Evaluating Memory Requirements. Characterizing the amount of memory required by winning strategies according to the winning conditions has been widely studied, in different frameworks.

The first result in this direction is about Muller conditions: in [5], the authors show how to compute the exact memory requirements by looking at the so-called Zielonka tree of a Muller condition. This is orthogonal to our results, as the Muller conditions only specify the limit behaviour (what is seen infinitely often), whereas we consider here only the behaviours in the finite.

In a different direction, Hugo Gimbert and Eryk Kopczyński independently investigated necessary and sufficient conditions for positional determinacy. We refer to their respective PhD theses [8, 11] for more details. A submitted result of Eryk Kopczyński [10] shows that one can compute the exact *chromatic* memory requirements of an ω-regular condition. Our results are also orthogonal to this result, as we characterize the exact memory requirements for all safety conditions, including *non-regular* ones.

2 Definitions

The games we consider are played on an *arena* $\mathcal{A} = (V, (V_\exists, V_\forall), E, c)$, consisting of a (finite or infinite) graph (V, E), a partition $(V_\exists, V_\forall)$ of the vertex set V: a vertex in $V_\exists$ belongs to Eve and in $V_\forall$ to Adam, and a coloring function $c : E \to A$ mapping edges to a color from a finite alphabet A. When drawing arenas, we will use circles for vertices owned by Eve and squares for those owned by Adam.

Throughout this paper, we make two assumptions:

- There are no dead-ends: for every vertex $v \in V$, there exists an edge $(v, v') \in E$;
- The degree is finite: for every vertex $v \in V$, the set $\{v' \mid (v, v') \in E\}$ is finite.

The first assumption is cosmetic. The second assumption, however, will be crucial: in particular, we will give a counter-example showing that our results do not hold without this assumption.

Game. A *play* π is an infinite word of edges $e_0 \cdot e_1 \cdots$ that are consecutive: for all i, $e_i = (\_, v) \in E$ and $e_{i+1} = (v, \_) \in E$ for some $v \in V$. The prefix of length k of π is denoted π_k. Note that we will always assume that our arenas have no dead-ends: from every vertex $v \in V$, there exists $v' \in V$ such that $(v, v') \in E$. A play π induces an infinite sequence of colors $c(\pi)$, obtaining by applying the coloring function c component-wise. We define *winning conditions* for a player by giving a set of infinite sequences of colors $W \subseteq A^\omega$. As we are interested in zero-sum games, *i.e.* where the winning conditions of the two players are opposite, if the winning condition for Eve is W, then the winning condition for Adam is $A^\omega \setminus W$. A *game* is a couple $\mathcal{G} = (\mathcal{A}, W)$ where $\mathcal{A}$ is an arena and W a winning condition.

Strategy. A *strategy* for a player is a function that prescribes, given a finite history of the play, the next move. Formally, a strategy for Eve is a function $\sigma : E^* \cdot V_\exists \to E$ such that for all $\pi \in E^*$ and $v \in V_\exists$ we have $\sigma(\pi \cdot v) = (v, \_) \in E$. Strategies for Adam are defined similarly, and usually denoted τ. Once a game $\mathcal{G} = (\mathcal{A}, W)$, a starting vertex v_0 and strategies σ for Eve and τ for Adam are fixed, there is a unique play $\pi(v_0, \sigma, \tau)$, which is said winning for Eve if its image by c belongs to W.

A strategy σ for Eve is *winning* if for all strategies τ for Adam, $\pi(q_0, \sigma, \tau)$ is winning. We say that Eve wins the game $\mathcal{G}$ from v_0 if she has a winning strategy from v_0, and denote $\mathcal{W}_E(\mathcal{G})$ the set of vertices from where Eve wins; we often say that $v \in \mathcal{W}_E(\mathcal{G})$ is winning. We define similarly $\mathcal{W}_A(\mathcal{G})$ for Adam to be the set of vertices from where Adam wins.

Memory. A *memory structure* is a deterministic state machine that reads the sequence of edges and abstracts its relevant informations into a memory state. Formally, a memory structure $\mathcal{M} = (M, m_0, \mu)$ for an arena consists of a set M of memory states, an initial memory state $m_0 \in M$ and an update function $\mu : M \times E \to M$. The update function takes as input the current memory state and the chosen edge to compute the next memory state. It can be extended to a function $\mu^* : E^* \to M$ by defining $\mu^*(\varepsilon) = m_0$ and $\mu^*(\pi \cdot e) = \mu(\mu^*(\pi), e)$. Given a memory structure $\mathcal{M}$ and a next-move function $\nu : V_\exists \times M \to E$, we can define a strategy σ for Eve by $\sigma(\pi \cdot v) = \nu(v, \mu^*(\pi \cdot v))$. A strategy with memory structure $\mathcal{M}$ has finite memory if M is a finite set. It is *memoryless*, or *positional* if M is a singleton: it only depends on the current vertex. Hence a memoryless strategy can be described as a function $\sigma : V_\exists \to E$. We denote $\mathrm{mem}(\sigma)$ the number of memory states used by the strategy σ.

