History Determinism vs. Good for Gameness in Quantitative Automata

Udi Boker
Reichman University, Herzliya, Israel

Karoliina Lehtinen
CNRS, Marseille-Aix Université, Université de Toulon, LIS, Marseille, France

Abstract

Automata models between determinism and nondeterminism/alternations can retain some of the algorithmic properties of deterministic automata while enjoying some of the expressiveness and succinctness of nondeterminism. We study three closely related such models – history determinism, good for gameness and determinisability by pruning – on quantitative automata.

While in the Boolean setting, history determinism and good for gameness coincide, we show that this is no longer the case in the quantitative setting: good for gameness is broader than history determinism, and coincides with a relaxed version of it, defined with respect to thresholds. We further identify criteria in which history determinism, which is generally broader than determinisability by pruning, coincides with it, which we then apply to typical quantitative automata types.

As a key application of good for games and history deterministic automata is synthesis, we clarify the relationship between the two notions and various quantitative synthesis problems. We show that good-for-games automata are central for "global" (classical) synthesis, while "local" (good-enough) synthesis reduces to deciding whether a nondeterministic automaton is history deterministic.

2012 ACM Subject Classification Theory of computation → Logic and verification

Keywords and phrases Good for games, history determinism, alternation, quantitative automata

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.38

Acknowledgements We thank Jan Otop for discussing Borel definability in quantitative automata.

1 Introduction

Boolean automata recognise languages of finite or infinite words, often used in verification to describe system behaviours. In contrast, quantitative automata define functions from words to values, and can describe system properties such as energy usage, battery-life or costs. Like Boolean automata, quantitative automata can have nondeterministic choices (disjunctions) and universal choices (conjunctions), which make them more powerful than deterministic models. Alternating automata combine both nondeterministic and universal choices.

However, not all nondeterminism is born equal. Generally, nondeterminism increases the expressiveness and succinctness of an automata model, but at the cost of also increasing the complexity of algorithmic problems on it, sometimes even rendering them undecidable. However, restricted forms of nondeterministic and even alternating automata can enjoy some of the good algorithmic properties of deterministic automata while also gaining in expressiveness and succinctness.

We focus on three closely related restrictions on nondeterminism and alternations, relevant to the synthesis problem. *History determinism* [11] postulates that the choices in the automaton – whether they be nondeterministic or universal – should not depend on the future of the input word. That is, one should be able to construct runs letter by letter while reading the input word, so that the resulting run is as good as one constructed with the knowledge of the full word. The notion of *good for games* automata comes from solving two-player games without determinisation [14]. It postulates that the composition of such an automaton $\mathcal{A}$ with games whose payoff function is described by $\mathcal{A}$ should be an equivalent game – that is, one with the same winner in the Boolean setting, or the same value in the quantitative setting. Finally, an automaton is *determinisable by pruning* if it embeds an equivalent deterministic automaton and, at least in the nondeterministic case, this notion can be seen as a (stronger) "semi-syntactic" version of history determinism.

The three notions are well studied in the Boolean setting. There, history determinism and good for gameness coincide, and are broader than determinisability by pruning in general, but coincide with it for some automata types [8].

We generalize these notions to the quantitative setting and study the relations between them. Some versions of these notions already appear in the literature with respect to quantitative automata, as we elaborate on in the related-work paragraph, however not in a systematic and consistent way, and without analysis of the relations between them.

We start with general results concerning arbitrary quantitative automata and then provide a more specific analysis of the following most common types of quantitative automata: $\mathsf{Sum}, \mathsf{Avg}, \mathsf{Inf}, \mathsf{Sup}$, discounted sum ($\mathsf{DSum}$), $\mathsf{LimInf}, \mathsf{LimSup}, \mathsf{LimInfAvg}$ and $\mathsf{LimSupAvg}$.

Surprisingly, it turns out that good for gameness and history determinism no longer coincide in the quantitative setting. The surprise comes from the fact that the two names are used interchangeably in the Boolean setting and are already starting to mix in the quantitative setting. (In the Boolean setting, even the seminal paper of Henzinger and Piterman [14], which named the "good for games" notion, defined history deterministic automata and showed that they are indeed good for games, while the other direction was only shown later [8]. In the quantitative setting, [16, 17, 18] speak of good for games quantitative automata, although their definition is closer to history determinism.)

We first observe that in the quantitative setting, the three notions need sub-notions, relating to whether one considers automata/games equivalence with respect to values or thresholds. (See Section 3 for the exact definitions.)

We then show that while good for gameness coincides with threshold good for gameness, history determinism is stricter than threshold history determinism, and only the latter, under some assumptions, is equivalent to good for gameness. (See Figure 2 for a detailed scheme of the relations.) The assumption for the equivalence of threshold history determinism and good for gameness is that the "letter game" played on the quantitative automaton (which defines whether or not it is history deterministic) is determined. We show that this is guaranteed for quantitative automata whose threshold versions define Borel sets.

Determinizability by pruning, which has an appealing structural definition, is generally stricter than history determinism for nondeterministic automata, already in the Boolean setting, while equivalent to it for some automata types. We observe that the two notions are incomparable for alternating automata, already in the Boolean setting (see Figure 5). We then analyse general properties of value functions that guarantee the equivalence of determinizability by pruning and history determinism for all nondeterministic quantitative automata whose value function has these properties. We apply these results to specific automata types. Specifically, we show the equivalence for $\mathsf{Sum}, \mathsf{Avg}, \mathsf{Inf}$ and Sup automata on finite words and DSum automata on finite and infinite words.

Finally, we discuss how the different notions are relevant for different quantitative synthesis problems. In quantitative synthesis [9, 2], the specification is a function f that maps sequences of input-output pairs onto values. The goal of the system is to respond to input letters by producing output letters while maximising the value of the resulting input-output sequence. Given a function f, one can ask several questions: (i) what is the best value a system can guarantee over all inputs [4]? (ii) can it guarantee at least a threshold value? (iii) can it guarantee for each input sequence I the best value that an input-output sequence including I has [12]? (iv) can it achieve a threshold value t for all inputs that appear in an input-output sequence with value at least t? In a nutshell, we show that on one hand, (threshold) good for games alternating quantitative automata can be used to solve (i) and (ii) via a product construction similar to the one used for deterministic automata [4]; and on the other hand, (iii) and (iv) for (threshold) history deterministic nondeterministic automata are linearly inter-reducible with deciding the (threshold) history-determinism of an automaton.

Related work. Thomas Colcombet's original definition of history determinism [11] also considered non-Boolean automata, namely cost automata. While the restriction of his definition to ω-regular automata coincides with the original definition in [14] of good for games automata [8], in the quantitative setting his definition is different from what we provide here. His notion can be viewed as 'approximated history-deterministic with respect to a threshold' as it asks for an approximation ratio that describes the difference between the value achieved by a strategy without the knowledge of the full input word and the actual value of the word. Another notion of approximative history determinism appears in [16, 17, 18] under the name of r-GFGness, where r is a bound on the difference of the two values. Zero-regret determinizability [3, 17] on the other hand lies somewhere between approximative determinizability by pruning and approximative history determinism. It requires an automaton to be approximatively equivalent to a deterministic automaton obtained by taking the product of the input automaton with a finite memory, with both the size of the memory and permitted regret as parameters. When both are set to zero, we have determinizability by pruning.

Observe that we use the term "quantitative automata" rather than "weighted automata". The latter usually relates to the algebraic definition, whereby the value of a nondeterministic automaton on a word is the semiring sum (or valuation-monoid sum) of its accepting runs' values. It is generally not defined for alternating automata. The former defines the value of a nondeterministic or alternating automaton on a word to be the supremum/infimum of its runs' values, having the "choice" and "obligation" interpretation of nondeterminism and universality, respectively. (See [5] for a discussion on the differences between the two.) Since history determinism naturally relates to "choice" and "obligation" in nondeterministic and alternating automata, quantitative automata better fit the present work.

Due to space constraints, some of the proofs appear in the appendix.

2 Preliminaries

Words. An *alphabet* Σ is a finite nonempty set of letters. A finite (resp. infinite) *word* $u = u_0 \dots u_k \in \Sigma^*$ (resp. $w = w_0 w_1 \dots \in \Sigma^\omega$) is a finite (resp. infinite) sequence of letters from Σ. We write Σ^∞ for $\Sigma^* \cup \Sigma^\omega$. We use $[i..j]$ to denote a set $\{i, \dots, j\}$ of integers, $[i]$ for $[i..i]$, $[..j]$ for $[0..j]$, and $[i..]$ for integers equal to or larger than i. We write $w[i..j], w[..j]$, and $w[i..]$ for the infix $w_i \dots w_j$, prefix $w_0 \dots w_j$, and suffix $w_i \dots$ of w. A *language* is a set of words, and the empty word is written ε.

Games. We consider turn-based zero-sum games between Adam and Eve, with Σ-labelled transitions. A play generates a word, and each word has a value, given by the game's payoff function. Eve tries to maximise the value of the play, while Adam tries to minimise it. Formally, for a payoff function f, an f *game* is defined on an *arena* $(V, E, V_E, V_A, L : E \to \Sigma \cup \{\varepsilon\})$, which consists of a (potentially infinite) set of positions V, partitioned into Eve's positions V_E and Adam's positions V_A, and a set of edges $E \subseteq V \times V$, labelled by L with letters from $\Sigma \cup \{\varepsilon\}$. In infinite-duration games every position has at least one outgoing edge. A play is a maximal path over V; its non-ε labels induce a word $w \in \Sigma^*$ or Σ^ω. The payoff of a play is the value of this word, given by the payoff function f.

