Checking History Determinism for Parity Automata Is in NP

Karoliina Lehtinen
CNRS, Aix Marseille Univ., LIS, Marseille, France

Keya Prakash
Aix Marseille Univ., CNRS, LIS, Marseille, France

Michał Skrzypczak
University of Warsaw, Poland

Abstract

History-deterministic automata, often also called good-for-games, are an intermediate model between deterministic and nondeterministic automata, which are particularly well-suited for applications in verification and reactive synthesis. We show that deciding whether a parity automaton is history-deterministic is in NP. Our result matches an NP-hardness lower bound (Prakash 2024) and builds on insights from a fixed-parameter tractable algorithm (Lehtinen and Prakash 2025). This settles the complexity of the problem, which has been open since 2006.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases History-determinism, parity automata, good-for-games

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.64

Funding *Keya Prakash*: Supported by ANR grant QuaSy (ANR 23-CE48-0008).
Michał Skrzypczak: Supported by Poland's NCN grant Opus no. 2024/55/B/ST6/00318.

1 Introduction

Background

History-deterministic automata were originally introduced by Henzinger and Piterman [17], under the name of *good-for-games* automata, as a way to model systems that can resolve their nondeterministic choices based on the history of the run so far, but without knowledge of the future input. The original motivation for the model was to provide a more efficient alternative to full determinisation of ω-automata in the context of synthesis [9]. The model was independently developed by Colcombet and Löding [11] with the aim of replacing deterministic automata in a quantitative context, where determinisation is not always possible. Kupferman, Safra, and Vardi in 1996 also studied and introduced what is currently known as good-for-trees automata, which are equivalent to history determinism for ω-regular automata [20, 5].

Initially, little was known about the relations between history deterministic and deterministic automata. A first breakthrough was achieved by Boker et al. [5], who showed that some history-deterministic automata are not *determinisable by pruning* (DBP), i.e., they cannot be made deterministic by removing some of their transitions. Then, Kuperberg and Skrzypczak showed that in the case of coBüchi automata history determinism can provide exponential succinctness over deterministic automata [18]. This showed that history determinism could capture some of power of nondeterminism, such as succinctness, while retaining some of the algorithmic properties of determinism, such as effective decision procedures for synthesis, universality, and inclusion. Since then, history determinism has been studied extensively both in the ω-regular setting and beyond (see surveys [19, 7]). For example, for ω-regular automata, history determinism has been shown to facilitate minimisation of coBüchi automata [27, 1], has been used to construct canonical representations for ω-regular languages [12], and recently, for learning of coBüchi languages [22]. History determinism has also been studied beyond the regular setting, where they are more expressive and more succinct than the deterministic versions, without the same cost to complexity and decidability as with full nondeterminism, for example for timed and pushdown automata [8, 14].

Yet, despite all the attention that history determinism has received over the past decades, the central question of the complexity of deciding whether a parity automaton is history deterministic has remained stubbornly open since the introduction of the model in 2006 by Henzinger and Piterman [17]. This question is not only central to understanding history-deterministic parity automata, it has also been a bottleneck for deciding history determinism of more general models, such as timed automata over infinite words. In their original paper, Henzinger and Piterman provided an EXPTIME decision procedure, consisting of checking whether the automaton simulates an equivalent deterministic automaton [17, Section 4]. Its complexity comes from the cost of determinisation.

The first indication that history-deterministic automata could be recognised with faster algorithms came in 2015, when Kuperberg and Skrzypczak gave a polynomial-time algorithm for recognising HD *coBüchi* automata [18]. Then, in 2018, Bagnol and Kuperberg [4] provided a polynomial-time algorithm for the dual case, that is, recognising HD Büchi automata. This algorithm introduced and used the *2-token games* [4], which Bagnol and Kuperberg conjectured to characterise history determinism for parity automata in general. They showed that this indeed holds for the Büchi case [4], while Boker et al. showed it for the dual coBüchi case [6]. However, atypically, the Büchi and coBüchi proofs could not be combined to prove the general case. In 2024, Prakash showed that the problem is NP-hard [25, Theorem 15] and in 2025 Lehtinen and Prakash showed that it is fixed-parameter tractable with respect to the index of the parity automaton [21], by showing that the 2-token games characterise history determinism for parity automata. Their procedure is in PSPACE when the index is not fixed, as they show that 2-token games can be solved in PSPACE [26, The 2-Token Theorem].

Here we show that the problem of checking history determinism is in NP. Together with the NP-hardness result for the problem due to Prakash [25, Theorem 15], we obtain NP-completeness.

▶ **Theorem 1.** *The problem of deciding whether a given parity automaton is history deterministic is in* NP.

Proof overview. We build on insights and technical tools from [21] to provide a normal form for history-deterministic parity automata. The relevant lemmata are stated in Section 2.4. Then, our core construction further refines the structure of history-deterministic parity automata to obtain the NP witness. Our construction is done inductively, showing how to witness that a given automaton is HD by witnessing that some other simpler automata (with fewer priotities) are HD. This involves two steps, corresponding to whether the most important priority of the given automaton is odd or even, and are handled in Section 3 and Section 4 respectively. The case of a most significant odd priority is the more difficult one, and the focus of most of our efforts and novel insights. Then, Section 5 argues that this inductively nested structure of witnesses in fact can be encoded as a single, polynomial-size witness for the original automaton being HD, that can be verified in polynomial time.

2 Preliminaries

$\mathbb{N} = \{0, 1, \ldots\}$ is the set of natural numbers. For $i, j \in \mathbb{N}$ we let $[i, j] = \{i, i+1, \ldots, j\}$. An *index* I is any set of the form $[i, j]$ for some $i, j \in \mathbb{N}$ with $i \leq j$ (typically $i < j$, we will discuss this later). By an *alphabet* Σ, we refer to a finite set whose elements we call *letters*. We use Σ^* and Σ^ω to denote the set of finite and infinite words respectively. A language of finite (resp. infinite) words is a subset of Σ^* (resp. Σ^ω).

If $L \subseteq A^\omega$ and $w \in A^*$ then by $w^{-1} \cdot L$ we denote the set $\{x \in A^\omega \mid wx \in L\}$.

2.1 Automata

Parity automata. A nondeterministic parity automaton has the syntax of a nondeterministic finite state automaton, except that each of its transitions is additionally labelled by a finite natural number, which we call the *priority* of that transition. Parity automata take as input an infinite word, and accordingly, the runs of parity automata on infinite words have infinite length. We say that such a run is *accepting* if the *least priority occurring infinitely often in that run is even*, and that a run is *rejecting* otherwise.

Formally, a parity automaton $\mathcal{A}$ of index I is a tuple $\mathcal{A} = \langle Q, \Sigma, \Delta, q_\iota \rangle$, where Q is a finite set of states, Σ is an alphabet, $q_\iota \in Q$ is an initial state, and $\Delta \subseteq Q \times \Sigma \times I \times Q$ is a set of transitions. We use $p \xrightarrow{a:c} q$ to denote the transition (p, a, c, q), from $p \in Q$ to $q \in Q$ on a letter $a \in \Sigma$ with priority $c \in I$. If the priority c is irrelevant then we sometimes skip it, writing just $p \xrightarrow{a} q$. A run of $\mathcal{A}$ over an infinite word $(a_i)_{i \in \mathbb{N}}$ is a sequence $(q_i, a_i, c_i, q_{i+1})_{i \in \mathbb{N}} \in \Delta^\omega$ of transitions. An infinite run is accepting if it starts from $q_0 = q_\iota$ and if $\liminf (c_i)_{i \in \mathbb{N}}$ is even; otherwise it is rejecting. Runs over finite words are defined analogously, but are neither accepting nor rejecting. For an automaton $\mathcal{A}$ and a state q, we use $(\mathcal{A}, q)$ to denote the automaton $\mathcal{A}$ with the initial state set as q. A word $w \in \Sigma^\omega$ is accepted by an automaton $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on w, and the language $L(\mathcal{A})$ of an automaton is the set of words accepted by $\mathcal{A}$.

Index. The *index* of $\mathcal{A}$ is the smallest interval $I = [i, j]$ containing all of the priorities of $\mathcal{A}$. In this case, we say that $\mathcal{A}$ is an I-*automaton*. We can assume, without loss of generality, that the least element $\min(I)$ of I is either 0 or 1, since we can always shift the index of a parity automaton by any even number without affecting the acceptance of runs. A parity automaton $\mathcal{A}$ is of *odd index* (resp. *even index*) if its index is of the form $[1, k]$ (resp. $[0, k]$) for some k. *Büchi automata* are $[0, 1]$ automata, and *coBüchi automata* are $[1, 2]$ automata.

2.2 Games on Automata

Games. In this work, we will reason with ω-regular games. These are games played between the players *Eve* and *Adam* on a finite directed graph $G = (V, E)$ called the *arena*. The game is defined by this arena, a partition $V = V_\exists \uplus V_\forall$ of vertices into Eve vertices and Adam vertices, respectively, and a winning condition $W \subseteq E^\omega$. A *play* of this game starts at a designated initial vertex v_I, and when the play is at a vertex v, the player whose vertex it is chooses an outgoing edge (v, v') from v, and the play goes to position v'. Such a play ρ, in the limit, forms an infinite word in E^ω. We say that ρ is winning for *Eve* if ρ is in W, and otherwise it is winning for *Adam*. A *strategy* for a player is a function that, given the history (prefix) of the play so far ending at that player's vertex, prescribes the next outgoing edge for that player. A strategy is *winning* for a player if all plays consistent with that strategy are winning for that player.

In ω-regular games, the winning objective W is the language of a nondeterministic Büchi automaton over the alphabet E. Every ω-regular game $\mathcal{G}$ is determined, i.e., there is a *winner* who has a *winning strategy* in $\mathcal{G}$ [23, 15].

▶ Remark 2. In the games we reason with, the formal definition of the arena is typically cumbersome and unenlightening, so we prefer to work with more succinct and instructive higher-level descriptions. The next two games, the HD game and the 1-token game are examples of this.

History-deterministic automata. We define history-deterministic automata via the HD game (short for history determinism game), which is defined below.

▶ **Definition 3** (HD game). *For a parity automaton $\mathcal{A}$, the HD game on $\mathcal{A}$, denoted by $HD(\mathcal{A})$, starts with a token for Eve at the initial state of the automaton, and proceeds in infinitely many rounds. In round i, when Eve's token is at q_i:*

1. *Adam selects a letter a_i,*
2. *Eve selects a transition $q_i \xrightarrow{a_i:c_i} q_{i+1}$, and the token moves to the state q_{i+1}.*

This way, a play results in Adam building a word $(a_i)_{i\in\mathbb{N}}$, and Eve building a run $(q_i \xrightarrow{a_i:c_i} q_{i+1})_{i\in\mathbb{N}}$ on that word, corresponding to the path taken by her token. We say that Eve wins this play if either Adam's word is rejected (i.e., does not belong to the language of $\mathcal{A}$) or Eve's run is accepting.

We say that an automaton is history deterministic if Eve has a winning strategy in the HD game on that automaton.

Token games. The following 1-*token game*, by now a standard tool in the HD literature, is a variant of the HD game in which Adam, in addition to building letter-by-letter a word in the language, has to build transition-by-transition an accepting run to witness this. This makes the game easier for Eve, since Adam gives her more information than in the HD game. Unlike the HD game, which is played on a single automaton, we define the 1-token game to be played on two potentially (but not necessarily) different automata. This also makes it very similar to the *fair simulation game* [16], except that here Adam plays his transition second, while in a fair simulation game he plays it first along with the letter[1].