An arena and a memory structure induce an expanded arena where the current memory state is computed online. Formally, the arena $\mathcal{A} = (V, (V_\exists, V_\forall), E, c)$, the memory structure $\mathcal{M}$ for $\mathcal{A}$ and a new coloring function $c' : E \times M \to A$ induce an expanded arena $\mathcal{A} \times \mathcal{M} = (V \times M, (V_\exists \times M, V_\forall \times M), E \times \mu, c')$, where $E \times \mu$ is defined by: $((v, m), (v', m')) \in E \times \mu$ if $(v, v') \in E$ and $\mu(m, (v, v')) = m'$.

From a memoryless strategy in $\mathcal{A} \times \mathcal{M}$, we can build a strategy in $\mathcal{A}$ using $\mathcal{M}$ as memory structure, which behaves as the original strategy. This key observation will be used several times in the paper.

3 Tight Bounds on the Memory for Safety Conditions

In this section, we consider a safety condition[1] W and compute the following quantity:

$$\mathrm{mem}(W) \doteq \sup_{\substack{\mathcal{G}=(\mathcal{A},W) \text{ game} \\ v_0 \text{ initial vertex}}} \inf_{\substack{\sigma \text{ winning} \\ \text{strategy}}} \mathrm{mem}(\sigma) \ .$$

In words, $\mathrm{mem}(W)$ is the necessary and sufficient number of memory states for constructing a winning strategy in games with condition W. Equivalently:

- *upper bound:* for all games $\mathcal{G} = (\mathcal{A}, W)$, if Eve has a winning strategy from an initial vertex v_0, then she has a winning strategy using at most $\mathrm{mem}(W)$ memory states,
- *lower bound:* there exists a game $\mathcal{G} = (\mathcal{A}, W)$ and an initial vertex v_0 where Eve has a winning strategy, but no winning strategy using less than $\mathrm{mem}(W)$ memory states.

The reader with a background in game theory may be surprised, as it is well known that "safety games are positionally determined", implying that the quantity above is constant equal to one. The subtlety here is that our setting is (much) more general than the classical notion of safety games. Specifically, consider an arena $\mathcal{A}$ and a coloring function $c : E \to A$:

- an *internal* safety condition is given by a subset $B \subseteq A$ of forbidden colors, inducing the winning condition

$$\mathrm{Safe}(B) = \{\rho = a_0 \cdot a_1 \cdots \in A^\omega \mid \text{for all } i,\ a_i \notin B\} \ ,$$

- an *external* safety condition is given by a subset $P \subseteq A^*$ of forbidden prefixes of colors, inducing the winning condition

$$\mathrm{Safe}(P) = \{\rho = a_0 \cdot a_1 \cdots \in A^\omega \mid \text{for all } i,\ a_0 \cdot a_1 \cdots a_i \notin P\} \ .$$

The term "internal" refers to the idea that the set B of forbidden edges can be thought of as a set of forbidden edges in the graph, giving rise to the classical notion of safety games, where Eve tries to ensure never to reach the forbidden parts in the graph.

▶ **Lemma 1** (Folklore). *Let $\mathcal{G}$ be a game with an internal safety condition, and v_0 an initial vertex. If Eve has a winning strategy, then she has a positional winning strategy.*

On the other hand, the external safety conditions describe much more, as we will demonstrate in Section 4.

From now on, as we are mostly interested in external safety conditions, we will drop the prefix "external". The notion of safety condition originates from topological studies of the set of infinite words: the safety conditions are the closed sets for the Cantor topology.

For the remainder of this section, we fix a safety condition $W = \mathrm{Safe}(P)$ induced by $P \subseteq A^*$.

A First Upper Bound

We first give an upper bound on $\mathrm{mem}(W)$. Let $w \in A^*$, define its left quotient as:

$$w^{-1}W = \{\rho \in A^\omega \mid w \cdot \rho \in W\}.$$

We denote $\mathrm{Res}(W)$ the set of left quotients of W. We mention some special left quotients: the initial one, $\varepsilon^{-1}W$ (equal to W), and the empty one, obtained as $w^{-1}W$ for any $w \in P$.

[1] To be defined in this section.

From a left quotient $w^{-1}W$ and a letter $a \in A$, we define $(w^{-1}W) \cdot a$ as $(w \cdot a)^{-1}W$: it is easy to check that this is well defined (independent of the representant w chosen). Recall that $\mathrm{Res}(W)$ is finite if and only if W is regular, and in such case it can be used to describe the set of states of the minimal deterministic automaton recognizing W.

