State Complexity of Chromatic Memory in Infinite-Duration Games

Alexander Kozachinskiy ✉
IMFD Chile & CENIA Chile

Abstract

We study finite-memory strategies in games over edge-colored graphs. Usually, these strategies are defined using finite automata that have a set of edges as an input alphabet. However, several recent results in strategy complexity are only known for finite-memory strategies that use chromatic memory (meaning that this means that these strategies do not distinguish edges of the same colors). We study the cost of transforming general finite-memory strategies into strategies with chromatic memory.

For every winning condition and 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 $n+3$ nodes such that (a) there exists a winning strategy with q states of general memory, (b) there exists no winning strategy with less than q^n states of chromatic memory.

2012 ACM Subject Classification Theory of computation → Logic

Keywords and phrases games on graphs, finite-memory strategies, self-verifying automata

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

Games on graphs are a standard tool in many areas of computer science, from decidability of logical theories [7] to reactive synthesis [1]. This situation raises various questions for games on graphs, and one of them is *strategy complexity*. In strategy complexity, one seeks to understand which winning conditions admit "simple" winning strategies (meaning that whenever you have a winning strategy, you also have a simple one). In this paper, we study one of the standard complexity measure of strategies – their memory.

Games in question are played over edge-colored graphs, with winning conditions defined as sets of infinite sequences of colors. During the game, two players move a token over nodes of a graph along its edges. We focus on *infinite-duration turn-based* games. "Infinite-duration" means that the game proceeds for infinitely many turns, giving us an infinite sequence of colors. Whether or not this sequence belongs to the winning condition determines who is the winner of the play. Now, "turn-based" means that in each turn, one of the players fully controls the token (there is a predetermined partition of the nodes between the players).

We are interested in strategies with *finite memory*. They are defined through so-called memory structures. A memory structure is a finite automaton whose input alphabet is the set of edges of the graph. Whenever the token passes an edge, this edge is fed to the memory structure. A strategy, built on top of a memory structure $\mathcal{M}$, makes its decisions based on two things – first, the current node, and second, the current state of $\mathcal{M}$. When a strategy can be built on top of some memory structure with q states, we say that this strategy has q states of general memory.

There is a specific class of memory structures called *chromatic* memory structures. These are memory structures that do not distinguish edges of the same color. They can be presented as finite automata not over the set of edges but over the set of colors. This makes them more uniform than general memory structures – one chromatic memory structure can be used in

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

different game graphs. Another reason to consider them is that our winning conditions are defined solely in terms of colors. Thus, they collect only those information which is directly related to a winning condition. When a strategy can be built on top of some chromatic memory structure with q states, we say that this strategy has q states of chromatic memory.

We study the cost of transforming strategies with general memory into strategies with chromatic memory. More specifically, for a given game graph G and a winning condition W, we study the relationship between the following two parameters. The first one, Q_{gen}, is the minimal q for which in G there exists a strategy with q states of general memory, which is winning w.r.t. W. The second one, Q_{chr}, is defined similarly, but with "general memory" replaced by "chromatic memory".

There are two motivations of this problem. First, several results in strategy complexity are known only for strategies with chromatic memory. For example, Kopczyński [4] gave an algorithm, computing chromatic memory requirements of prefix-independent ω-regular winning conditions. No such algorithm is known for general memory requirements. In turn, Bouyer et al. [6] obtained a characterization of winning conditions, for which there exists some constant q such that both players need at most q of states of chromatic memory in all games graphs. Again, no such characterization is known for general memory. This situation motivates studying the relationship between chromatic and general memory more deeply.

Second, there is a connection to *self-verifying* automata. A self-verifying automaton is a non-deterministic automaton with the following property: for every input word either there exists an accepting run or a rejecting run, but not both. In [3], Jirásková and Pighizzini study the cost of transforming a self-verifying automaton into an equivalent deterministic automaton. As we discuss below, this is a special instance of the problem that we study in this paper. More specifically, given a self-verifying automaton A, the minimal size of an equivalent deterministic automaton equals the minimal number of states of chromatic memory, needed to simulate a certain strategy with 2 states of general memory (played over a transition graph of A).

Results.

Let us go back to the relationship between Q_{gen} and Q_{chr}. Since every strategy with q states of chromatic memory is also a strategy with q states of general memory, we have $Q_{gen} \leq Q_{chr}$. The following bound in the other direction was obtained by Le Roux [5]:

$$Q_{chr} \leq 2^{Q_{gen} \cdot (n^2+1)},$$

where n is the number of nodes of the underlying game graph. Our first result is the following improvement of this bound:

$$Q_{chr} \leq (Q_{gen} + 1)^n.$$

Note that our bound is no longer exponential in Q_{gen}. This shows that there is no much difference between chromatic and general memory, when one of them is at least exponential in n. This applies, for example, to energy parity conditions [2].

▶ Remark 1. Le Roux establishes his upper bound for *concurrent games*. Our upper holds for concurrent games as well. However, we only present the argument for turn-based games, for two reasons: (1) the argument is less technical in the turn-based case; (2) the difference between general and chromatic memory has been mostly studied for turn-based games.

We also show that our upper bound is tight. Namely, for every n and q we provide a winning condition W and a game graph G with $n + 3$ nodes, such that $Q_{gen} \leq q$ and $Q_{chr} \geq q^n$. To obtain this separation, we adapt technique from self-verifying automata.

Unfortunately, our upper bound is unsatisfactory for winning conditions, where Q_{gen} is small in n (for example, independent of n). A future work might be to search classes of winning conditions, for which our upper bound can be further improved.

The rest of the paper is organized as follows. In Section 2 we give preliminaries. In Section 3 we give exact statements and a technical overview of our results. In Section 4 we prove our main upper bound. In Appendix A we prove our lower bound. In Appendix B, we establish a version of our upper bound for *preference relations*.

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 $B \cap C = \varnothing$. We let $\circ$ denote the function composition. The set of positive integral numbers is denoted by $\mathbb{Z}^+$.