▶ **Definition 4** (The 1-token game). *Given two automata $\mathcal{A}$ and $\mathcal{B}$, the 1-token game of $\mathcal{A}$ against $\mathcal{B}$, denoted $G1(\mathcal{A};\mathcal{B})$ is defined as follows. The game starts with Eve's token at the initial state of $\mathcal{A}$ and Adam's token at the initial state of $\mathcal{B}$. In round i, when Eve's token is at p_i in $\mathcal{A}$ and Adam's token is at q_i in $\mathcal{B}$:*

1. *Adam chooses a letter $a_i \in \Sigma$,*
2. *Eve chooses a transition $p_i \xrightarrow{a_i:c_i} p_{i+1}$ in $\mathcal{A}$ from p_i over a_i,*
3. *Adam chooses a transition $q_i \xrightarrow{a_i:c_i'} q_{i+1}$ in $\mathcal{B}$ from q_i over a_i.*

The game then proceeds to the next round with Eve and Adam's tokens at p_{i+1} and q_{i+1}, respectively. Thus, a play consisting of infinitely many rounds results in Adam building a word $(a_i)_{i\in\mathbb{N}}$, Eve building a run $(p_i \xrightarrow{a_i:c_i} p_{i+1})_{i\in\mathbb{N}}$ traced by her token in $\mathcal{A}$, and Adam building a run on $(q_i \xrightarrow{a_i:c_i'} q_{i+1})_{i\in\mathbb{N}}$ traced by his token in $\mathcal{B}$. We say that Eve wins such a play if either her run is accepting, or Adam's run is rejecting. *If Eve has a winning strategy in $G1(\mathcal{A};\mathcal{B})$, then we say that* Eve wins $G1(\mathcal{A};\mathcal{B})$.

Two automata $\mathcal{A}$ and $\mathcal{B}$ are $G1$-equivalent if Eve wins both $G1(\mathcal{A};\mathcal{B})$ and $G1(\mathcal{B};\mathcal{A})$.

[1] We note that the present results can be obtained with either of the 1-token game or the simulation game, since they are equivalent on HD automata.

We write $G1(p;q)$ in $\mathcal{A}$ to denote the game $G1((\mathcal{A},p);(\mathcal{A},q))$. The following observation is easy to make, since Eve's winning strategy constructs an accepting run in $\mathcal{A}$ against every accepting run in $\mathcal{B}$ on the same word in $G1(\mathcal{A};\mathcal{B})$.

▶ **Lemma 5.** *If Eve wins $G1(\mathcal{A};\mathcal{B})$ for two parity automata $\mathcal{A}$ and $\mathcal{B}$, then $L(\mathcal{B}) \subseteq L(\mathcal{A})$.*

The following allows us to reduce language-inclusion for HD automata to 1-token games.

▶ **Lemma 6.** *If $\mathcal{A}$ is history deterministic and $\mathrm{L}(\mathcal{A}) \supseteq \mathrm{L}(\mathcal{B})$ then Eve wins $\mathrm{G1}(\mathcal{A};\mathcal{B})$.*

Proof. If Eve has a winning strategy in the HD game on $\mathcal{A}$, then she can use her winning strategy in $\mathcal{A}$ to play $G1(\mathcal{A};\mathcal{B})$, ignoring Adam's transitions in $\mathcal{B}$. If Adam's run on $\mathcal{B}$ in $G1(\mathcal{A};\mathcal{B})$ is accepting, then Adam's word $w \in L(\mathcal{B}) \subseteq L(\mathcal{A})$, and thus Eve's HD-game strategy builds an accepting run on w in $\mathcal{A}$. ◀

An automaton for which Eve wins $G1$ against language-equivalent HD automata is also HD.

▶ **Lemma 7.** *Let $\mathcal{A}$ and $\mathcal{B}$ be two parity automata that are language-equivalent, such that $\mathcal{B}$ is HD. If Eve wins $G1(\mathcal{A};\mathcal{B})$, then $\mathcal{A}$ is HD.*

Proof. Let σ_{G1} be a winning strategy for Eve in $G1(\mathcal{A};\mathcal{B})$, and $\sigma_{\mathcal{B}}$ be a winning strategy for Eve in the HD game on $\mathcal{B}$. We will construct a winning strategy for Eve in the HD game on $\mathcal{A}$ as follows. During the HD game on $\mathcal{A}$, Eve keeps a corresponding play of $G1(\mathcal{A};\mathcal{B})$ in her memory, where Adam is playing the same letters as in the HD game on $\mathcal{A}$ and choosing transitions for his token in $\mathcal{B}$ according to $\sigma_{\mathcal{B}}$, while she herself chooses transitions according to σ_{G1}. Then, Eve chooses the same transitions for her token in the HD game on $\mathcal{A}$ as her strategy σ_{G1} in $G1(\mathcal{A};\mathcal{B})$ against Adam's token. This way, whenever Adam's word w in the HD game is in $L(\mathcal{A}) = L(\mathcal{B})$, Adam's run on $\mathcal{B}$ in Eve's memory is accepting, and thus, her run in the HD game on $\mathcal{A}$ must be accepting as well. ◀

▶ **Lemma 8** (G1 transitivity)**.** *Given three parity automata $\mathcal{A},\mathcal{B}$ and $\mathcal{C}$, if Eve wins $G1(\mathcal{A};\mathcal{B})$ and $G1(\mathcal{B};\mathcal{C})$ then she also wins $G1(\mathcal{A};\mathcal{C})$.*

Proof. To win in $G1(\mathcal{A};\mathcal{C})$, Eve can compose her winning strategies σ_1 and σ_2 in $G1(\mathcal{A};\mathcal{B})$ and $G1(\mathcal{B};\mathcal{C})$, respectively. That is, she keeps in her memory positions of both games $G1(\mathcal{A};\mathcal{B})$ and $G1(\mathcal{B};\mathcal{C})$. When Adam plays a letter a_i at a round i, she responds with the transition t played by σ_1 in $G1(\mathcal{A};\mathcal{B})$. She then updates the position of $G1(\mathcal{B};\mathcal{C})$ in her memory by having σ_2 pick a transition t' of $\mathcal{B}$. She then updates the position of $G1(\mathcal{A};\mathcal{B})$ by first playing t and then having Adam play t' in $\mathcal{B}$. When Adam plays a transition in $\mathcal{C}$, she uses this to also update the position of the game $G1(\mathcal{B};\mathcal{C})$.

Then, if Adam builds an accepting run in $\mathcal{C}$ in $G1(\mathcal{A};\mathcal{C})$, the strategy σ_2 builds an accepting run of $\mathcal{B}$ in Eve's memory, and σ_1 guarantees that Eve buils an accepting run in $\mathcal{A}$. ◀

Computational Complexity of 1-token games. The game $G1(\mathcal{A};\mathcal{B})$, when written out as an ω-regular game, has as its winning condition for Eve as the disjunction between two parity conditions, which is a Rabin condition [10], and such games have positional strategies for Eve [13].

Therefore, if Eve wins the 1-token game $G1(\mathcal{A};\mathcal{B})$ for two parity automata $\mathcal{A}$ and $\mathcal{B}$, then she wins using a *positional strategy*; i.e., a strategy that only depends on the current states of her and Adam's token, and the letter chosen by Adam. Furthermore, given such a positional strategy for Eve in $G1(\mathcal{A};\mathcal{B})$, we can verify in polynomial time if it is winning, as this corresponds to checking whether all paths induced by the strategy in the arena (consisting broadly of a product of $\mathcal{A}$ and $\mathcal{B}$ with some intermediate positions to capture the steps of each round) satisfy the disjunction of two parity conditions.

▶ **Lemma 9** ([26, Proposition 3.16]). *There is an NP procedure to decide whether Eve wins the* 1*-token game* $G1(\mathcal{A};\mathcal{B})$ *for two given parity automata, where the polynomial witness is a positional strategy for Eve.*

We note that the above complexity is tight: the problem of solving 1-token games is NP-hard [25, Theorem 4.1].

2.3 State relations

In this section we recall some useful notions that relate states with each other, which we will use throughout the paper.

The first such relationship is *weak coreachability*, which corresponds to states being reachable with the same word, and the transitive closure thereof. Formally, given any automaton $\mathcal{A}$, define $\mathrm{WCR}(\mathcal{A}) \subseteq Q \times Q$ as the least equivalence relation over Q such that if $(p,q) \in \mathrm{WCR}(\mathcal{A})$ and $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ then $(p',q') \in \mathrm{WCR}(\mathcal{A})$. Note that this relation can be computed in polynomial time. This relation is symmetric and we write $p \in \mathrm{WCR}(\mathcal{A},q)$ if $(p,q) \in \mathrm{WCR}(\mathcal{A})$.

We say that an automaton $\mathcal{A}$ is *semantically deterministic* if for every transition $p \xrightarrow{a} p'$, the following condition is true:

$$a^{-1} \cdot \mathrm{L}(\mathcal{A},p) = \mathrm{L}(\mathcal{A},p').$$

In other words, these are automata in which every transition from q on a letter a is language-maximal, and could therefore potentially be used by a strategy in the HD game.

▶ **Lemma 10.** *If* $\mathcal{A}$ *is semantically deterministic then* $\mathrm{L}(\mathcal{A},p) = \mathrm{L}(\mathcal{A},q)$ *holds for every* $(p,q) \in \mathrm{WCR}(\mathcal{A})$.

Proof. By the definition of WCR it is enough to show that if $(p,q) \in \mathrm{WCR}(\mathcal{A})$ and $p \xrightarrow{a} p'$ and $q \xrightarrow{a} q'$ then $\mathrm{L}(\mathcal{A},p') = \mathrm{L}(\mathcal{A},q')$. But this follows directly from the assumption that $\mathcal{A}$ is semantically deterministic:

$$\mathrm{L}(\mathcal{A},p') = a^{-1} \cdot \mathrm{L}(\mathcal{A},p) = a^{-1} \cdot \mathrm{L}(\mathcal{A},q) = \mathrm{L}(\mathcal{A},q'). \qquad ◀$$

Thus, for a semantically deterministic automaton, if $(p,q) \in \mathrm{WCR}(\mathcal{A})$ then p and q are language-equivalent in the sense that $\mathrm{L}(\mathcal{A},p) = \mathrm{L}(\mathcal{A},q)$.

Given a parity automaton $\mathcal{A}$, we say that Eve wins the 1*-token game from everywhere* in $\mathcal{A}$ if Eve wins $G1((\mathcal{A},p);(\mathcal{A},q))$ for all $(p,q) \in \mathrm{WCR}(\mathcal{A})$. This is stronger than semantic determinacy as in addition to the semantic condition, every pair in WCR has to also satisfy the more syntactic, simulation-like relation given by the 1-token game.

▶ **Lemma 11** ([26, Lemma 3.36]). *Every parity automaton on which Eve wins the* 1*-token game from everywhere is semantically deterministic.*

2.4 Structural results on HD automata

We will need two lemmata from Lehtinen and Prakash [21]. We first define $\mathcal{A}_{>0}$ and $\mathcal{A}_{>1}$ for $\mathcal{A}$ being a $[0,k]$ and $[1,k]$-automaton, respectively, as the automata obtained by removing (and redirecting into an appropriate sink state) the transitions of the lowest priorities. These are the sub-automata that will enable us to reason inductively.

For a $[0,k]$-automaton $\mathcal{A}$, we define the automata $\mathcal{A}_{>0}$ as the automaton obtained from $\mathcal{A}$ by adding an accepting sink state $q_\top$, with self-loops $q_\top \xrightarrow{a:2} q_\top$ on all letters $a \in \Sigma$, and replacing every transition $q \xrightarrow{a:0} q'$ by $q \xrightarrow{a:2} q_\top$. The initial state of $\mathcal{A}_{>0}$ is same as the initial state of $\mathcal{A}$. Note that if $k > 1$, then $\mathcal{A}_{>0}$ is a $[1,k]$-automaton. The automaton $\mathcal{A}_{>0}$ accepts words on which there is a run of $\mathcal{A}$ that is either accepting, or contains a 0-transition. The following observation follows.

▶ **Observation 12.** *For every $[0,k]$ automaton $\mathcal{A}$, the following language-containment relation holds: $L(\mathcal{A}) \subseteq L(\mathcal{A}_{>0})$.*

For a $[1,k]$ automaton $\mathcal{A}$ with, we define a $[2,k]$-automaton $\mathcal{A}_{>1}$ from $\mathcal{A}$ analogously, by adding a rejecting sink state $q_\perp$ with self-loops $q_\perp \xrightarrow{a:3} q_\perp$ on all letters $a \in \Sigma$ and replacing each transition $q \xrightarrow{a:1} q'$ in $\mathcal{A}$ by $q \xrightarrow{a:3} q_\perp$. The initial state of $\mathcal{A}_{>1}$ is the same as the initial state of $\mathcal{A}$. Note that if $k > 2$, then $\mathcal{A}_{>1}$ is a $[2,k]$-automaton (and equivalent to a $[0,k-2]$ one). The automaton $\mathcal{A}_{>1}$ accepts words on which there is an accepting run of $\mathcal{A}$ that does not contain a 1-transition. Thus, analogous to Observation 12, we have the following observation.

