

State Complexity of Chromatic Memory in Infinite-Duration Games

Alexander Kozachinskiy*

January 25, 2022

Abstract

A major open problem in the area of infinite-duration games is to characterize winning conditions that are determined in finite-memory strategies. Infinite-duration games are usually studied over edge-colored graphs, with winning conditions that are defined in terms of sequences of colors. In this paper, we investigate a restricted class of finite-memory strategies called *chromatic* finite-memory strategies. While general finite-memory strategies operate with sequences of edges of a game graph, chromatic finite-memory strategies observe only colors of these edges.

Recent results in this area show that studying finite-memory determinacy is more tractable when we restrict ourselves to chromatic strategies. On the other hand, as was shown by Le Roux (CiE 2020), determinacy in general finite-memory strategies implies determinacy in chromatic finite-memory strategies. Unfortunately, this result is quite inefficient in terms of the state complexity: to replace a winning strategy with few states of general memory, we might need much more states of chromatic memory. The goal of the present paper is to find out the exact state complexity of this transformation.

For every winning condition and for every game graph with n nodes we show the following: if this game graph has a winning strategy with q states of general memory, then it also has a winning strategy with $(q + 1)^n$ states of chromatic memory. We also show that this bound is almost tight. For every q and n , we construct a winning condition and a game graph with $n + O(1)$ nodes, where one can win with q states of general memory, but not with $q^n - 1$ states of chromatic memory.

1 Introduction

We study deterministic, two-player, turn-based, infinite-duration games on finite arenas. An arena is a finite directed graph whose set of nodes V is partitioned into two subsets

*kozlach@mail.ru

$V_0, V_1 \subseteq V$. A game over such an arena is played as follows. Two players (in this paper, we call them Player 0 and Player 1) are traveling over the nodes of the graph. When they are in a node from V_0 , Player 0 decides where to go next. Similarly, when they are in a node from V_1 , Player 1 decides where to go next. Players can move only along edges of the graph.

After infinitely many turns, this process produces an infinite path in our graph. A *winning condition* determines, which infinite paths are winning for Player 0 and which are for Player 1. In the literature, one usually defines winning conditions through *colors*. Namely, fix a set C of colors. Then color edges of an arena into elements of C . Once this is done, any subset $W \subseteq C^\omega$ defines a winning condition in the following way. Take an infinite path in our graph. If its “coloring” (that is, the sequence of colors of its edges) belongs to W , we make this path winning for Player 0. Otherwise, we make this path winning for Player 1. In this paper, we consider only winning conditions that are defined through colors.

A utility of colors is that we do not have to define a winning condition for each arena separately. Indeed, once we fix a set of colors C and a subset $W \subseteq C^\omega$, we can treat W as a winning condition in any arena with edges colored by elements of C .

A major open problem in this area is to characterize winning conditions that are determined in *finite-memory strategies*. This is motivated by applications to reactive synthesis [13], where infinite-duration games on graphs model an interaction between a system and the environment. Finite-memory determinacy over *infinite* arenas has also been used in decidability of logical theories [12].

Let us discuss the concept of a finite-memory strategy in more detail. The games we are studying are of infinite duration. This means that as a game goes on, the players need more and more memory to remember everything what has happened so far. However, a player might have a smart strategy which does not require the whole “transcript” of the previous development of the game. For instance, a strategy might only require to know whether the number of steps made so far is even or odd. In this case, 1 bit of memory is sufficient to “implement” this strategy (we simply reverse the value of this bit each time we make a step). In general, if there exists a constant $k > 0$ such that our strategy never stores more than k bits of information during the play, then we call this strategy finite-memory.

More formally, finite-memory strategies store information through *memory structures*. A memory structure \mathcal{M} is a finite automaton whose input alphabet is the set of edges of an arena. We call a strategy of one of the players an \mathcal{M} -*strategy* if this strategy stores information according to \mathcal{M} . More specifically, one can imagine that, say, Player 0 carries \mathcal{M} with them during the play. In each turn, Player 0 takes the edge along which the game goes on in this turn, and feeds this edge into \mathcal{M} . Then, in each moment of the game, the current state of \mathcal{M} informally represents the memory of Player 0. If a strategy S of this player never requires anything besides the current state of \mathcal{M} , then we call S an \mathcal{M} -strategy.

We call S a *finite-memory strategy* if it is an \mathcal{M} -strategy for some memory structure \mathcal{M} . If a memory structure \mathcal{M} has q states, then we call any \mathcal{M} -strategy a q -state

strategy.

We say that a winning condition $W \subseteq C^\omega$ is *finite-memory determined* for Player $i \in \{0, 1\}$ if in any arena, where this player has a winning strategy w.r.t. W , this player also has a finite-memory winning strategy. In practice, we are interested in finite-memory winning strategies with as few states as possible. This leads to a concept of *memory complexity*. To define it, we first have to fix a parametrization of arenas by natural numbers. For every $n \in \mathbb{N}$, there has to be only finitely many arenas where the value of the parameter does not exceed n . When the set of colors is finite, one natural example of such a parameter is the number of nodes. Now, the memory complexity of a finite-memory determined winning condition W for Player $i \in \{0, 1\}$ is the following function. For every $n \in \mathbb{N}$, it maps n to smallest q such that in all arenas, where Player i can win w.r.t. W and where the value of the parameter does not exceed n , Player i has a q -state winning strategy.

There has been the following progress in the problem of characterizing finite-memory determined winning conditions. In 2005, Gimbert and Zielonka [7] obtained a sufficient and necessary condition for determinacy in *memory-less strategies* (that is, in strategies that are defined over “useless” 1-state memory structures). Since then, several conditions that are sufficient (but not necessary) for finite-memory determinacy were established in the literature [11, 4].

Recently, there have been advances in studying *chromatic* memory complexity. In contrast to the general memory complexity, the chromatic memory complexity takes into account only so-called chromatic strategies (we define this kind of strategies in the next subsection). Bouyer et al. [3] obtained a complete characterization of winning conditions (and, more generally, of preference relations) with constant chromatic memory complexity. It is open to extend this result to winning conditions with constant¹ general memory complexity. This motivates a study of the relationship between chromatic and general memory complexities – as, potentially, there might be a way of converting results about chromatic memory into results about general memory. The present paper is devoted to this kind of questions.

1.1 Chromatic strategies

A chromatic strategy is a strategy which, to make its moves, only needs to know the sequence of colors along a current play. That is, chromatic strategies use only a part of the description of a current play; the full description consists of the sequence of edges that were played so far.

We are interested in chromatic strategies that are additionally finite-memory. Such strategies can be defined through chromatic memory structures. A memory structure

¹As we mentioned, memory complexity depends on the parametrization of the arenas. However, the class of winning conditions with constant memory complexity is independent of the choice of the parametrization. Indeed, these are winning conditions for which there exists a constant q such that all arenas require at most q states of memory.

is chromatic if its transition function does not distinguish edges of the same color. Alternatively, a chromatic memory structure can be viewed as a finite automaton whose input alphabet is not the set of edges, but rather the set of colors. That is, each time we feed an edge into a chromatic memory structure, it only takes as an input the color of this edge.

In this paper, we study the following problem. Fix a winning condition and an arena. Suppose that Player i has a winning strategy with q states of general memory in this arena. What is the minimal Q such that Player i has a winning strategy with Q states of chromatic memory in this arena?

It is not even obvious whether Q is always finite (in other words, whether determinacy in general finite-memory strategies implies determinacy in chromatic finite-memory strategies). The finiteness of Q was proved by Le Roux [10]. In this paper, we obtain a tight bound on Q in terms of q and the number of nodes of an arena. But before presenting this in more detail, let us give an example where Q is strictly larger than q .

1.2 Example

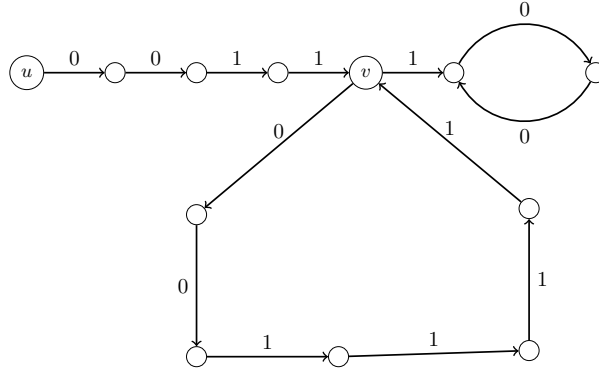


Figure 1: Example.

In this example, the set of colors is $\{0, 1\}$, the starting node is u , and Player 0 is the one to move everywhere. Assume that the goal of Player 0 is to maximize the maximum number of consecutive 1's in the play. The best number Player 0 can achieve is 5. Namely, Player 0 can go to v , then to the loop and, after returning to v , to the right. Note that we do different things at v when we are there for the first time and when for the second time. That is, the content of our memory has to be different at these moments. This can be realized with a 2-state memory structure which changes its state from “has not been to v ” to “has been to v ” after seeing any edge from v . However, such a memory structure is not chromatic – it does different things for edges of the same color.

In fact, one can show that to attain 5 in this problem, we require $Q \geq 3$ states of chromatic memory. Indeed, when we first come to v , we have to go to the loop – otherwise the maximum number of consecutive 1's will be 3. Now, when we return to v ,

our memory has to be different from what it was the first time – otherwise we will stay on the loop forever, and the maximum number of consecutive 1’s will be 4. Therefore, our chromatic memory structure has to come into different states on 0011 and on 0011001111. It can be checked via a computer search that no 2-state deterministic automaton over the alphabet $\{0, 1\}$ can distinguish these two words.