2.1 Arenas

We call our players Protagonist and Antagonist. Graphs over which they play are called *arenas*.

▶ **Definition 2.** *Let C be any set. A tuple $\mathcal{A} = \langle V, V_P, V_A, E\rangle$, where V, V_P, V_A, E are four finite sets such that $V \neq \varnothing$, $V = V_P \sqcup V_A$ 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_P and V_A will be called Protagonist's nodes and Antagonist's nodes, respectively. Elements of E will be called edges of $\mathcal{A}$. For an edge $e = (s, c, t) \in E$, we define $\mathsf{source}(e) = s$, $\mathsf{col}(e) = c$ and $\mathsf{target}(e) = t$. We imagine $e \in E$ as an arrow, colored into $\mathsf{col}(e)$ and going from $\mathsf{source}(e)$ to $\mathsf{target}(e)$.

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

$$\mathsf{col}(e_1e_2e_3\ldots) = \mathsf{col}(e_1)\mathsf{col}(e_2)\mathsf{col}(e_3)\ldots, \qquad e_1, e_2, e_3, \ldots \in E.$$

A non-empty sequence $p = e_1e_2e_3\ldots \in E^* \cup E^\omega$ is called a path if for every $1 \leq i < |p|$ we have $\mathsf{target}(e_i) = \mathsf{source}(e_{i+1})$. We set $\mathsf{source}(p) = \mathsf{source}(e_1)$ and, if p is finite, $\mathsf{target}(p) = \mathsf{target}(e_{|p|})$. For technical convenience, for every node $v \in V$ we define a 0-length path λ_v with $\mathsf{source}(\lambda_v) = \mathsf{target}(\lambda_v) = v$.

We will concatenate paths. Non-empty paths are just sequences of edges. So if p and q are two non-empty paths, their concatenations pq is defined in the usual way. However, pq is a path if and only if p is finite and $\mathsf{target}(p) = \mathsf{source}(q)$. We also have to define concatenation for 0-length paths. For every $v \in V$, we set $\lambda_v q = q$ if $\mathsf{source}(q) = v$. Otherwise, $\lambda_v q$ is undefined. Similarly, we let $q\lambda_v = q$ if q is finite and $\mathsf{target}(q) = v$. Otherwise, $q\lambda_v$ is undefined.

2.2 The game

Let $\mathcal{A} = \langle V, V_P, V_A, E\rangle$ be an arena over the set of colors C. We define a game, associated with $\mathcal{A}$. The set of positions of the game is the set of finite paths in $\mathcal{A}$. Next, take a finite path p. Protagonist is the one to move from p if and only if $\mathsf{target}(p) \in V_P$. The set of moves, available to Protagonist at p, is the set of out-going edges of $\mathsf{target}(p)$, i.e., the set of $e \in E$

with $\mathsf{source}(e) = \mathsf{target}(p)$ (by definition of an arena, this set is non-empty). If Protagonist plays an edge e in the position p, the next position is pe. Antagonist plays similarly in positions with $\mathsf{source}(p) \in V_A$. The game can start in an arbitrary node v (more formally, possible starting positions are paths $\lambda_v, v \in V$). We assume that it continues for infinitely many turns, giving us some infinite path.

A Protagonist's strategy maps each position, from where Protagonist is the one to move, to some move, available at this position. Antagonist's strategies are defined likewise, but we do not mention them in this paper.

We use the following notation. For $u \in V$ and for a Protagonist's strategy S, we let $\mathsf{InfPlays}(S, u)$ be the set of infinite paths $P = (e_1, e_2, e_3, \ldots)$ that can be obtained in a play with S from u. Formally, P can be obtained in a play with S from u if $\mathsf{source}(P) = u$ and if the following holds:

- $u \in V_P \implies S(\lambda_u) = e_1$;
- for every $i \geq 1$, we have that $\mathsf{target}(e_i) \in V_P \implies S(e_1 \ldots e_i) = e_{i+1}$.

Additionally, we define $\mathsf{col}(S, u) = \mathsf{col}\big(\mathsf{InfPlays}(S, u)\big)$. In other words, $\mathsf{col}(S, u)$ is the set of all infinite sequences of colors that can be obtained in a play with S from the node u. For $U \subseteq V$, we define $\mathsf{col}(S, U) = \bigcup_{u \in U} \mathsf{col}(S, u)$.

2.3 Winning conditions and preference relations

A winning condition is any set $W \subseteq C^\omega$. We say that a Protagonist's strategy S is winning from $u \in V$ w.r.t. to W if $\mathsf{col}(S, u) \subseteq W$. In other words, any infinite play from u against S must give a sequence of colors from W.

We also consider a more general class of objectives called *preference relations*. A preference relation is a total preorder[1] $\sqsubseteq$ on the set C^ω. Intuitively, when given a preference relation $\sqsubseteq$, the goal of Protagonist is to maximize the sequence of colors w.r.t. $\sqsubseteq$.

Any two strategies S_1, S_2 of Protagonist can be compared w.r.t. $\sqsubseteq$ (from the Protagonist's perspective). Namely, we say that S_2 is *no worse* than S_1 from $u \in V$ if for every $\beta \in \mathsf{col}(S_2, u)$ there exists $\alpha \in \mathsf{col}(S_1, u)$ such that $\alpha \sqsubseteq \beta$. In other words, if the game starts at u, then for any play with S_2 there exists a play with S_1, which is the same w.r.t. $\sqsubseteq$ or worse.

It is instructive to observe that the relation "no worse from u" over Protagonist's strategies is a total preorder. Transitivity and reflexivity is immediate. To show totality, assume for contradiction that there are two Protagonist's strategies such that two statements "S_2 is no worse than S_1 from u" and "S_1 is no worse than S_2 from u" are both false. Hence, there exists $\beta_2 \in \mathsf{col}(S_2, u)$ such that $\alpha \not\sqsubseteq \beta_2$ for all $\alpha \in \mathsf{col}(S_1, u)$. Similarly, there exists $\beta_1 \in \mathsf{col}(S_1, u)$ such that $\alpha \not\sqsubseteq \beta_1$ for all $\alpha \in \mathsf{col}(S_2, u)$. Comparing β_1 and β_2, we get a contradiction with the totality of $\sqsubseteq$.