▶ **Observation 13.** *For every $[1,k]$ automaton $\mathcal{A}$, the following language-containment relation holds: $L(\mathcal{A}_{>1}) \subseteq L(\mathcal{A})$.*

We now state the two lemmata we need from [21], which give us a normal form for HD automata with even and odd indices, respectively. In both cases, Eve wins the 1-token game from everywhere, that is, (weakly) co-reachable states are not only languague-equivalent, but also $G1$-equivalent. Furthermore, the normal form identifies states q that are HD in the subautomaton $\mathsf{A}_{>i}$ without the most significant priority ($i \in \{0,1\}$, as appropriate), and requires every state p of $\mathcal{A}$ to be weakly-coreachable to such a state q in the original automaton $\mathcal{A}$. These two lemmata additonally provide a $G1$-relation between these states p and q, the order of which crucially differs depending on whether i is 0 or 1.

▶ **Lemma 14.** *For every $k > 1$ and every $[0,k]$-automaton $\mathcal{A}'$ that is HD, $\mathcal{A}'$ can be converted to a language-equivalent HD $[0,k]$-automaton $\mathcal{A}$ without increasing the number of states, so that the following two conditions are true.*

1. *Eve wins the 1-token game from everywhere in $\mathcal{A}$.*
2. *For every state p, there is a state $q \in \mathrm{WCR}(\mathcal{A}, p)$ such that Eve wins $G1((\mathcal{A}_{>0}, p); (\mathcal{A}_{>0}, q))$ and $(\mathcal{A}_{>0}, q)$ is HD.*

▶ **Lemma 15.** *For every $k > 2$ and every $[1,k]$-automaton $\mathcal{A}'$ that is HD, $\mathcal{A}'$ can be converted to a language-equivalent HD $[1,k]$-automaton $\mathcal{A}$ without increasing the number of states, so that the following two conditions are true.*

1. *Eve wins the 1-token game from everywhere in $\mathcal{A}$.*
2. *For every state p, there is a state $q \in \mathrm{WCR}(\mathcal{A}, p)$ such that Eve wins $G1((\mathcal{A}_{>1}, q); (\mathcal{A}_{>1}, p))$ and $(\mathcal{A}_{>1}, q)$ is HD.*

Both these lemmata were originally proved in [21] and [26, Lemma 6.4, Theorem I; Lemma 7.6]. The statements there are stronger, as they claim that Eve wins the 2-token games $G2$ from respective states in the restricted automata. In the present work we do not use the 2-token games (for a definition, see [26, Definition 3.22]); but the fact that Eve winning $G2$ implies her winning $G1$ is obvious (see, e.g., [26, Lemma 3.25]) and therefore the above statements directly follow.

Similar conditions have appeared in earlier literature for HD coBüchi or Büchi automata too, e.g., in [18, Lemma 41 and 54 in full version], [2, Lemma 26], and in [28, Lemma 7], and they are at the heart of the proof of the $G2$ characterisation of history determinism.

While Lemma 14 allows us to obtain a polynomial witness fairly directly for $[0, k]$ automata (Section 4), the normal form provided by Lemma 15 is not by itself sufficient for the $[1, k]$ case, but requires the additional notion of *fullness*, in Definition 17, key to the technical development of the next section.

3 Even to Odd

We proceed towards building an NP procedure for checking history determinism. As we mentioned earlier in Section 1, this will involve witnessing the history determinism of the automata in question by witnessing history determinism for automata with fewer priorities. Additionally, we will also require Eve to win some 1-token games in relevant automata; these are witnessed by positional strategies for Eve (see Lemma 9).

In this section, we will present how we can characterise the history determinism for $[1, k]$ automata by history determinism for $[2, k]$ automata. In the next section, we will present how we can characterise the history determinism for $[0, k]$ automata in terms of history determinism for $[1, k]$ automata.

We fix $k > 2$ for the reset of this section. Towards characterising history determinism of $[1, k]$-automata in terms of history determinism for $[2, k]$-automata, we will first aim to define *normalised* $[1, k]$*-automata*, for which the following three properties are true.

1. Every HD $[1, k]$-automaton $\mathcal{A}$ has a language-equivalent normalised $[1, k]$-automaton $\mathcal{A}'$.
2. Every normalised $[1, k]$-automaton is HD.
3. Verifying whether a $[1, k]$-automaton is normalised can by done by checking history determinism for linearly many $[2, k]$-automata and linearly many positional-strategies in 1-token games on automata.

We will present the exact definition of normalised automata later, in Definition 17, which will allow us to use Lemmas 7 and 9 to meet the third criterion above.

Our notion of normalised automata aims to generalise properties of minimal coBüchi automata [1], including *safe determinism*, which means that removing the rejecting transitions results in a deterministic safety automaton. Here, we replace determinism by history determinism. To develop an intuition of this form, we first present a failed attempt at defining normalised automata, and we will then use the intuition obtained to present our definition (Definition 17).

A first attempt. Given a $[1, k']$-automaton $\mathcal{A}$, we say that a state q is 1*-safe HD* if the automaton $(\mathcal{A}_{>1}, q)$ is HD. Abusing the notation slightly, we say that $\mathcal{A}$ is 1*-safe HD*, if every state of $\mathcal{A}$ is 1-safe HD.

Consider $[1, k']$-automata in which:

1. Eve wins the 1-token game from everywhere, and
2. that are 1-safe HD.

If $k' = 2$, i.e., we are working with coBüchi automata, such automata are HD [21, Theorem 3.2]. Conversely, every HD coBüchi automaton can be converted, without increasing the number of states, into a language-equivalent HD coBüchi automaton that satisfies the above two properties (as we will prove in Lemma 21 for $[1, k']$-automata).

These properties can be checked by verifying whether there is a winning positional strategy for Eve in the 1-token game from everywhere (Lemma 9) and verifying history determinism for $[2, k']$-automata. Unfortunately, they does not guarantee history determinism when $k' > 2$ (i.e., the case considered in this section), as the following example shows.

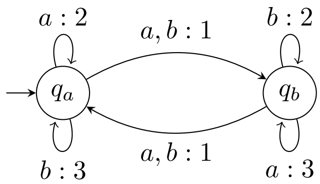

Figure 1 A 1-safe HD $[1, 3]$-automaton $\mathcal{A}$ in which Eve wins the 1-token game from everywhere.

▶ **Lemma 16.** *In the automaton $\mathcal{A}$ in Figure 1,*
1. *Eve wins the* 1*-token game from everywhere;*
2. *$\mathcal{A}$ is* 1*-safe HD*
3. *and $\mathcal{A}$ is not HD.*

Proof.

$\mathcal{A}$ is not HD. We note that the language of $\mathcal{A}$ is

$$\mathrm{L}(\mathcal{A}) = (a + b)^\omega,$$

i.e., the set of all words over $\{a, b\}$. Indeed, every infinite word over $\{a, b\}$ contains either infinitely many a's or infinitely many b's, and runs that eventually only take self-loops on q_a (resp. q_b) are accepting runs for words that contain infinitely many as (resp. bs). The automaton $\mathcal{A}$ is not HD, as Adam has the following winning strategy in its HD-game: when Eve's token is at q_a, Adam selects b, and when Eve's token is at q_b, Adam selects a. This way, Eve's token only takes transitions of odd priority and her run is rejecting, while Adam's word is in the language.

$\mathcal{A}$ satisfies properties (1) and (2). Note that $\mathcal{A}_{>1}$ is a deterministic automaton, and thus, $\mathcal{A}$ is 1-safe HD. Eve furthermore wins the 1-token game from everywhere in $\mathcal{A}$, using the following strategy: in every round, she takes the transition to Adam's token. If Adam's run is accepting, then Adam's run eventually only takes self-loops on either q_a or q_b, and thus, Eve's run eventually coincides with Adam's run and is accepting as well. ◀

The above example illustrates the need for a stronger condition for guaranteeing history determinism. Informally, once Eve wins the 1-token game from everywhere and for all states q, the automaton $(\mathcal{A}_{>1}, q)$ is HD, we further require that Eve does not have to resolve the nondeterminism between choosing to follow the HD strategy in $\mathcal{A}_{>1}$ or taking a 1-transition. That is, in the HD game, if Adam is giving a word in $\mathrm{L}(\mathcal{A})$, we desire that if Eve follows an HD strategy in $\mathcal{A}_{>1}$, either she builds an accepting run or she is forced to see priority 1.

Towards formalising this, we define, for every $[1, k]$-automaton $\mathcal{A}$, the safety automaton $\mathcal{A}_{\mathrm{safe}}$, which accepts words with a run of $\mathcal{A}$ that avoids priority 1, as follows. The automaton $\mathcal{A}_{\mathrm{safe}}$ has the same states as $\mathcal{A}$ and an additional rejecting sink state $q_\perp$. The transitions of $\mathcal{A}$ that have priority at least 2 are kept and have priority 2 in $\mathcal{A}_{\mathrm{safe}}$, while transitions of priority 1 in $\mathcal{A}$ are redirected to $q_\perp$ in $\mathcal{A}_{\mathrm{safe}}$. The rejecting sink state $q_\perp$ has a self-loop of priority 1 on every letter.

▶ **Definition 17.** *We say that a $[1, k]$-automaton $\mathcal{A}$ is normalised if it satisfies the following three conditions.*
1. *Eve wins the* 1*-token game from everywhere in $\mathcal{A}$.*
2. *$\mathcal{A}$ is* 1*-safe HD, i.e., for every state q in $\mathcal{A}$, $(\mathcal{A}_{>1}, q)$ is HD.*
3. Fullness*: for every state q in $\mathcal{A}$, the following holds.*

$$\mathrm{L}(\mathcal{A}_{>1}, q) = \mathrm{L}(\mathcal{A}, q) \cap \mathrm{L}(\mathcal{A}_{\mathrm{safe}}, q). \tag{1}$$

We call states satisfying the language-equivalence of 3 full states *in $\mathcal{A}$.*

Note that fullness is precisely what is false for the automaton in Figure 1.

▶ **Fact 18.** *For every $[1,k]$-automaton $\mathcal{A}$, the following two language-inclusions hold:*

$$\mathrm{L}(\mathcal{A}_{>1}, q) \subseteq \mathrm{L}(\mathcal{A}_{\text{safe}}, q); \quad \mathrm{L}(\mathcal{A}_{>1}, q) \subseteq \mathrm{L}(\mathcal{A}, q).$$

The fullness condition captures what we needed: in the HD game on a normalised automaton $\mathcal{A}$, if Eve is following an HD strategy in $\mathcal{A}_{>1}$ against an accepting word given by Adam, then either she is forced to take a 1-transition, or her run is accepting.

We next describe our witness for deciding history determinism of $[1,k]$-automata.

3.1 Even-to-odd witness

We claim that a given $[1,k]$-automaton $\mathcal{A}$ is HD if and only if there exists a normalised $[1,k]$-automaton $\mathcal{B}$ with at most as many states as $\mathcal{A}$, which satisfies the following *even-to-odd witness conditions.*

- $\mathcal{A}$ and $\mathcal{B}$ are $G1$-equivalent.
- $\mathcal{B}$ is normalised.

We will show the soundness and correctness of the proposed even-to-odd witness condition in Sections 3.2 and 3.3, respectively.

3.2 Soundness

▶ **Lemma 19** (Soundness)**.** *If a $[1,k]$-automaton $\mathcal{A}$ has a $G1$-equivalent normalised $[1,k]$-automaton $\mathcal{B}$, then $\mathcal{A}$ is HD.*

Proof. Due to Lemmas 5 and 7, it suffices to prove that $\mathcal{B}$ is HD. We will do so by describing a winning strategy γ for Eve in the HD game on $\mathcal{B}$. Since Eve wins the HD game on $(\mathcal{B}_{>1}, q)$ for all states q, we fix a winning strategy $\gamma_{>1}$ for Eve in the HD game from all states in $\mathcal{B}_{>1}$. We also fix a positional strategy σ_{G1} in $G1(\mathcal{B};\mathcal{B})$ that is winning from every pair of weakly coreachable states.