1.3 Our contribution

As we mentioned, Le Roux has shown [10] that determinacy in general finite-memory strategies implies determinacy in chromatic finite-memory strategies. More specifically, Le Roux obtained the following bound. Fix an arena \mathcal{A} with n nodes and a winning condition W . Assume that Player 0 has in \mathcal{A} a q -state strategy which is winning w.r.t. W . Then Player 0 also has in \mathcal{A} a $2^{q(n^2+1)}$ -state chromatic strategy which is winning w.r.t. W .

In fact, Le Roux obtained this as a corollary of a more general result. Namely, he studied a problem of the uniformization of winning strategies. In this problem, given an equivalence relation on the set of finite paths of the arena, we want a winning strategy which does the same thing on paths from the same equivalence class. In our case, the equivalence relation is “to have the same sequence of colors and the same endpoint”. Let us also mention that Le Roux obtained this result for a more general class of games than we consider in this paper – for deterministic *concurrent* games. In these games, in each turn two players make two moves *simultaneously*, and then one computes the next node via some predetermined transition function.

The summary of our results. First, we obtain an improvement of the aforementioned result of Le Roux. Namely, for any winning condition W and for any arena \mathcal{A} with n nodes we show the following: if Player 0 has in \mathcal{A} a q -state winning strategy w.r.t. W , then Player 0 also has in \mathcal{A} a $(q + 1)^n$ -state chromatic winning strategy w.r.t. W . Of course, the same result holds for Player 1. We also provide an analog of this result for preference relations, with slightly worse bound $Q \leq (qn + 1)^n$. Finally, we show that our upper bound is essentially tight. Namely, for every n and q we provide a winning condition W and an arena \mathcal{A} with $n + 3$ nodes, where Player 0 has a winning q -state strategy w.r.t. W , but no winning chromatic strategy with less than q^n states.

In fact, our upper bounds hold for deterministic concurrent games as well. However, to simplify the exposition, we prove them only for turn-based games.

What consequences does it have for the memory complexity? The bound of Le Roux means that the chromatic memory complexity is at most exponential in the general memory complexity and in the number of nodes. Our improvement is a removal of the exponential dependency on the general memory complexity. Still, there is an exponential dependency on the number of nodes. Note that this exponential dependency turns into just a linear dependency if we measure memory not in states but in bits. As a consequence, we obtain that for a class “PSPACE” of winning conditions with polynomial bit-memory complexity (in the number of nodes), it is unimportant whether we use chromatic or general strategies.

Unfortunately, our lower bound does not provide any separation between the chromatic and the general memory complexity. We only provide an exponential separation between the chromatic and the general memory in individual arenas, using different winning conditions for different arenas. It is not clear how to combine these winning conditions into a single one for which the general memory grows more slowly than the chromatic memory. We return to this problem in the next subsection.

1.4 Other related works and open problems

Chromatic finite-memory strategies were first studied by Kopczyński [9] in 2008. He introduced the following notation for winning conditions with constant memory complexity. For $i \in \{0, 1\}$ and for a winning condition $W \subseteq C^\omega$, let $\text{mm}_i(W)$ denote the minimal q such that for every finite arena the following holds: if Player i has a winning strategy w.r.t. W in this arena, then Player i has also a q -state winning strategy w.r.t. W in this arena. If no such q exists, then we set $\text{mm}_i(W) = +\infty$. That is, W has constant memory complexity for Player i if and only if $\text{mm}_i(W)$ is finite. Kopczyński also introduced an analogue of this notation for chromatic strategies. Namely, let $\text{mm}_i^X(W)$ denote the minimal q such that for every finite arena the following holds: if Player i has a winning strategy w.r.t. W in this arena, then Player i has also a chromatic q -state winning strategy w.r.t. W in this arena. Again, if no such q exists, we set $\text{mm}_i^X(W) = +\infty$. Clearly, we have $\text{mm}_i(W) \leq \text{mm}_i^X(W)$ for every winning condition W .

It is classical [5] that all ω -regular winning conditions $W \subseteq C^\omega$ have constant chromatic memory complexity: $\text{mm}_0^X(W) < +\infty, \text{mm}_1^X(W) < +\infty$. Kopczyński gave an exponential-time algorithm which computes $\text{mm}_i^X(W)$ for an ω -regular W , provided that W is given as a deterministic parity automaton. He then conjectured that $\text{mm}_i^X(W) = \text{mm}_i(W)$ for every winning condition W . His motivation was as follows: if this equality holds, then we automatically have an exponential-time algorithm to compute $\text{mm}_i(W)$ for an ω -regular W .

As we mentioned, in 2020 Bouyer et al. [3] obtained a *complete characterization* of winning conditions with constant chromatic memory complexity for both players, that is, of winning conditions satisfying $\text{mm}_0^X(W) < +\infty, \text{mm}_1^X(W) < +\infty$. Broadly speaking, they gave a graph-theoretic sufficient and necessary condition for determinacy with constant chromatic memory complexity. In other words, they reduced a game-theoretic reasoning in this case to a reasoning about graphs.

This result motivates the following modification of the Kopczyński's conjecture. Is it true for every winning condition W that $\text{mm}_i(W) < +\infty \iff \text{mm}_i^X(W) < +\infty$? In other words, is it true that a winning condition with constant general memory complexity also has constant chromatic memory complexity? If that were true, we would have had a complete characterization of winning conditions with constant general memory complexity, due to results of Bouyer et al.

Recently, Casares [6] disproved the original conjecture of Kopczyński. More specifically, for every n he constructed a winning condition $W_n \subseteq \{1, 2, \dots, n\}^\omega$ such that $\text{mm}_0^X(W_n) = n$ and $\text{mm}_0(W_n) = 2$. This winning condition is arranged as follows. We

put $\alpha \in \{1, 2, \dots, n\}^\omega$ into W_n if and only if there are exactly two elements of $\{1, 2, \dots, n\}$ that occur infinitely often in α . Let us note that W_n is ω -regular.

One can notice that this result of Casares also gives an example of a winning condition W with constant general memory complexity, but super-constant chromatic memory complexity. Thus, it answers negatively to our modification of the Kopczyński's conjecture. However, this W is over an *infinite* set of colors. Namely, set $C = \mathbb{N}$. Let $W \subseteq \mathbb{N}^\omega$ be the set of all $\alpha \in \mathbb{N}^\omega$ such that there are exactly 2 elements of \mathbb{N} occurring infinitely often in α . On one hand, W is at least as hard (in terms of memory requirements) as W_n , for every n , so we have $\text{mm}_0^x(W) = +\infty$. On the other hand, since any finite arena involves only finitely many colors, and since $\text{mm}_0(W_n) = 2$ for every n , we have $\text{mm}_0(W) = 2$.

It is open if there exists a winning condition W over a *finite* set of colors with $\text{mm}_0(W) < +\infty$ and $\text{mm}_0^x(W) = +\infty$. Let us note here that such a separation can be easily obtained for *one-player* games (that is, for games over arenas where one of the players controls all the nodes), via the example from subsection 1.2. Namely, let the set of colors be $\{0, 1\}$, and assume that the goal of Player 0 is to maximize the maximum number of consecutive 1's in a sequence of colors. It is not hard to see that Player 0 can play optimally with just 2 states of memory in every one-player arena. An idea is that we first have to reach the starting point of the longest all-ones path p , and then we have to go through p . We may assume that p is simple because otherwise there is a simple all-ones cycle which allows us to win trivially. There is a slight problem that the path to the starting node of p and the path p itself may have common nodes, and we have to do different things at these nodes at different moments. So we just have to remember whether we have already been to the starting node of p or not. However, this kind of memory is essentially non-chromatic. In fact, by considering arenas as on Figure 1, one can show that there is no constant q such that Player 0 can play optimally with q states of chromatic memory in all one-player arenas. Unfortunately, this example has super-constant general memory complexity in two-player arenas, so it does not provide a separation between the chromatic and the general memory complexity in the two-player regime.

The question of the relationship between the chromatic and the general memory is not settled in infinite arenas. In contrast to finite arenas, it is unclear for infinite arenas whether determinacy in general finite-memory strategies implies determinacy in chromatic finite-memory strategies. Note that our upper bounds are inapplicable as they involve the number of nodes. It was recently shown by Bouyer, Randour and Vandenhove [2] that the class of winning conditions that are chromatic finite-memory determined in infinite arenas coincides with the class ω -regular conditions.

Finally, it would be interesting to extend our upper bounds on chromatic memory to *stochastic* games. Seemingly, our proofs do not work in the stochastic setting, so we require new ideas here. It is worth to mention in this context a paper [1], studying winning conditions that have constant chromatic memory complexity in stochastic games.

The rest of the paper is organized as follows. In the next section we give preliminaries. In Section 3 we give the exact statements of our results, equipped with brief overviews

of the proofs. Complete proofs of our results are given in the subsequent sections.

2 Preliminaries

Notation. For a set A , we let A^* (resp., A^ω) stand for the set of all finite (resp., infinite) sequences of elements of A . For $x \in A^*$, we let $|x|$ denote the length of x (we also set $|x| = +\infty$ for $x \in A^\omega$). We write $A = B \sqcup C$ for three sets A, B, C when $A = B \cup C$ and $B \cap C = \emptyset$. We let \circ denote the function composition. The set of positive integral numbers is denoted by \mathbb{Z}^+ .

2.1 Arenas

Definition 1. Let C be any set. A tuple $\mathcal{A} = \langle V, V_0, V_1, E \rangle$, where V, V_0, V_1, E are four finite sets such that $V = V_0 \sqcup V_1$ and $E \subseteq V \times C \times V$, is called an **arena** over the set of colors C if for every $s \in V$ there exist $c \in C$ and $t \in V$ such that $(s, c, t) \in E$.