Strategies for Adam and Eve map partial plays ending in a position v in V_A and V_E respectively to outgoing edges from v. A play or partial play π agrees with a strategy s_P, written $\pi \in s_P$, for a player $P \in \{A, E\}$, if whenever its prefix p ends in a position in V_P, the next edge is $s_P(p)$. The value $f(s_E)$ of a strategy s_E for Eve is $\inf_{\pi \in s_E} f(\pi)$ and the value $f(s_A)$ of a strategy s_A for Adam is $\sup_{\pi \in s_A} f(\pi)$. Let S_E and S_A be the sets of strategies for Eve and Adam respectively. If $\sup_{s \in S_E} f(s)$ (the best Eve can do) coincides with $\inf_{s \in S_A} f(s)$ (the best Adam can do), we say that G is determined and $\sup_{s \in S_E} f(s) = \inf_{s \in S_A} f(s)$ is called the value of G. Eve wins the t-threshold game on G, for some $t \in \mathbb{R}$, if the value of G is at least t; else Adam wins. Eve wins the strict t-threshold game on G if the value of G is greater than t. Two games are equivalent in this context if they have the same value. We restrict the scope of this article to determined games.

Quantitative Automata. An *alternating quantitative automaton* on words is a tuple $\mathcal{A} = (\Sigma, Q, \iota, \delta)$, where: Σ is an alphabet; Q is a finite nonempty set of states; $\iota \in Q$ is an initial state; and $\delta \colon Q \times \Sigma \to \mathsf{B}^+(\mathbb{Q} \times Q)$ is a transition function, where $\mathsf{B}^+(\mathbb{Q} \times Q)$ is the set of positive Boolean formulas (*transition conditions*) over weight-state pairs.

A *transition* is a tuple $(q, a, x, q') \in Q \times \Sigma \times \mathbb{Q} \times Q$, sometimes also written $q \xrightarrow{a:x} q'$. (Note that there might be several transitions with different weights over the same letter between the same pair of states[1].) We write $\gamma(t) = x$ for the weight of a transition $t = (q, a, x, q')$.

An automaton $\mathcal{A}$ is nondeterministic (resp. universal) if all its transition conditions are disjunctions (resp. conjunctions), and it is deterministic if all its transition conditions are just weight-state pairs. We represent the transition function of nondeterministic and universal automata as $\delta \colon Q \times \Sigma \to 2^{(\mathbb{Q} \times Q)}$, and of a deterministic automaton as $\delta \colon Q \times \Sigma \to \mathbb{Q} \times Q$.

We require that the automaton $\mathcal{A}$ is *total*, namely that for every state $q \in Q$ and letter $a \in \Sigma$, there is at least one state q' and a transition $q \xrightarrow{a:x} q'$. For a state $q \in Q$, we denote by $\mathcal{A}^q$ the automaton that is derived from $\mathcal{A}$ by setting its initial state ι to q.

A run of the automaton on a word w is intuitively a play between Adam and Eve. It starts in the initial state ι, and in each round, when the automaton is in state q and the next letter of w is a, Eve resolves the nondeterminism (disjunctions) of the transition condition $\delta(q, a)$ and Adam resolves its universality (conjunctions), yielding a transition $q \xrightarrow{a:x} q'$. The output of a play is thus a sequence $\pi = t_0 t_1 t_2 \ldots$ of transitions. As each transition t_i carries a weight $\gamma(t_i) \in \mathbb{Q}$, the sequence π provides a weight sequence $\gamma(\pi) = \gamma(t_0)\gamma(t_1)\gamma(t_2)\ldots$. More formally, given the automaton $\mathcal{A} = (\Sigma, Q, \iota, \delta)$ and a word $w \in \Sigma^*$ (resp. $w \in \Sigma^\omega$), we define the arena $G(\mathcal{A}, w)$ with positions $Q \times \Sigma^* \times \mathsf{B}^+(\mathbb{Q} \times Q)$ (resp. $Q \times \Sigma^\omega \times \mathsf{B}^+(\mathbb{Q} \times Q)$), the initial position $(\iota, w, \delta(\iota, w[0]))$, ε-labelled edges from (q, u, b) to (q, u, b') when b' is an immediate subformula of b, and x-labelled edges from $(q, u, (x, q'))$ to $(q', u[1..], \delta(q', u[1]))$. Conjunctive positions belong to Adam while disjunctive ones belong to Eve.

A Val automaton (for example a Sum automaton) is one equipped with a *value function* $\mathsf{Val} : \mathbb{Q}^* \to \mathbb{R}$ or $\mathsf{Val} : \mathbb{Q}^\omega \to \mathbb{R}$. The corresponding game is the Val game on the arena $G(\mathcal{A}, w)$: each run π (play in $G(\mathcal{A}, w)$) has a real value $\mathsf{Val}(\gamma(\pi))$, which we abbreviate by

[1] This extra flexibility of allowing for "parallel" transitions with different weights is often omitted (e.g., in [10]) since it is redundant for some value functions while important for others.

$\mathsf{Val}(\pi)$. When this game is determined, we say that the value of $\mathcal{A}(w)$ is the value of $G(\mathcal{A}, w)$, and if $G(\mathcal{A}, w)$ is determined for all $w \in \Sigma^\omega$, we say that $\mathcal{A}$ realizes a function from words to real numbers. We restrict the scope of this article to automata realizing functions.

Two automata $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent*, denoted by $\mathcal{A} \equiv \mathcal{A}'$, if they realize the same function. For a threshold $t \in \mathbb{R}$ and a Val automaton $\mathcal{A}$, we also speak of a corresponding Boolean t-threshold Val automaton $\mathcal{A}'$ that accepts the words w such that $\mathcal{A}(w) \geq t$.

Observe that when $\mathcal{A}$ is nondeterministic, a run of $\mathcal{A}$ on a word w is a sequence π of transitions, and the value of $\mathcal{A}$ on w is the supremum of $\mathsf{Val}(\pi)$ over all these runs π.

Value functions. We list here the most common value functions for quantitative automata on finite/infinite words, defined over sequences of rational weights[2]:

- For finite sequences $v = v_0 v_1 \dots v_{n-1}$:
  - $\mathsf{Sum}(v) = \sum_{i=0}^{n-1} v_i$
  - $\mathsf{Avg}(v) = \frac{1}{n} \sum_{i=0}^{n-1} v_i$
- For finite and infinite sequences $v = v_0 v_1 \dots$:
  - $\mathsf{Inf}(v) = \inf\{v_n \mid n \geq 0\}$
  - $\mathsf{Sup}(v) = \sup\{v_n \mid n \geq 0\}$
  - For a discount factor $\lambda \in \mathbb{Q} \cap (0, 1)$, $\mathsf{DSum}(v) = \sum_{i \geq 0} \lambda^i v_i$
- For infinite sequences $v = v_0 v_1 \dots$:
  - $\mathsf{LimInf}(v) = \lim_{n\to\infty} \inf\{v_i \mid i \geq n\}$
  - $\mathsf{LimSup}(v) = \lim_{n\to\infty} \sup\{v_i \mid i \geq n\}$
  - $\mathsf{LimInfAvg}(v) = \mathsf{LimInf}(\mathsf{Avg}(v_0), \mathsf{Avg}(v_0, v_1), \mathsf{Avg}(v_0, v_1, v_2), \dots)$
  - $\mathsf{LimSupAvg}(v) = \mathsf{LimSup}(\mathsf{Avg}(v_0), \mathsf{Avg}(v_0, v_1), \mathsf{Avg}(v_0, v_1, v_2), \dots)$

($\mathsf{LimInfAvg}$ and $\mathsf{LimSupAvg}$ are also called $\underline{\mathsf{MeanPayoff}}$ and $\overline{\mathsf{MeanPayoff}}$.)

Products. The synchronized product of a Σ-labelled game G and an automaton $\mathcal{A}$ over alphabet Σ is (like in the Boolean setting, see e.g., [8, Definition 1]) a game $G \times \mathcal{A}$ obtained by taking the product of the positions of G and the states and transition conditions of $\mathcal{A}$, and their corresponding transitions. Positions with nondeterminism are of Eve and positions with universality are of Adam. Transitions carry their weight from the corresponding transition in $\mathcal{A}$. The payoff function of the game is the value function of $\mathcal{A}$.

3 Good For Gameness, History Determinism, and Determinizability By Pruning

In the Boolean setting, "good for gameness" and "history determinism", stemming from different concepts, coincide both for nondeterministic and alternating automata [8].

We generalize these definitions to quantitative automata, observing that under this setting they need some sub-variants, relating to whether one considers automata/games equivalence with respect to all values or some threshold[3]. As shown in Section 4, the two main notions, as well as some of their variants, are generally not equivalent in the quantitative setting.

[2] There are also value functions that are more naturally defined over sequences of tuples of rational numbers, for example discounted-summation with multiple discount factors [6].

[3] For a threshold $t \in \mathbb{R}$, we provide the definitions with respect to a non-strict inequality $\geq t$. Using strict inequality $> t$ instead, yields the same relations between the notions, as stated in Theorem 4.

The notion of determinizability by pruning, which has an appealing structural definition, is generally stricter than good for gameness and history determinism in the setting of nondeterministic automata, already in the Boolean setting, yet we show that for some value functions it is equivalent to history determinism. For alternating automata, we show that it is incomparable with history determinism and good for gameness.