The strategy γ for Eve in $\mathrm{HD}(\mathcal{B})$ will keep track of additional auxiliary runs. More formally, in round i when Eve's token in $\mathrm{HD}(\mathcal{B})$ is at state p_i, Eve will have in her memory:

1. A nonempty set of states S_i, together with a particular state q_i in S_i,
2. for every state s in S_i, a history of a play ρ_s of $HD(\mathcal{B}_{>1})$ ending at s played according to $\gamma_{>1}$.

Initially, we set $p_0 = q_{\mathrm{I}}$ and $S_0 = \mathrm{WCR}(\mathcal{B}, p_0)$, with q_0 being any state in S_0. The history for each state in S_0 is ϵ at this stage.

Assume that a round of $\mathrm{HD}(\mathcal{B})$ begins with Eve's token in p_i, her memory set being S_i, a distinguished state $q_i \in S_i$, and finite plays of the HD game on $\mathcal{B}_{>1}$ ending at states in S_i. Suppose Adam plays a_i as his letter in $\mathrm{HD}(\mathcal{B})$. Then, Eve responds by picking the transition $p_i \xrightarrow{a_i} p_{i+1}$ as the transition given by σ_{G1} for the game $G1((\mathcal{B}, p_i); (\mathcal{B}, q_i))$.

We now explain how we update the auxiliary states. For each state $s \in S_i$ with history ρ_s, consider the transition δ_s given by $\gamma_{>1}$ for Eve in the HD game on $\mathcal{B}_{>1}$ in response to Adam's letter a_i

Then, δ_s can either be a transition that leads to the rejecting sink state in $\mathcal{B}_{>1}$, or an *actual transition* $s \xrightarrow{a_i} s'$ that is also a transition in $\mathcal{B}$. We let S_{i+1} be the set of targets of the actual transitions δ_s, for each s in S_i. The histories for each state s' in S_{i+1} is updated by concatenating them with δ_s. In case there are multiple histories ending at s', we pick one arbitrarily: we call such a round a *merge*.

If the transition given by $\gamma_{>1}$ from the designated state q_i is an actual transition $q_i \xrightarrow{a_i} q_{i+1}$, we let the designated state of S_{i+1} be q_{i+1}. Otherwise, if S_{i+1} is non-empty, then we pick as q_{i+1} an arbitrary state from S_{i+1}. We call such a round a *retry*. If S_{i+1} is empty, then we *reset* by letting $S_{i+1} = \mathrm{WCR}(\mathcal{B}, p_{i+1})$. This concludes the description of Eve's strategy γ in the HD game on $\mathcal{B}$.

We will prove that γ is a winning strategy. Take a play of the HD game on $\mathcal{B}$ following γ with $w = a_0a_1 \cdots$ being the word played by Adam. Assume that w is in $\mathrm{L}(\mathcal{B})$ and that $\rho' = (q'_j \xrightarrow{a_j} q'_{j+1})_{j\geq 0}$ is an accepting run of $\mathcal{B}$ on w.

We first show that we cannot have infinitely many resets. Since the run ρ' on Adam's word w is accepting, there is a state q'_i such that $a_ia_{i+1} \cdots \in \mathrm{L}(\mathcal{B}_{>1}, q'_i)$. Then, for every $j \geq i$, the strategy $\gamma_{>1}$ in the HD game on $(\mathcal{B}_{>1}, q'_j)$ on the word $a_ja_{j+1} \cdots$ will construct an accepting run from q'_j. It follows that there can be at most one reset after round i.

Thus, there are only finitely many resets during the play. This implies that we have finitely many *retries*, as the cardinality of S_i strictly decreases with each *retry* without a *reset*. Furthermore, whenever we *merge*, the cardinality of S_i strictly decreases as well, and thus we also have only finitely many merges. Thus, from some moment on, Eve plays $G1$ against a single play that is a run in $\mathrm{HD}(\mathcal{B}_{>1}, \bar{q})$ for some state $\bar{q}$, which is weakly coreachable to Eve's token in that round, with the transitions being played $(q_j \xrightarrow{a_j} q_{j+1})_{j\geq i}$ and $q_i = \bar{q}$. As none of these transitions have priority 1, we know that $a_ia_{i+1} \cdots \in \mathrm{L}(\mathcal{B}_{\mathrm{safe}}, \bar{q})$. Since $\mathcal{B}$ is semantically deterministic (Lemma 11), $a_ia_{i+1} \cdots \in \mathrm{L}(\mathcal{B}, \bar{q})$. Thus, $a_ia_{i+1} \cdots \in \mathrm{L}(\mathcal{B}_{>1}, \bar{q})$ because of fullness of the state q_i, which implies that the run of Eve $(q_j \xrightarrow{a_j} q_{j+1})_{j\geq i}$ constructed by $\gamma_{>1}$ is accepting. This implies that also the run $(p_j \xrightarrow{a_j} p_{j+1})_{j\geq i}$ constructed by σ_{G1} in the HD game of $\mathcal{B}$ by playing $G1$ against $(q_j \xrightarrow{a_j} q_{j+1})_{j\geq i}$ is also accepting. ◀

3.3 Completeness

To show completeness for our even-to-odd witness condition, it suffices to prove the following result. We recall that $k > 2$, as is assumed in this whole section.

▶ **Lemma 20.** *If a $[1, k]$-automaton $\mathcal{A}$ is HD then there exists a $[1, k]$-automaton $\mathcal{B}$ with at most as many states as $\mathcal{A}$ such that $\mathcal{B}$ is normalised and $\mathcal{B}$ is $G1$-equivalent to $\mathcal{A}$.*

We will split our transformation from a HD $[1, k]$-automaton into a normalised automaton into the following four steps. First, we will describe a conversion from HD $[1, k]$-automata to 1-safe HD $[1, k]$-automata where Eve wins $G1$ from everywhere in Section 3.3.1. Second, we will describe a priority-normalisation procedure in Section 3.3.2. Third, we will combine these two procedures to obtain automata that are both priority-normalised and 1-safe HD (Section 3.3.3). Finally, we will describe a conversion from priority-normalised HD $[1, k]$-automata to normalised $[1, k]$-automata in Section 3.3.4, which will conclude the proof of Lemma 20.

3.3.1 Obtaining 1-safe HD automata

Recall that we say that an $[1, k]$-automaton $\mathcal{A}$ is 1-safe HD if for every state q, $(\mathcal{A}_{>1}, q)$ is HD. We begin by describing our conversion to 1-safe HD automata.

▶ **Lemma 21.** *For every HD $[1, k]$-automaton $\mathcal{A}$, there is a $G1$-equivalent HD automaton $\mathcal{B}$ that has at most as many states as $\mathcal{A}$, is 1-safe HD, and on which Eve wins the 1-token game from everywhere.*

Proof. Let $\mathcal{A}$ be a $[1,k]$-automaton that is HD. From Lemma 15, there is a language-equivalent $[1,K]$ HD automaton $\mathcal{A}'$ with at most any states, such that:

1. Eve wins 1-token game from everywhere in $\mathcal{A}'$, and
2. for every state p, there is a state $q \in \mathrm{WCR}(\mathcal{A}', p)$ such that Eve wins $G1((\mathcal{A}'_{>1}, q); (\mathcal{A}'_{>1}, p))$ and $(\mathcal{A}'_{>1}, q)$ is HD.

By, Lemma 6, we note that $\mathcal{A}$ and $\mathcal{A}'$ are $G1$-equivalent. Since $G1$-equivalence is transitive (Lemma 8), we can assume without loss of generality that $\mathcal{A}$ satisfies the two conditions above.

Recall that a state q in $\mathcal{A}$ is 1-*safe HD* if $(\mathcal{A}_{>1}, q)$ is HD. We construct an automaton $\mathcal{B}$ such that its states are the 1-safe HD states of $\mathcal{A}$. The transitions of $\mathcal{B}$ are the transitions of $\mathcal{A}$ between 1-safe HD states, and additionally, we maximally add 1-transitions that do not change the weak-coreachability relation. More precisely, $\mathcal{B}$ consists of the following transitions.

1. $\delta = p \xrightarrow{a:c} q$ is a transition in $\mathcal{B}$ if δ is a transition in $\mathcal{A}$, p and q are 1-safe HD states in $\mathcal{A}$,
2. For every two 1-safe HD states p, q in $\mathcal{A}$, every letter a, and every transition $p \xrightarrow{a} q'$ of $\mathcal{A}$: if q is in $\mathrm{WCR}(\mathcal{A}, p')$ then we have the transition $p \xrightarrow{a:1} q$ in $\mathcal{B}$.

We let the initial state of $\mathcal{B}$ be a 1-safe HD state q such that q is language-equivalent in $\mathcal{A}$ to q_ι, the initial state of $\mathcal{A}$. We know such a-state exists due to Lemma 15.

We will prove that $\mathcal{B}$ is the desired automaton. Towards this, we need to prove the following three statements.

1. Eve wins 1-token game from everywhere in $\mathcal{B}$.
2. $\mathcal{B}$ and $\mathcal{A}$ are $G1$-equivalent.
3. For every state q in $\mathcal{B}$, $(\mathcal{B}_{>1}, q)$ is HD.

For 3, we first note that every transition $\delta = p \xrightarrow{a:c} p'$ used by some winning strategy for Eve in any HD game on $(\mathcal{A}_{>1}, q)$ is included in the automaton $\mathcal{B}$. Indeed, this follows because the considered Eve's winning strategy wins the HD game from $(\mathcal{A}_{>1}, p')$, and thus p' is 1-safe HD.

For such a transition $p \xrightarrow{a:c} p'$, we also have $\mathrm{L}(\mathcal{A}_{>1}, p') = a^{-1}\mathrm{L}(\mathcal{A}_{>1}, p)$: otherwise, if there is a word w in $a^{-1}\mathrm{L}(\mathcal{A}_{>1}, p) \setminus \mathrm{L}(\mathcal{A}_{>1}, p')$, then Eve loses the HD game on $(\mathcal{A}_{>1}, p)$ if Adam plays the word aw, leading to a contradiction to the fact that $(\mathcal{A}_{>1}, p)$ is HD.

Thus, it follows that whenever a state q of $\mathcal{A}$ is 1-safe HD then $\mathrm{L}(\mathcal{A}_{>1}, q) = \mathrm{L}(\mathcal{B}_{>1}, q)$ (the transitions of priority 1 added to $\mathcal{B}$ do not intervene here), and that $(\mathcal{B}_{>1}, q)$ is HD for every state q in $\mathcal{B}_{>1}$.

For 1 and 2, note that by construction, states p and q are weakly coreachable in $\mathcal{B}$ if and only if $(p, q) \in \mathrm{WCR}(\mathcal{A})$ and both p and q are 1-safe HD states in $\mathcal{A}$. In Claim 22, we will prove in that for every two states $(p, q) \in \mathrm{WCR}(\mathcal{B})$, the automata $(\mathcal{A}, p)$ and $(\mathcal{B}, q)$ are $G1$-equivalent. This will prove that $\mathcal{B}$ and $\mathcal{A}$ are $G1$-equivalent. Because $\mathcal{A}$ is HD and Eve wins 1-token game from everywhere on $\mathcal{A}$, it will follow, due to Lemmas 5 and 7 that $\mathcal{B}$ is HD, and due to $G1$-transitivity (Lemma 8), that Eve wins the 1-token game from everywhere on $\mathcal{B}$. Thus, proving Claim 22 below will finish our proof.

$\triangleright$ Claim 22. For every pair of states p and q that are weakly coreachable in $\mathcal{B}$, $(\mathcal{A}, p)$ and $(\mathcal{B}, q)$ are $G1$-equivalent.

Eve wins $G1((\mathcal{A}, p); (\mathcal{B}, q))$. Fix a positional winning strategy σ using which Eve wins the 1-token game from every pair of weakly coreachable states, due to the assumption that Eve wins 1-token game from everywhere in $\mathcal{A}$. Consider the strategy for Eve using which she picks the transitions in $G1((\mathcal{A}, p); (\mathcal{B}, q))$ using σ. In every play following σ, if Adam's run is accepting, then Adam's run in $\mathcal{B}$ contains finitely many 1-transitions, and therefore, some suffix of Adam's run is also a run in $\mathcal{A}$. Since σ is a winning strategy for Eve in 1-token game from everywhere in $\mathcal{A}$, it follows that Eve's run is accepting.