Elements of V will be called nodes of \mathcal{A} . Elements of V_0 (resp., V_1) will be called nodes of Player 0 (resp., Player 1). Elements of E will be called edges of \mathcal{A} . For an edge $e = (s, c, t) \in E$ we define $\mathbf{source}(e) = s$, $\mathbf{col}(e) = c$ and $\mathbf{target}(e) = t$. We imagine $e \in E$ as an arrow which is drawn from the node $\mathbf{source}(e)$ to the node $\mathbf{target}(e)$ and which is colored by $\mathbf{col}(e)$. The out-degree of a node $s \in V$ is the number of $e \in E$ with $s = \mathbf{source}(e)$. By definition of an arena, the out-degree of every node must be positive.

We extend the function \mathbf{col} to a function $\mathbf{col}: E^* \cup E^\omega \rightarrow C^* \cup C^\omega$ by setting:

$$\mathbf{col}(e_1 e_2 e_3 \dots) = \mathbf{col}(e_1) \mathbf{col}(e_2) \mathbf{col}(e_3) \dots, \quad e_1, e_2, e_3, \dots \in E.$$

A non-empty sequence $p = e_1 e_2 e_3 \dots \in E^* \cup E^\omega$ is called a path if for any $1 \leq i < |p|$ we have $\mathbf{target}(e_i) = \mathbf{source}(e_{i+1})$. We set $\mathbf{source}(p) = \mathbf{source}(e_1)$ and, if p is finite, $\mathbf{target}(p) = \mathbf{target}(e_{|p|})$. For technical convenience, with every node $v \in V$ we associate a 0-length path λ_v , for which we set $\mathbf{source}(\lambda_v) = \mathbf{target}(\lambda_v) = v$.

2.2 Strategies

Let $\mathcal{A} = \langle V, V_0, V_1, E \rangle$ be an arena over the set of colors C . A strategy of Player 0 in \mathcal{A} is any function

$$S: \{p \mid p \text{ is a finite path in } \mathcal{A} \text{ with } \mathbf{target}(p) \in V_0\} \rightarrow E,$$

such that for every p from the domain of S we have $\mathbf{source}(S(p)) = \mathbf{target}(p)$.

Intuitively, finite paths in \mathcal{A} represent possible developments of the game in \mathcal{A} . When the game starts in a node $s \in V$, the starting position² is represented by the 0-length path λ_s . Player 0 is the one to move after a finite path p if and only if $t = \mathbf{target}(p)$

²We stress that in our setting, the starting node is not fixed. The same strategy S can be applied for different starting nodes.

is a node of Player 0. In this situation, Player 0 must choose some edge from t . We then append this edge to p . A strategy of Player 0 fixes the choices of Player 0 in all situations when this player is the one to move.

In this paper, we only mention strategies of Player 0, but, of course, strategies of Player 1 can be defined similarly.

Let us define, what does it mean that a path is consistent with a strategy S of Player 0. First, any 0-length path λ_v is consistent with S . Now, a non-empty path $p = e_1e_2e_3\dots$ (which may be finite or infinite) is consistent with S if the following holds:

- if $\text{source}(p) \in V_0$, then $e_1 = S(\lambda_{\text{source}(p)})$;
- for any $1 \leq i < |p|$, if $\text{target}(e_i) \in V_0$, then $e_{i+1} = S(e_1e_2\dots e_i)$.

For any node $v \in V$ and for any strategy S of Player 0, we let $\text{col}(S, v) \subseteq C^\omega$ be the set of all $\text{col}(p)$ over all infinite paths p such that $\text{source}(p) = v$ and p is consistent with S . Less formally, $\text{col}(S, v)$ is the set of all infinite sequences of colors that can be obtained in a play with S from the node v . For $U \subseteq V$, we define $\text{col}(S, U) = \bigcup_{v \in U} \text{col}(S, v)$.

2.3 Winning conditions and preference relations

A winning condition is any set $W \subseteq C^\omega$. We say that a strategy S of Player 0 is winning from $u \in V$ w.r.t. to W if $\text{col}(S, u) \subseteq W$. In other words, any infinite play from u against S must give a sequence of colors belonging to W .

We also consider a more general class of objectives called *preference relations*. A preference relation is a total preorder \sqsubseteq on the set C^ω . Intuitively, when given a preference relation \sqsubseteq , the goal of Player 0 is to maximize the sequence of colors in the play w.r.t. \sqsubseteq .

Any two strategies S_1, S_2 of Player 0 can be compared w.r.t. \sqsubseteq (from the perspective of Player 0). Namely, we say that S_1 is *better* than S_2 from $u \in V$ if there exists $\beta \in \text{col}(S_2, u)$ such that $\alpha \not\sqsubseteq \beta$ for every $\alpha \in \text{col}(S_1, u)$. This means that there exists a play against S_2 which is strictly worse from the viewpoint of Player 0 than any play against S_1 . Correspondingly, we say that S_2 is *at least as good* as S_1 from $u \in V$ if it is not true that S_1 is better than S_2 from u , that is, if for every $\beta \in \text{col}(S_2, u)$ there exists $\alpha \in \text{col}(S_1, u)$ such that $\alpha \sqsubseteq \beta$.

It is straightforward to check that for any $u \in V$, the relation “at least as good from u ” is a total preorder on the set of strategies of Player 0.

2.4 Memory structures

Let $\mathcal{A} = \langle V, V_0, V_1, E \rangle$ be an arena over the set of colors C . A memory structure in \mathcal{A} is a tuple $\mathcal{M} = \langle M, m_{\text{init}}, \delta \rangle$, where M is a finite set, $m_{\text{init}} \in M$ and $\delta: M \times E \rightarrow M$. In other words, a memory structure \mathcal{M} is a deterministic finite automaton whose input alphabet is the set of edges of \mathcal{A} . Thus, M serves as the set of states of our memory structure,

m_{init} serves as its initial state, and δ as its transition function. Given $m \in M$, we inductively extend the function $\delta(m, \cdot)$ to arbitrary finite sequences of edges as follows:

$$\begin{aligned}\delta(m, \text{empty sequence}) &= m, \\ \delta(m, se) &= \delta(\delta(m, s), e), \quad s \in E^*, e \in E.\end{aligned}$$

Thus, $\delta(m, s)$ for $s \in E^*$ is the state into which our memory structure comes from the state m after reading the sequence s .

We say that a memory structure $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ is chromatic if there exists a function $\sigma: M \times C \rightarrow M$ such that $\delta(m, e) = \sigma(m, \text{col}(e))$ for every $e \in E$ and $m \in M$. In other words, a chromatic memory structure does not distinguish edges of the same color. Correspondingly, it will be sometimes convenient to view chromatic memory structures as finite automata over the set C (and not over the set of edges of \mathcal{A}).

Let S be a strategy of Player 0 and $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ be a memory structure. We say that S is an \mathcal{M} -strategy if for any two paths p_1, p_2 with $\text{target}(p_1) = \text{target}(p_2) \in V_0$ it holds that:

$$\delta(m_{init}, p_1) = \delta(m_{init}, p_2) \implies S(p_1) = S(p_2)$$

In other words, the value of $S(p)$ for an \mathcal{M} -strategy S solely depends on $\text{target}(p)$ and on a state of \mathcal{M} after p (assuming that initially \mathcal{M} is in m_{init}). Thus, with any \mathcal{M} -strategy S one can associate the *next-move* function of S . This is a function $n_S: V_0 \times M \rightarrow E$ defined as follows: to determine $n_S(v, m)$, we take an arbitrary finite path p with $\text{target}(p) = v$ and $\delta(m_{init}, p) = m$, and set $n_S(v, m) = S(p)$. If there is no such path p at all, we define $n_S(v, m)$ arbitrarily. Less formally, $n_S(v, m)$ is the move of S from the node v when the state of S is m . Note that the next-move function completely determines the corresponding strategy. For the sake of brevity, in the paper we will use the same letter for a strategy and for its next-move function. That is, when S is an \mathcal{M} -strategy, we use the letter S in two different ways. First, $S(p)$ denotes the move of S after a finite path p . Second, $S(v, m)$ for $v \in V_0, m \in M$ denotes the value of the next-move function of S on the pair (v, m) .

For brevity, if S is an \mathcal{M} -strategy, we let *the state of S after a finite path p* stand for $\delta(m_{init}, p)$, where δ is the transition function of \mathcal{M} .

We say that S is a q -state strategy if S is an \mathcal{M} -strategy for some memory structure $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ with $|M| = q$. If \mathcal{M} is additionally chromatic, then we say that S is a chromatic q -state strategy.

3 Exact Statements of the Results and Overviews of the Proofs

The exact statement of our main upper bound is the following

Theorem 1. *For any $n, q \in \mathbb{Z}^+$, for any arena $\mathcal{A} = \langle V, V_0, V_1, E \rangle$ with n nodes, for any set $U \subseteq V$, and for any q -state strategy S_1 of Player 0 in \mathcal{A} , there exists a chromatic $(q + 1)^n$ -state strategy S_2 of Player 0 such that $\text{col}(S_2, U) \subseteq \text{col}(S_1, U)$.*

It should not be confusing that this theorem does not mention winning conditions. One can notice that $\text{col}(S_1, U)$ is the minimal winning condition w.r.t. which S_1 is winning from all nodes of U . So, if S_1 and S_2 are as in Theorem 1, and W is a winning condition w.r.t. which S_1 is winning from all nodes of U , then S_2 is also winning w.r.t. W from all nodes of U . That is, we obtain the following corollary.