▶ **Definition 1** (Good for gameness). *An automaton $\mathcal{A}$ realizing a function $f : \Sigma^* \to \mathbb{R}$ or $f : \Sigma^\omega \to \mathbb{R}$ is*

- good for games *if for every determined*[4] *game G with a Σ-labelled arena and payoff function f, we have that G and $G \times \mathcal{A}$ have the same value;*
- good for t-threshold games*, for some $t \in \mathbb{R}$, if for every determined game G with a Σ-labelled arena and payoff function f, Eve wins the t-threshold game on G if and only if she wins the t-threshold game on $G \times \mathcal{A}$;*
- good for threshold games *if it is good for t-threshold games for all $t \in \mathbb{R}$.*

An automaton is history deterministic if there are strategies to resolve its nondeterminism and universality, such that for every word, the (threshold) value remains the same.

▶ **Definition 2** (History-determinism). *Consider an alternating* Val *automaton $\mathcal{A} = (\Sigma, Q, \iota, \delta)$ realizing a function $f : \Sigma^* \to \mathbb{R}$ or $f : \Sigma^\omega \to \mathbb{R}$. Formally,* history determinism *is defined via* letter games*, detailed below.*

- *$\mathcal{A}$ is* history deterministic *if Eve and Adam win their letter games.*
- *$\mathcal{A}$ is* t-threshold history deterministic*, for some $t \in \mathbb{R}$, if Eve and Adam win their t-threshold letter games.*
- *$\mathcal{A}$ is* threshold history deterministic *if it is t-threshold history deterministic for all $t \in \mathbb{R}$.*

Eve's (Adam's) letter games are the following win-lose games, in which Adam (Eve) chooses the next letter and Eve and Adam resolve the nondeterminism and universality, aiming to construct a run whose value is (threshold) equivalent to the generated word's value.

Eve's letter game: *A configuration is a pair (σ, b) where $b \in \mathsf{B}^+(Q)$ is a transition condition and $\sigma \in \Sigma \cup \{\varepsilon\}$ is a letter. (We abuse ε to also be an empty letter.) A play begins in $(\sigma_0, b_0) = (\varepsilon, \iota)$ and consists of an infinite sequence of configurations $(\sigma_0, b_0)(\sigma_1, b_1)\dots$. In a configuration (σ_i, b_i), the play proceeds to the next configuration (σ_{i+1}, b_{i+1}) as follows.*
  - *If b_i is a state of Q, Adam picks a letter a from Σ, and $(\sigma_{i+1}, b_{i+1}) = (a, \delta(b_i, a))$.*
  - *If b_i is a conjunction $b_i = b' \wedge b''$, Adam chooses between (ε, b') and (ε, b'').*
  - *If b_i is a disjunction $b_i = b' \vee b''$, Eve chooses between (ε, b') and (ε, b'').*

  In the limit, a play consists of an infinite word w that is derived from the concatenation of $\sigma_0, \sigma_1, \dots$, as well as an infinite sequence $b_0, b_1, \dots$ of transition conditions, which yields an infinite sequence $\pi = t_0, t_1, \dots$ of transitions.

  If $\mathcal{A}$ is over infinite words, Eve wins a play in the letter-game if $\mathsf{Val}(\pi) \geq \mathcal{A}(w)$. In the t-threshold letter game, Eve wins if $\mathcal{A}(w) \geq t \implies \mathsf{Val}(\pi) \geq t$. For $\mathcal{A}$ over finite words, Eve wins if $\mathsf{Val}(\pi[0..i]) \geq \mathcal{A}(w[0..i])$ or $\mathcal{A}(w[0..i]) \geq t \implies \mathsf{Val}(\pi[0..i]) \geq t$ for all i.

Adam's letter game *is similar to Eve's game, except that Eve chooses the letters instead of Adam, and Adam wins a play in his letter game if $\mathsf{Val}(\pi) \leq \mathcal{A}(w)$ and in his t-threshold letter game if $\mathcal{A}(w) < t \implies \mathsf{Val}(\pi) < \mathcal{A}(w)$. (The asymmetry of $<$ and $\leq$ is intended).*

Intuitively, an automaton is determinizable by pruning if it can be determinized to an equivalent (w.r.t. a threshold) deterministic automaton by removing some of its states and transitions. (In an alternating automaton, "removing transitions" means removing some disjunctive and conjunctive choices.)

---

[4] We discuss in the conclusion questions that arise if this restriction is lifted

▶ **Definition 3** (Determinizability by Pruning). *A* Val *automaton* $\mathcal{A}$ *is*
- determinizable by pruning *if there exists a deterministic* Val *automaton* $\mathcal{A}'$ *that is derived from* $\mathcal{A}$ *by pruning, such that* $\mathcal{A}' \equiv \mathcal{A}$*;*
- t-threshold determinizable by pruning *if there is a deterministic* Val *automaton* $\mathcal{A}'$ *that is derived from* $\mathcal{A}$ *by pruning, such that for every word* w*, we have* $\mathcal{A}'(w) \geq t$ *iff* $\mathcal{A}(w) \geq t$*;*
- threshold determinizable by pruning *if it is* t*-threshold determinizable by pruning* $\forall t \in \mathbb{R}$*.*

Observe that a Val-automaton can be good for games, history deterministic, or determinizable by pruning when interpreted on infinite words, but not when interpreted on finite words, as demonstrated in Figure 1 .

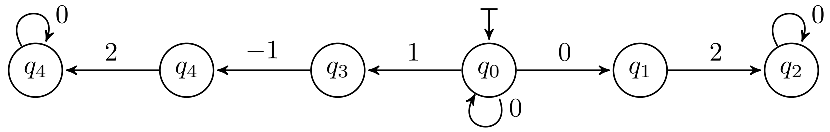

Figure 1 A nondeterministic DSum-automaton with discount factor $\frac{1}{2}$ over a unary alphabet that is determinizable by pruning, good for games, and history deterministic with respect to infinite words, but none of them with respect to finite words: For the single infinite word, the initial choice of going from q_0 to q_1 provides the optimal value of 1, making it all of the above. On finite words, on the other hand, it is not even threshold history deterministic (and by Theorem 4 neither of the rest), since in order to guarantee a value of at least 1, the first transition should be different for the word of length 1 and the word of length 2, going to q_3 for the former and to q_1 for the latter.

4 The Relations Between Notions

Having defined these notions, we now establish which inclusions hold in general, and which are conditional on characteristics of the value function, as summarised in Figure 2.

▶ **Theorem 4.** *(Threshold) good for gameness, (threshold) history determinism, and (threshold) determinizability by pruning of quantitative automata are related as described in Figure 2.*

Considering good for gameness, if an automaton $\mathcal{A}$ is good for all games then it is obviously good for all threshold games. The implication for the other direction stems from the fact that every concrete game G has a single value v. Then for G, it is enough to be good for v-threshold games, and for all automata, it is enough to be good for all threshold games.

▶ **Lemma 5.** *Good for Gameness* $\iff$ *Threshold Good for Gameness.*

For a t-threshold history deterministic automaton $\mathcal{A}$, Eve and Adam have strategies to win their t-letter games on $\mathcal{A}$. Thus, whenever Eve or Adam win some t-threshold game G, they can combine their two winning strategies to win $G \times \mathcal{A}$.

▶ **Lemma 6.** *Threshold History Determinism* $\implies$ *Threshold Good for Gameness*

For the other direction, we generalize proofs from [7, 8]: assuming that the automaton $\mathcal{A}$ is not threshold history deterministic we construct a threshold game G with respect to which $\mathcal{A}$ is not good for composition (namely, the product of G with $\mathcal{A}$ does not have the same winner as G). However, to build this game, we assume that either Adam wins Eve's letter game on $\mathcal{A}$ or Eve wins Adam's letter game on $\mathcal{A}$, that is, we assume that the letter games on $\mathcal{A}$ are determined. We later show that this determinacy requirement holds for all the specific value functions that we consider in the paper.

Good For Gameness $=^1$ Threshold Good For Gameness $\cong^2$ Threshold History Determinism

(for nondet.) $\subsetneq^4$ $\subsetneq^3$

Threshold Determinizability by Pruning $\neq^5$ History Determinism

$\subsetneq^3$ $\subsetneq^4$ (for nondet.)

Determinizability by Pruning

1. Always holds (Lemma 5).
2. The $\Longleftarrow$ implication always holds (Lemma 6); The $\Longrightarrow$ implication holds at least for all Val automata whose threshold letter games are determined (Lemma 7), e.g., for $\mathsf{Inf}, \mathsf{Sup}, \mathsf{LimInf}, \mathsf{LimSup}, \mathsf{DSum}$ and all functions on finite words (Theorem 9).
3. Strict containment for all non-trivial value functions with at least three values (Lemma 10); Equal (the same notion) for value functions with two values.
4. Strict containment, in general, for nondeterministic automata (Propositions 11 and 12); Equivalent notions for some nondeterministic Val automata (Section 4.1); Incomparable for alteranting automata (Proposition 13).
5. Incomparable, in general, for value functions with at least three values (Lemma 10 and Propositions 12 and 13); For value functions with two values, as relation 4 above.

Figure 2 The relations between the different notions.

▶ **Lemma 7.** *For* Val *automata whose threshold letter games are determined, Threshold Good for Gameness* $\Longrightarrow$ *Threshold History Determinism.*

Proof. Consider a Val automaton $\mathcal{A}$ whose threshold letter games are determined. Then, if $\mathcal{A}$ is not threshold history deterministic, it follows that Adam wins Eve's t-letter game on $\mathcal{A}$ for some threshold t, or Eve wins Adam's t-letter game on $\mathcal{A}$ for some t. We show below that in both cases $\mathcal{A}$ is not good for threshold games, proving the contra-positive of the claim.