2.4 Memory structures

Let $\mathcal{A} = \langle V, V_P, V_A, E\rangle$ be an arena over the set of colors C. A memory structure in $\mathcal{A}$ is a tuple $\mathcal{M} = \langle M, m_{init}, \delta\rangle$, where M is a finite set, $m_{init} \in M$ and $\delta \colon M \times E \to M$. 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

[1] A total preorder is a transitive and reflexive binary relation which is "total" in a sense that for every a, b from its domain, either (a, b) or (b, a) belongs to it.

extend the function $\delta(m, \cdot)$ to arbitrary finite sequences of edges as follows:

$$\delta(m, \text{empty sequence}) = m,$$
$$\delta(m, e_1 e_2 \ldots e_{n+1}) = \delta(\delta(m, e_1 \ldots e_n), e_{n+1}), \qquad n \geq 1,\ e_1, e_2, \ldots, e_{n+1} \in E.$$

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

We say that a memory structure $\mathcal{M} = \langle M, m_{init}, \delta \rangle$ is chromatic if there exists a function $\sigma \colon M \times C \to M$ such that $\delta(m, e) = \sigma(m, \mathsf{col}(e))$ for every $e \in E$ and $m \in M$. In other words, chromatic memory structures do 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 Protagonist's strategy 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 $\mathsf{target}(p_1) = \mathsf{target}(p_2) \in V_P$ it holds that:

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

In other words, if S is an $\mathcal{M}$-strategy, then the value of $S(p)$ for an $\mathcal{M}$-strategy S solely depends on $\mathsf{target}(p)$ and on $\delta(m_{init}, p)$. Sometimes, to avoid overusing formulas, we will refer to $\delta(m_{init}, p)$ as *the state of S after p*.

With any $\mathcal{M}$-strategy S one can associate the *next-move* function of S. This is a function $n_S \colon V_P \times M \to E$, defined as follows: to determine $n_S(v, m)$, we take an arbitrary finite path p with $\mathsf{target}(p) = v$ 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 $v \in V_P, m \in M$ denotes the value of the next-move function of S on the pair (v, m).

We say that S is a strategy with q states of general memory if S is an $\mathcal{M}$-strategy for some memory structure $\mathcal{M}$ with q states. If $\mathcal{M}$ is chromatic, we say that S is a strategy with q states of chromatic memory.

3 Technical Overview

3.1 An example

We start with a simple example, where there is a gap between general and chromatic memory. More specifically, we give a winning condition and an arena, where (a) there is a winning strategy of Protagonist with 2 states of general memory, (b) there is no winning strategy of Protagonist with 2 states of chromatic memory. The arena is depicted below.

All nodes are controlled by Protagonist. The set of colors is $C = \{S, D\}$, where S means "solid" and D means "dashed". Consider a winning condition $W \subseteq C^\omega$, consisting of all sequences from C^ω that have $SSSSS$ (the letter S five times) as a subword. Assume that the game starts at u.

Protagonist has the following winning strategy: go to v, then go to the cycle, then, after returning to v, go to the right. This is a strategy with 2 states of general memory. Indeed, consider a memory structure with 2 states: "have not been to v" and "have been to v". It

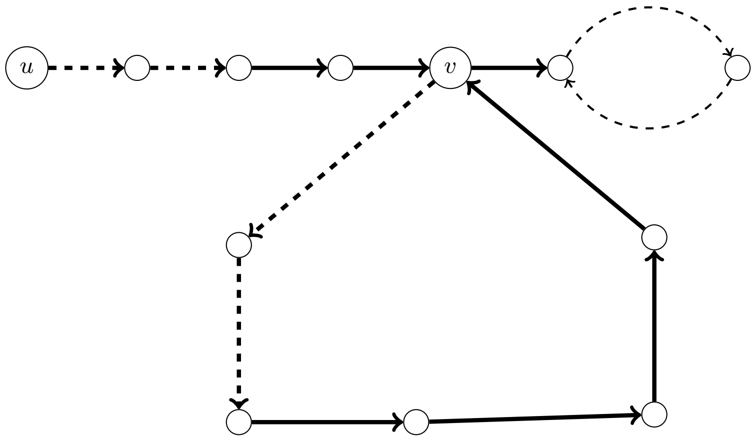

Figure 1 An example.

switches from the first state to the second state after receiving any edge starting at v. The only choice our strategy has is at v. In the state "have not been to v", it goes to the cycle. In the state "have been to v", it goes to the right.

We now argue that there is no winning strategy of Protagonist with 2 states of chromatic memory. Assume for contradiction that such strategy exists. First it goes to v, as there is no other choice. Let q_1 be its state when it reaches v. If it then goes to the right, it loses. So it has to go to the cycle. When it returns to v, it is in some state q_2. If $q_1 = q_2$, then our strategy stays on the cycle forever and hence loses. Therefore, $q_1 \neq q_2$. However, the memory structure of our strategy is chromatic. In other words, it can be presented as a deterministic finite automaton (DFA) over C. Now, q_1 is the state of this automaton on the word $DDSS$, and q_2 is the state of this automaton on the word $DDSSDDSSSS$. It can be checked via the computer search that no DFA with 2 states can distinguish $DDSS$ and $DDSSDDSSSS$.