Corollary 2. *Let $W \subseteq C^\omega$ be any winning condition. Then for any $n, q \in \mathbb{Z}^+$ the following holds. Take any n -node arena \mathcal{A} and any q -state strategy S_1 of Player 0 in \mathcal{A} . Then in \mathcal{A} there exists a chromatic $(q + 1)^n$ -state strategy S_2 of Player 0 such that for any node v of \mathcal{A} the following holds: if S_1 is winning from v w.r.t. W , then so is S_2 .*

Proof. Apply Theorem 1 to the set U of nodes from where S_1 is winning w.r.t. W . \square

Let us now give a proof sketch of Theorem 1. Its full proof is given in Section 4

Proof sketch of Theorem 1. At each moment, S_2 stores a function $f: V \rightarrow M \cup \{\perp\}$, where M is the set of states of S_1 and $\perp \notin M$ is a special symbol meaning “undefined”. There are $(q + 1)^n$ such functions, so S_2 will have $(q + 1)^n$ states. Strategy S_2 maintains the following two invariants called *soundness* and *completeness*:

- (*soundness*) for any $v \in V$ where f is defined, there exists a path from U to v such that, first, this path is colored exactly as our current play with S_2 , second, this path is consistent with S_1 , and third, the state of S_1 after this path is $f(v)$;
- (*completeness*) f is defined in the last node of our current play with S_2 .

To show that $\text{col}(S_2, U) \subseteq \text{col}(S_1, U)$, we have to show that for any infinite play P_2 with S_2 from U there exists an infinite play P_1 with S_1 from U such that P_1 has the same sequence of colors as P_2 . An analog of this statement for finite plays easily follows from the soundness and the completeness. Indeed, if we have a finite play p_2 with S_2 , then, by the completeness, f is defined on its endpoint, so we can take a finite play p_1 with S_1 which establishes the soundness for the endpoint of p_2 . We then extend this to infinite plays via the Kőnig’s lemma, by considering all finite prefixes of P_2 .

It remains to define the memory structure of S_2 (which has to be chromatic) and the next-move function of S_2 in such a way that the soundness and the completeness hold.

We start with the next-move function of S_2 . Assume that S_2 has to make a move from $v \in V$. If $f(v) \neq \perp$, then S_2 makes the same move as S_1 does from v when its state is $f(v)$. If $f(v) = \perp$, then S_2 makes an arbitrary move (however, the completeness invariant ensures that this situation never occurs).

We now define the memory structure of S_2 . To ensure our invariants in the beginning, we let the initial state of S_2 be the function $f_{init}: V \rightarrow M \cup \{\perp\}$, which maps nodes from U to m_{init} (the initial state of S_1) and other nodes to \perp . Now, assume that the current state of S_2 is represented by a function $f: V \rightarrow M \cup \{\perp\}$, and then we receive an edge whose color is $c \in C$. Our current play against S_2 became one edge longer. We have to define a new state $\hat{f}: V \rightarrow M \cup \{\perp\}$ of S_2 in some way which preserves the soundness

and the completeness. Moreover, since S_2 has to be chromatic, \hat{f} has to be a function of f and the color c .

On a high level, we consider all paths which establish the soundness for f , and then try all possible ways to extend them by c -colored edges without violating consistency with S_1 . Of course, we do not store all these paths in our memory, but the knowledge of f is sufficient. For example, assume that we have a c -colored edge from a node u of Player 1 with $f(u) \neq \perp$ to a node v . Then we can safely define $\hat{f}(v)$ to be the state into which the memory structure of S_1 transits from the state $f(u)$ after reading the edge (u, c, v) . Now, if u is a node of Player 0, there is one more restriction: the edge (u, c, v) has to be the move of S_1 from u in the state $f(u)$. This is needed to preserve the consistency with S_1 .

For the completeness, we have to ensure that there is a c -colored edge allowing us to define \hat{f} at the endpoint of our current play with S_2 . It is not hard to see that one such edge is simply the last edge of this play. It is not a problem that we might actually use some other edge to define \hat{f} at the last node of our current play – we should only ensure that \hat{f} is defined there. □

To obtain an analog of Corollary 2 for preference relations, we establish the following result.

Theorem 3. *For any $n, q \in \mathbb{Z}^+$, for any arena $\mathcal{A} = \langle V, V_0, V_1, E \rangle$ with n nodes, for any total preorder \preceq on the set V and for any q -state strategy S_1 of Player 0 in \mathcal{A} , there exists a chromatic $(qn + 1)^n$ -state strategy S_2 of Player 0 such that for any $v \in V$ we have:*

$$\text{col}(S_2, v) \subseteq \bigcup_{u \in V, v \preceq u} \text{col}(S_1, u).$$

Corollary 4. *Let \sqsubseteq be any preference relation on C^ω . Then for any $n, q \in \mathbb{Z}^+$ the following holds. Take any n -node arena \mathcal{A} and any q -state strategy S_1 of Player 0 in \mathcal{A} . Then there exists a chromatic $(qn + 1)^n$ -state strategy S_2 of Player 0 such that, for any node v of \mathcal{A} , we have that S_2 is at least as good as S_1 w.r.t. \sqsubseteq from v .*

Let us discuss why do we need slightly more states in Corollary 4 than in Corollary 2. The reason is that we want S_2 to be as good as S_1 from *every* node of \mathcal{A} . When we were dealing with winning condition, we could just forget about the nodes where S_1 is not winning. Now there is a finer classification of the nodes, depending on what Player 0 can achieve in these nodes w.r.t. \sqsubseteq , and it is slightly harder to deal with this classification.

Let us now formally derive Corollary 4 from Theorem 3.

Proof of Corollary 4. We write $u \preceq v$ for two nodes u, v of \mathcal{A} if for any $\beta \in \text{col}(S_1, v)$ there exists $\alpha \in \text{col}(S_1, u)$ such that $\alpha \sqsubseteq \beta$. Let us verify that \preceq is a total preorder on the set V of nodes of \mathcal{A} . The transitivity of \preceq follows from the transitivity \sqsubseteq . The reflexivity of \preceq is obvious. Now we show the totality of \preceq , that is, we show that $u \not\preceq v \implies v \preceq u$. Since $u \not\preceq v$, there exists $\beta \in \text{col}(S_1, v)$ such that $\alpha \not\sqsubseteq \beta$ for every $\alpha \in \text{col}(S_1, u)$. By the totality of \sqsubseteq , we have $\beta \sqsubseteq \alpha$ for every $\alpha \in \text{col}(S_1, u)$. This implies that $v \preceq u$.

We then apply Theorem 3 to \preceq . Consider the resulting chromatic $(qn + 1)^n$ -state strategy S_2 . We show, for every $v \in V$, that S_2 is at least as good as S_1 w.r.t. \sqsubseteq from v . That is, we show, for every $\beta \in \text{col}(S_2, v)$, that there exists $\alpha \in \text{col}(S_1, v)$ with $\alpha \sqsubseteq \beta$. By the conclusion of Theorem 3, we have that β belongs to $\text{col}(S_1, u)$ for some $v \preceq u$. By definition of \preceq , there exists some $\alpha \in \text{col}(S_1, v)$ such that $\alpha \sqsubseteq \beta$, as required. \square

Next we provide a proof sketch of Theorem 3, whose full proof is given in Section 5.

Proof sketch of Theorem 3. This time, S_2 will store slightly more information than in the proof of Theorem 1. Namely, at each moment, S_2 stores a function of the form $f: V \rightarrow V \times M \cup \{\perp\}$ (observe that this requires $(qn + 1)^n$ states). That is, whenever f is defined at v , the value $f(v)$ is an ordered pair in which the first coordinate is a node and the second coordinate is a state of S_1 . We introduce the first coordinate to strengthen the soundness condition: a path which establishes the soundness at v must be from the first coordinate of $f(v)$. More precisely, the soundness condition is formulated as follows:

- (*soundness*) for any $v \in V$ where $f(v) = (u, m) \neq \perp$ is defined, there exists a path from u to v such that, first, this path is colored exactly as our current play with S_2 , second, this path is consistent with S_1 , and third, the state of S_1 after this path is m ;

To take into account the preorder \preceq , we modify the completeness condition as follows:

- (*completeness*) f is defined at the last node of our current play with S_2 ; moreover, the first coordinate of f at this node is at least as large w.r.t. \preceq as the starting node of our play with S_2 .

These two conditions imply that for any infinite play P_2 with S_2 there exists an infinite play P_1 with S_1 such that, first, $\text{col}(P_2) = \text{col}(P_1)$, and second, $\text{source}(P_2) \preceq \text{source}(P_1)$ (again, we first derive this for finite plays, and then extend to infinite plays via the König's lemma). In turn, this obviously implies that

$$\text{col}(S_2, v) \subseteq \bigcup_{u \in V, v \preceq u} \text{col}(S_1, u).$$

It remains to define S_2 satisfying the soundness and the completeness. We give essentially the same definition as in the proof of Theorem 1. The only difference is that we have to additionally care about the first coordinates of f . For example, assume that our current state is f , and then we have to determine the new state \hat{f} at a node v . If there is more than one edge of the form (u, c, v) allowing to define \hat{f} at v , we choose one maximizing the first coordinate of $f(u)$ w.r.t. \preceq (and we set the first coordinate of $\hat{f}(v)$ to be equal to this maximum). This ensures that the value of the first coordinate of $f(v_{cur})$, where v_{cur} is the last node in our current play with S_2 , can only increase over time w.r.t. \preceq . In particular, it is always at least as large as the starting node of our play, since in the beginning we set $f_{init}(v) = (v, m_{init})$ for every $v \in V$. This easily

establishes our modified completeness condition, while the soundness requires almost no new argument. □

Finally, the exact statement of our lower bound showing the tightness of Theorem 1 is the following:

Theorem 5. *For any $n, q \in \mathbb{Z}^+$ there exists an arena \mathcal{A} with $n + 3$ nodes, a node u of \mathcal{A} and a q -state strategy S_1 of Player 0 in \mathcal{A} such that for any chromatic Q -state strategy S_2 of Player 0 it holds that $\text{col}(S_2, u) \subseteq \text{col}(S_1, u) \implies Q \geq q^n$.*

Our argument has a connection to a work of Jirásková and Pighizzini [8] on *self-verifying* automata. It turns out that from one of their results one can directly derive a weaker version of Theorem 5. Namely, one can get an arena with $n + O(1)$ nodes and a 2-state strategy S_1 of Player 0 in it such that for some node u of this arena the following holds: if S_2 is a chromatic Q -state strategy with $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$, then $Q = \Omega(3^{n/2})$. We show this derivation below in this section. Of course, when S_1 has 2 states, Theorem 5 gives a better bound $Q = \Omega(2^n)$, let alone that q can be arbitrary in Theorem 5). In fact, for the strong version of Theorem 5 it is not sufficient to use results of Jirásková and Pighizzini as a black box – we have to slightly modify their construction. The full proof of Theorem 5 is given in Section 6.