▶ **Lemma 2.** *For all games $\mathcal{G} = (\mathcal{A}, W)$, if Eve has a winning strategy from an initial vertex v_0, then she has a winning strategy using at most $|\mathrm{Res}(W)|$ memory states.*

Consequently,

$$\mathrm{mem}(W) \ \leq \ |\mathrm{Res}(W)| \ .$$

Proof. We construct a memory structure $\mathcal{M}$, as follows: $\mathcal{M} = (\mathrm{Res}(W), W, \nu)$, where $\nu(w^{-1}W, a) = (w^{-1}W) \cdot a$. At any point in the game, the memory state computed by $\mathcal{M}$ is the current left quotient. Let $\mathcal{A}$ be an arena. We construct the expanded arena $\mathcal{A} \times \mathcal{M}$ equipped with the coloring function $c' : E \times \mathrm{Res}(W) \to \{0, 1\}$ defined by:

$$c'(\_, w^{-1}W) = \begin{cases} 0 & \text{if } w^{-1}W = \emptyset \text{ (equivalently, } w \in P) \ , \\ 1 & \text{otherwise .} \end{cases}$$

We attach to $\mathcal{A} \times \mathcal{M}$ the *internal* safety condition induced by $B = \{0\}$, giving rise to the game $\mathcal{G} \times \mathcal{M} = (\mathcal{A} \times \mathcal{M}, \mathrm{Safe}(B))$. First observe that by construction, the plays in $\mathcal{A} \times \mathcal{M}$ are of the form $(e_0, c(e_0)^{-1}W) \cdot (e_1, c(e_0 \cdot e_1)^{-1}W) \cdots (e_k, c(e_0 \cdot e_1 \cdots e_k)^{-1}W)$, so by definition of c' a play is winning is $\mathcal{G} \times \mathcal{M}$ if and only if its projection (on the first component) is winning in $\mathcal{G}$.

It follows that a winning strategy for Eve in $\mathcal{G}$ from v_0 induces a winning strategy in $\mathcal{G} \times \mathcal{M}$ from (v_0, W). Now, thanks to Lemma 1, since Eve wins in $\mathcal{G} \times \mathcal{M}$, she has a positional winning strategy. This induces a winning strategy in $\mathcal{G}$ using $\mathcal{M}$ as memory structure, concluding the proof of Lemma 2. ∎

The game $\mathcal{G} \times \mathcal{M}$ defined above will be an important tool in the proofs to follow. We will also rely on the following remark: assume we want to prove that a strategy σ is winning. Then it is enough to show that for all plays π consistent with σ, for all k, $c(\pi_k)^{-1}W \neq \emptyset$, where π_k is the prefix of π of length k. This simple observation follows from the definition of safety conditions.

A Tighter Upper Bound

The memory structure $\mathcal{M}$ is not optimal. A first remark is that the empty left quotient (which exists if $W \neq A^\omega$) can be removed from the memory states as the game is lost. From now on by "left quotient" we mean "non-empty left quotient of W", and in particular $\mathrm{Res}(W)$ denotes the set of non-empty left quotients.

The second remark is the following: let L_1 and L_2 be two left quotients of W, such that $L_1 \subseteq L_2$. With the same notations as above, consider a vertex v in the arena $\mathcal{A}$. If Eve wins from (v, L_1) in $\mathcal{G} \times \mathcal{M}$, then she also wins from (v, L_2): indeed, she can play as she would have played from (v, L_1). Since this ensures from v that all plays are winning for L_1, then a fortiori they are winning for L_2.

This suggests to restrict the memory states only to *minimally winning* left quotients with respect to inclusion. Two issues arise:

- which left quotients are winning depends on the current vertex, so the semantics of a memory state can no longer be *one* left quotient, but rather a left quotient for each possible vertex,

- there may not exist *minimally winning* left quotients.

For the sake of presentation, we first show how to deal with the first issue, assuming the second issue does not appear. Specifically, in the following lemma, we assume that $\mathrm{Res}(W)$ is finite (*i.e.* W is regular), implying the existence of minimally winning left quotients. We will later drop this assumption.

We define the width of an ordered set $(E, \leq)$ as the cardinal of the maximal antichain of E with respect to $\leq$, *i.e.* the cardinal of the largest set of pairwise incomparable elements.