Assume that Adam wins Eve's t-letter game $G_{\mathcal{A},t}$ on $\mathcal{A}$ for some threshold t with a strategy s. We can build a one-player Σ-labelled (infinite) game G_s in which the positions, which all belong to Adam, are the finite words that can be constructed along plays of $G_{\mathcal{A},t}$ that agree with s, and where for every positions u and $u \cdot a$, there is an a-labelled edge from the position u to the position $u \cdot a$. The empty word ε is the initial position. In other words, this is the one-player arena in which plays correspond to (infinite) words that occur in the letter game if Adam uses the strategy s. Notice that since s is a winning strategy in the t-letter game, all words w that are plays of G_s have $\mathcal{A}(w) \geq t$. The t-threshold game on G_s is therefore winning for Eve.

We now argue that Adam wins the product game $G_s \times \mathcal{A}$. Indeed, Adam can now use the strategy s to choose directions in G_s according to the run constructed so far in $\mathcal{A}$, and resolve conjunctions in $\mathcal{A}$ according to the history of the word and run so far. Since s is a winning strategy for Adam in the letter game, this guarantees that the resulting run ρ is such that $\mathsf{Val}(\rho) < t$. Then $\mathcal{A}$ is not threshold-good-for-games, as witnessed by G_s.

By a similar argument, if Eve wins Adam's t-letter game for some t with a strategy s, then we can construct a one-player game G_s in which all positions belong to Eve such that G_s is winning for Adam (i.e., all words have value strictly smaller than t), but in the product

$G \times \mathcal{A}$, Eve wins, i.e., can force value at least t.

Hence if either player has a winning strategy in the other player's threshold letter game for some threshold, then the automaton is not good for threshold games. ◀

We now show that letter games on Val automata whose threshold variants define Borel sets are determined. This stems from the fact that their winning condition is a union between two conditions that can be defined by threshold Val automata or their complement.

▶ **Proposition 8.** *If for some value function* Val*, all threshold* Val *automata define Borel sets, then threshold letter games on* Val *automata are determined.*

Proof. Consider Eve's t-letter game on a Val automaton $\mathcal{A}$, for some threshold $t \in \mathbb{R}$. A play of the game generates a sequence $\rho \in (\Sigma \times V)^\omega$, where Σ is $\mathcal{A}$'s alphabet and V is the finite set of its weights. We may view ρ as a pair of sequences (ρ_Σ, ρ_V), where $\rho_\Sigma \in \Sigma^\omega$ and $\rho_V \in V^\omega$. Then the winning set of Eve is $\{\rho \mid \mathsf{Val}(\rho_V) \geq t \text{ or } \mathcal{A}(\rho_\Sigma) < t\}$.

Observe that the set $S_V = \{\rho \mid \mathsf{Val}(\rho_V) \geq t\}$ can be defined by a t-threshold deterministic Val automaton $\mathcal{B}$, in which the weight of a transition over the input letter (σ, v) is v. Let $\mathcal{A}'$ be a t-threshold Val automaton that is identical to $\mathcal{A}$, except that its alphabet is $\Sigma \times V$, while the transitions are sensitive, as in $\mathcal{A}$, only to the Σ component of the input. Then the set $S_\Sigma = \{\rho \mid \mathcal{A}(\rho_\Sigma) \geq t\}$ is defined by $\mathcal{A}'$.

As the winning condition of Eve's letter game is the union of S_V and the complement of S_Σ, and as both are Borel sets, so is the winning condition. Hence, by [20] the game is determined.

The argument regarding Adam's letter game is analogous. ◀

A direct corollary of Proposition 8 is that for most of the common quantitative automata, we have that good for gameness is equivalent to threshold history determinism. In particular, this is the case for all the concrete value functions that are considered in this paper.

▶ **Theorem 9.** *Good For Gameness $\iff$ Threshold History Determinism for all* Val *automata on finite words, and* Inf, Sup, LimInf, LimSup, LimInfAvg, LimSupAvg *and* DSum *automata on infinite words.*

Proof. It is enough to show that threshold automata of these types define Borel sets, and then the claim directly follows from Lemmas 5 and 6 and Proposition 8.

Automata on finite words. Every threshold Val automaton on finite words defines a set of finite words, which is a countable union of singletons and thus a Borel set.

Inf and Sup automata. Observe that Inf, Sup automata on infinite words are "almost" like automata on finite words, in the sense that the value of the automaton on a word is equal to its value on some prefix of the word. Formally, for a Sup automaton $\mathcal{A}$, we have that the set of infinite words $\{w \in \Sigma^\omega \mid \mathcal{A}(w) \geq t\}$ is equal to the set of infinite words $\{w \in \Sigma^\omega \mid \text{ exists } p \in \mathbb{N} \text{ such that } \mathcal{A}(w[..p]) \geq t\}$ (when considering $\mathcal{A}$ to operate on finite words). Observe that it is indeed a Borel set, since it is a countable union of open sets. The argument for Inf-automata is analogous, having a countable intersection of closed sets.

LimInf and LimSup automata. Observe that threshold LimInf and LimSup automata are equivalent to coBüchi and Büchi automata, respectively, thus defining ω-regular languages, which are known to be Borel sets [22].

LimInfAvg and LimSupAvg automata. Directly follows from [15, Corollaries 6 and 10].

DSum automata. For DSum automata the argument stems from the continuity with respect to the Cantor topology of functions defined by DSum automata.

Consider a DSum automaton $\mathcal{A}$ and a threshold $t \in \mathbb{R}$. Define the following set of infinite words $B_t = \{w \in \Sigma^\omega \mid \text{ for every } n \in \mathbb{N} \text{ and } p_0 \in \mathbb{N}, \text{ there exists } p > p_0, \text{ such that } \mathcal{A}(w[..p]) \geq t - \frac{1}{n}\}$ (when considering $\mathcal{A}$ to operate on finite words).

Observe that B_t is a Borel set, since $\{w \mid \mathcal{A}(w[..p]) \geq t - \frac{1}{n}\}$ is an open set, and the existential and universal quantifiers can be defined by countable unions and intersections.

We claim that B_t is equivalent to the set $A_t = \{w \mid \mathcal{A}(w) \geq t\}$, which will prove the required statement. One direction is immediate – if a word w is in A_t then by the definition of $\mathcal{A}(w)$, there are runs of $\mathcal{A}$ on w whose supremum is at least t, admitting the membership of w in B_t.

As for the other direction, we show that for every $n \in \mathbb{N}$, there is a run r of $\mathcal{A}$ on w, such that $\mathsf{Val}(r) \geq t - \frac{1}{n}$, proving that w is in A_t. (One can then even combine these runs to create a single run that attains a value at least t.) Consider some $n \in \mathbb{N}$, and let R be the infinite set of finite runs $r_1, r_2, \ldots$ that witness the membership of w in B_t with respect to $2n$. That is, r_i is a run on a prefix of w of length at least i, whose value is at least $t - \frac{1}{2n}$. We create a single run r from R in a "Konig's lemma" approach (for simplicity, we detail the construction for a nondeterministic automaton, and later explain how to extend it to an alternating automaton):

We choose the first transition t_1 in r to be a transition that appears as the first transition in infinitely many r_i's. We then choose the next transition t_2 to be a transition that appears as the second transition, where t_1 is the first transition, in infinitely many r_i's, and so on. Notice that r is indeed a run of $\mathcal{A}$ and its value is at least $t - \frac{1}{n}$: By the discounted-sum value function, if the value of a long enough prefix is at least $t - \frac{1}{2n}$, the value of the entire run cannot be smaller than $t - \frac{1}{n}$.

Now, for an alternating automaton, rather than "choosing transitions" we need to "resolve the nondeterminism", while ensuring that the choice we make appears in infinitely many runs after the previous nondeterministic and *universal* choices that were already made. ◀

History determinism and determinizability by pruning obviously imply their threshold versions; Figure 3 demonstrates that the converse does not hold.

▶ **Lemma 10.**

- *History Determinism* $\overset{\Longrightarrow}{\not\Longleftarrow}$ *Threshold History Determinism;*
- *Determinizability by Pruning* $\overset{\Longrightarrow}{\not\Longleftarrow}$ *Threshold Determinizability by Pruning;*
- *History Determinism* $\not\Longleftarrow$ *Threshold Determinizability by Pruning*

Proof. The implications are straightforward: a winning strategy for each player in their letter game is also a winning strategy in their t-threshold letter game, for every threshold $t \in \mathbb{R}$; further, if an automaton $\mathcal{A}'$ that results from pruning $\mathcal{A}$ is equivalent to $\mathcal{A}$, then for every threshold t and word w, if $\mathcal{A}(w) \geq t$ then $\mathcal{A}'(w) \geq t$.

As for the non-implications, Figure 3 provides such counter examples, which hold, with some variations, with respect to every non-trivial value function with at least 3 values, and in particular with respect to all value functions discussed in the paper.

Consider, for example, the automaton $\mathcal{A}$ of Figure 3 with respect to the Sup value function. It is not history deterministic, since if the nondeterminism in q_0 is resolved by going to q_1, the resulting automaton is not equivalent to $\mathcal{A}$ with respect to the finite word aa and infinite word a^ω, and if it is resolved by going to q_2, the resulting automaton fails on ab and ab^ω.