Eve wins $G1((\mathcal{B}, q); (\mathcal{A}, p))$. For states q' in $\mathcal{B}$ and p' in $\mathcal{A}$ that are weakly coreachable in $\mathcal{A}$, we say that the position $((\mathcal{B}, q'); (\mathcal{A}, p'))$ for Eve's and Adam's tokens in this 1-token game is *desirable* if

$$\mathrm{L}(\mathcal{A}_{>1}, p') \subseteq \mathrm{L}(\mathcal{B}_{>1}, q').$$

Note that because of our assumption due to Lemma 15 at the start of our proof and construction of $\mathcal{B}$, we know that such a q' exists for every p'.

We describe a winning strategy for Eve in $G1((\mathcal{B}, q); (\mathcal{A}, p))$ as follows. Whenever the position in the 1-token game is desirable, Eve picks transitions on her token in $\mathcal{B}$ according to a winning strategy $\sigma_{>1}$ in the HD game on $\mathcal{B}_{>1}$, unless the transition in $\mathcal{B}_{>1}$ leads to the rejecting sink state, in which case she picks an arbitrary transition.

Otherwise, if the position of the 1-token game $((\mathcal{B}, q'); (\mathcal{A}, p'))$ is not desirable, then Eve picks a transition as follows. We know there is a state r in $\mathcal{B}$ such that $\mathrm{L}(\mathcal{B}_{>1}, r) \supseteq \mathrm{L}(\mathcal{A}_{>1}, p')$. When Adam picks a letter a, let $r \xrightarrow{a} r'$ be the transition given by her strategy $\sigma_{>1}$ in HD game on $(\mathcal{B}_{>1}, r)$. If r' is not the rejecting sink state, then Eve picks the 1-transition to r' on her token, and otherwise picks an arbitrary transition.

We claim that the above described strategy is winning for Eve. Indeed, if the run on Adam's token is accepting, then eventually, Adam's run does not contain a 1-transition. Suppose this happens after round i, when Adam's token is at p_i (in $\mathcal{A}$) and Eve's token is at q_i (in $\mathcal{B}$). In round $(i+1)$, when Adam selects a letter a_{i+1}, Eve's strategy above guarantees that her token would be at q_{i+1} and would be such that

$$\mathrm{L}(\mathcal{B}_{>1}, q_{i+1}) \supseteq (a_{i+1})^{-1}\mathrm{L}(\mathcal{A}_{>1}, p_i)$$

We know that Adam's word $w' = a_i a_{i+1} \dots$ from round i is in $\mathrm{L}(\mathcal{A}_{>1}, p_i)$. Since Eve's token is following a winning strategy in the HD game from $(\mathcal{B}_{>1}, q_{i+1})$ henceforth, Eve's run on her token is accepting, as desired. This proves Claim 22, and hence Lemma 21. ◀

We make the following key observation in our proof above.

▶ **Observation 23.** *Every HD $[1, k]$-automaton $\mathcal{A}$ can be converted to a $G1$-equivalent HD automaton $\mathcal{B}$ that is 1-safe HD and where Eve wins the 1-token game from everywhere with at most as many states. Additionally, this conversion can be achieved by the following three kinds of operations in sequence.*

1. *Deletion of certain states, and thus of transitions involving those states.*
2. *Deletion of certain transitions that have priority at least 2.*
3. *Addition of some priority 1 transitions.*

3.3.2 Priority normalisation

We now present a priority-normalisation procedure, which relabels the priorities of $[1, k]$-automata without introducing new accepting or new rejecting runs. The goal of this procedure is to obtain an automaton in which every maximally strongly connected component without a 1-transtion contains at least one 2-transition.

▶ **Definition 24** (Priority-normalised automata)**.** *For a $[1, k]$-automaton $\mathcal{B}$, consider the directed graph $G_{>1}(\mathcal{B})$ whose vertices consist of the states of $\mathcal{B}$ and its edges are the transitions of $\mathcal{B}$ that have priority at least 2. We say that a $[1, k]$-automaton $\mathcal{B}$ is* priority-normalised *if the following two conditions hold.*

1. *Every transition of $\mathcal{B}$ which has priority at least 2 is in a strongly connected component (SCC) of $G_{>1}(\mathcal{B})$.*
2. *Each SCC of $G_{>1}(\mathcal{B})$ contains at least one transition of priority 2.*

We next describe a procedure to priority-normalise a given $[1, k]$-automaton. This is done by a relabelling of priorities via the iterative procedure presented below.

Priority-normalisation procedure

Let $\mathcal{A}$ be a $[1, k]$-automaton, and let $\mathcal{A}_0 = \mathcal{A}$. We perform the following two steps for each $i \in \{0, 1, 2, \dots\}$ until $\mathcal{A}_{i+1} = \mathcal{A}_i$.

1. For every transition in $\mathcal{A}_i$ that does not appear in a strongly connected component of $G_{>1}(\mathcal{A}_i)$ we change its priority to 1, to obtain $\mathcal{A}'_i$.
2. For every SCC in $G_{>1}(\mathcal{A}'_i)$ in which the priorities of all transitions from $\mathcal{A}'_i$ are strictly greater than 2, we decrease the priority of every transition of $\mathcal{A}'_i$ which is in that SCC by 2 to obtain $\mathcal{A}_{i+1}$.

The above procedure eventually terminates, since in each step we only decrease the priorities of certain transitions. When the above procedure terminates, it is clear that the resulting automaton is priority-normalised.

Furthermore, each iteration of the priority-normalisation procedure does not introduce any new accepting or rejecting runs, since it does not change the parity of the lowest priority occurring in any cycle. We refer the reader to [26, Lemma 6.3] for more details. The following result follows.

▶ **Lemma 25** ([24, Lemma 12],[26, Lemma 6.3]). *Every $[1, k]$-automaton $\mathcal{A}$ can be converted to a priority-normalised $[1, k]$-automaton $\mathcal{B}$ by relabelling the priorities of $\mathcal{A}$ such that for every run ρ, ρ is accepting in $\mathcal{A}$ if and only if it is accepting in $\mathcal{B}$.*

We make the following observation about the priority-normalisation procedure.

▶ **Observation 26.** *The priority-normalisation procedure only decreases priorities of transitions.*

3.3.3 Priority-normalised and 1-safe HD automata

Our ultimate goal is to construct automata that are normalised. We do so by applying three procedures: the above described procedure of obtaining a *priority-normalised* automaton; the procedure of making an automaton *1-safe HD* presented in subsection 3.3.1; and the procedure of obtaining a *normalised* automaton as explained in Section 3.3.4.

Unfortunately, our procedure of converting HD $[1, k]$-automata to 1-safe HD $[1, k]$-automata in Lemma 21 need not preserve priority-normality of an automaton, and similarly, our priority-normalisation procedure (see Section 3.3.2) need not preserve the property of being 1-safe HD. Thus, we can't apply this procedures one-after-the-other to obtain an automaton that is both priority-normalised and 1-safe HD. The priority-normalisation procedure does preserve the property of history determinism and winning 1-token game from everywhere, however, since it preserves the acceptance of runs.

To obtain an automaton that is priority-normalised, 1-safe HD, and on which Eve wins the 1-token game from everywhere, we will iteratively apply our procedure to convert HD $[1, k]$-automata to 1-safe HD automata (that remain HD) and then our priority-normalisation procedure in alternation, until the automaton stabilises on both of these procedures. Once this is done, we can turn the resulting automaton (which is both priority-normalised and 1-safe HD) into a normalised automaton.

This iterative procedure will terminate, because from Observations 23 and 26, we only delete states, so the states of the automaton should eventually remain the same. Furthermore, once the states have stabilised, we note that we either delete transitions of priority at least 2, or increase the number of transitions with priority 1 in both of these procedures. Thus, eventually, the set of transitions that have priority 1 and the set of transitions that have priority at least 2 also stabilises. Since the priority-normalisation procedure then only decreases the priorities of transitions, it follows that the overall iterative alternating procedure has to stabilise. We thus conclude the following result from Lemmas 21 and 25.

▶ **Lemma 27.** *For every HD $[1,k]$-automaton $\mathcal{A}$, there is a $G1$-equivalent HD $[1,k]$-automaton $\mathcal{B}$ that has at most as many states and that is 1-safe HD, priority-normalised, and on which Eve wins the 1-token game from everywhere.*

3.3.4 Normalised automata

We proceed towards proving that every HD $[1,k]$-automaton has a $G1$-equivalent normalised $[1,k]$-automaton with at most as many states (Lemma 19). Towards this, for an $[1,k]$-automaton $\mathcal{A}$ we define the following relation $\star(p,q)$ on pairs of states:

$$\star(p,q):\ (p,q)\in \mathrm{WCR}(\mathcal{A}) \text{ and } \mathrm{L}(\mathcal{A}_{\mathrm{safe}},p)\cap \mathrm{L}(\mathcal{A},p)\subseteq \mathrm{L}(\mathcal{A}_{>1},q)$$

We will first prove that for every state p in every 1-safe HD and priority-normalised HD $[1,k]$-automaton $\mathcal{A}$, there is a state q weakly coreachable to p in $\mathcal{A}$ such that $\star(p,q)$ is true. The relation $(\star)$ is transitive, and so the finiteness of states of $\mathcal{A}$ implies that there are states q for which $\mathrm{L}(\mathcal{A}_{\mathrm{safe}},q)\cap \mathrm{L}(\mathcal{A},q)=\mathrm{L}(\mathcal{A}_{>1},q)$, i.e., that are full. Restricting the automaton $\mathcal{A}$ to such states q and adding appropriate 1-transitions would then give us the desired normalised automaton.

▶ **Lemma 28.** *Let $\mathcal{A}$ be a HD $[1,k]$-automaton that is 1-safe HD and priority-normalised. For every state p, there is a state q such that $\star(p,q)$ is true.*

Proof. We will prove that in a priority-normalised and 1-safe HD automaton as above, if there is a state p such that $\star(p,q)$ is not true for every state q, then Adam wins $HD(\mathcal{A},p)$.

We do so by describing a winning strategy for Adam in $HD(\mathcal{A},p)$, that forces Eve to either visit states that are weakly-coreachable to p infinitely often, or to build a run that is rejecting while Adam's word is accepting. Suppose that after i rounds of play, Eve's run is at state q in $\mathrm{WCR}(\mathcal{A},p)$. We will build an auxiliary run $\rho_\forall$ for Adam in $\mathcal{A}$ that starts at p, contains no transition of priority 1, and is currently at p after having read the word $u=a_0a_1\dots a_i$ formed by Adam's letters in the HD game. Unlike our previous proofs, however, the auxiliary run $\rho_\forall$ won't be built in paralled to Eve's run.

Since $\star(p,q)$ is not true, we know that there is a word $w'\in \mathrm{L}(\mathcal{A}_{\mathrm{safe}},p)\cap\mathrm{L}(\mathcal{A},p)\setminus\mathrm{L}(\mathcal{A}_{>1},q)$. We let Adam play consecutive letters of w' as his letters in the HD game on $\mathcal{A}$, until Eve's token picks a transition of priority 1. If Eve's token never picks a transition of priority 1, then note that her run from q on w' is also a run in $(\mathcal{A}_{>1},q)$. Since $w'\notin(\mathcal{A}_{>1},q)$, her run is rejecting, whereas Adam's word is accepting, and thus Eve loses the HD game.

Therefore, suppose Eve picks a transition of priority 1, after reading a finite prefix u of w'. Then, since $w'\in \mathrm{L}(\mathcal{A}_{\mathrm{safe}},p)$, we can extend Adam's auxiliary run $\rho_\forall$ from p to some state p' on uv, without $\rho_\forall$ containing any 1-transition. Since $\mathcal{A}$ is priority-normalised, there is a word v' on which there is a run from p' to p that contains a-transition of priority 2 but no transition of priority 1. We extend Adam's auxiliary run by this run, and we let Adam play the letters of v' in sequence. After having read uvv', Eve's token will be in some state q' that is weakly-coreachable to p, and Adam can inductively repeat this strategy.

If Eve's run contains finitely many 1-transitions, then Adam's strategy produces a word in the HD game that is accepting while her run is rejecting. Otherwise, if Eve's run contains infinitely many 1-transitions, then we argue inductively that Adam's auxiliary run contains infinitely many 2-transitions and no 1-transition. Thus, Adam's word is accepting with Adam's auxiliary run as a witness run, while Eve's run is rejecting. In both cases, Eve loses the HD game, as we wanted to prove. ◀

We observe that the relation $\star$ is transitive.