Sketch of the proof of Theorem 5 (weak version). A self-verifying automaton is a non-deterministic finite automaton with a property that, for any input word w , exactly one of the following two statements holds:

- there exists a computation which accepts w ;
- there exists a computation which rejects w .

For every $n \in \mathbb{N}$, Jirásková and Pighizzini construct a self-verifying automaton A_n with n states such that any deterministic automaton, recognizing the same language as A_n , has $\Omega(3^{n/2})$ states (they also show that this bound is tight). Using A_n , we construct an arena with $n + O(1)$ nodes and a 2-state strategy S_1 of Player 0 in it such that for some node u of this arena the following holds: if S_2 is a chromatic Q -state strategy with $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$, then $Q = \Omega(3^{n/2})$. Namely, consider the transition graph of A_n ; it can be viewed as an arena with edges colored by the input letters of A_n . Assume that Player 1 is the one to move everywhere in this transition graph. Now, add a node t controlled by Player 0. Draw edges to t from all accepting and rejecting states of A_n . Color all these edges into a single color $\#$. Finally, take two colors c and d that do not belong to the input alphabet of A_n , and draw two edges from t to the initial state of A_n , one colored by c and the other one by d . We define u as a node corresponding to the initial state of A_n .

Consider the following strategy S_1 of Player 0. If we come to t from an accepting state, then S_1 goes to the initial state by the c -colored edge. Otherwise, S_1 uses the

d -colored edge. Note that S_1 is a 2-state strategy – it just has to remember, whether the last node in a play was some accepting state of A_n .

We now show that from any chromatic strategy S_2 with $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$ one can extract a deterministic finite automaton recognizing the language of A_n (below, we denote this language by $L(A_n)$). This means that any such S_2 must have $\Omega(3^{n/2})$ states.

First, notice that, for every word w , Player 1 has a path to t which is colored by $w\#$. Indeed, for any w there is a computation over w which brings either to an accepting or to a rejecting state, from where we can go to t by a $\#$ -colored edge. Now, since A_n is self-verifying, we have the following. For $w \in L(A_n)$, there is no $w\#$ -colored path to t which goes through a rejecting state (otherwise it would give a computation which rejects w). Similarly, for $w \notin L(A_n)$, there is no $w\#$ -colored path to t which goes through an accepting state. This means that if $w\#$ is a prefix of some infinite sequence from $\text{col}(S_1, u)$, then it is followed by c in this sequence if and only if $w \in L(A_n)$. Thus, if S_2 is chromatic and $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$, then its memory structure, for any $w_1 \in L(A_n), w_2 \notin L(A_n)$, must come into different states on w_1 and on w_2 . Indeed, otherwise it makes the same move after $w_1\#$ and after $w_2\#$. □

4 Proof of Theorem 1

Let $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ be the memory structure of S_1 . We have that $|M| = q$. The set of states of S_2 will be the set of functions $f: V \rightarrow M \cup \{\perp\}$, where $\perp \notin M$. Thus, S_2 will be a $(q + 1)^n$ -state strategy. The initial state of S_2 is the function $f_{init}: V \rightarrow M \cup \{\perp\}$,

$$f_{init}(v) = \begin{cases} m_{init} & v \in U, \\ \perp & \text{otherwise} \end{cases}$$

We will define S_2 in such a way that for any finite path p the following holds. Assume that p is consistent with S_2 and $\text{source}(p) \in U$. Let $f: V \rightarrow M \cup \{\perp\}$ be the state of S_2 after p . Then we have the following two properties called *soundness* and *completeness*:

- (*soundness*) for any $v \in V$, if $f(v) = m \neq \perp$, then there exists a finite path p_1 with $\text{source}(p_1) \in U, \text{target}(p_1) = v$, such that, first, p_1 is consistent with S_1 , second, $\text{col}(p_1) = \text{col}(p)$, and third, $\delta(m_{init}, p_1) = m$.
- (*completeness*) $f(\text{target}(p)) \neq \perp$.

Let us first show that for any S_2 with these properties we have $\text{col}(S_2, U) \subseteq \text{col}(S_1, U)$. For that it is sufficient to establish the following. Let P be an arbitrary infinite path such that P is consistent with S_2 and $\text{source}(P) \in U$. Then there exists an infinite path P_1 such that P_1 is consistent with S_1 , $\text{source}(P_1) \in U$ and $\text{col}(P_1) = \text{col}(P)$.

Take an arbitrary $v \in V$. Consider an infinite tree of all finite paths from v that are consistent with S_1 . Now, delete from this tree all paths that are inconsistent with the

coloring of P . That is, we delete a path q if $\text{col}(q) \neq \text{col}(p)$, where p is a prefix of P with $|p| = |q|$. Let the resulting tree be T_v .

It is sufficient to show that for some $v \in U$, there is an infinite branch in T_v . By König's lemma, we have this as long as there exists $v \in U$ such that T_v is infinite (since we consider only finite arenas, T_v has finite branching for every v). To show this, we show that for any $k \in \mathbb{Z}^+$ there exists $v \in U$ such that T_v has a node of depth k . Indeed, let p be a prefix of P of length k . Since P is consistent with S_2 , so is p . Moreover, $\text{source}(p) = \text{source}(P) \in U$. Let $f: V \rightarrow M \cup \{\perp\}$ be the state of S_2 after reading p . By the completeness property, we have $f(\text{target}(p)) = m \neq \perp$. By the soundness property, there exists a finite path p_1 with $\text{source}(p_1) \in U$ such that p_1 is consistent with S_1 and $\text{col}(p_1) = \text{col}(p)$. Observe then that p_1 is a depth- k node of $T_{\text{source}(p_1)}$.

We now show how to define S_2 with properties as above. We first describe the transition function of the memory structure of S_2 . This memory structure has to be chromatic. So when its transition function receives an edge, it will only use the color of this edge to produce a new state.

Assume the current state of this memory structure is $f: V \rightarrow M \cup \{\perp\}$, and it receives an edge whose color is $c \in C$. We determine the new state $g: V \rightarrow M \cup \{\perp\}$ according to the following algorithm. To determine $g(v)$ for $v \in V$, we introduce a notion of a (f, v, c) -good edge. An edge $e \in E$ is (f, v, c) -good if

$$\text{target}(e) = v, \text{col}(e) = c \text{ and } f(\text{source}(e)) \neq \perp; \quad (1)$$

$$\text{if } \text{source}(e) \in V_0, \text{ then } e = S_1(\text{source}(e), f(\text{source}(e))). \quad (2)$$

If no (f, v, c) -good edge exists, we set $g(v) = \perp$. Otherwise, we take an arbitrary (f, v, c) -good edge e and set $g(v) = \delta(f(\text{source}(e)), e)$.

We now describe the next-move function of S_2 . Consider an arbitrary state $f: V \rightarrow M \cup \{\perp\}$ of S_2 and an arbitrary node $v \in V_0$. Define $S_2(v, f)$ as follows. Assume first that $f(v) \neq \perp$. Then set $S_2(v, f) = S_1(v, f(v))$. If $f(v) = \perp$, define $S_2(v, f)$ arbitrarily.

Definition of S_2 is finished. It remains to verify that it satisfies the soundness and the completeness properties. We show this by induction on the length of p .

We start with the induction base. Assume that p is a 0-length path (then it is automatically consistent with any strategy) and that $\text{source}(p) \in U$. We have to check the soundness and the completeness properties for p and for the initial state f_{init} . Let us start with the soundness. If $f_{\text{init}}(v) \neq \perp$, then, by definition, $v \in U$ and $f_{\text{init}}(v) = m_{\text{init}}$. Therefore, we can set $p_1 = \lambda_v$. As for the completeness, we have $f_{\text{init}}(\text{source}(p)) \neq \perp$ because $\text{source}(p) \in U$.

We now perform the induction step. Assume that we have verified the soundness and the completeness properties for all paths of length k . We extend this to paths of length $k + 1$. Consider any path $p = p'e'$ of length $k + 1$. Here $e' \in E$ is the last edge of p so that p' is of length k . Assume that p is consistent with S_2 and $\text{source}(p) \in U$. Then p' is also consistent with S_2 and $\text{source}(p') = \text{source}(p) \in U$. Let f be the state of S_2 after p' . By the induction hypothesis, we have that the soundness and the completeness hold for

p' and f . Now, let $g: V \rightarrow M \cup \{\perp\}$ be the state of S_2 after p . Alternatively, g is the state into which the memory structure of S_2 transits from the state f when it receives e' . Let $c = \text{col}(e')$ be the color of e' .

We first show that p and g satisfy the soundness property (see Figure 2).