On the other hand, $\mathcal{A}$ is threshold determinizable by pruning and threshold history deterministic: For a threshold up to 1, the nondeterminism is resolved by going to q_1 and for the threshold 2 by going to q_2. ◀

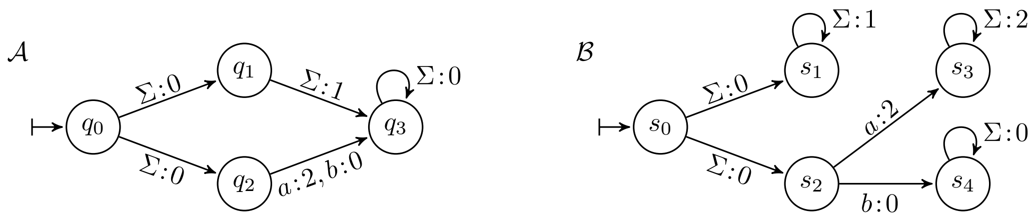

Figure 3 Nondeterministic automata that are threshold history deterministic and threshold determinizable by pruning, but not history deterministic and not determinizable by pruning. The automaton $\mathcal{A}$ has this property with respect, for example, to the Sum/DSum/Sup value functions, and $\mathcal{B}$ with respect, for example, to Avg/LimSup/LimInf/LimSupAvg/LimInfAvg.

History Determinism $\neq$ Determinizability by Pruning

For nondeterministic automata, it is clear that determinizability by pruning implies history determinism: the pruning provides a strategy for Eve in her letter game.

▶ **Proposition 11.** *For nondeterministic automata, determinizability by pruning $\implies$ history determinism.*

The converse was shown to be false for Büchi and coBüchi automata [7], directly implying the same for LimSup and LimInf automata. Considering LimInfAvg and LimSupAvg automata, the automaton depicted in Figure 4, which is similar to the coBüchi automaton in [7, Figure 3], is history deterministic but not determinizable by pruning.

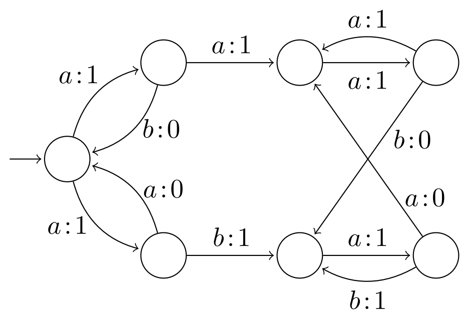

Figure 4 (Similar to [7, Figure 3].) A history deterministic LimInfAvg or LimSupAvg automaton that is not determinizable by pruning. (Missing transitions lead to a sink with a 0-weighted self loop on both a and b.) It is history deterministic by a strategy that chooses in the initial state to go up if and only if it went down the previous time. Following this strategy in the letter game, Eve returns infinitely often to the initial state only on $(aaab)^\omega$, getting a value $\frac{1}{2}$, which is also the automaton's value on it. For every other word, the run of Eve moves to the right part of the automaton, which is deterministic, guaranteeing Eve the optimal value on the word. On the other hand, every pruning of it yields an automaton whose value on either a^ω or $(ab)^\omega$ is $\frac{1}{2}$ instead of 1.

▶ **Proposition 12.** *For nondeterministic* LimInf, LimSup, LimInfAvg, *and* LimSupAvg *automata, history determinism $\not\Longrightarrow$ determinizability by pruning.*

For alternating automata, it turns out that (threshold) history determinism and determinizability by pruning are incomparable, as demonstrated in Figure 5.

▶ **Proposition 13.** *For (Boolean and quantitative) alternating automata, determinizability by pruning $\not\Rightarrow$ history determinism, good for gameness.*

Proof. The claim holds for Boolean automata as well as quantitative automata with every non-trivial value function. Consider the alternating finite automaton on finite words (which can also be viewed, for example as a Sup automaton) in Figure 5. It does not accept any word, and can be determinized by pruning the right nondeterministic transition. However, it is not history deterministic: Eve wins Adam's letter game, by choosing the right nondeterministic transition. ◀

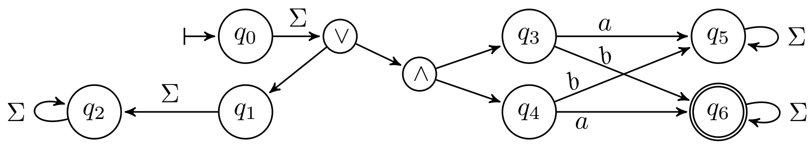

Figure 5 An alternating finite automaton on finite words that is determinizable by pruning, but not history deterministic nor good for games.

4.1 When (History Determinism = Determinizability by Pruning)

In general for nondeterministic automata, determinizability by pruning is strictly contained in history determinism; here we study when the two notions coincide. In the Boolean setting, they are equivalent for nondeterministic finite automata on finite words (NFAs) [19] as well as for nondeterministic weak automata on infinite words [21]. Here we analyse general properties of value functions that guarantee this equivalence, and then consider specific value functions on finite and infinite words. The general properties that we analyze relate to how "sensitive" the value function is to the prefix, current position, and suffix of the weight sequence.

We begin by defining *cautious*[5] *strategies* for Eve in the letter game, that we then use to define value functions that are "present focused". Intuitively, a strategy is cautious if it avoids mistakes, that is, it only builds run prefixes that can still achieve the maximal value of any continuation of the word so far.

▶ **Definition 14** (Cautious strategies). *Consider Eve's letter game on a* Val *automaton $\mathcal{A}$. A move (transition) $t = q \xrightarrow{\sigma:x} q'$ of Eve, played after some run ρ ending in a state q, is* non-cautious *if for some word w, there is a run π' from q over σw such that $\mathsf{Val}(\rho\pi')$ is strictly greater than the value of $\mathsf{Val}(\rho\pi)$ for any π starting with t.*

A strategy is cautious *if it makes no non-cautious moves.*

We call a value function *present focused* if, morally, it depends on the prefixes of the value sequence, formalized by winning the letter game via cautious strategies.

▶ **Definition 15** (Present-focused value functions). *A value function* Val*, on finite or infinite sequences, is* present focused *if for all automata $\mathcal{A}$ with value function* Val*, every cautious strategy in the letter game on $\mathcal{A}$ is also a winning strategy in that game.*

Value functions on finite sequences are present focused, as they can only depend on prefixes.

[5] Similar transitions are sometimes called "residual" in the literature.

▶ **Lemma 16.** *Every value function* Val *on finite sequences is present focused.*

Proof. Assume Eve plays a cautious strategy s in some letter game on an automaton $\mathcal{A}$ on finite words. Towards a contradiction, assume that there is a finite play π, in which Adam plays some word w and Eve plays a run ρ over w such that $\mathsf{Val}(\rho) < \mathcal{A}(w)$. Then, let ρ' be the longest prefix of ρ such that the highest value of a run over w starting with ρ' is $\mathcal{A}(w)$. Since ρ is not a run with value $\mathcal{A}(w)$, ρ' is a strict prefix of ρ. However, since ρ' is the longest prefix that could be continued into a run with value $\mathcal{A}(w)$, Eve's next move after ρ' must be non-cautious, contradicting that s never plays non-cautious moves. ◀

▶ Remark 17. Value functions on infinite sequences are not necessarily present focused. For example, consider the automaton depicted in Figure 1, but viewed as a Sup automaton on infinite words rather than a DSum automaton. Observe that Eve can forever stay in q_0, always having the potential to continue to an optimal run with value 2, but never fulfilling this potential.

We now define "suffix monotonicity" of value functions, which, with present-focus, will guarantee the equivalence of history determinism and determinizability by pruning.

▶ **Definition 18.** *A value function* Val *is* suffix monotonic *if for every finite set $S \subset \mathbb{Q}$, sequence $\alpha \in S^*$ and sequences $\beta, \beta' \in S^\infty$, we have $\mathsf{Val}(\beta) \geq \mathsf{Val}(\beta')$ iff $\mathsf{Val}(\alpha\beta) \geq \mathsf{Val}(\alpha\beta')$.*

Observe that the above definition does not consider arbitrary sequences of rational numbers, but rather sequences of finitely many different rational numbers, which is the case in sequences of weights that are generated by runs of quantitative automata.

Value functions that are *suffix dependent* (namely Val functions such that for every finite set $S \subset \mathbb{Q}$, sequences $\alpha, \alpha' \in S^*$ and sequence $\beta \in S^\infty \setminus \{\varepsilon\}$, we have $\mathsf{Val}(\alpha\beta) = \mathsf{Val}(\alpha'\beta)$) are obviously suffix monotonic. Examples for such value functions are the acceptance condition of NFAs (i.e, a "last" value function, that depends only on the last weight of 0 for rejection and 1 for acceptance), all ω-regular conditions (which depend on the states/transitions that are visited infinitely often), LimInf, LimSup, LimInfAvg, and LimSupAvg. Examples for value functions that are suffix monotonic but not suffix dependent are Sum, Avg and DSum, and examples for value functions that are not suffix monotonic are Inf and Sup.

We next show that suffix monotonicity together with present-focus guarantee the equivalence of history determinism and determinizability by pruning. The idea is that under these conditions, every cautious strategy in the letter game can be arbitrarily pruned into a positional strategy (with respect to the automaton states).