▶ **Lemma 29.** *If p, q, r are three states in a $[1, k]$-automaton $\mathcal{A}$ such that $\star(p, q)$ and $\star(q, r)$, then $\star(p, r)$.*

Proof. Observe that for every state s, $\mathrm{L}(\mathcal{A}_{>1}, s) \subseteq \mathrm{L}(\mathcal{A}_{\text{safe}}, s) \cap \mathrm{L}(\mathcal{A}, s)$, due to Fact 18. Thus, we have the following chain of language-inclusions, which gives us the desired result.

$$\begin{aligned} \mathrm{L}(\mathcal{A}_{\text{safe}}, p) \cap \mathrm{L}(\mathcal{A}, p) &\subseteq \mathrm{L}(\mathcal{A}_{>1}, q) \\ &\subseteq \mathrm{L}(\mathcal{A}_{\text{safe}}, q) \cap \mathrm{L}(\mathcal{A}, q) \\ &\subseteq \mathrm{L}(\mathcal{A}_{>1}, r) \end{aligned}$$ ◀

The following result thus follows from Lemmas 28 and 29 and the fact that our automata have finitely many states.

▶ **Lemma 30.** *For every priority-normalised HD $[1, k]$-automaton $\mathcal{A}$ that is 1-safe HD and on which Eve wins the 1-token game from everywhere, for every state p in $\mathcal{A}$, there is a state q such that $\star(p, q)$ and $\star(q, q)$.*

We now have the tools to prove Lemma 20, the result that every HD $[1, K]$ automaton has a $G1$-equivalent normalised automaton that has at most as many states.

Proof of Lemma 20. Due to Lemma 27, we will assume that $\mathcal{A}$ is 1-safe HD, priority-normalised, and Eve wins the 1-token game from everywhere on $\mathcal{A}$. Recall that we call a state q in $\mathcal{A}$ *full* if $\star(q, q)$, i.e., if $\mathrm{L}(\mathcal{A}_{>1}, q) \supseteq \mathrm{L}(\mathcal{A}, q) \cap \mathrm{L}(\mathcal{A}_{\text{safe}}, q)$, or equivalently, due to Fact 18, that $\mathrm{L}(\mathcal{A}_{>1}, q) = \mathrm{L}(\mathcal{A}, q) \cap \mathrm{L}(\mathcal{A}_{\text{safe}}, q)$.

We construct an automaton $\mathcal{B}$ whose states are the full states of $\mathcal{A}$, and whose transitions are as follows.

1. $\delta = p \xrightarrow{a:c} q$ is a transition in $\mathcal{B}$ if δ is a transition in $\mathcal{A}$, and p and q are full states in $\mathcal{A}$.
2. For every two full states p, q in $\mathcal{A}$, every letter a and every transition $p \xrightarrow{a} p'$ of $\mathcal{A}$: if $q \in \mathrm{WCR}(\mathcal{A}, p')$, we have the transition $p \xrightarrow{a:1} q$ in $\mathcal{B}$.

We claim that $\mathcal{B}$ is normalised. To show this, we need to show the following three statements.

1. $\mathcal{B}$ is 1-safe HD (i.e., for every state $(\mathcal{B}_{>1}, p)$ is HD),
2. every state in $\mathcal{B}$ is full,
3. Eve wins the 1-token game from everywhere on $\mathcal{B}$.

To show 1, we will first show that if p is a full state in $\mathcal{A}$ and $p \xrightarrow{a:c} p'$ is a transition given by a winning strategy for Eve in the HD game on $(\mathcal{A}_{>1}, p)$, then p' is also full. This follows from the following chain of language-inclusions.

$$\begin{aligned} \mathrm{L}(\mathcal{A}_{>1}, p') &= a^{-1}\mathrm{L}(\mathcal{A}_{>1}, p) \\ &\supseteq a^{-1}\mathrm{L}(\mathcal{A}, p) \cap a^{-1}\mathrm{L}(\mathcal{A}_{\text{safe}}, p) \\ &\supseteq \mathrm{L}(\mathcal{A}, p') \cap \mathrm{L}(\mathcal{A}_{\text{safe}}, p'). \end{aligned}$$

Thus, it follows that $\mathrm{L}(\mathcal{A}_{>1}, p) = \mathrm{L}(\mathcal{B}_{>1}, p)$ and that $(\mathcal{B}_{>1}, p)$ is HD, since the transitions given by the HD strategy in $\mathcal{A}_{>1}$ are all present in $\mathcal{B}$.

We will show the following key claim to prove the rest.

▷ Claim 31. For every pair of states p and q that are weakly coreachable in $\mathcal{B}$, $(\mathcal{A}, p)$ and $(\mathcal{B}, q)$ are $G1$-equivalent.

Note that by construction, p and q are weakly coreachable states in $\mathcal{B}$ if and only if they are weakly coreachable 1-safe HD states in $\mathcal{A}$. Thus, following Claim 31 and the transitivity of $G1$ (Lemma 8), we obtain that Eve wins the 1-token game from everywhere in $\mathcal{B}$, proving 2.

For 3, note that from Lemma 5, $\mathrm{L}(\mathcal{A}, p) = \mathrm{L}(\mathcal{B}, p)$ for all states p in $\mathcal{B}$. Furthermore, note that $\mathcal{B}_{\text{safe}}$ is a subautomaton of $\mathcal{A}_{\text{safe}}$, since $\mathcal{B}$ is obtained by removing states of $\mathcal{A}$ and adding some 1-transitions. Fullness for $\mathcal{B}$ thus follows from the following chain of inequalities.

$$\begin{aligned}\mathrm{L}(\mathcal{B}_{>1}, p) &= \mathrm{L}(\mathcal{A}_{>1}, p)\\ &= \mathrm{L}(\mathcal{A}, p) \cap \mathrm{L}(\mathcal{A}_{\text{safe}}, p)\\ &\supseteq \mathrm{L}(\mathcal{B}, p) \cap \mathrm{L}(\mathcal{B}_{>1}, p)\end{aligned}$$

Proving Claim 31 will therefore finish our proof of Lemma 20. Claim 31 is proved similarly to the proof of Claim 22.

Proof of Claim 31.

Eve wins $G1((\mathcal{A}, p); (\mathcal{B}, q))$. Fix a positional winning strategy σ using which Eve wins the 1-token game from every pair of weakly coreachable states in $\mathcal{A}$. Consider the strategy for Eve using which she pick the transitions in $G1((\mathcal{A}, p); (\mathcal{B}, q))$ using σ. In every play following σ, if Adam's run is accepting, then Adam's run in $\mathcal{B}$ contains finitely many 1-transitions, and therefore, some suffix of Adam's run is also a run in $\mathcal{A}$. Since σ is a winning strategy for Eve in 1-token game from everywhere in $\mathcal{A}$, it follows that Eve's run is accepting.

Eve wins $G1((\mathcal{B}, q); (\mathcal{A}, p))$. For states q' in $\mathcal{B}$ and p' in $\mathcal{A}$ that are weakly coreachable in $\mathcal{A}$, we say that the position $((\mathcal{B}, q'); (\mathcal{A}, p'))$ for Eve's and Adam's tokens in this 1-token game is *desirable* if

$$\mathrm{L}(\mathcal{A}_{>1}, p') \subseteq \mathrm{L}(\mathcal{B}_{>1}, q').$$

Note that because of Lemma 30, we know that such a q' exists for every p'.

We describe a winning strategy for Eve in $G1((\mathcal{B}, q); (\mathcal{A}, p))$ as follows. Whenever the position in the 1-token game is desirable, Eve picks transitions on her token in $\mathcal{B}$ according to a winning strategy $\sigma_{>1}$ in the HD game on $\mathcal{B}_{>1}$, unless the transition in $\mathcal{B}_{>1}$ leads to the rejecting sink state, in which case she picks an arbitrary transition.

Otherwise, if the position of the 1-token game $((\mathcal{B}, q'); (\mathcal{A}, p'))$ is not desirable, then Eve picks a transition as follows. We know there is a state r in $\mathcal{B}$ such that $\mathrm{L}(\mathcal{B}_{>1}, r) \supseteq \mathrm{L}(\mathcal{A}_{>1}, p')$. When Adam picks the letter a, let $r \xrightarrow{a} r'$ be the transition given by her strategy $\sigma_{>1}$ in HD game on $(\mathcal{B}_{>1}, r)$. If r' is not the rejecting sink state, then Eve picks the 1-transition to r' on her token, and otherwise picks an arbitrary transition.

We claim that the above described strategy is winning for Eve. Indeed, if the run on Adam's token is accepting, then eventually, Adam's run does not contain a 1. Suppose this happens after round i, when Adam's token is at p_i (in $\mathcal{A}$) and Eve's token is at q_i (in $\mathcal{B}$). In round $(i+1)$, when Adam selects a letter a_{i+1}, Eve's strategy above guarantees that her token would be at q_{i+1} and the new position would be desirable, i.e.,

$$\mathrm{L}(\mathcal{B}_{>1}, q_{i+1}) \supseteq (a_{i+1})^{-1}\mathrm{L}(\mathcal{A}_{>1}, p_i)$$

We know that Adam's word $w' = a_i a_{i+1} \dots$ from round i is in $\mathrm{L}(\mathcal{A}_{>1}, p_i)$. Since Eve's token is following a winning strategy in the HD game from $(\mathcal{B}_{>1}, q_{i+1})$ henceforth, Eve's run on her token is accepting, as desired. This proves Claim 31, and hence Lemma 20. ◁

◀

4 Odd to Even

In this section, we will present how we can characterise the history determinism of $[0,k]$-automata in terms of the history determinism of $[1,k]$-automata, for every $k > 1$. Similar to the witness condition from Section 3.1 for $[1,k]$-automata, this will involve $G1$-equivalent automata that have some structural properties. These properties will enable us to verify history determinism for $[0,k]$-automata by verifying some strategies for Eve in 1-token games and by checking history determinism for $[1,k]$-automata. Our witness condition for this odd-to-even lifting step comes from Lemma 14, which we present next.

4.1 Odd-to-even witness

We claim that a given $[0,k]$-automaton $\mathcal{A}$ is HD if and only if there exists a witness $[0,k]$-automaton $\mathcal{B}$ with at most as many states as $\mathcal{A}$, such that it satisfies to following *odd-to-even witness conditions*:

- $\mathcal{A}$ and $\mathcal{B}$ are $G1$-equivalent,
- Eve wins 1-token game from everywhere in $\mathcal{B}$
- for every state p of $\mathcal{B}$ there exists a state q of $\mathcal{B}$ such that:
  $(\star_0)(p,q)$: $(p,q) \in \mathrm{WCR}(\mathcal{B})$, Eve wins $G1(p;q)$ in $\mathcal{B}_{>0}$, and $(\mathcal{B}_{>0}, q)$ is HD.

We show the completeness and soundness for the odd-to-even witness conditions next. Completeness follows rather easily from Lemma 14.

4.2 Completeness

▶ **Lemma 32.** *If a $[0,k]$-automaton $\mathcal{A}$ is HD then there exists an $[0,k]$-automaton $\mathcal{B}$ with at most as many states as $\mathcal{A}$ that satisfies the odd-to-even witness condition.*

We obtain $\mathcal{B}$ from Lemma 14, which ensures that $\mathcal{B}$ is language equivalent to $\mathcal{A}$, no larger than $\mathcal{A}$, HD, that Eve wins $G1$ from everywhere and that for every state p there is a state $q \in WCR(\mathcal{B}, p)$ such that Eve wins $G1(\mathcal{B}_{>0}, p, q)$ and $\mathrm{HD}(\mathcal{B}_{>0}, q)$.

Since both automata $\mathcal{A}$ and $\mathcal{B}$ are HD and share the same language, they are $G1$-equivalent: Eve can use the HD strategy to move her token in both $G1$-games.