▶ **Lemma 3** (Upper bound in the regular case). *Assume that* $\mathrm{Res}(W)$ *is finite.*

For all games $\mathcal{G} = (\mathcal{A}, W)$*, if Eve has a winning strategy from an initial vertex* v_0*, then she has a winning strategy using at most* K *memory states, where* K *is the width of* $(\mathrm{Res}(W), \subseteq)$*.*

Proof. We use the same notations as for the proof of Lemma 2, and construct a smaller memory structure together with a winning strategy using this memory structure. In this proof, by winning we mean winning in the game $\mathcal{G} \times \mathcal{M}$.

Let K be the cardinal of the maximal antichain of left quotients of W. We construct the memory structure $\mathcal{M}^* = (\{1, \dots, K\}, 1, \mu)$, and the strategy σ induced by the next-move function ν.

Let v be a vertex in $\mathcal{A}$. We consider the set of minimal left quotients L such that (v, L) is winning. (Here we use the finiteness of $\mathrm{Res}(W)$ to guarantee the existence of such left quotients.) This is an antichain, so there are at most K of them, we denote them by $L_1(v), \dots, L_p(v)$, for some $p \leq K$. The *key* property is that for every left quotient L such that (v, L) is winning, there exists i such that $L_i(v) \subseteq L$. Furthermore, we choose $L_1(v_0)$ such that $L_1(v_0) \subseteq W$. (Indeed, by assumption (v_0, W) is winning.)

We define the update function: $\mu(i, (v, v'))$ is a j such that $L_j(v') \subseteq L_i(v) \cdot c(v, v')$. Note that in general, such a j may not exist; it does exist if $(v', L_i(v) \cdot c(v, v'))$ is winning, and we will prove that this will always be the case when playing the strategy σ.

We define the next-move function ν (inducing σ). Let $v \in V_\exists$, and consider $(v, L_i(v))$: since Eve wins from there, there exists an edge $(v, v') \in E$ such that $(v', L_i(v) \cdot c(v, v'))$ is winning. Define $\nu(v, i)$ to be this v'.

We show that the strategy σ is winning. Consider a play $\pi = (v_0, v_1) \cdot (v_1, v_2) \cdots$ consistent with σ, and $i_0 \cdot i_1 \cdots$ the sequence of memory states assumed along this play. Denote π_k the prefix of π of length k, we prove that for all k, $L_{i_k}(v_k) \subseteq c(\pi_k)^{-1}W$. Note that by definition, $(v_k, L_{i_k}(v_k))$ is winning, so $L_{i_k}(v_k) \neq \emptyset$, implying that $c(\pi_k)^{-1}W \neq \emptyset$.

We proceed by induction. For $k = 0$, it follows from $L_1(v_0) \subseteq W$. Let $k > 0$, the induction hypothesis is $L_{i_{k-1}}(v_{k-1}) \subseteq c(\pi_{k-1})^{-1}W$. We distinguish two cases.

- Either v_{k-1} belongs to Eve, then by construction of σ we have that $(v_k, L_{i_{k-1}}(v_{k-1}) \cdot c(v_{k-1}, v_k))$ is winning. It follows that the update function is well defined, and $L_{i_k}(v_k) \subseteq L_{i_{k-1}}(v_{k-1}) \cdot c(v_{k-1}, v_k)$, which together with the induction hypothesis implies $L_{i_k}(v_k) \subseteq c(\pi_k)^{-1}W$.
- Or v_{k-1} belongs to Adam. Since Adam cannot escape $\mathcal{W}_E(\mathcal{G} \times \mathcal{M})$, we have that $(v_k, L_{i_{k-1}}(v_{k-1}) \cdot c(v_{k-1}, v_k))$ is winning, and the same reasoning concludes.

It follows that the strategy σ is winning, concluding the proof of Lemma 3. ∎

We now get rid of the regularity assumption. This means that for a vertex v, there may not be a minimal left quotient L such that (v, L) is winning. To get around this difficulty, the semantics of a memory state is a not anymore a left quotient for each vertex, but rather a decreasing sequence of left quotients for each vertex.

Note that the proof of Lemma 4 uses the finite degree assumption, and is the only proof in the paper to do so. We will show in Section 4 that the result fails without this assumption.

▶ **Lemma 4** (Upper bound). *For all games $\mathcal{G} = (\mathcal{A}, W)$, if Eve has a winning strategy from an initial vertex v_0, then she has a winning strategy using at most K memory states, where K is the width of* $(\mathrm{Res}(W), \subseteq)$.

Consequently, $\mathrm{mem}(W)$ *is smaller than or equal to the width of* $(\mathrm{Res}(W), \subseteq)$.

Proof. We use the same notations as for the proof of Lemma 3, and construct a memory structure together with a winning strategy using this memory structure.

Let K be the cardinal of the maximal antichain of left quotients of W. We construct the memory structure $\mathcal{M}^* = (\{1, \dots, K\}, 1, \mu)$, and the strategy σ induced by the next-move function ν.