▶ **Theorem 19.** *For nondeterministic* Val *automata, where* Val *is a present-focused and suffix-monotonic value function, we have that history determinism $\iff$ determinizability by pruning.*

Proof. We show that Eve wins her letter game on $\mathcal{A}$ with a positional strategy, which implies that $\mathcal{A}$ is determinizable by pruning.

Let s be a cautious strategy for Eve in the letter game on $\mathcal{A}$. Let $\hat{s}$ be an arbitrary positional strategy that only uses transitions also used by s. We argue that $\hat{s}$ is also cautious. Indeed, if $\hat{s}$ chooses $\tau = q \xrightarrow{\sigma:x} q'$ after a play $(\hat{w}, \hat{\rho})$ of the letter game, there is some play (w, ρ) from which s plays τ. Since s is cautious, for every word v and every run π' from q over σv, there is a run π from q starting with τ such that $\mathsf{Val}(\rho\pi) \geq \mathsf{Val}(\rho\pi')$. Thus, by suffix monotonicity, we have $\mathsf{Val}(\pi) \geq \mathsf{Val}(\pi')$, and then again by the other direction of suffix monotonicity, we get that $\mathsf{Val}(\hat{\rho}\pi) \geq \mathsf{Val}(\hat{\rho}\pi')$, implying that $\hat{s}$ choosing τ is a cautious move.

Then $\mathcal{A}$ is determinisable by pruning: the subautomaton $\mathcal{A}_{\hat{s}}$ that only has transitions used by $\hat{s}$ is equivalent to $\mathcal{A}$. Indeed, for every word w, $\mathcal{A}_{\hat{s}}(w)$ is $\mathsf{Val}(\rho_w)$, where ρ_w is the unique run of $\mathcal{A}_{\hat{s}}$ over w. The run ρ_w is also the run built by $\hat{s}$ in the letter game over w. Since $\hat{s}$ is cautious and Val is present focused, we have that $\hat{s}$ is a history-deterministic strategy, which guarantees that $\mathsf{Val}(\rho_w) = \mathcal{A}(w)$, giving us the equivalence of $\mathcal{A}$ and $\mathcal{A}_{\hat{s}}$. ◀

▶ Remark 20. Both present-focus and suffix-monotonicity are necessary in Theorem 19. For example LimInf is suffix monotonic, but LimInf automata are not determinizable by pruning. On the other hand, Figure 6 demonstrates a present-focused value function whose history deterministic automata on finite words are not determinizable by pruning.

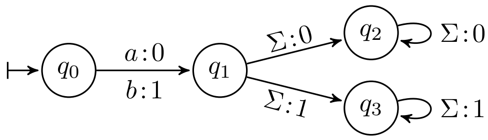

Figure 6 A nondeterministic Val automaton $\mathcal{A}$ on finite words with the value function $\mathsf{Val}(\rho) = 1$ if ρ has both even and odd values, and 0 otherwise. Notice that Val is present focused and $\mathcal{A}$ is history deterministic but not determinizable by pruning.

We now apply these results to specific value functions.

▶ **Theorem 21.** [6] *For nondeterministic* Sum *and* Avg *automata (on finite words), history determinism $\iff$ determinizability by pruning.*

Proof. From Lemma 16 and Theorem 19 and the suffix monotonicity of these value functions. ◀

We continue with showing that DSum is present focused due to the function's continuity.

▶ **Lemma 22.** DSum *on infinite sequences is a present-focused value function.*

▶ **Theorem 23** ([17, Section 5])**.** *For nondeterministic* DSum *automata on finite and infinite words, history determinism $\iff$ determinizability by pruning.*

Proof. The claim, which was also proved in [17, Section 5], is a direct consequence of Lemmas 16 and 22 and Theorem 19 and the suffix monotonicity of the DSum value functions. ◀

The Inf and Sup value function are not suffix monotonic, and indeed the proof of Theorem 19 does not hold for them – not every cautious transition of a history deterministic Sup automaton on finite words can be used for pruning it into a deterministic automaton. Yet, also for Inf and Sup automata on finite words we have that history determinism is equivalent to determinizability by pruning, using other characteristics of these value functions – we can prune the automaton, by choosing the transitions that are used by the strategy of the letter game after reading words with minimal values for Sup and maximal value for Inf.

▶ **Theorem 24.** *For nondeterministic* Inf *and* Sup *automata on finite words, history determinism $\iff$ determinizability by pruning.*

[6] A slightly weaker result is given in [3, Theorem 5.1]: a Sum automaton is history deterministic *with a finite-memory strategy for resolving the nondeterminism* if and only if it is determinizable by pruning.

5 Applications to Quantitative Synthesis

Establishing the non-equivalence of history determinism, good for gameness and their threshold versions leaves us with the question of which definitions, if any, are the most useful or interesting ones. We explore this question from the perspective of quantitative synthesis.

In the Boolean setting, Church's classical synthesis problem asks for a transducer $\mathcal{T}$ that produces, letter by letter, for every input sequence $I \in \Sigma_I^\omega$ an output sequence $\mathcal{T}(I) \in \Sigma_O^\omega$ such that $I \otimes \mathcal{T}(I) \in L$ for some specification language $L \in (\Sigma_I \otimes \Sigma_O)^\omega$. This synthesis requirement is *global*, in the sense that the output of all input sequences should satisfy the same constraint. A *local* variant of the problem, termed "good enough synthesis" in [1], considers each input sequence I separately, requiring that the output $\mathcal{T}(I)$ of the transducer on the input I satisfies $I \otimes \mathcal{T}(I) \in L$ only if $I \otimes O \in L$ for some sequence $O \in \Sigma_O^\omega$.

In quantitative synthesis, the specification is a function $f : (\Sigma_I \times \Sigma_O)^\omega \to \mathbb{R}$ (generalizing languages $L : (\Sigma_I \times \Sigma_O)^\omega \to \{\mathtt{true}, \mathtt{false}\}$), and the two synthesis problems above naturally generalize into two quantitative variants each – requiring either the best possible value or a value matching a given threshold. We thus have four variants of quantitative synthesis: Global/Local Threshold/Best-value synthesis. It turns out that **good for gameness is closely related to global synthesis, while history determinism is closely related to local synthesis**, both for the threshold and best-value settings.

Global Threshold and Best-value Synthesis. The global threshold variant is the closest to Church synthesis: given a function f and a threshold $t \in \mathbb{R}$, it requires that $f(I \otimes \mathcal{T}(I)) \geq t$ for all input sequences I. In the best-value version, t is not given and we are interested in what is the highest threshold that the system can guarantee.

Analogously to the Boolean setting, a t-threshold good for games Val automaton $\mathcal{A}$ realizing f can be used instead of a deterministic automaton to solve the global threshold synthesis problem: $\mathcal{A}$ is turned into a t-threshold Val game $G_\mathcal{A}$, in which Adam controls the input letters and Eve controls the output letters. Then, the synthesis problem is realizable if and only if Eve has a winning strategy in $G_\mathcal{A}$. If $\mathcal{A}$ is nondeterministic, Eve's winning strategy in $G_\mathcal{A}$ induces a transducer for the synthesis problem. In the best-value case, the same is true, but $\mathcal{A}$ must be good for games, rather than just for t-threshold games, and it is Eve's optimal strategy, if it exists, that induces the solution transducer.

Local Best-value and Threshold Synthesis. We define $\mathtt{Best}_f(I) = \sup_{O \in \Sigma_O^\omega} f(I \otimes O)$ for $I \in \Sigma_I^\omega$, i.e., the best value that the input I can get, or converge to, according to f. The local best value synthesis problem requires that for every $I \in \Sigma_I^\omega$, we have $f(I \otimes \mathcal{T}(I)) = \mathtt{Best}_f(I)$. Since $\mathtt{Best}_f(I)$ is a supremum, it need not be attained by any word; then the synthesis problem is unrealisable, even if the system could force a value arbitrarily close to $\mathtt{Best}_f(I)$. The threshold variant requires that for every $I \in \Sigma_I^\omega$, such that $\mathtt{Best}_f(I) \geq t$, we have $f(I \otimes \mathcal{T}(I)) \geq t$, for a given threshold $t \in \mathbb{R}$.

The local best value (or t-threshold) synthesis problem of a function given by deterministic (or even history-deterministic nondeterministic) automata and the problem of whether a nondeterministic automaton is (t-threshold) history deterministic reduce to each other. The relationship between good-enough synthesis [1] and history determinism was noted for visibly pushdown automata in [13]; a similar reduction in [12] reduces the approximative local best-value synthesis of deterministic quantitative automata over finite words by finite transducers to the notion of r-regret determinisability, that is, whether a nondeterministic automaton is close enough to a deterministic automaton obtained by pruning its product with a finite memory. Our reductions are in the same spirit, but relate the synthesis problem to history determinism rather than determinisability, and obtain a two-way correspondence for all history-deterministic nondeterministic quantitative automata. In the alternating case, only one direction is preserved, and only for realisability, rather than synthesis.

▶ **Proposition 25.** *Deciding the local best value (resp. t-threshold) synthesis problem with respect to a function f given by a (t-threshold) history deterministic nondeterministic* Val*-automaton $\mathcal{A}$ and deciding whether a nondeterministic* Val*-automaton $\mathcal{A}'$ is (t-threshold) history deterministic are linearly inter-reducible. Furthermore, the witness of (t-threshold) history determinism of $\mathcal{A}'$ is implementable by the same computational models as a solution to the best-value (t-threshold) synthesis of $\mathcal{A}$.*

6 Conclusions

We have painted a picture of how definitions of good for gameness and history determinism behave in the quantitative setting, and how they relate to quantitative synthesis. Our work opens up many directions for further work, of which we name a few.