4.3 Soundness

▶ **Lemma 33.** *If a $[0,k]$-automaton $\mathcal{A}$ admits an automaton $\mathcal{B}$ satisfying the odd-to-even witness conditions, then $\mathcal{A}$ is HD.*

The $G1$-equivalence of $\mathcal{A}$ and $\mathcal{B}$ implies that $\mathcal{A}$ and $\mathcal{B}$ are language-equivalent (Lemma 5). Thus, since Eve wins $G1(\mathcal{A}; \mathcal{B})$, it is enough to show that $\mathcal{B}$ is HD to conclude that $\mathcal{A}$ is HD (Lemma 7). It remains to describe a winning strategy for Eve in the game $\mathrm{HD}(\mathcal{B})$.

Intuition: To play from a state p of $\mathcal{B}$, Eve keeps track of an auxiliary token at a state q of $\mathcal{B}$ that satisfies $\star_0(p,q)$. The auxiliary token follows a winning strategy in $\mathrm{HD}(\mathcal{B}_{>0}, q)$ and Eve plays a winning strategy on her token from p against q using $G1(\mathcal{B}_{>0}, p; \mathcal{B}_{>0}, q)$. She does this until her token takes a 0-transition, at which point she resets the auxiliary token in order to maintain the $\star_0$ invariant. We will show that she either takes infinitely many 0-transitions, or builds an otherwise accepting run against every accepting word of Adam.

More formally, fix a positional strategy σ_{G1} for Eve in $G1(\mathcal{B}_{>0})$ using which she wins from configuration $(p;q)$ that satisfy $\star_0(p,q)$. We also fix a strategy $\tau_{>0}$ for Eve to win the HD game on $(\mathcal{B}_{>0})$ from states q such that $(\mathcal{B}_{>0},q)$ are HD.

We will call $(p_i)_{i\geq 0}$ the consecutive configurations of a play of HD($\mathcal{B}$), with p_0 as the initial state q_{I}. Eve, while playing the HD game on $\mathcal{B}$ from a state p_i, will keep in her memory, an auxiliary state q_i such that either $\star_0(p_i,q_i)$ holds, or q_i is the accepting sink state $q_\top$ of $\mathcal{B}_{>0}$ and Eve wins $G1(p_i;q_\top)$ in $\mathcal{B}_{>0}$. Additionally, Eve will keep track of a play of the HD game on her auxiliary state on $\mathcal{B}_{>0}$ following $\tau_{>0}$ beginning at the state after last *reset* and played against Adam's letters. We let q_0 be some state such that $\star_0(p_0,q_0)$ holds, and the first *reset* to be at round 0 of the HD game on $\mathcal{B}$.

In round i of the HD game on $\mathcal{B}$, when Eve's token is at p_i and the auxiliary state is at q_i, suppose Adam selects a letter a_i. Let $\delta = p_i \xrightarrow{a_i} p'$ be the transition given by σ_{G1} from $(p_i;q_i)$ in the game $G1(\mathcal{B}{>}\ 0)$.

If δ leads to the accepting sink state in $\mathcal{B}_{>0}$, then there is a transition $\delta' = p_i \xrightarrow{a_i:0} p_{i+1}$ in $\mathcal{B}$, which Eve picks in the HD game on $\mathcal{B}$. Eve then *resets* her auxiliary state to be some q_{i+1} such that $\star_0(p_{i+1};q_{i+1})$ holds.

Otherwise, δ is a transition of priority at least 1, and we let Eve pick this transition for her token in the HD game on $\mathcal{B}$. She updates the auxiliary state to be the target of the transition $\delta_{>0}$ given by her strategy $\tau_{>0}$ in the HD game on $\mathcal{B}_{>0}$ (since the last reset). In her memory, she also updates the play of the HD game on $\mathcal{B}_{>0}$ since the last reset by concatenating it with $\delta_{>0}$.

This concludes the description of Eve's strategy in the HD game on $\mathcal{B}$, which we will prove to be winning.

Consider an infinite play of the HD game on $\mathcal{B}$ according to the above strategy. If there are infinitely many resets, then Eve's token takes infinitely many transitions of priority 0, and her run is accepting. Otherwise, suppose the last reset occurred at round i, when Eve's token is at p_i and her auxiliary state is q_i, and suppose that Adam has the word u so far. Then, if Adam's word uv in the HD game on $\mathcal{B}$ is in $L(\mathcal{B})$, then we must have $v \in L(\mathcal{B},q_i)$, since $\mathcal{B}$ is semantically-deterministic (due to Lemma 11). Recall that $L(\mathcal{B},q_i) \subseteq L(\mathcal{B}_{>0},q_i)$ (Observation 12), and thus $v \in L(\mathcal{B}_{>0},q_i)$. Therefore, $\tau_{>0}$ builds an accepting run from q_i in $\mathcal{B}_{>0}$. Since Eve is playing according to a winning strategy in $G1(\mathcal{B}_{>0})$ on her token against her auxiliary run, her run is accepting too, as desired. This concludes our proof of Lemma 33.

▶ Remark 34. Note that our assumptions above do not exclude the scenario that after round i for some i, the auxiliary run in Eve's memory reaches an accepting sink state, while her run in $\mathcal{B}$ does not contain a transition of priority 0 (no reset); in this case, her run must be simply accepting due to the standard parity condition.

5 Summing up

We are now in a position to prove Theorem 1: we show an NP procedure to check history determinism for parity automata. Towards this, we will use the two witness conditions we presented in Sections 3.1 and 4.1 to build an overall polynomial-sized witness to verify history determinism for parity automata in polynomial time.

Intuitively, our two witness conditions allow us, in order to verify the history determinism of a parity automaton, to verify the history determinism of another witness automaton that has fewer priorities, and the other conditions of the witnesses. The latter is done by encoding the other conditions as 1-token games (where Eve wins if the condition is true), for which we will include postional strategies for Eve in the witness (see Lemma 9).

Before describing our witness in more detail, we need to present one more reduction. Note that in each of our odd-to-even or even-to-odd witness conditions, we need to check history determinism of an automaton that has fewer priorities with *multiple* initial states.

Multiple-state HD problem. Given a parity automaton $\mathcal{A}$ and a subset S of states of $\mathcal{A}$, is $(\mathcal{A}, s)$ HD for every state s in S?

For $\mathcal{A}$ and S as above, we use $\mathrm{Mult}-\mathrm{HD}(\mathcal{A}, S)$ to denote that $(\mathcal{A}, s)$ is history deterministic for all states s in S. Multiple-state HD problem is easily reduced to the problem of checking history determinism for a single parity automaton.

▶ **Lemma 35.** *There is a LOGSPACE algorithm that reduces instances of multiple-state HD problem to the problem of deciding history determinism of a parity automaton with one additional state and the same parity-index.*

Proof. Let $(\mathcal{A}, S)$ be an instance of a multiple-state HD problem. We construct $\mathcal{A}'$ by adding a new initial state s_ι to $\mathcal{A}$, adding a new letter σ_s for each state s in S, and adding transitions $s_\iota \xrightarrow{\sigma_s:i} s$, where i is the least priority of $\mathcal{A}$.

It is easy to see that $\mathcal{A}'$ is HD if and only if $\mathrm{Mult}-\mathrm{HD}(\mathcal{A}, S)$ holds. If $\mathcal{A}'$ is HD, then Eve's winning strategy in the HD game on $\mathcal{A}$ can be used to play the HD game from $(\mathcal{A}, s)$ easily, since Eve can play as if the first letter of Adam of HD game on $\mathcal{A}'$ was σ_s. Conversely, if $\mathrm{Mult}-\mathrm{HD}(\mathcal{A}, S)$ holds, then Eve has the following strategy to win the HD game on $\mathcal{A}'$: upon letter σ_s from s_ι, she takes the only available transition to s, from where she plays according to a winning strategy in the HD game on $(\mathcal{A}, s)$. ◀

We are now ready to formally describe the NP-procedure for checking history determinism of parity automata. For convenience (of indexing), we will describe a polynomial witness for a $[0, k]$ parity automaton $\mathcal{A}$ where $k > 1$, such that the polynomial witness can be used to verify history determinism of $\mathcal{A}$ in polynomial time.

Our witness consists of a sequence of automata $\mathcal{A} = \mathcal{A}^0, \mathcal{B}^0, \mathcal{A}^1, \mathcal{B}^1, \ldots, \mathcal{A}^{k-1}$, where each $\mathcal{A}^i$ and $\mathcal{B}^i$ are $[i, k]$ automata, when defined. Additionally, for each index i in $[0, k-1]$, we will describe a polynomial-sized witness $\mathcal{W}^i$.

Broadly, for each $i < k-1$, the automata $\mathcal{B}^i$ is the witness automaton for $\mathcal{A}^i$ from Section 3.1 or Section 4.1, depending on whether i is odd or even, respectively. The witness $\mathcal{W}^i$ consists of additional information that helps verify the history determinism of $\mathcal{A}^i$ and $\mathcal{B}^i$, provided that $\mathcal{A}^{i+1}$ is HD. The history determinism of $\mathcal{A}^{k-1}$ can be checked in polynomial time, since it is a $[k-1, k]$-automaton, or equivalently a coBüchi or Büchi automaton, and thus its history determinism can be verified in polynomial time, see [6, Section 6] and [4, Theorem 22] respectively.

To formally describe the witness, for each $i < k-1$, given $\mathcal{A}^i$, we will describe the automata $\mathcal{B}^i, \mathcal{A}^{i+1}$, and the witness $\mathcal{W}^i$. Naturally, we split this into two cases, based on parity of i.

If i is even. If i is even, the automata $\mathcal{A}^i$ is an $[i, k]$ automaton, or, equivalently, an $[0, k-i]$ automaton. Thus, from Section 4.1, $\mathcal{A}^i$ is HD if and only if there is a $[0, k-i]$ automaton $\mathcal{B}$ of the same size that satisfies the following three conditions:

1. $\mathcal{B}$ and $\mathcal{A}^i$ are $G1$-equivalent
2. Eve wins 1-token game from everywhere in $\mathcal{B}$
3. For every state p in $\mathcal{B}$, there is a state $q \in \mathrm{WCR}(\mathcal{B}, p)$ such that Eve wins $G1(p; q)$ in $\mathcal{B}_{>0}$ and $(\mathcal{B}_{>0}, q)$ is HD.

Thus, the witness automaton $\mathcal{B}^i$ has the role of $\mathcal{B}$ as above. The witness $\mathcal{W}^i$ consists of the following components.

1. Positional strategies for Eve in $G1(\mathcal{B}^i; \mathcal{A}^i)$ and $G1(\mathcal{A}^i; \mathcal{B}^i)$: we use this to verify $G1$ equivalence of $\mathcal{A}^i$ and $\mathcal{B}^i$.
2. A positional strategy for Eve in $G1(\mathcal{B}^i)$ from weakly coreachable states: we use this to verify that Eve wins the 1-token game from everywhere in $\mathcal{B}$.
3. A function f_i from states of $\mathcal{B}^i$ to states of $\mathcal{B}^i$, such that f maps each state to a weakly coreachable state, and a positional strategy for Eve in $G1(\mathcal{B}^i_{>0})$: we use f_i and this positional strategy to verify that Eve wins $G1(p; f(p))$ in $\mathcal{B}^i_{>0}$ for every state p in $\mathcal{B}^i$.

We also need to verify the history determinism of states that are in the image $Im(f)$ of f in $\mathcal{B}^i_{>0}$. This is where our witness automaton $\mathcal{A}^{i+1}$ in the overall certificate comes in the picture: the automaton $\mathcal{A}^{i+1}$ is obtained by the reduction from Lemma 35 to reduce the corresponding multiple-state HD problem to the problem of checking history determinism of $\mathcal{A}^{i+1}$. We note that $\mathcal{A}^{i+1}$ is an $[i+1, k]$ automaton and has at most one more state than $\mathcal{A}^i$.

In our witness $\mathcal{W}^i$, we additionally include an injective function g_i from the states of $\mathcal{B}^i_{>0}$ to states of $\mathcal{A}^{i+1}$: g_i is used to verify, in polynomial time, that $\mathcal{A}^{i+1}$ is indeed obtained by reducing the multiple-state HD problem of $\mathcal{B}^i_{>0}$ from $f(I)$ to its history determinism, as in Lemma 35.

If $\mathcal{A}^i$ is HD, it is clear that $\mathcal{W}^i$, $\mathcal{B}^i$, and $\mathcal{A}^{i+1}$ as above exist due to the completeness (Lemma 32) of our odd-to-even witness criterion (Section 4.1). Conversely, it is clear that $\mathcal{W}^i$, $\mathcal{B}^i$, and $\mathcal{B}^{i+1}$ can be used to verify history determinism of $\mathcal{A}^i$ (due to Lemma 33).