Let v be a vertex in $\mathcal{A}$. We consider the set $\mathcal{W}(v)$ of left quotients L such that (v, L) is winning. We split $\mathcal{W}(v)$ into maximal decreasing (finite or infinite) sequences of left quotients, denoted $\ell_1(v), \dots, \ell_p(v)$, for some $p \leq K$. Furthermore, we choose $\ell_1(v_0)$ such that $W \in \ell_1(v_0)$. (Indeed, by assumption (v_0, W) is winning.)

We say that (v, ℓ) is winning if for all $L \in \ell$, we have that (v, L) is winning. For ℓ a sequence of left quotients and $a \in A$, we define $\ell \cdot a$ component-wise. Note that even if ℓ is infinite, it may be that $\ell \cdot a$ is finite.

We define the update function: $\mu(i, (v, v'))$ is a j as follows.

- If $\ell_i(v) \cdot c(v, v')$ is finite, denote $L \cdot c(v, v')$ its last element. Choose j such that $L \cdot c(v, v') \in \ell_j(v')$. Note that in general, such a j may not exist; it does exist if $(v', \ell_i(v) \cdot c(v, v'))$ is winning, and we will prove that this will always be the case when playing the strategy σ.
- If $\ell_i(v) \cdot c(v, v')$ is infinite, then choose j such that $\ell_j(v')$ has an infinite intersection with $\ell_i(v) \cdot c(v, v')$. Such a j exists without any assumption.

We define the next-move function ν (inducing σ). Let $v \in V_\exists$, and consider $(v, \ell_i(v))$. Let $L \in \ell_i(v)$, Eve wins from (v, L), so there exists an edge $(v, v') \in E$ such that $(v', L \cdot c(v, v'))$ is winning, we say that $(v, v') \in E$ is good for L. Since $\mathcal{W}(v')$ is upward closed, if $(v, v') \in E$ is good for L, then it is good for every L' such that $L \subseteq L'$. We argue that there exists an edge $(v, v') \in E$ that is good for all $L \in \ell_i(v)$, *i.e.* such that $(v', \ell_i(v) \cdot c(v, v'))$ is winning; define $\nu(v, i)$ to be this v'. There are two cases:

- Either $\ell_i(v)$ is finite, denote L its last element. Since $\ell_i(v)$ is decreasing, an edge good for L is good for all $L' \in \ell_i(v)$.
- Or $\ell_i(v)$ is infinite. The vertex v has finite degree, so there exists an edge which is good for infinitely many $L \in \ell_i(v)$. Since $\ell_i(v)$ is decreasing, it is good for all $L' \in \ell_i(v)$.

We show that the strategy σ is winning. Consider a play $\pi = (v_0, v_1) \cdot (v_1, v_2) \cdots$ consistent with σ, and $i_0 \cdot i_1 \cdots$ the sequence of memory states assumed along this play. Denote π_k the prefix of π of length k, we prove that for all k, there exists $L \in \ell_{i_k}(v_k)$ such that $L \subseteq c(\pi_k)^{-1} W$. Note that by definition, $(v_k, \ell_{i_k}(v_k))$ is winning, so $c(\pi_k)^{-1} W \neq \emptyset$.

We proceed by induction. For $k = 0$, it follows from $W \in \ell_1(v_0)$. Let $k > 0$, the induction hypothesis implies the existence of $L \in \ell_{i_{k-1}}(v_{k-1})$ such that $L \subseteq c(\pi_{k-1})^{-1} W$. We distinguish two cases, and denote $c_k = c(v_{k-1}, v_k)$.

- Either v_{k-1} belongs to Eve, then by construction of σ we have that $(v_k, \ell_{i_{k-1}}(v_{k-1}) \cdot c_k)$ is winning. It follows that the update function is well defined, and:
  1. If $\ell_{i_{k-1}}(v_{k-1}) \cdot c_k$ is finite, denote $L' \cdot c_k$ its last element, we have $L' \cdot c_k \in \ell_{i_k}(v_k)$. Since $L' \cdot c_k$ is the last element of $\ell_{i_{k-1}}(v_{k-1}) \cdot c_k$, it follows that $L' \subseteq L$. We have thus $L' \cdot c_k \subseteq L \cdot c_k$, so $L' \subseteq c(\pi_k)^{-1} W$, and $L' \cdot c_k \in \ell_{i_k}(v_k)$.
  2. If $\ell_{i_{k-1}}(v_{k-1}) \cdot c_k$ is infinite, $\ell_{i_k}(v_k)$ has an infinite intersection with $\ell_{i_{k-1}}(v_{k-1}) \cdot c_k$. So there exists $L' \subseteq L$ with $L' \in \ell_{i_{k-1}}(v_{k-1})$ such that $L' \cdot c_k$ is in this intersection. We have $L' \cdot c_k \in \ell_{i_k}(v_k)$ and $L' \subseteq c(\pi_k)^{-1} W$.