- The reductions between local synthesis and history determinism motivate expanding methods used to decide history determinism of ω-regular automata to quantitative ones.
- So far, we have restricted our attention to determined games, but one could also consider more general classes of games and study the effect of composition in that setting.
- One appeal of good for games and history deterministic automata is that they can be more expressive and more succinct than deterministic ones, while their synthesis problems retain the same complexity. The expressivity and succinctness of quantitative good for games and history deterministic automata is open for most value functions.
- It is natural to look at approximative versions of the discussed notions (as has been done, see the related work section); we expect our results to also generalise in that direction.

References

1. Shaull Almagor and Orna Kupferman. Good-enough synthesis. In *CAV*, volume 12225 of *Lecture Notes in Computer Science*, pages 541–563. Springer, 2020.
2. Shaull Almagor, Orna Kupferman, Jan Oliver Ringert, and Yaron Velner. Quantitative assume guarantee synthesis. In *International Conference on Computer Aided Verification*, pages 353–374. Springer, 2017.
3. Benjamin Aminof, Orna Kupferman, and Robby Lampert. Reasoning about online algorithms with weighted automata. *ACM Trans. Algorithms*, 6(2):28:1–28:36, 2010.
4. Roderick Bloem, Krishnendu Chatterjee, Thomas A Henzinger, and Barbara Jobstmann. Better quality in synthesis through quantitative objectives. In *International Conference on Computer Aided Verification*, pages 140–156. Springer, 2009.
5. Udi Boker. Quantitative vs. weighted automata. In *Proc. of Reachbility Problems*, pages 1–16, 2021.
6. Udi Boker and Guy Hefetz. Discounted-sum automata with multiple discount factors. In *Proc. of CSL*, volume 183 of *LIPIcs*, pages 12:1–12:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
7. Udi Boker, Denis Kuperberg, Orna Kupferman, and Michał Skrzypczak. Nondeterminism in the presence of a diverse or unknown future. In *Proceedings of ICALP*, pages 89–100, 2013.
8. Udi Boker and Karoliina Lehtinen. Good for games automata: From nondeterminism to alternation. In *Proceedings of CONCUR*, volume 140 of *LIPIcs*, pages 19:1–19:16, 2019.

9 Romain Brenguier, Lorenzo Clemente, Paul Hunter, Guillermo A Pérez, Mickael Randour, Jean-François Raskin, Ocan Sankur, and Mathieu Sassolas. Non-zero sum games for reactive synthesis. In *Language and Automata Theory and Applications*, pages 3–23. Springer, 2016.

10 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Alternating weighted automata. In *Proceedings of FCT*, pages 3–13, 2009.

11 Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Proceedings of ICALP*, pages 139–150, 2009.

12 Emmanuel Filiot, Christof Löding, and Sarah Winter. Synthesis from weighted specifications with partial domains over finite words. In Nitin Saxena and Sunil Simon, editors, *FSTTCS*, volume 182 of *LIPIcs*, pages 46:1–46:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

13 Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. A bit of nondeterminism makes pushdown automata expressive and succinct. In *MFCS*, volume 202 of *LIPIcs*, pages 53:1–53:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.MFCS.2021.53`.

14 Thomas Henzinger and Nir Piterman. Solving games without determinization. In *Proceedings of CSL*, pages 395–410, 2006.

15 Paul Hunter, Arno Pauly, Guillermo A. Pérez, and Jean-François Raskin. Mean-payoff games with partial observation. *Theor. Comput. Sci.*, 735:82–110, 2018.

16 Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Reactive synthesis without regret. In Luca Aceto and David de Frutos-Escrig, editors, *CONCUR*, volume 42 of *LIPIcs*, pages 114–127, 2015.

17 Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Minimizing regret in discounted-sum games. In Jean-Marc Talbot and Laurent Regnier, editors, *CSL*, volume 62 of *LIPIcs*, pages 30:1–30:17, 2016.

18 Paul Hunter, Guillermo A. Pérez, and Jean-François Raskin. Reactive synthesis without regret. *Acta Informatica*, 54(1):3–39, 2017.

19 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.

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

21 Gila Morgenstern. Expressiveness results at the bottom of the ω-regular hierarchy. M.Sc. Thesis, The Hebrew University, 2003.

22 Wolfgang Thomas. Languages, automata, and logic. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages, Volume 3: Beyond Words*, pages 389–455. Springer, 1997.

A Proofs of Section 4

Proof of Lemma 5. One direction is immediate: if an automaton $\mathcal{A}$ is good for all games then it is also good for all threshold games. Indeed, assuming that $\mathcal{A}$ is good for games, if the value of a game G is v, then the value of the product game $G \times \mathcal{A}$ is also v. Then, for all thresholds t, Eve wins the t-threshold game on both G and $G \times \mathcal{A}$ if and only if $v \geq t$.

As for the other direction, assume $\mathcal{A}$ is good for threshold games. Let G be a game with value v. Since $\mathcal{A}$ composes with threshold games, considering the v-threshold game on G, we know that Eve can achieve at least v in the product v-threshold game $G \times A$. Conversely, let $v' \geq v$ be the value of $G \times A$. Since Eve wins the v'-threshold game on $G \times A$, and $\mathcal{A}$ is good for threshold games, Eve can also achieve at least v' in G, i.e., $v' = v$, the value of G. ◀

Proof of Lemma 6. Consider a threshold history deterministic automaton $\mathcal{A}$ over an alphabet Σ, realizing a function f. Then for every threshold $t \in \mathbb{R}$, Eve has a winning strategy s' in the t-threshold letter game on $\mathcal{A}$.

Now, consider a Σ-labelled t-threshold game G with payoff function f, in which Eve has a winning strategy s. Then in the product game $G \times \mathcal{A}$, Eve can combine s and s' into a strategy $\hat{s}$, so that s guarantees that any play $\pi = (w, \rho)$ that agrees with $\hat{s}$ reads a word w such that $\mathcal{A}(w) \geq t$, and s' guarantees that $\mathsf{Val}(\rho) \geq t$ (since $\mathcal{A}(w) \geq t$).

By a similar argument, if Adam has a winning strategy in his threshold letter game, he can combine it with his winning strategy in a threshold game for getting a winning strategy in the product threshold game. ◀

A.1 Proofs of Section 4.1

Proof of Lemma 22. Consider a λ-DSum Val automaton $\mathcal{A}$ and let m be the maximal absolute transition weight in $\mathcal{A}$. Observe that for every word w and state q of $\mathcal{A}$, we have $|\mathcal{A}^q(w)| \leq \frac{m}{1-\lambda}$.

Let s be a cautious strategy of Eve in the letter game on $\mathcal{A}$. By the definition of a cautious strategy, for every finite word u, playing according to s on u generates a finite run ρ that ends in some state q, such that for every infinite word v, there is an infinite run π on v from q, such that $\mathsf{Val}(\rho\pi) = \mathcal{A}(uv)$.

Now, consider a word w, let r be the run of $\mathcal{A}$ on w that is generated by following s, and let r' be an optimal run of $\mathcal{A}$ on w. For every position i, let q_i be the state that $r[0..i]$ ends in and q'_i be the state that $r'[0..i]$ ends in. By the cautiousness of s, for every position i, there is a run π from q_i on $w[i+1..]$, such that for every run π' from q'_i on $w[i+1..]$, we have $\mathsf{Val}(r[0..i]) + \lambda^i \mathsf{Val}(\pi) \geq \mathsf{Val}(r'[0..i]) + \lambda^i \mathsf{Val}(\pi')$.

Since $\mathsf{Val}(\pi) \leq \frac{m}{1-\lambda}$ and $\mathsf{Val}(\pi') \geq -\frac{m}{1-\lambda}$, we get that $\mathsf{Val}(r'[0..i]) - \mathsf{Val}(r[0..i]) \leq \frac{2m \cdot \lambda^i}{1-\lambda}$. Since $\lim_{i\to\infty} \frac{2m\cdot\lambda^i}{1-\lambda} = 0$, we get that $\mathsf{Val}(r) = \mathsf{Val}(r')$, implying that Eve wins the letter game. ◀

Proof of Theorem 24. We provide the proof for Sup automata and then describe the required changes for adapting it to Inf automata.

Consider a history deterministic Sup automaton $\mathcal{A}$ on finite words in Σ^*, whose history determinism is witnessed by a strategy s. We derive from s a positional strategy s', by taking for every state q of $\mathcal{A}$ and letter $\sigma \in \Sigma$, the transition that s chooses over a minimal prefix, where minimality is with respect to the Sup function.

Formally, for every state q, let $m(q)$ be a Sup-minimal run that reaches q along s; namely $m(q) = \rho$, such that ρ is a run of $\mathcal{A}$ that agrees with s and ends in q, and such that for every run ρ' of $\mathcal{A}$ that agrees with s and ends in q, we have $\mathsf{Sup}(\rho) \leq \mathsf{Sup}(\rho')$. (Notice that since there are finitely many weights in $\mathcal{A}$, such a minimal run, which need not be unique, always exists.) For every state q of $\mathcal{A}$ and letter $\sigma \in \Sigma$, we define $s'(q, \sigma) = t$, such that s chooses t over the prefix run $m(q)$ and current letter σ.

We claim that s' is cautious. Indeed, for the correctness proof, we shall change s into s' iteratively, considering in each iteration a single state q and letter σ. Assume by way of contradiction that exists a word $u \in \Sigma^*$ on which s' generates a path τ that ends in a state q, such that $s'(u\sigma) = t$ for a non-cautious transition t. Without loss of generality, we may assume that this is not the case for any strict prefix of u, as otherwise we can consider that prefix instead of u.