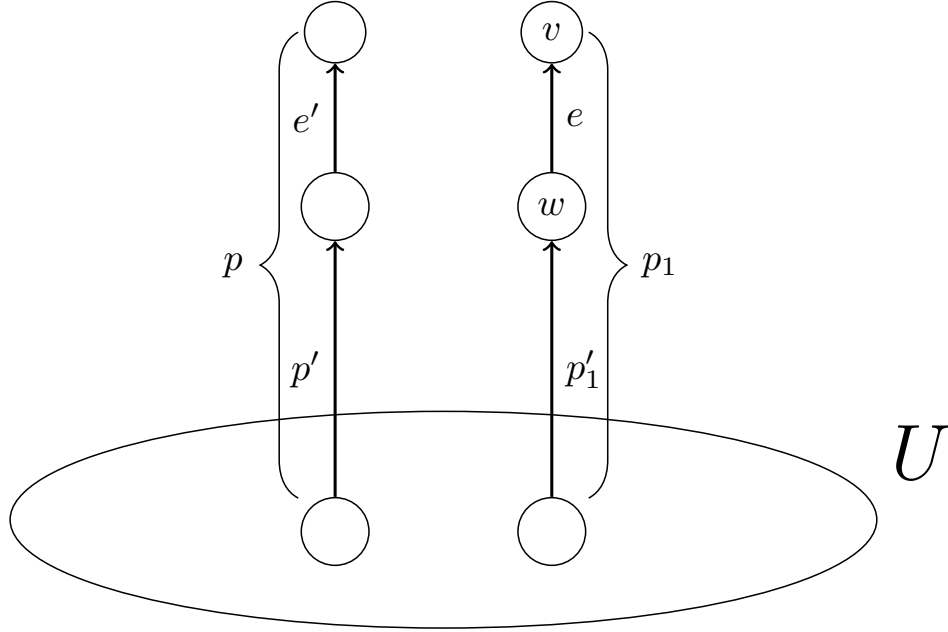


Figure 2: The argument for the soundness.

Consider any $v \in V$ such that $g(v) \neq \perp$. There must be an (f, v, c) -good edge. Let e be an (f, v, c) -good edge which was used to determine $g(v)$. Denote $w = \text{source}(e)$. By (1), we have $f(w) \neq \perp$. Hence, by the soundness for p' and f , there exists a finite path p'_1 with $\text{source}(p'_1) \in U$, $\text{target}(p'_1) = w$, such that **(a)** p'_1 is consistent with S_1 ; **(b)** $\text{col}(p'_1) = \text{col}(p')$; **(c)** $\delta(m_{init}, p'_1) = f(w)$. Set $p_1 = p'_1 e$. Since $\text{target}(p'_1) = w = \text{source}(e)$, we have that p_1 is a path. We show that p_1 verifies the soundness property for $g(v)$. Obviously, $\text{source}(p_1) = \text{source}(p'_1) \in U$. Since e is (f, v, c) -good, we have by (1) that $\text{target}(e) = v$. Hence, $\text{target}(p_1) = \text{target}(e) = v$. Let us now check that p_1 is consistent with S_1 . This is obvious if $w = \text{target}(p'_1) \in V_1$, because p'_1 is consistent with S_1 . Now, if $w = \text{target}(p'_1) \in V_0$, we have to show that $e = S_1(p'_1)$. Since $f(w) = \delta(m_{init}, p'_1)$ is the state of S_1 after p'_1 , we have $S_1(p'_1) = S_1(w, f(w))$. In turn, since e is (f, v, c) -good, by (2) we have $S_1(w, f(w)) = S_1(\text{source}(e), f(\text{source}(e))) = e$. It remains to show that $\text{col}(p_1) = \text{col}(p)$ and $\delta(m_{init}, p_1) = g(v)$. Indeed, $\text{col}(p_1) = \text{col}(p'_1 e) = \text{col}(p'_1) \text{col}(e) = \text{col}(p') c = \text{col}(p') \text{col}(e') = \text{col}(p' e') = \text{col}(p)$. Here we use a fact that $\text{col}(e) = c$ due to (1). In turn, $\delta(m_{init}, p_1) = \delta(m_{init}, p'_1 e) = \delta(\delta(m_{init}, p'_1), e) = \delta(f(w), e)$. It remains to recall that by definition, $g(v) = \delta(f(\text{source}(e)), e) = \delta(f(w), e)$.

Now we show that p and g satisfy the completeness property. In other words, we show that $g(\text{target}(p)) \neq \perp$. By definition, this holds as long as there exists an $(f, \text{target}(p), c)$ -good edge. We claim that e' , the last edge of p , is $(f, \text{target}(p), c)$ -

good. Let us first verify that e' satisfies (1). Obviously, $\text{target}(e') = \text{target}(p)$. Now, $\text{col}(e') = c$ by definition. Finally, we have $f(\text{source}(e')) = f(\text{target}(p')) \neq \perp$ due to the completeness property for p' and f . Let us now check that e' satisfies (2). Assume that $\text{source}(e') = \text{target}(p') \in V_0$. Since p is consistent with S_2 , we have $e' = S_2(p')$. Now, by definition, f is the state of S_2 after p' . Therefore, $e' = S_2(p') = S_2(\text{target}(p'), f) = S_2(\text{source}(e'), f)$. Again, since the completeness property holds for p' and f , we have $f(\text{source}(e')) = f(\text{target}(p')) \neq \perp$. Hence, by definition of S_2 , we have that $e' = S_2(\text{source}(e'), f) = S_1(\text{source}(e'), f(\text{source}(e')))$. Thus, (2) is established for e' .

5 Proof of Theorem 3

Let $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ be the memory structure of S_1 . We have that $|M| = q$. The set of states of S_2 will be the set of functions $f: V \rightarrow V \times M \cup \{\perp\}$, where $\perp \notin V \times M$. Thus, S_2 is a $(qn+1)^n$ -state strategy. The initial state of S_2 is the function $f_{init}: V \rightarrow V \times M \cup \{\perp\}$, defined by $f(v) = (v, m_{init})$ for every $v \in V$.

We use the following notation in the proof. Take any $f: V \rightarrow V \times M \cup \{\perp\}$ and $v \in V$, and assume that $f(v) = (u, m) \neq \perp$. Then we set $f_1(v) = u$ and $f_2(v) = m$. That is, f_1 is the projection of f to the first coordinate (its values are nodes of our arena) and f_2 is the projection of f to the second coordinate (its values are states of S_1). If $f(v) = \perp$, we set $f_1(v) = f_2(v) = \perp$.

Our goal is to define S_2 in a such a way that, for any finite path p which is consistent with S_2 , and for the state f of S_2 after p , the following holds:

- (*soundness*) for any $v \in V$, if $f(v) \neq \perp$, then there exists a finite path p_1 from $f_1(v)$ to v such that, first, p_1 is consistent with S_1 , second, $\text{col}(p_1) = \text{col}(p)$, and third, $\delta(m_{init}, p_1) = f_2(v)$.
- (*completeness*) $f(\text{target}(p)) \neq \perp$ and $\text{source}(p) \preceq f_2(v)$.

It is not hard to see that for any S_2 with these properties we have

$$\text{col}(S_2, v) \subseteq \bigcup_{u \in V, v \preceq u} \text{col}(S_1, u).$$

Indeed, to establish this, we have to show that for any infinite path P which is consistent with S_2 there exists an infinite path P_1 which is consistent with S_1 such that $\text{col}(P) = \text{col}(P_1)$ and $\text{source}(P) \preceq \text{source}(P_1)$. For this we define the trees T_v , $v \in V$ as in the proof of Theorem 1. By König's lemma, it is sufficient to show that T_u is infinite for some u with $\text{source}(P) \preceq u$. We take an arbitrary finite prefix p of P . Since p is consistent with S_2 , from the completeness we get that $f(\text{target}(p)) \neq \perp$. By applying the soundness to the node $\text{target}(p)$, we get a finite path p_1 from $f_1(\text{target}(p))$ to $\text{target}(p)$ such that, first, p_1 is consistent with S_1 , and second, $\text{col}(p_1) = \text{col}(p)$. Hence, p_1 is a node of $T_{f_1(\text{target}(p))}$. Moreover, by the completeness we have $\text{source}(P) = \text{source}(p) \preceq f_1(\text{target}(p))$. Thus, for

some u with $\text{source}(P) \preceq u$ there is a node of depth $|p|$ in T_u . It remains to note that $|p|$ can be arbitrarily large.

We now explain how to define S_2 in a way which guaranties the soundness and the completeness properties. We start with the transition function of S_2 . Assume that the current state of S_2 is $f: V \rightarrow V \times M \cup \{\perp\}$, and then it receives an edge whose color is c . We define the new state $g: V \rightarrow V \times M \cup \{\perp\}$ as follows (we stress that S_2 has to be chromatic, so g will be a function of f and c). Take any $v \in V$ for which we want to determine $g(v)$. Note that $f_2: V \rightarrow M \cup \{\perp\}$. If there is no (f_2, v, c) -good edge, in a sense of (1-2), then we set $g(v) = \perp$. Otherwise, we take an (f_2, v, c) -good edge e , maximizing $f_1(\text{source}(e))$ w.r.t. \preceq , and set $g_1(v) = f_1(\text{source}(e))$, $g_2(v) = \delta(f_2(\text{source}(e)), e)$.

We now define the next-move function of S_2 . Let $f: V \rightarrow V \times M \cup \{\perp\}$ be a state and $v \in V_0$ be a node of Player 0. If $f(v) \neq \perp$, we set $S_2(v, f) = S_1(v, f_2(v))$. Otherwise, we define $S_2(v, f)$ arbitrarily.

It remains to establish the soundness and the completeness properties for all finite paths p that are consistent with S_2 . As before, we do so by induction on $|p|$.

We start with the induction base. Assume that $|p| = 0$. The initial state of S_2 is the function f_{init} . Recall that we have $f_{init}(v) = (v, m_{init})$ for every $v \in V$. So, to establish the soundness, we can set $p_1 = \lambda_v$ for every $v \in V$. For the completeness, observe that $f_{init}(\text{target}(p)) = (\text{target}(p), m_{init}) \neq \perp$ and, obviously, $\text{source}(p) \preceq \text{target}(p)$ (just because p is a 0-length path so that $\text{source}(p) = \text{target}(p)$).