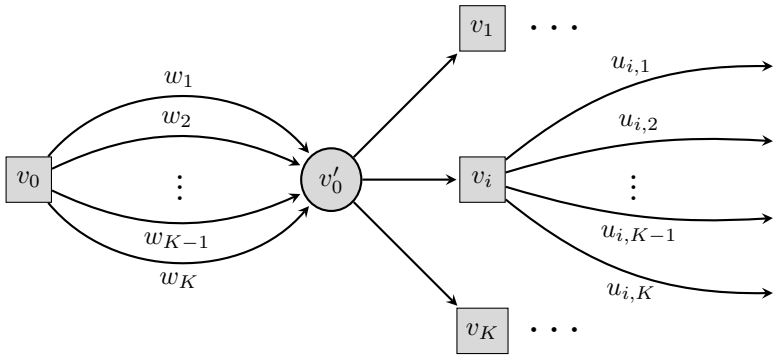

Figure 1 The lower bound.

- Or v_{k-1} belongs to Adam. Since Adam cannot escape $\mathcal{W}_E(\mathcal{G} \times \mathcal{M})$, we have that $(v_k, \ell_{i_{k-1}}(v_{k-1}) \cdot c_k)$ is winning, and the same reasoning concludes.

It follows that the strategy σ is winning, concluding the proof of Lemma 4. ∎

A Matching Lower Bound

▶ **Lemma 5** (Lower bound). *There exists a game $\mathcal{G} = (\mathcal{A}, W)$, and an initial vertex v_0 where Eve has a winning strategy, but no winning strategy using less than K memory states where K is the width of* $(\mathrm{Res}(W), \subseteq)$.

Consequently, $\mathrm{mem}(W)$ *is greater than or equal to the width of* $(\mathrm{Res}(W), \subseteq)$.

Proof. Consider $\{w_1^{-1}W, \ldots, w_K^{-1}W\}$ an antichain of left quotients of W. For $i \neq j$, there exists $u_{i,j} \in A^\omega$ such that $u_{i,j} \in w_i^{-1}W$ and $u_{i,j} \notin w_j^{-1}W$.

We describe the game, illustrated in Figure 1. A play consists in three steps:

1. From v_0 to v'_0: Adam chooses a word in $\{w_1, \ldots, w_K\}$;
2. Eve chooses between K options;
3. say Eve chose the i^{th} option, then Adam chooses between the $K-1$ words $u_{i,j}$ for $j \neq i$.

We first show that Eve has a winning strategy from v_0, using K memory states. It consists in choosing the i^{th} option whenever Adam chooses the word u_i: whatever Adam chooses at the third step, $w_i \cdot u_{i,j} \in W$.

We now show that there exists no winning strategy using less than K memory states. Indeed, such a strategy will not comply with the above strategy and for some $i \neq j$, choose the j^{th} option if Adam chooses w_i. Then Adam wins by playing $u_{i,j}$, since $w_i \cdot u_{i,j} \notin W$. ∎

Tight Bounds

Putting together upper and lower bounds, we proved the following result:

▶ **Theorem 6.** *For all safety conditions W,* $\mathrm{mem}(W)$ *is the width of* $(\mathrm{Res}(W), \subseteq)$.

A condition W is half-positional if $\mathrm{mem}(W) = 1$. Characterizing half-positional conditions has been a fruitful topic over the last few years [8, 11]. In the case of safety conditions, we obtain the following characterization:

▶ **Corollary 7.** *For all safety conditions W, W is half-positional if and only if the inclusion is a linear order over* $\mathrm{Res}(W)$.

4 Examples and Applications

In this section, we instantiate Theorem 6 on different examples. We chose four examples:

- The outbidding condition shows the difference between graphs with finite degree and graphs with infinite degree; in particular, it gives a counter example to Lemma 4 when dropping the finite degree assumption,
- The energy condition is a non-regular half-positional safety condition,
- The generalized safety condition is a regular safety condition for which the partially ordered set of left quotients has a nice well-known combinatorical structure,
- The boundedness condition is a central piece in the theory of regular cost functions.

When representing the partial order $(\mathrm{Res}(W), \subseteq)$ for a given W, we use the following convention: a black edge from L to L' means that $L \subseteq L'$, and a red edge labeled a from L to L' means that $L' = L \cdot a$, so the red structure is the minimal (although possibly infinite) deterministic automaton recognizing W.