3.2 Upper bounds

The exact statement of our main upper bound is the following

▶ **Theorem 3.** *For any $n, q \in \mathbb{Z}^+$ the following holds. Consider an arbitrary arena $\mathcal{A} = \langle V, V_P, V_A, E\rangle$ with n nodes, an arbitrary set $U \subseteq V$, and an arbitrary Protagonist's strategy S_1 with q states of general memory in this arena. Then there exists a Protagonist's strategy S_2 with $(q+1)^n$ states of chromatic memory such that $\mathsf{col}(S_2, U) \subseteq \mathsf{col}(S_1, U)$.*

It should not be confusing that this theorem does not mention winning conditions. One can notice that $\mathsf{col}(S_1, U)$ 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 3, 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 4.** *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 Protagonist's strategy S_1 with q states of general memory in it. Then in $\mathcal{A}$ there exists a Protagonist's strategy S_2 with $(q+1)^n$ states of chromatic memory such that for every 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 3 to the set U of nodes from where S_1 is winning w.r.t. W. ◀

We also obtain an analog of Corollary 4 for preference relations. For that, we first establish the following technical result:

▶ **Theorem 5.** *For any $n, q \in \mathbb{Z}^+$, for any arena $\mathcal{A} = \langle V, V_P, V_A, E\rangle$ with n nodes, for any total preorder $\preceq$ on the set V and for any Protagonist's strategy S_1 with q states of general memory in $\mathcal{A}$, there exists a Protagonist's strategy S_2 with $(qn+1)^n$ states of chromatic memory such that for any $v \in V$ we have:*

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

▶ **Corollary 6.** *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 Protagonist's strategy S_1 with q states of general memory in $\mathcal{A}$. Then there exists a Protagonist's strategy S_2 with $(qn+1)^n$ states of chromatic memory such that, for every 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 6 than in Corollary 4. 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 Protagonist 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 6 from Theorem 5.

Proof of Corollary 6. We write $u \preceq v$ for two nodes u, v of $\mathcal{A}$ if for any $\beta \in \mathsf{col}(S_1, v)$ there exists $\alpha \in \mathsf{col}(S_1, u)$ 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 $\beta \in \mathsf{col}(S_1, v)$ such that $\alpha \not\sqsubseteq \beta$ for every $\alpha \in \mathsf{col}(S_1, u)$. By the totality of $\sqsubseteq$, we have $\beta \sqsubseteq \alpha$ for every $\alpha \in \mathsf{col}(S_1, u)$. This implies that $v \preceq u$.

We then apply Theorem 5 to $\preceq$. Consider the resulting 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 $\beta \in \mathsf{col}(S_2, v)$, that there exists $\alpha \in \mathsf{col}(S_1, v)$ with $\alpha \sqsubseteq \beta$. By the conclusion of Theorem 5, we have that β belongs to $\mathsf{col}(S_1, u)$ for some $v \preceq u$. By definition of $\preceq$, there exists some $\alpha \in \mathsf{col}(S_1, v)$ such that $\alpha \sqsubseteq \beta$, as required. ◀

3.3 A lower bound

Finally, the exact statement of our lower bound, showing tightness of Theorem 3, is the following:

▶ **Theorem 7.** *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 Protagonist's strategy S_1 with q states of general memory such that for any Q and for any Protagonist's strategy S_2 with Q states of chromatic memory the following holds: $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u) \implies Q \geq q^n$.*

Our argument has a connection to a work of Jirásková and Pighizzini [3] on *self-verifying* automata. It turns out that from one of their results one can directly derive a weaker version of Theorem 7. Namely, one can get an arena with $n + O(1)$ nodes and a Protagonist's strategy with 2 states of general memory such that for some node u of this arena the following holds: if S_2 is a Protagonist's strategy with Q states of chromatic memory such that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$, then $Q = \Omega(3^{n/2})$. We show this derivation below in this section. Of course, when S_1 has 2 states, Theorem 7 gives a better bound $Q = \Omega(2^n)$, let alone that q can be arbitrary in Theorem 7). In fact, for the strong version of Theorem 7 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 7 is given in Appendix A.

Sketch of the proof of Theorem 7 (weak version). Consider a non-deterministic finite automaton $\mathcal{A}$ which, in addition to the set of accepting states, has a set of *rejecting* states which is disjoint from the set of accepting states. Such an automaton is *self-verifying* if for every finite word w, exactly one of the following two statements is true:

- there exists a run of $\mathcal{A}$ on w which ends in an accepting state;
- there exists a run of $\mathcal{A}$ on w which ends in a rejecting state.

The language, recognized by such an automaton, consists of all finite words for which the first statement is true. The complement of this language consists of all words for which the second statement is true.

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 Protagonist's strategy with 2 states of general memory such that for some node u of this arena the following holds: if S_2 is a Protagonist's strategy with Q states of chromatic memory such that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$, 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 Antagonist is the one to move everywhere in this transition graph. Now, add a node t controlled by Protagonist. Draw edges to t from all accepting and rejecting states of A_n. Color all these edges into a new color $\#$. Finally, take two more new colors c and d, and draw two edges from t to the initial state of A_n, one colored by c, and the other one by d. Let u be the initial state of A_n.

Consider the following Protagonist's strategy S_1. The only node where S_1 has to do something is t. If we come to t from an accepting state, then S_1 goes to u via the c-colored edge. Otherwise, S_1 goes to u via the d-colored edge. Note that S_1 is a strategy with 2 states of general memory – it only needs to remember whether the last edges in the current play starts at an accepting state of A_n.

Assume now that there is a Protagonist's strategy S_2 with Q states of chromatic memory such that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$. Its memory structure can be presented as a deterministic finite automaton with Q states whose input alphabet contains the input alphabet of A_n and also $\#, c$ and d. We show this memory structure recognizes the language of A_n (below, we denote this language by $L(A_n)$). This means that $Q = \Omega(3^{n/2})$.