Let us now perform the induction step. Assume that our claim is proved for all p of length up to k . Take any $p = p'e'$ of length $k + 1$ which is consistent with S_2 . Here e' is the last edge of p . Then p' is consistent with S_2 and has length k . Hence, we have the induction hypothesis for p' and for a function $f: V \rightarrow V \times M \cup \{\perp\}$ which is the state of S_2 after p' . Next, let the state of S_2 after p be $g: V \rightarrow V \times M \cup \{\perp\}$. Note that g is the value of the transition function of S_2 on f and $c = \text{col}(e') \in C$.

To check the soundness for p and g , one can use exactly the same argument as in Theorem 1 for f_2 and g_2 . That is, for any $v \in V$ with $g(v) \neq \perp$, we consider an (f_2, v, c) -good edge e which was used to define $g(v)$. By (1), we have $f_2(\text{source}(e)) \neq \perp \implies f(\text{source}(e)) \neq \perp$. Then, using the induction hypothesis for p' , we take a path p'_1 establishing the soundness for f and p' at $\text{source}(e)$. Finally, we define $p_1 = p'_1e$ and show that p_1 establishes the soundness for g and p at v . Obviously, p_1 is a path to v . By the same routine check as in the proof of Theorem 1, we have that, first, p_1 is consistent with S_1 , second, $\text{col}(p_1) = \text{col}(p)$, and third, $\delta(m_{init}, p_1) = g_2(v)$. The only thing we have to additionally check is that p_1 starts in $g_1(v)$. Indeed, by definition, $g_1(v) = f_1(\text{source}(e))$. Note that p'_1 is a prefix of p_1 , so these paths have the same starting node. In turn, since p'_1 establishes the soundness for f and p' at $\text{source}(e)$, the starting node of p'_1 must be $f_1(\text{source}(e)) = g_1(v)$, as required.

We now check the completeness property for p and g . It is sufficient to show the existence of an $(f_2, \text{target}(p), c)$ -good edge e with $\text{source}(p) \preceq f_1(\text{source}(e))$. Indeed, $g(\text{target}(p)) \neq \perp$ if and only if $(f_2, \text{target}(p), c)$ -good edges exist, and $g_1(\text{target}(p))$ is the maximum of $f_1(\text{source}(e))$ w.r.t. \preceq over such edges.

We claim that e' , the last edge of p , satisfies these conditions. To show this, recall that by the induction hypothesis we have the completeness for p' and f . Let us first demonstrate that e' satisfies (1) for f_2 , $v = \mathbf{target}(p)$ and c . Indeed, $\mathbf{target}(e') = \mathbf{target}(p)$ because e' is the last edge of p , $\mathbf{col}(e') = c$ by definition of c , and $f_2(\mathbf{source}(e')) = f_2(\mathbf{target}(p')) \neq \perp$ by the completeness for p' and f . Let us now verify (2). Assume that $\mathbf{source}(e') \in V_0$. Then, since $p = p'e'$ is consistent with S_2 , we have $e' = S_2(p')$. The state of S_2 after p' is f , so $S_2(p') = S_2(\mathbf{target}(p'), f) = S_2(\mathbf{source}(e'), f)$. Note that $f(\mathbf{source}(e')) = f(\mathbf{target}(p')) \neq \perp$ by the completeness for p' and f . Hence, by definition of S_2 , we have $S_2(\mathbf{source}(e'), f) = S_1(\mathbf{source}(e'), f_2(\mathbf{source}(e')))$, and, thus, e' satisfies (2). Finally, we have to show that $\mathbf{source}(p) \preceq f_1(\mathbf{source}(e))$. This is because, by the completeness for p' and f , we have $\mathbf{source}(p') \preceq f_1(\mathbf{target}(p'))$. It remains to note that $\mathbf{source}(p) = \mathbf{source}(p')$ and $\mathbf{source}(e') = \mathbf{target}(p')$ – recall that e' is the last edge of p and p' is the part of p which precedes e' .

6 Proof of Theorem 5

Let \mathcal{A} be as on Figure 3.

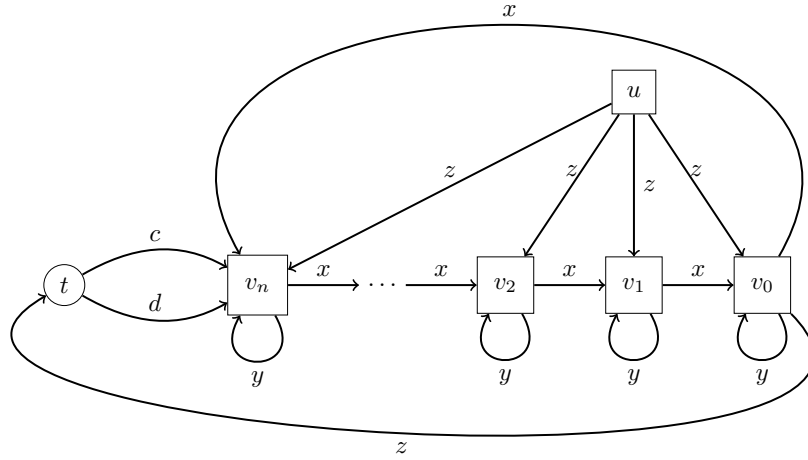


Figure 3: Arena \mathcal{A} . The set of colors is $C = \{x, y, z, c, d\}$. The partition of the nodes between the players is given by $V_0 = \{t\}$, $V_1 = \{u, v_0, v_1, \dots, v_n\}$.

We define S_1 as follows. Its memory structure maintains a number $\mathbf{count} \in \{0, 1, \dots, q-1\}$. Initially, $\mathbf{count} = 0$. When the memory structure of S_1 passes through any y -colored edge, it increments \mathbf{count} by 1 modulo q . In turn, when we go from v_0 to v_n , it sets $\mathbf{count} = 0$. In all the other cases, the value of \mathbf{count} does not change. It remains to define how S_1 acts at t (this is the only node from where Player 0 is the one

to move). There are two edges from t , both go to v_0 , but one is c -colored and the other is d -colored. If $\text{count} = 0$, then S_1 uses the c -colored edge. If $\text{count} \neq 0$, then S_1 uses the d -colored edge.

For brevity, if p is a finite path in \mathcal{A} , then by $\text{count}(p)$ we denote the value of count after p .

We need the following definition and the following lemma about paths in the arena \mathcal{A} .

Definition 2. Define a function $f: \{x, y\}^* \rightarrow \{0, 1, \dots, q-1\}$ as follows. Take any $w \in \{x, y\}^*$. To define $f(w)$, first define a word $w' \in \{x, y\}^*$. Namely, if w has at most n occurrences of x , then set $w' = w$. Otherwise, take the $(n+1)$ st occurrence of x from the right, erase it and everything to its left, and let the remaining word be w' . Finally, let $f(w)$ be the number of y 's in w' modulo q .

Lemma 6. For any $w \in \{x, y\}^*$ the following holds:

- (a) there exists a finite path p with $\text{source}(p) = u$, $\text{target}(p) = t$ and $\text{col}(p) = zwz$;
- (b) for any finite path p , if $\text{source}(p) = u$ and $\text{col}(p) = zwz$, then $\text{target}(p) = t$ and $\text{count}(p) = f(w)$.

Proof. Let X be the number of occurrences of x in w . Set i to be the remainder of X when divided by $n+1$.

We start by showing (a). To construct p , we first go from u to v_i . Then we start reading letters of w one by one from left to right. Every time we read a new letter, we move from our current location via some edge colored by this letter. It remains to show that after reading the whole w we end up in v_0 , which has an out-going z -colored edge to t . Indeed, if we forget about y 's, then we are just rotating counterclockwise along the cycle $v_n \rightarrow \dots \rightarrow v_1 \rightarrow v_0 \rightarrow v_n$. The length of this cycle is $n+1$, and the distance from v_i to v_0 , measured counterclockwise, is i . Thus, since in w there are $X \equiv i \pmod{n+1}$ occurrences of x , we end up in v_0 .

We now show (b). Consider any finite path p with $\text{source}(p) = u$ and $\text{col}(p) = zwz$. Observe that once we left u , it is impossible to come back to it again. Therefore, since the last edge of p is z -colored, this edge must be from v_0 to t . Hence, $\text{target}(p) = t$.

It remains to show that $\text{count}(p) = f(w)$. Assume first that p never goes from v_0 to v_n . Then $\text{count}(p)$ is the number of y 's modulo q in w , because $\text{col}(p) = zwz$. Thus, to show that $\text{count}(p) = f(w)$ in this case, it is enough to show $X \leq n$. Indeed, the first edge of p is from u to v_j , for some $j \in \{0, 1, \dots, n\}$. Then it makes X steps along the cycle $v_n \rightarrow \dots \rightarrow v_1 \rightarrow v_0 \rightarrow v_n$. If X were at least $n+1$, then p had to go from v_0 to v_n at least once, contradiction.

Now, assume that p contains edges from v_0 to v_n . By definition, $\text{count}(p)$ equals the number of y -colored edges in p modulo q after the last time p went from v_0 to v_n . To show that $\text{count}(p) = f(w)$, we have to show that the number of x -colored edges in p after the last time p went from v_0 to v_n is n (then the last edge from v_0 to v_n in p corresponds to the $(n+1)$ st occurrence of x in w from the right). Indeed, as we discussed above, the

last edge of p must be from v_0 to t . Obviously, if we go from v_n to v_0 without going to v_n again after this, then the number of times we pass an x -colored edge is exactly n . \square

This gives the following fact about the set $\text{col}(S_1, u)$.