By the definition of non-cautiousness, there exists a word w, such that the maximal value of $\mathsf{Sup}(\tau\pi)$ for a run π from q over σw starting with t is strictly smaller than the maximal value of $\mathsf{Sup}(\tau\pi')$ where π' is a run from q over σw that does not start with t.

It thus follows that $\mathsf{Sup}(\pi') > \mathsf{Sup}(\tau)$ and that for every run π from q over σw starting with t, we have $\mathsf{Sup}(\pi') > \mathsf{Sup}(\pi)$. Now, let ρ be a run that witnesses t's minimality in the definition of s', namely s chooses t when reading σ after reaching q over ρ, and for every run ρ' that ends in q, we have $\mathsf{Sup}(\rho) \leq \mathsf{Sup}(\rho')$.

Then, in particular, $\mathsf{Sup}(\rho) \leq \mathsf{Sup}(\tau)$. Hence, $\mathsf{Sup}(\pi') > \mathsf{Sup}(\rho)$. Therefore, for every run π from q over σw starting with t, we have $\mathsf{Sup}(\rho\pi') > \mathsf{Sup}(\rho\pi)$, contradicting the cautiousness of s.

Having that s' is cautious, we get from Lemma 16 that it is also winning in the letter game, implying that the deterministic automaton that results from pruning $\mathcal{A}$ along s' is indeed equivalent to $\mathcal{A}$.

Now, for Inf automata, the proof is analogous, choosing the Inf-maximal run rather than the Sup-minimal run, switching between some $\geq$ and $\leq$ and between some $<$ and $>$, and providing the following final argument: For every run π from q over σw starting with t, we have $\mathsf{Inf}(\pi) < \mathsf{Inf}(\tau)$ and $\mathsf{Inf}(\pi) < \mathsf{Inf}(\pi')$. Now, let ρ be a run that witnesses t's maximality in the definition of s', namely s chooses t when reading σ after reaching q over ρ, and for every run ρ' that ends in q, we have $\mathsf{Inf}(\rho) \geq \mathsf{Inf}(\rho')$.

Then, in particular, $\mathsf{Inf}(\rho) \geq \mathsf{Inf}(\tau)$. Hence, for every run π from q over σw starting with t, we have $\mathsf{Inf}(\pi) < \mathsf{Inf}(\rho)$ and $\mathsf{Inf}(\pi) < \mathsf{Inf}(\pi')$. Hence, for every run π from q over σw starting with t, we have $\mathsf{Inf}(\rho\pi) < \mathsf{Inf}(\rho\pi')$, contradicting the cautiousness of s. ◀

B Proofs of Section 5

Proof of Proposition 25.

$\Longrightarrow$: Reducing the synthesis problem to the history-determinism problem.

The idea of the reduction (both in the best-value and t-threshold case) is to turn output letter choices in $\mathcal{A}$ into nondeterministic choices in $\mathcal{A}'$. Then $\mathcal{A}'$ maps $I \in \Sigma_I^\omega$ onto $\mathtt{Best}_\mathcal{A}(I)$. A solution to the synthesis problem for $\mathcal{A}$ corresponds exactly to a function that resolves the nondeterminism of $\mathcal{A}'$ on the fly to build a run with value $\mathtt{Best}_\mathcal{A}(I)$, that is, a witnesses of the history determinism of $\mathcal{A}'$. If $\mathcal{A}$ is itself nondeterministic, then $\mathcal{A}'$ will have both the nondeterminism of $\mathcal{A}$ and the nondeterminism that stems from the choice of output letters. As long as the nondeterminism of $\mathcal{A}$ is history deterministic, the nondeterminism of $\mathcal{A}'$ is history deterministic if and only if $\mathcal{A}$ is local best value realisable.

More formally, first let us define formally the projection of $\mathcal{A}$ onto its first component: $\mathcal{A}' = (\Sigma_I, Q, \iota, \delta')$, where $\delta'(q,a) = \bigvee_{b\in\Sigma_O} \delta(q,(a,b))$. In other words, the automaton $\mathcal{A}'$ moves the Σ_O letters from the input word into a nondeterministic choice. It implements a mapping of inputs $I \in \Sigma_I^\omega$ onto $\mathtt{Best}_\mathcal{A}(I)$. We now argue that witnesses of history determinism for $\mathcal{A}'$ coincide with solutions to the best-value synthesis problem for $\mathcal{A}$. Let s be the witness of the history determinism of $\mathcal{A}$.

We first argue that a solution s' to the best-value synthesis problem for $\mathcal{A}$, combined with s is a witness that $\mathcal{A}'$ is history-deterministic. Indeed, in Eve's letter game on $\mathcal{A}'$, Eve has two types of choices: a choice $\bigvee_{b\in\Sigma_O} \delta(q,(a,b))$ of an Σ_O-letter, and the choice in $\delta(q,(a,b))$ that stems from $\mathcal{A}$. Let $\hat{s}$ be the strategy that after a run prefix ρ ending in a state q over a word $w \in \Sigma_I$ chooses the letter $s'(w)$, that is, the disjunct $\delta(q,(a,s'(w)))$ in the disjunction $\bigvee_{b\in\Sigma_O} \delta(q,(a,b))$. Then, from $\delta(q,(a,s'(w)))$, $\hat{s}$ behaves as s would after a run of $\mathcal{A}$ over $w \otimes \overline{s}(w)$.

First, observe that a run ρ of $\mathcal{A}'$ over $I \in \Sigma_I^\omega$, labelled with the choices of Σ_O-letters forming some $O \in \Sigma_O^\omega$, corresponds to a run of $\mathcal{A}$ over $I \otimes O$ with the same value.

Then, since s' is a solution to the best value synthesis problem, it guarantees that given an input word Σ_I, the sequence of Σ_O letters chosen by $\hat{s}$ is $\bar{s}(I)$, and $\mathcal{A}(I \otimes \bar{s}(I)) = \mathtt{Best}_{\mathcal{A}}(I)$. Then, as s witnesses the history determinism of $\mathcal{A}$, $\hat{s}$ guarantees that ρ has value $\mathcal{A}(I \otimes \bar{s}(I)$, that is, $\hat{s}$ witnesses the history determinism of $\mathcal{A}'$.

For the converse direction, assume $\mathcal{A}'$ is history deterministic, as witnessed by some strategy s. We claim that s induces a solution s' to the synthesis problem for $\mathcal{A}$ as follows: after reading an finite sequence of inputs $Ia \in \Sigma_I^*$, s has built some run ρ over I that ends in a state q, after which s resolves a disjunction $\bigvee_{b \in \Sigma_O} \delta(q, \binom{a}{b})$ by choosing some $b \in \Sigma_O$. We then set $s'(Ia) = b$. Then, as s witnesses that $\mathcal{A}'$ is history-deterministic, the run chosen by s over an input $I \in \Sigma_I^\omega$ has the value $\mathtt{Best}_{\mathcal{A}}(I)$. By construction of $\mathcal{A}'$ and s', this is the value $\mathcal{A}(I \otimes \bar{s'}(I)$, that is, s' is indeed a solution to the synthesis problem on $\mathcal{A}$. Furthermore, observe that an implementation of s also implements s' by ignoring the outputs of s that do not choose Σ_O letters, so the memory of the solution to the synthesis problem is bounded by the memory required by a witness of history determinism.

$\Longleftarrow$: Reducing the history-determinism problem to the synthesis problem.

Dually to the previous translation, we turn the nondeterminism in an automaton $\mathcal{A}$ into choices of output letters in the best-value synthesis problem. We build a deterministic automaton $\mathcal{A}'$ that is similar to $\mathcal{A}$ except that it reads both an input letter and a transition; then a transition can only be chosen if it is the second element of the input (that is, the output letter). Then $\mathcal{A}'$ maps valid runs of $\mathcal{A}$ to their value and a solution to the local best value synthesis problem of $\mathcal{A}'$ corresponds exactly to a witness of history-determinism for $\mathcal{A}$.

Formally, let $\mathcal{A}'$ be the Val automaton $(\Sigma \times \Delta, Q, \iota, \delta')$ where $\delta'(q, (a, q \xrightarrow{a:x} q')) = (x, q')$ if $(x, q') \in \delta(q, a)$. $\mathcal{A}'$ maps valid runs of $\mathcal{A}$ written as pairs (w, r) where r is a run of $\mathcal{A}$ over w, onto $\mathsf{Val}(r)$ and in particular $\mathtt{Best}_{\mathcal{A}'}(I) = \mathcal{A}(I)$.

We claim that $\mathcal{A}'$ is best-value realisable if and only $\mathcal{A}$ is history-deterministic. Indeed, a solution s to the best value synthesis problem of $\mathcal{A}'$ corresponds to a function building a run of $\mathcal{A}$ over the input I transition by transition such that the value of the run is $\mathtt{Best}_{\mathcal{A}'}(I)$. Since $\mathtt{Best}_{\mathcal{A}'}(I)$ is $\mathcal{A}(I)$, s is precisely a witness of history-determinism in $\mathcal{A}$. Similarly, a witness of history-determinism in $\mathcal{A}$ induces a solution to the best value synthesis problem for $\mathcal{A}'$ since it builds a run of $\mathcal{A}$ over I with value at least $\mathcal{A}(I)$, exactly what is required from a solution to the best value synthesis. ◀