Assume for contradiction that there are two words $w_1 \in L(A_n), w_2 \notin L(A_n)$ over the input alphabet of A_n such that the memory structure of S_2 comes into the same state on them. Notice that, for every word w over the input alphabet of A_n, Antagonist has a path from u to t, colored by $w\#$. Indeed, for any w there is a run over w which brings us to an accepting or to a rejecting state, from where we can go to t by a $\#$-colored edge. In particular, there are 2 paths from u to t that are colored by $w_1\#$ and $w_2\#$. Note that S_2 does the same thing after these two paths because its memory structure does not distinguish w_1 and w_2. Assume that at this moment S_2 goes to u via the c-colored edge (if it goes via the d-colored edge, the argument is the same). This means that $\mathsf{col}(S_2, u)$ has some infinite sequence, starting with $w_2\#c$. Due to the fact that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$, this infinite sequence can be obtained in some play with S_1. Since we have c after $\#$ in this play, we came to t via some accepting state. This means that there exists an accepting run for w_2. On the other hand, since $w_2 \notin L(A_n)$, there exists a rejecting run for w_2. This is a contradiction, because A_n is self-verifying. ◀

As we indicated in the introduction, this argument, for every self-verifying automaton A, establishes equality of the following two parameters: (a) the minimal number of states in a deterministic finite automaton, recognizing the same language as A, (b) the minimal Q for which there exists a strategy S_2 with Q states of chromatic memory such that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$ (where u and S_1 are constructed from A as above).

4 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 $f\colon V \to M \cup \{\perp\}$, where $\perp \notin M$. Thus, S_2 will be a $(q+1)^n$-state strategy. The initial state of S_2 is the function $f_{init}\colon V \to M \cup \{\perp\}$,

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

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 $\mathsf{source}(p) \in U$. Let $f\colon V \to M \cup \{\perp\}$ 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 $\mathsf{source}(p_1) \in U, \mathsf{target}(p_1) = v$, such that, first, p_1 is consistent with S_1, second, $\mathsf{col}(p_1) = \mathsf{col}(p)$, and third, $\delta(m_{init}, p_1) = m$.
- *(completeness)* $f(\mathsf{target}(p)) \neq \perp$.

Let us first show that for any S_2 with these properties we have $\mathsf{col}(S_2, U) \subseteq \mathsf{col}(S_1, U)$. 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 $\mathsf{source}(P) \in U$. Then there exists an infinite path P_1 such that P_1 is consistent with S_1, $\mathsf{source}(P_1) \in U$ and $\mathsf{col}(P_1) = \mathsf{col}(P)$.

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 $\mathsf{col}(q) \neq \mathsf{col}(p)$, 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 Kőnig'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, $\mathsf{source}(p) = \mathsf{source}(P) \in U$. Let $f\colon V \to M \cup \{\perp\}$ be the state of S_2 after reading p. By the completeness property, we have $f(\mathsf{target}(p)) = m \neq \perp$. By the soundness property, there exists a finite path p_1 with $\mathsf{source}(p_1) \in U$ such that p_1 is consistent with S_1 and $\mathsf{col}(p_1) = \mathsf{col}(p)$. Observe then that p_1 is a depth-k node of $T_{\mathsf{source}(p_1)}$.

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 $f\colon V \to M \cup \{\perp\}$, and it receives an edge whose color is $c \in C$. We determine the new state $g\colon V \to M \cup \{\perp\}$ 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

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

$$\text{if } \mathsf{source}(e) \in V_P, \text{ then } e = S_1\big(\mathsf{source}(e), f(\mathsf{source}(e))\big). \tag{2}$$

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

We now describe the next-move function of S_2. Consider an arbitrary state $f\colon V \to M \cup \{\bot\}$ of S_2 and an arbitrary node $v \in V_P$. Define $S_2(v, f)$ as follows. Assume first that $f(v) \neq \bot$. Then set $S_2(v, f) = S_1\big(v, f(v)\big)$. If $f(v) = \bot$, 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 $\mathsf{source}(p) \in U$. 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 $f_{init}(v) \neq \bot$, then, by definition, $v \in U$ and $f_{init}(v) = m_{init}$. Therefore, we can set $p_1 = \lambda_v$. As for the completeness, we have $f_{init}(\mathsf{source}(p)) \neq \bot$ because $\mathsf{source}(p) \in U$.

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 $\mathsf{source}(p) \in U$. Then p' is also consistent with S_2 and $\mathsf{source}(p') = \mathsf{source}(p) \in U$. 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 $g\colon V \to M \cup \{\bot\}$ 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 $c = \mathsf{col}(e')$ be the color of e'.

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

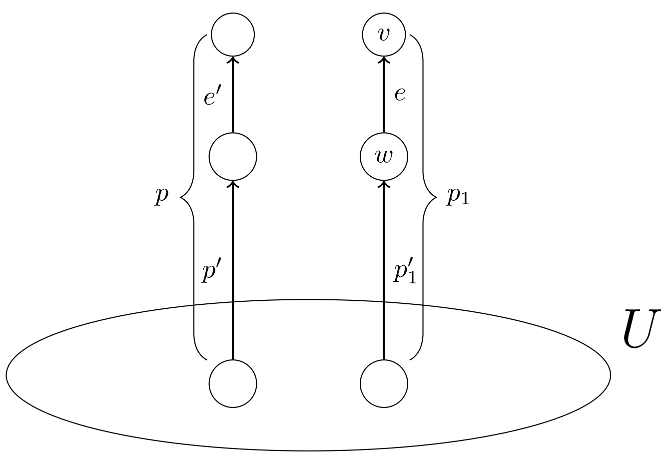

Figure 2 The argument for the soundness.

Consider any $v \in V$ such that $g(v) \neq \bot$. 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 $w = \mathsf{source}(e)$. By (1), we have $f(w) \neq \bot$. Hence, by the soundness for p' and f, there exists a finite path p'_1 with $\mathsf{source}(p'_1) \in U, \mathsf{target}(p'_1) = w$, such that ***(a)*** p'_1 is consistent with S_1; ***(b)***