Corollary 7. *For any $w \in \{x, y\}^*$ the following holds. If $zwzc$ is a prefix of some sequence from $\text{col}(S_1, u)$, then $f(w) = 0$. In turn, if $zwzd$ is a prefix of some sequence from $\text{col}(S_1, u)$, then $f(w) \neq 0$.*

Proof. Fix $h \in \{c, d\}$. Take any $w \in \{x, y\}^*$ such that $zwzh$ is a prefix of some sequence of $\text{col}(S_1, u)$. We show that $h = c \iff f(w) = 0$.

By definition of $\text{col}(S_1, u)$, there exists a finite path p with $\text{source}(p) = u$, $\text{col}(p) = zwzh$ which is consistent with S_1 . Let p_1 be the part of p which precedes its last edge. Since $\text{source}(p_1) = u$ and $\text{col}(p_1) = zwz$, we have by the item **(b)** of Lemma 6 that $\text{target}(p_1) = t$ and $\text{count}(p_1) = f(w)$.

The node t is controlled by Player 0. Hence, since p is consistent with S_1 , the last edge of p must be equal to $S_1(p_1)$. The color of $S_1(p_1)$ is h . In turn, $\text{count}(p_1)$ is the state of S_1 after p_1 . Therefore, by definition of S_1 , the color of $S_1(p_1)$ is c if and only if $\text{count}(p_1) = f(w) = 0$. The lemma is proved. \square

Consider now any chromatic Q -state strategy S_2 such that $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$. Let $\mathcal{M} = \langle M, m_{\text{init}}, \sigma: M \times \{x, y, z, c, d\} \rightarrow M \rangle$ be its (chromatic) memory structure. To show that $Q \geq q^n$, in Lemma 8 we provide q^n words from $\{x, y, z, c, d\}^*$ such that $\sigma(m_{\text{init}}, \cdot)$ must take different values on these words.

Definition 3. *Let $g: \{0, 1, \dots, q-1\}^n \rightarrow \{x, y\}^*$ be the following function:*

$$g: (i_1, i_2, \dots, i_n) \mapsto xy^{i_1}xy^{i_2} \dots xy^{i_n}.$$

Lemma 8. *For any $\kappa_1, \kappa_2 \in \{0, 1, \dots, q-1\}^n$ such that $\kappa_1 \neq \kappa_2$ we have $\sigma(m_{\text{init}}, zg(\kappa_1)) \neq \sigma(m_{\text{init}}, zg(\kappa_2))$.*

To establish Lemma 8, we first need the following lemma.

Lemma 9. *For any $\kappa_1, \kappa_2 \in \{0, 1, \dots, q-1\}^n$ such that $\kappa_1 \neq \kappa_2$ there exists a word $w \in \{x, y\}^*$ such that $f(g(\kappa_1)w) = 0$ and $f(g(\kappa_2)w) \neq 0$.*

Proof. Assume that $\kappa_1 = (i_1, i_2, \dots, i_n)$ and $\kappa_2 = (j_1, \dots, j_n)$. Take the largest $k \in \{1, 2, \dots, n\}$ such that $i_k \neq j_k$. Let $r \in \{0, 1, \dots, q-1\}$ be such that $i_k + i_{k+1} + \dots + i_n + r \equiv 0 \pmod{q}$. Define $w = x^k y^r$. Thus,

$$\begin{aligned} g(\kappa_1)w &= xy^{i_1}xy^{i_2} \dots xy^{i_k} \dots xy^{i_n}x^k y^r, \\ g(\kappa_2)w &= xy^{j_1}xy^{j_2} \dots xy^{j_k} \dots xy^{j_n}x^k y^r. \end{aligned}$$

Observe that the $(n + 1)$ st occurrence of x in $g(\kappa_1)w$ is one before y^{i_k} . Similarly, the $(n + 1)$ st occurrence of x in $g(\kappa_2)w$ is one before y^{j_k} . Hence, by definition of f , we have:

$$\begin{aligned} f(g(\kappa_1)w) &\equiv i_k + i_{k+1} + \dots + i_n + r \pmod{q}, \\ f(g(\kappa_2)w) &\equiv j_k + j_{k+1} + \dots + j_n + r \pmod{q}. \end{aligned}$$

By definition of r , we have $f(g(\kappa_1)w) = 0$. In turn, by definition of k , we have $i_k \neq j_k$ and $i_{k+1} = j_{k+1}, \dots, i_n = j_n$. The numbers i_k, j_k are different elements of $\{0, 1, \dots, q-1\}$, which means that their difference is not divisible by q . Hence, $f(g(\kappa_2)w) \neq f(g(\kappa_1)w) = 0$. \square

To conclude the proof of the theorem, it remains to derive Lemma 8 from Lemma 9. Assume for contradiction that

$$\sigma(m_{init}, zg(\kappa_1)) = \sigma(m_{init}, zg(\kappa_2)) \quad (3)$$

for some $\kappa_1, \kappa_2 \in \{0, 1, \dots, q-1\}^n$, $\kappa_1 \neq \kappa_2$. By Lemma 9 there exists $w \in \{x, y\}^*$ such that

$$f(g(\kappa_1)w) = 0, \quad f(g(\kappa_2)w) \neq 0. \quad (4)$$

By the item (a) of Lemma 6 there exist two finite paths p_1 and p_2 such that

$$\begin{aligned} \text{source}(p_1) &= \text{source}(p_2) = u, \\ \text{target}(p_1) &= \text{target}(p_2) = t, \\ \text{col}(p_1) &= zg(\kappa_1)wz, \quad \text{col}(p_2) = zg(\kappa_2)wz. \end{aligned}$$

The paths p_1 and p_2 do not have c, d -colored edges. That is, they do not have edges that start at t . This means that these paths are consistent with S_2 . We claim that $S_2(p_1) = S_2(p_2)$. Indeed, since S_2 is chromatic, the value of $S_2(p)$ for a finite path p with $\text{target}(p) = t$ is completely determined by $\sigma(m_{init}, \text{col}(p))$. Now, by (3) we have that:

$$\sigma(m_{init}, \text{col}(p_1)) = \sigma(m_{init}, zg(\kappa_1)wz) = \sigma(m_{init}, zg(\kappa_2)wz) = \sigma(m_{init}, \text{col}(p_2)).$$

So let $e = S_2(p_1) = S_2(p_2)$. The paths p_1e and p_2e are both consistent with S_2 . Since $\text{col}(S_2, u) \subseteq \text{col}(S_1, u)$, we have that $\text{col}(p_1e)$ is a prefix of some sequence from $\text{col}(S_1, u)$, and so is $\text{col}(p_2e)$. This gives a contradiction with Corollary 7. Indeed, assume first that $\text{col}(e) = c$. Then $\text{col}(p_2e) = zg(\kappa_2)wzc$ is a prefix of some sequence from $\text{col}(S_1, u)$, but $f(g(\kappa_2)w) \neq 0$ by (4), contradiction. Similarly, if $\text{col}(e) = d$, then $\text{col}(p_1e) = zg(\kappa_1)wzd$ is a prefix of some sequence from $\text{col}(S_1, u)$, but $f(g(\kappa_1)w) = 0$ by (4), contradiction.

References

- [1] BOUYER, P., OUALHADJ, Y., RANDOUR, M., AND VANDENHOVE, P. Arena-Independent Finite-Memory Determinacy in Stochastic Games. In *32nd International Conference on Concurrency Theory (CONCUR 2021)* (Dagstuhl, Germany, 2021), S. Haddad and D. Varacca, Eds., vol. 203 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 26:1–26:18.

- [2] BOUYER, P., RANDOUR, M., AND VANDENHOVE, P. Characterizing omega-regularity through finite-memory determinacy of games on infinite graphs. *arXiv preprint arXiv:2110.01276* (2021).
- [3] BOUYER, P., ROUX, S. L., OUALHADJ, Y., RANDOUR, M., AND VANDENHOVE, P. Games Where You Can Play Optimally with Arena-Independent Finite Memory. In *31st International Conference on Concurrency Theory (CONCUR 2020)* (Dagstuhl, Germany, 2020), I. Konnov and L. Kovács, Eds., vol. 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 24:1–24:22.
- [4] BOUYER, P., ROUX, S. L., AND THOMASSET, N. Finite-memory strategies in two-player infinite games. *arXiv preprint arXiv:2107.09945* (2021).
- [5] BÜCHI, J. R., AND LANDWEBER, L. H. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society* 138 (1969), 295–311.
- [6] CASARES, A. On the minimisation of transition-based rabin automata and the chromatic memory requirements of muller conditions. *arXiv preprint arXiv:2105.12009* (2021).
- [7] GIMBERT, H., AND ZIELONKA, W. Games where you can play optimally without any memory. In *International Conference on Concurrency Theory* (2005), Springer, pp. 428–442.
- [8] JIRÁSKOVÁ, G., AND PIGHIZZINI, G. Optimal simulation of self-verifying automata by deterministic automata. *Information and Computation* 209, 3 (2011), 528–535.
- [9] KOPCZYŃSKI, E. *Half-positional determinacy of infite games*. PhD thesis, Warsaw University, 2008.
- [10] LE ROUX, S. Time-aware uniformization of winning strategies. In *Conference on Computability in Europe* (2020), Springer, pp. 193–204.
- [11] LE ROUX, S., PAULY, A., AND RANDOUR, M. Extending finite-memory determinacy by boolean combination of winning conditions. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science* (2018).
- [12] MUCHNIK, A. A. Games on infinite trees and automata with dead-ends: a new proof for the decidability of the monadic second order theory of two successors. *BULLETIN-EUROPEAN ASSOCIATION FOR THEORETICAL COMPUTER SCIENCE* 48 (1992), 219–219.
- [13] THOMAS, W., AND WILKE, T. Automata, logics, and infinite games: A guide to current research. *Bulletin of Symbolic Logic* 10, 1 (2004).