Outbidding Games

Let $A = \{a, b, c\}$ and $W = \{a^n \cdot b^p \cdot c^\omega \mid n \leq p\} \cup \{a^\omega\} \cup a^* \cdot b^\omega$. It is a non-regular safety condition, called the outbidding condition. The figure 2 represents the partial order $(\mathrm{Res}(W), \subseteq)$. Its width is three: there are two incomparable infinite increasing sequences of left quotients, $((a^n)^{-1}W)_{n\in\mathbb{N}}$ and $((b \cdot a^n)^{-1}W)_{n\in\mathbb{N}}$, and $c^{-1}W$.

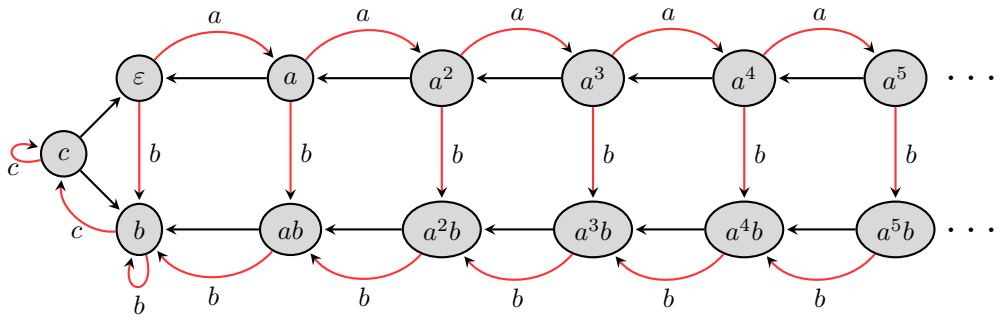

Figure 2 The outbidding condition: more b's than a's.

Hence thanks to Theorem 6, $\mathrm{mem}(W) = 3$. However, there exists an outbidding game where Eve wins but needs infinite memory for this. This does not contradict Theorem 6, as this game, represented in Figure 3, has a vertex of infinite degree. It goes as follows: first Adam picks a number n, and then Eve takes over: she has to pick a number p, higher than or equal to n. A finite memory strategy can only choose from finitely many options, hence cannot win against all strategies of Adam.

Energy Games

The setup for the energy condition is the following: assume we are monitoring a resource. We denote by A the set of actions on this resource, which is any monotonic function $f : \mathbb{N} \to \mathbb{N}$, as for instance:

- consuming one unit of the resource,
- reloading by one unit,

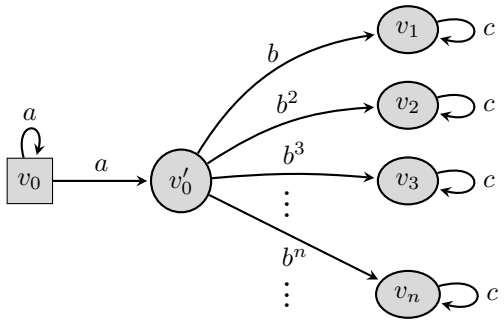

Figure 3 An outbidding game with infinite degree where Eve needs infinite memory to win.

- emptying the resource,
- consuming half of the current energy level.

Define the energy condition by

$$W = \{w = w_0 w_1 \cdots \mid \text{ the energy level remains always non-negative}\} .$$

It is a non-regular safety condition. Energy games and several variants have been extensively studied [1, 3, 2]. The Figure 4 represents the partial order $(\mathrm{Res}(W), \subseteq)$, with only two actions: a reloads by one unit, and b consumes one unit. In general, if the actions are monotonic, then the left quotients are totally ordered by inclusion, so thanks to Theorem 6 we have $\mathrm{mem}(W) = 1$.

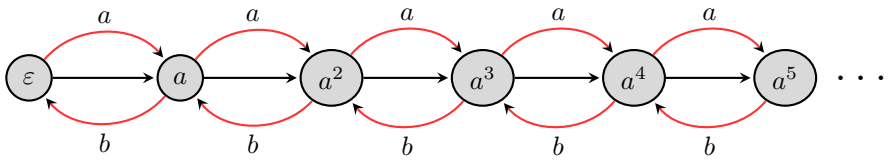

Figure 4 The energy condition: always more a's than b's.

▶ **Corollary 8.** *The energy games are half-positional.*

Generalized Safety Games

This example originates from the study of generalized reachability games [6, 7]. A generalized reachability condition is a (finite) conjunction of reachability conditions. Here we take the opponent's vantage point: a generalized safety condition is a (finite) disjunction of (internal) safety conditions. Specifically, let $A = \{\bot, 1, \ldots, k\}$: each letter is a color, and $\bot$ is uncolored. Let $W = \{w = w_0 w_1 \cdots \mid \exists i \in \{1, \ldots, k\}, \forall n, w_n \neq i\}$, it is satisfied if at least one color is not seen along the play. It is a safety condition. The figure 4 represents the partial order $(\mathrm{Res}(W), \subseteq)$ for $k = 3$. The left quotients are all the strict subsets of $\{1, \ldots, k\}$. The width of this partial order is $\binom{k}{\lfloor k/2 \rfloor}$, according to the well known Sperner's Lemma from combinatorics.