$\mathsf{col}(p_1') = \mathsf{col}(p')$; ***(c)*** $\delta(m_{init}, p_1') = f(w)$. Set $p_1 = p_1'e$. Since $\mathsf{target}(p_1') = w = \mathsf{source}(e)$, we have that p_1 is a path. We show that p_1 verifies the soundness property for $g(v)$. Obviously, $\mathsf{source}(p_1) = \mathsf{source}(p_1') \in U$. Since e is (f, v, c)-good, we have by (1) that $\mathsf{target}(e) = v$. Hence, $\mathsf{target}(p_1) = \mathsf{target}(e) = v$. Let us now check that p_1 is consistent with S_1. This is obvious if $w = \mathsf{target}(p_1') \in V_A$, because p_1' is consistent with S_1. Now, if $w = \mathsf{target}(p_1') \in V_P$, 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 $S_1(w, f(w)) = S_1\big(\mathsf{source}(e), f(\mathsf{source}(e))\big) = e$. It remains to show that $\mathsf{col}(p_1) = \mathsf{col}(p)$ and $\delta(m_{init}, p_1) = g(v)$. Indeed, $\mathsf{col}(p_1) = \mathsf{col}(p_1'e) = \mathsf{col}(p_1')\mathsf{col}(e) = \mathsf{col}(p')c = \mathsf{col}(p')\mathsf{col}(e') = \mathsf{col}(p'e') = \mathsf{col}(p)$. Here we use a fact that $\mathsf{col}(e) = c$ 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, $g(v) = \delta(f(\mathsf{source}(e)), e) = \delta(f(w), e)$.

Now we show that p and g satisfy the completeness property. In other words, we show that $g(\mathsf{target}(p)) \neq \bot$. By definition, this holds as long as there exists an $(f, \mathsf{target}(p), c)$-good edge. We claim that e', the last edge of p, is $(f, \mathsf{target}(p), c)$-good. Let us first verify that e' satisfies (1). Obviously, $\mathsf{target}(e') = \mathsf{target}(p)$. Now, $\mathsf{col}(e') = c$ by definition. Finally, we have $f(\mathsf{source}(e')) = f(\mathsf{target}(p')) \neq \bot$ due to the completeness property for p' and f. Let us now check that e' satisfies (2). Assume that $\mathsf{source}(e') = \mathsf{target}(p') \in V_P$. 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, $e' = S_2(p') = S_2(\mathsf{target}(p'), f) = S_2(\mathsf{source}(e'), f)$. Again, since the completeness property holds for p' and f, we have $f(\mathsf{source}(e')) = f(\mathsf{target}(p')) \neq \bot$. Hence, by definition of S_2, we have that $e' = S_2(\mathsf{source}(e'), f) = S_1\big(\mathsf{source}(e'), f(\mathsf{source}(e'))\big)$. Thus, (2) is established for e'.

References

1. Roderick Bloem, Krishnendu Chatterjee, and Barbara Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking*, pages 921–962. Springer, 2018.
2. Krishnendu Chatterjee and Laurent Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.
3. Galina Jirásková and Giovanni Pighizzini. Optimal simulation of self-verifying automata by deterministic automata. *Information and Computation*, 209(3):528–535, 2011.
4. Eryk Kopczyński. *Half-positional determinacy of infite games*. PhD thesis, Warsaw University, 2008.
5. Stéphane Le Roux. Time-aware uniformization of winning strategies. In *Conference on Computability in Europe*, pages 193–204. Springer, 2020.
6. Pierre Vandenhove, Mickael Randour, Youssouf Oualhadj, Stéphane Le Roux, and Patricia Bouyer. Games where you can play optimally with arena-independent finite memory. *Logical Methods in Computer Science*, 18, 2022.
7. Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200(1-2):135–183, 1998.

A Proof of Theorem 7

Let $\mathcal{A} = \langle V, V_P, V_A, E\rangle$ be as on Figure 3.

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

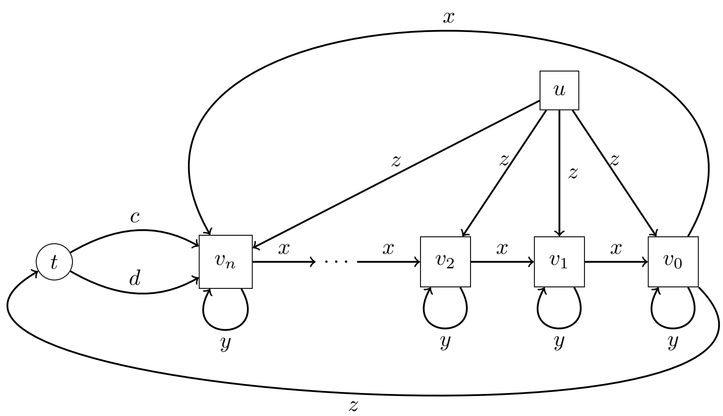

Figure 3 Arena $\mathcal{A}$. The set of colors is $C = \{x, y, z, c, d\}$. The partition of the nodes between the players is given by $V_P = \{t\}, V_A = \{u, v_0, v_1, \ldots, v_n\}$.

t, both go to v_0, but one is c-colored and the other is d-colored. If $\mathsf{count} = 0$, then S_1 uses the c-colored edge. If $\mathsf{count} \neq 0$, then S_1 uses the d-colored edge.

For brevity, if p is a finite path in $\mathcal{A}$, then by $\mathsf{count}(p)$ 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 8.** *Define a function $f\colon \{x, y\}^* \to \{0, 1, \ldots, q-1\}$ 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 9.** *For any $w \in \{x, y\}^*$ the following holds:*

(a) *there exists a finite path p with $\mathsf{source}(p) = u$, $\mathsf{target}(p) = t$ and $\mathsf{col}(p) = zwz$;*

(b) *for any finite path p, if $\mathsf{source}(p) = u$ and $\mathsf{col}(p) = zwz$, then $\mathsf{target}(p) = t$ and $\mathsf{count}(p) = f(w)$.*

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 $v_n \to \ldots \to v_1 \to v_0 \to v_n$. 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 $\mathsf{source}(p) = u$ and $\mathsf{col}(p) = zwz$. 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, $\mathsf{target}(p) = t$.

It remains to show that $\mathsf{count}(p) = f(w)$. Assume first that p never goes from v_0 to v_n. Then $\mathsf{count}(p)$ is the number of y's modulo q in w, because $\mathsf{col}(p) = zwz$. Thus, to show that $\mathsf{count}(p) = f(w)$ in this case, it is enough to show $X \le n$. Indeed, the first edge of p is from u to v_j, for some $j \in \{0, 1, \ldots, n\}$. Then it makes X steps along the cycle $v_n \to \ldots \to v_1 \to v_0 \to v_n$. 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, $\mathsf{count}(p)$ 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 $\mathsf{count}(p) = f(w)$, 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.