If i is odd. If i is odd, then $\mathcal{A}^i$ is an $[i, k]$ automaton, or equivalently, a $[1, k-i+1]$ automaton. Thus, from Section 3.1, $\mathcal{A}^i$ is HD if and only if there is an $[1, k-i+1]$ automaton $\mathcal{B}$ with at most as many states that satisfies the following four conditions.

1. $\mathcal{A}^i$ and $\mathcal{B}$ are $G1$-equivalent.
2. Eve wins $G1$ from everywhere on $\mathcal{B}$.
3. $\mathcal{B}$ is 1-safe HD, i.e., $(\mathcal{B}_{>1}, q)$ is HD for all states q of $\mathcal{B}$.
4. $\mathcal{B}$ has fullness, i.e., for all states q, $L(\mathcal{B}_{>1}, q) = L(\mathcal{B}_{\text{safe}}, q) \cap L(\mathcal{B}, q)$.

The witness automaton $\mathcal{B}^i$ in our overall certificate will have the role of $\mathcal{B}$ above. The witness $\mathcal{W}^i$ will help us verify the conditions 1,2, and 4 above, while $\mathcal{A}^{i+1}$ will be used to verify 3. More concretely, $\mathcal{W}^i$ consists of the following components, where the first two are the same as for when i is even.

1. Positional strategies for Eve in $G1(\mathcal{A}^i; \mathcal{B}^i)$ and $G1(\mathcal{B}^i; \mathcal{A}^i)$: this is used to verify the $G1$-equivalence of $\mathcal{A}^i$ and $\mathcal{B}^i$.
2. A positional strategy for Eve in $G1(\mathcal{B}^i)$ from weakly coreachable states: we use this to verify that Eve wins $G1$ from everywhere.
3. To verify fullness of $\mathcal{B}^i$, we need to check if $L(\mathcal{B}^i_{>1}, q) \supseteq L(\mathcal{B}^i_{\text{safe}}, q) \cap L(\mathcal{B}^i, q)$ for each q, since the other inclusion holds trivially (Fact 18). Towards this, we consider the product automaton $\mathcal{P}^i$ of the safety automaton $\mathcal{B}^i_{\text{safe}}$ and the parity automaton $\mathcal{B}^i$ that inherits its priorities from corresponding transitions of $\mathcal{B}^i$. It is clear that for every state q in $\mathcal{B}^i$, $L(\mathcal{P}^i, (q, q)) = L(\mathcal{B}^i, q) \cap L(\mathcal{B}^i_{\text{safe}}, q)$. Thus, we include in $\mathcal{W}^i$, a positional strategy for Eve in $G1(\mathcal{B}^i_{>1}; \mathcal{P}^i)$: we use this to verify is Eve wins from $((\mathcal{B}^i_{>0}, q); (\mathcal{P}^i, (q, q)))$ for every state q of $\mathcal{B}^i_{>0}$. Due to Lemmas 5 and 6, such a strategy exists and suffices to verify the corresponding langauge-containment if $\mathcal{B}^i_{>0}$ is HD from all states.

It remains to verify that $\mathcal{B}^i$ is 1-safe HD, i.e., $\mathcal{B}^i_{>0}$ is HD from all states. Similar to the case of i even, we use the automaton $\mathcal{A}^{i+1}$, together with a function g_i in $\mathcal{W}^i$ from the states of $\mathcal{B}^i_{>0}$ to states of $\mathcal{A}^{i+1}$, as a certificate for verifying that $\mathcal{A}^{i+1}$ is obtained by reducing from corresponding multiple-state HD problem as in Lemma 35. This concludes our description of the polynomial certificate.

Polynomiality of certificate and verification. The certificate is overall polynomial: for each i, the witnesses $\mathcal{W}^i$ have polynomial size, and there are $k-1$ many such witnesses, one for each layer. The automaton $\mathcal{A}_{k-1}$ at most as many states as the sum of the number of states of $\mathcal{A}$ and k, since the only reduction from Lemma 35 increases the number of state by one. The alphabet size increases by at most the sum of the number of states of $\mathcal{B}^1, \mathcal{B}^2, \dots, \mathcal{B}^{k-2}$, which is also a polynomial. Thus, the overall certificate is polynomial in the number of states, transitions, alphabet size, and number of priorities.

The verification of the certificate involves checking history determinism of $\mathcal{A}^{k-1}$, a Büchi or a coBüchi automaton (which can be done polynomially, see [21, 2-token theorem]), and verifying positional strategies for Eve in polynomially-many 1-token games, which also takes polynomial time (Lemma 9).

This concludes the proof of Theorem 1.

Discussion

Here ends the two-decade old endeavour to settle the complexity of deciding whether a parity automaton is history deterministic: it is NP-complete (where the NP-hardness was shown in [25]).

This NP completeness result extends to the variant of reactive synthesis known as *good-enough-synthesis* [3] with a specification given by a deterministic parity automaton, as both problems are LOGSPACE-equivalent.

Our proof technique shows how to make the conditions used in [21] nest so that HDness of an automaton can be witnessed by HDness of a simpler automaton. One of the crucial insights developed in the current paper is the notion of fullness of a state (see Equation (1)): it shows how to use HDness of the restricted automaton $\mathcal{A}_{>1}$ to provide strategies in the original automaton $\mathcal{A}$, in a way which guarantees acceptance whenever we avoid transitions of priority 1. One may view it as a refinement of the condition of 1-safe-coverage from [21]; however checking the latter is only known to be in PSPACE, while the condition of fullness can be witnessed in NP.

Hopefully, these insights will be useful in other contexts related to history determinism, for instance to provide randomised resolvers or construct history-deterministic automata.

Beyond the ω-regular languages, settling the complexity of deciding history determinism opens up the possibility to advance on its generalisations for timed automata and register automata over infinite words, which, so far, have seemed out of reach.

Within the ω-regular realm, the other major question raised by Kuperberg and Skrzypczak [18] was whether history-deterministic Büchi automata can be more succinct than deterministic ones. They admit a quadratic determinisation procedure, but whether this is at all tight remains one of the classic open problems surrounding history-deterministic automata.

References

1 Bader Abu Radi and Orna Kupferman. Minimization and Canonization of GFG Transition-Based Automata. *Log. Methods Comput. Sci.*, 18(3), 2022. doi:10.46298/lmcs-18(3:16)2022.

2 Rohan Acharya, Marcin Jurdzinski, and Keya Prakash. Lookahead Games and Efficient Determinisation of History-Deterministic Büchi Automata. In *ICALP*, volume 297 of *LIPIcs*, pages 124:1–124:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ICALP.2024.124.

3 Shaull Almagor and Orna Kupferman. Good-enough synthesis. In *CAV (2)*, volume 12225 of *Lecture Notes in Computer Science*, pages 541–563. Springer, 2020. doi:10.1007/978-3-030-53291-8_28.

4 Marc Bagnol and Denis Kuperberg. Büchi Good-for-Games Automata Are Efficiently Recognizable. In *FSTTCS*, volume 122 of *LIPIcs*, pages 16:1–16:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.FSTTCS.2018.16.

5 Udi Boker, Denis Kuperberg, Orna Kupferman, and Michal Skrzypczak. Nondeterminism in the Presence of a Diverse or Unknown Future. In *ICALP (2)*, volume 7966 of *Lecture Notes in Computer Science*, pages 89–100. Springer, 2013. doi:10.1007/978-3-642-39212-2_11.

6 Udi Boker, Denis Kuperberg, Karoliina Lehtinen, and Michał Skrzypczak. On Succinctness and Recognisability of Alternating Good-for-Games Automata. *arXiv*, 2020. arXiv:2002.07278.

7 Udi Boker and Karoliina Lehtinen. When a Little Nondeterminism Goes a Long Way: An Introduction to History-Determinism. *ACM SIGLOG News*, 10(1):24–51, 2023. doi:10.1145/3584676.3584682.

8 Sougata Bose, Thomas A. Henzinger, Karoliina Lehtinen, Sven Schewe, and Patrick Totzke. History-deterministic Timed Automata. *Log. Methods Comput. Sci.*, 20(4), 2024. doi:10.46298/LMCS-20(4:1)2024.

9 J Richard Büchi and Lawrence H Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

10 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Generalized parity games. In *FoSSaCS*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007. doi:10.1007/978-3-540-71389-0_12.

11 Thomas Colcombet and Christof Löding. The Non-deterministic Mostowski Hierarchy and Distance-Parity Automata. In *Proc. of ICALP*, volume 5126, pages 398–409, 2008. doi:10.1007/978-3-540-70583-3_33.

12 Rüdiger Ehlers and Sven Schewe. Natural colors of infinite words. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, IIT Madras, Chennai, India, December 18-20, 2022*, volume 250 of *LIPIcs*, pages 36:1–36:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.36.

13 E. Allen Emerson. Automata, Tableaux and Temporal Logics (Extended Abstract). In *Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 79–88. Springer, 1985. doi:10.1007/3-540-15648-8_7.

14 Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. A bit of nondeterminism makes pushdown automata expressive and succinct. In *Proc. of MFCS*, pages 53:1–53:20, 2021. doi:10.4230/LIPIcs.MFCS.2021.53.

15 Yuri Gurevich and Leo Harrington. Trees, Automata, and Games. In *STOC*, pages 60–65. ACM, 1982. doi:10.1145/800070.802177.

16 Thomas A. Henzinger, Orna Kupferman, and Sriram K. Rajamani. Fair Simulation. *Inf. Comput.*, 173(1):64–81, 2002. doi:10.1006/inco.2001.3085.

17 Thomas A. Henzinger and Nir Piterman. Solving Games Without Determinization. In *CSL*, volume 4207 of *Lecture Notes in Computer Science*, pages 395–410. Springer, 2006. doi:10.1007/11874683_26.

18 Denis Kuperberg and Michal Skrzypczak. On Determinisation of Good-for-Games Automata. In *ICALP (2)*, volume 9135 of *Lecture Notes in Computer Science*, pages 299–310. Springer, 2015. doi:10.1007/978-3-662-47666-6_24.

19 O. Kupferman. Using the past for resolving the future. To appear, 2023.

20 Orna Kupferman, Shmuel Safra, and Moshe Y Vardi. Relating Word and Tree Automata. *Ann. Pure Appl. Logic*, 138(1-3):126–146, 2006. Conference version in 1996. doi:10.1016/J.APAL.2005.06.009.

21 Karoliina Lehtinen and Keya Prakash. The 2-token theorem: Recognising history-deterministic parity automata efficiently. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 1839–1850. ACM, 2025. doi:10.1145/3717823.3718310.

22 Christof Löding and Igor Walukiewicz. Minimal history-deterministic co-büchi automata: Congruences and passive learning. In *LICS*, pages 431–443. IEEE, 2025. doi:10.1109/LICS65433.2025.00039.

23 Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2), 1975.

24 Damian Niwinski and Igor Walukiewicz. Relating hierarchies of word and tree automata. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *STACS 98, 15th Annual Symposium on Theoretical Aspects of Computer Science, Paris, France, February 25-27, 1998, Proceedings*, volume 1373 of *Lecture Notes in Computer Science*, pages 320–331. Springer, 1998. doi:10.1007/BFB0028571.

25 Keya Prakash. Checking History-Determinism is NP-hard for Parity Automata. In *FoSSaCS (1)*, volume 14574 of *Lecture Notes in Computer Science*, pages 212–233. Springer, 2024. doi:10.1007/978-3-031-57228-9_11.

26 Keya Prakash. History-deterministic parity automata : games, complexity, and the 2-token theorem. PhD thesis, January 2025. URL: http://webcat.warwick.ac.uk/record=b4197352.

27 Bader Abu Radi and Orna Kupferman. Minimizing GFG transition-based automata. In *ICALP*, volume 132 of *LIPIcs*, pages 100:1–100:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.100.

28 Bader Abu Radi, Orna Kupferman, and Ofer Leshkowitz. Easy complementation of history-deterministic büchi automata. In *ATVA*, volume 15054 of *Lecture Notes in Computer Science*, pages 67–88. Springer, 2024. doi:10.1007/978-3-031-78709-6_4.