▶ **Corollary 9.** *For all generalized safety games with k colors, if Adam has a winning strategy, then he has a winning strategy with $\binom{k}{\lfloor k/2 \rfloor}$ memory states.*

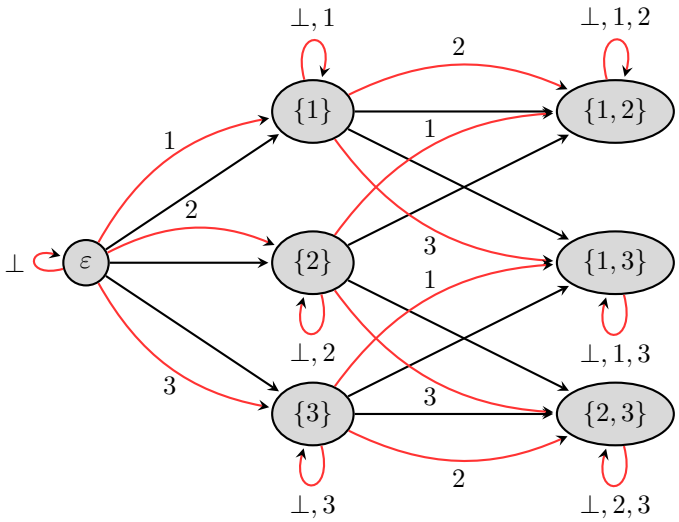

Figure 5 The generalized safety condition.

Furthermore, for all k, there exists a generalized safety game with k colors where Adam has a winning strategy using $\binom{k}{\lfloor k/2 \rfloor}$ memory states, but none using less memory states.

Games with Counters

This example originates from the theory of regular cost functions [4]. Let $N \in \mathbb{N}$, and define the boundedness condition W_N involving a counter as follows:

$$W_N = \{w \mid \text{the counter value in } w \text{ remains bounded by } N\} .$$

For the set of actions, we consider any monotonic action (even non-regular), *i.e.* function $f : \mathbb{N} \to \mathbb{N}$ such that if $i \leq j$ then $f(i) \leq f(j)$, as for instance:

- leaving the counter value unchanged,
- incrementing the counter value by one,
- resetting the counter value to zero,
- dividing the counter value by two, rounded down,
- increasing the counter value to the next power of two.

The condition W_N is a regular safety condition.

▶ **Corollary 10.** *The boundedness games are half-positional.*

Conclusion and Perspectives

We considered general safety conditions and characterized their memory requirements. Specifically, the memory requirements of a safety condition W is the width of the partially ordered set $(\mathrm{Res}(W), \subseteq)$. This is the first general result characterizing the memory requirements for *some* non-regular conditions, based on their topological properties. We hope that this is the first stone on the path of memory requirements characterizations for much more conditions.

References

1 Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jirí Srba. Infinite runs in weighted timed automata with energy constraints. In Franck Cassez and Claude Jard, editors, *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.

2 Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource interfaces. In Rajeev Alur and Insup Lee, editors, *EMSOFT*, volume 2855 of *Lecture Notes in Computer Science*, pages 117–133. Springer, 2003.

3 Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theor. Comput. Sci.*, 458:49–60, 2012.

4 Thomas Colcombet. Regular cost functions, part I: Logic and algebra over words. *Logical Methods in Computer Science*, 9(3), 2013.

5 Stefan Dziembowski, Marcin Jurdziński, and Igor Walukiewicz. How much memory is needed to win infinite games? In *LICS*, pages 99–110. IEEE Computer Society, 1997.

6 Nathanaël Fijalkow and Florian Horn. The surprizing complexity of reachability games. *CoRR*, abs/1010.2420, 2010.

7 Nathanaël Fijalkow and Florian Horn. Les jeux d'accessibilité généralisée. *Technique et Science Informatiques*, 32(9-10):931–949, 2013.

8 Hugo Gimbert. *Jeux Positionnels*. PhD thesis, Université Paris 7, 2007.

9 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

10 Eryk Kopczyński. Personal communication.

11 Eryk Kopczyński. *Half-Positional Determinacy of Infinite Games*. PhD thesis, University of Warsaw, 2009.

12 Orna Kupferman. Variations on safety. In Erika Ábrahám and Klaus Havelund, editors, *TACAS*, volume 8413 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2014.