◀

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

▶ **Corollary 10.** *For any $w \in \{x, y\}^*$ the following holds. If $zwzc$ is a prefix of some sequence from $\mathsf{col}(S_1, u)$, then $f(w) = 0$. In turn, if $zwzd$ is a prefix of some sequence from $\mathsf{col}(S_1, u)$, 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 $\mathsf{col}(S_1, u)$. We show that $h = c \iff f(w) = 0$.

By definition of $\mathsf{col}(S_1, u)$, there exists a finite path p with $\mathsf{source}(p) = u, \mathsf{col}(p) = zwzh$ which is consistent with S_1. Let p_1 be the part of p which precedes its last edge. Since $\mathsf{source}(p_1) = u$ and $\mathsf{col}(p_1) = zwz$, we have by the item **(b)** of Lemma 9 that $\mathsf{target}(p_1) = t$ and $\mathsf{count}(p_1) = f(w)$.

The node t is controlled by Protagonist. 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, $\mathsf{count}(p_1)$ 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 $\mathsf{count}(p_1) = f(w) = 0$. The lemma is proved. ◀

Consider any $Q \in \mathbb{Z}^+$ and any Protagonist's strategy S_2 with Q states of chromatic strategy such that $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$. We show that $Q \ge q^n$. Note that S_2 is an $\mathcal{M}$-strategy for some chromatic memory structure $\mathcal{M} = \langle M, m_{init} \in M, \delta \colon M \times E \to M \rangle$. By definition of chromatic memory structures, there exists some $\sigma \colon M \times \{x, y, z, c, d\} \to M$ such that

$$\delta(m, e) = \sigma(m, \mathsf{col}(e)), \qquad m \in M, e \in E. \tag{3}$$

To show that $Q \ge q^n$, in Lemma 12 we provide q^n words from $\{x, y, z, c, d\}^*$ such that $\sigma(m_{init}, \cdot)$ must take different values on these words.

▶ **Definition 11.** *Let $g \colon \{0, 1, \ldots, q-1\}^n \to \{x, y\}^*$ be the following function:*

$$g \colon (i_1, i_2, \ldots, i_n) \mapsto xy^{i_1} xy^{i_2} \ldots xy^{i_n}.$$

▶ **Lemma 12.** *For any $\kappa_1, \kappa_2 \in \{0, 1, \ldots, q-1\}^n$ such that $\kappa_1 \neq \kappa_2$ we have $\sigma(m_{init}, zg(\kappa_1)) \neq \sigma(m_{init}, zg(\kappa_2))$.*

To establish Lemma 12, we first need the following lemma.

▶ **Lemma 13.** *For any $\kappa_1, \kappa_2 \in \{0, 1, \ldots, q-1\}^n$ 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 $\kappa_1 = (i_1, i_2, \ldots, i_n)$ and $\kappa_2 = (j_1, \ldots, j_n)$. Take the largest $k \in \{1, 2, \ldots, n\}$ such that $i_k \neq j_k$. Let $r \in \{0, 1, \ldots, q-1\}$ be such that $i_k + i_{k+1} + \ldots + i_n + r \equiv 0 \pmod q$. Define $w = x^k y^r$. Thus,

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

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:

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

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 $i_{k+1} = j_{k+1}, \ldots, i_n = j_n$. The numbers i_k, j_k are different elements of $\{0, 1, \ldots, q-1\}$, which means that their difference is not divisible by q. Hence, $f(g(\kappa_2)w) \neq f(g(\kappa_1)w) = 0$. ◀

To conclude the proof of the theorem, it remains to derive Lemma 12 from Lemma 13.

Proof of Lemma 12. Assume for contradiction that

$$\sigma(m_{init}, zg(\kappa_1)) = \sigma(m_{init}, zg(\kappa_2)) \tag{4}$$

for some $\kappa_1, \kappa_2 \in \{0, 1, \ldots, q-1\}^n$, $\kappa_1 \neq \kappa_2$. By Lemma 13 there exists $w \in \{x, y\}^*$ such that

$$f(g(\kappa_1)w) = 0, \qquad f(g(\kappa_2)w) \neq 0. \tag{5}$$

By the item **(a)** of Lemma 9 there exist two finite paths p_1 and p_2 such that

$$\mathsf{source}(p_1) = \mathsf{source}(p_2) = u,$$
$$\mathsf{target}(p_1) = \mathsf{target}(p_2) = t,$$
$$\mathsf{col}(p_1) = zg(\kappa_1)wz, \qquad \mathsf{col}(p_2) = zg(\kappa_2)wz.$$

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 $S_2(p)$ is determined by $\mathsf{target}(p)$ and $\delta(m_{init}, p)$. Now, $\mathsf{target}(p_1) = \mathsf{target}(p_2) = t$. It remains to show that $\delta(m_{init}, p_1) = \delta(m_{init}, p_2)$. By (3), it is sufficient to show that $\sigma(m_{init}, \mathsf{col}(p_1)) = \sigma(m_{init}, \mathsf{col}(p_2))$ This follows from (4) because:

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

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 $\mathsf{col}(S_2, u) \subseteq \mathsf{col}(S_1, u)$, we have that $\mathsf{col}(p_1e)$ is a prefix of some sequence from $\mathsf{col}(S_1, u)$, and so is $\mathsf{col}(p_2e)$. This gives a contradiction with Corollary 10. Indeed, assume first that $\mathsf{col}(e) = c$. Then $\mathsf{col}(p_2e) = zg(\kappa_2)wzc$ is a prefix of some sequence from $\mathsf{col}(S_1, u)$, but $f(g(\kappa_2)w) \neq 0$ by (5), contradiction. Similarly, if $\mathsf{col}(e) = d$, then $\mathsf{col}(p_1e) = zg(\kappa_1)wzd$ is a prefix of some sequence from $\mathsf{col}(S_1, u)$, but $f(g(\kappa_1)w) = 0$ by (5), contradiction. ◀

B Proof of Theorem 5

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 $f: V \to V \times M \cup \{\perp\}$, where $\perp \notin V \times M$. Thus, S_2 is a strategy with $(qn+1)^n$ states. The initial state of S_2 is the function $f_{init}: V \to V \times M \cup \{\perp\}$, defined by $f(v) = (v, m_{init})$ for every $v \in V$.

We use the following notation in the proof. Take any $f: V \to V \times M \cup \{\perp\}$ 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, $\mathsf{col}(p_1) = \mathsf{col}(p)$, and third, $\delta(m_{init}, p_1) = f_2(v)$.
- *(completeness)* $f(\mathsf{target}(p)) \neq \perp$ and $\mathsf{source}(p) \preceq f_1(v)$.

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

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

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 $\mathsf{col}(P) = \mathsf{col}(P_1)$ and $\mathsf{source}(P) \preceq \mathsf{source}(P_1)$. For this we define the trees T_v, $v \in V$ as in the proof of Theorem 3. By Kőnig's lemma, it is sufficient to show that T_u is infinite for some u with $\mathsf{source}(P) \preceq u$. We take an arbitrary finite prefix p of P. Since p is consistent with S_2, from the completeness we get that $f(\mathsf{target}(p)) \neq \perp$. By applying the soundness to the node $\mathsf{target}(p)$, we get a finite path p_1 from $f_1(\mathsf{target}(p))$ to $\mathsf{target}(p)$ such that, first, p_1 is consistent with S_1, and second, $\mathsf{col}(p_1) = \mathsf{col}(p)$. Hence, p_1 is a node of $T_{f_1(\mathsf{target}(p))}$. Moreover, by the completeness we have $\mathsf{source}(P) = \mathsf{source}(p) \preceq f_1(\mathsf{target}(p))$. Thus, for some u with $\mathsf{source}(P) \preceq u$ 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 $f: V \to V \times M \cup \{\perp\}$, and then it receives an edge whose color is c. We define the new state $g: V \to V \times M \cup \{\perp\}$ 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 $f_2: V \to M \cup \{\perp\}$. 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 $f_1(\mathsf{source}(e))$ w.r.t. $\preceq$, and set $g_1(v) = f_1(\mathsf{source}(e)), g_2(v) = \delta(f_2(\mathsf{source}(e)), e)$.

We now define the next-move function of S_2. Let $f: V \to V \times M \cup \{\perp\}$ be a state and $v \in V_P$ be a node of Protagonist. 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

$f_{init}(\mathsf{target}(p)) = (\mathsf{target}(p), m_{init}) \neq \bot$ and, obviously, $\mathsf{source}(p) \preceq \mathsf{target}(p)$ (just because p is a 0-length path so that $\mathsf{source}(p) = \mathsf{target}(p)$).

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 $f\colon V \to V \times M \cup \{\bot\}$ which is the state of S_2 after p'. Next, let the state of S_2 after p be $g\colon V \to V \times M \cup \{\bot\}$. Note that g is the value of the transition function of S_2 on f and $c = \mathsf{col}(e') \in C$.

To check the soundness for p and g, one can use exactly the same argument as in Theorem 3 for f_2 and g_2. That is, for any $v \in V$ with $g(v) \neq \bot$, we consider an (f_2, v, c)-good edge e which was used to define $g(v)$. By (1), we have $f_2(\mathsf{source}(e)) \neq \bot \implies f(\mathsf{source}(e)) \neq \bot$. Then, using the induction hypothesis for p', we take a path p'_1 establishing the soundness for f and p' at $\mathsf{source}(e)$. Finally, we define $p_1 = p'_1 e$ 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 3, we have that, first, p_1 is consistent with S_1, second, $\mathsf{col}(p_1) = \mathsf{col}(p)$, 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, $g_1(v) = f_1(\mathsf{source}(e))$. 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 $\mathsf{source}(e)$, the starting node of p'_1 must be $f_1(\mathsf{source}(e)) = g_1(v)$, as required.

We now check the completeness property for p and g. It is sufficient to show the existence of an $(f_2, \mathsf{target}(p), c)$-good edge e with $\mathsf{source}(p) \preceq f_1(\mathsf{source}(e))$. Indeed, $g(\mathsf{target}(p)) \neq \bot$ if and only if $(f_2, \mathsf{target}(p), c)$-good edges exist, and $g_1(\mathsf{target}(p))$ is the maximum of $f_1(\mathsf{source}(e))$ 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, $v = \mathsf{target}(p)$ and c. Indeed, $\mathsf{target}(e') = \mathsf{target}(p)$ because e' is the last edge of p, $\mathsf{col}(e') = c$ by definition of c, and $f_2(\mathsf{source}(e')) = f_2(\mathsf{target}(p')) \neq \bot$ by the completeness for p' and f. Let us now verify (2). Assume that $\mathsf{source}(e') \in V_P$. 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 $S_2(p') = S_2(\mathsf{target}(p'), f) = S_2(\mathsf{source}(e'), f)$. Note that $f(\mathsf{source}(e')) = f(\mathsf{target}(p')) \neq \bot$ by the completeness for p' and f. Hence, by definition of S_2, we have $S_2(\mathsf{source}(e'), f) = S_1(\mathsf{source}(e'), f_2(\mathsf{source}(e')))$, and, thus, e' satisfies (2). Finally, we have to show that $\mathsf{source}(p) \preceq f_1(\mathsf{source}(e))$. This is because, by the completeness for p' and f, we have $\mathsf{source}(p') \preceq f_1(\mathsf{target}(p'))$. It remains to note that $\mathsf{source}(p) = \mathsf{source}(p')$ and $\mathsf{source}(e') = \mathsf{target}(p')$ – recall that e' is the last edge of p and p' is the part of p which precedes e'.