On Delay and Regret Determinization of Max-Plus Automata

Emmanuel Filiot*, Ismaël Jecker*, Nathan Lhote*†, Guillermo A. Pérez*, Jean-François Raskin*
*Université Libre de Bruxelles
{efiliot, ijecker, gperezme, jraskin}@ulb.ac.be
†Université de Bordeaux, LaBRI
nlhote@labri.fr

***Abstract*—Decidability of the determinization problem for weighted automata over the semiring $(\mathbb{Z} \cup \{-\infty\}, \max, +)$, WA for short, is a long-standing open question. We propose two ways of approaching it by constraining the search space of deterministic WA: k-delay and r-regret. A WA $\mathcal{N}$ is k-delay determinizable if there exists a deterministic automaton $\mathcal{D}$ that defines the same function as $\mathcal{N}$ and for all words α in the language of $\mathcal{N}$, the accepting run of $\mathcal{D}$ on α is always at most k-away from a maximal accepting run of $\mathcal{N}$ on α. That is, along all prefixes of the same length, the absolute difference between the running sums of weights of the two runs is at most k. A WA $\mathcal{N}$ is r-regret determinizable if for all words α in its language, its non-determinism can be resolved on the fly to construct a run of $\mathcal{N}$ such that the absolute difference between its value and the value assigned to α by $\mathcal{N}$ is at most r.**

We show that a WA is determinizable if and only if it is k-delay determinizable for some k. Hence deciding the existence of some k is as difficult as the general determinization problem. When k and r are given as input, the k-delay and r-regret determinization problems are shown to be EXPTIME-complete. We also show that determining whether a WA is r-regret determinizable for some r is in EXPTIME.

***Index Terms*—Weighted Automata, Determinization, Regret**

I. INTRODUCTION

Weighted automata. Weighted automata generalize finite automata with weights on transitions [1]. They generalize word languages to partial functions from words to values of a semiring. First introduced by Schützenberger and Chomsky in the 60s, they have been studied for long [1], with applications in natural language and image processing for instance. More recently, they have found new applications in computer-aided verification as a measure of system quality through quantitative properties [2], and in system synthesis, as objectives for quantitative games [3]. In this paper, we consider weighted automata $\mathcal{N}$ over the semiring $(\mathbb{Z} \cup \{-\infty\}, \max, +)$, and just call them weighted automata (WA). The value of a run is the sum of the weights occurring on its transitions, and the value of a word is the maximal value of all its accepting runs. Absent transitions have a weight of $-\infty$ and runs of value $-\infty$ are considered non-accepting. This defines a partial function denoted $[\![\mathcal{N}]\!] : \Sigma^* \to \mathbb{Z}$ whose domain is denoted by $\mathcal{L}_{\mathcal{N}}$.

This work was partially supported by the ERC Starting grant 279499 (inVEST), the ARC project Transform (Fédération Wallonie-Bruxelles), the Belgian FNRS CDR project Flare, and the French ANR ExStream project (ANR-13-JS02-0010). E. Filiot is an F.R.S.-FNRS research associate, I. Jecker an F.R.S.-FNRS Aspirant fellow, G. A. Pérez an F.R.S.-FNRS Aspirant and FWA post-doc fellow, J.-F. Raskin is Professeur Francqui de Recherche.

978-1-5090-3018-7/17/$31.00

Determinization problem. Most of the good algorithmic properties of finite automata do not transfer to WA. Notably, the (quantitative) language inclusion $[\![\mathcal{A}]\!] \leq [\![\mathcal{B}]\!]$ is undecidable for WA [4] (see also [5] and [6] for different proofs based on reductions from the halting problem for two-counter machines). This has triggered research on sub-classes or other formalisms for which this problem becomes decidable [7], [8]. This includes the class of deterministic WA (DWA, also known as sequential WA in the literature), which are the WA whose underlying (unweighted) automaton is deterministic. Another scenario where it is desirable to have a DWA is the quantitative synthesis problem, undecidable even for unambiguous WA, yet decidable for DWA [3]. However, and in contrast with finite unweighted automata, WA are not determinizable in general. For instance, the function which outputs the maximal value between the number of a's and the number of b's in a word $\alpha \in \{a, b\}^*$ is not realizable with a DWA. This motivates the determinization problem: given a WA $\mathcal{N}$, is it determinizable? I.e. is there a DWA defining the same (partial) function as $\mathcal{N}$?

The determinization problem for computational models is fundamental in theoretical computer science. For WA in particular, it is sometimes more natural (and at least exponentially more succinct) to specify a (non-deterministic) WA, even if some equivalent DWA exists. If the function is specified in a weighted logic equivalent to WA, such as weighted MSO [9], the logic-to-automata transformation may construct a non-deterministic, but determinizable, WA. However, despite many research efforts, the largest class for which this problem is known to be decidable is the class of polynomially ambiguous WA [10], and the decidability status for the full class of WA is a long-standing open problem. Other contributions and approaches to the determinization problem include the identification of sufficient conditions for determinizability [11], approximate determinizability (for unambiguous WA) where the DWA is required to produce values at most t times the value of the WA, for a given factor t [12], and (incomplete) approximation algorithms when the weights are non-negative [13].

Bounded-delay & regret determinizers. In this paper, we adopt another approach that consists in constraining the class of DWA that can be used for determinization. More precisely, we define a *class* of DWA $\mathfrak{C}$ as a function from WA to sets of DWA, and say that a DWA $\mathcal{D}$ is a $\mathfrak{C}$-determinizer of a WA $\mathcal{N}$ if (i) $\mathcal{D} \in \mathfrak{C}(\mathcal{N})$ and (ii) $[\![\mathcal{N}]\!] = [\![\mathcal{D}]\!]$. Then, $\mathcal{N}$ is said to be $\mathfrak{C}$-determinizable if it admits a $\mathfrak{C}$-determinizer. If $\mathbf{DWA}$ denotes the function mapping any WA to the whole set of DWA, then obviously the $\mathbf{DWA}$-determinization problem is the general (open) determinization problem. In this paper, we consider two restrictions.

First, given a bound $k \in \mathbb{N}$, we look for the class of k-delay DWA $\mathbf{Del}_k$, which maintain a strong relation with the sequence of values along some accepting run of the non-deterministic automaton. More precisely, a DWA $\mathcal{D}$ belongs to $\mathbf{Del}_k(\mathcal{N})$ if for all words $\alpha \in \mathcal{L}_\mathcal{D}$, there is an accepting run ϱ of $\mathcal{N}$ with maximal value such that the running sum of the prefixes of ϱ and the running sum of the prefixes of the unique run $\varrho_\mathcal{D}$ of $\mathcal{D}$ on α are constantly close in the following sense: for all lengths ℓ, the absolute value of the difference of the value of the prefix of ϱ of length ℓ and the value of the prefix of $\varrho_\mathcal{D}$ of length ℓ is at most k. Then the $\mathbf{Del}_k$-determinization problem amounts to deciding whether there exists $\mathcal{D} \in \mathbf{Del}_k(\mathcal{N})$ such that $[\![\mathcal{N}]\!] = [\![\mathcal{D}]\!]$. And if k is left unspecified, it amounts to decide whether there exists $\mathcal{D} \in \bigcup_{k\in\mathbb{N}} \mathbf{Del}_k(\mathcal{N})$ such that $[\![\mathcal{N}]\!] = [\![\mathcal{D}]\!]$. We note $\mathbf{Del}$ the function mapping any WA $\mathcal{N}$ to $\bigcup_k \mathbf{Del}_k(\mathcal{N})$. We will show that the $\mathbf{Del}_k$-determinization problem is complete for EXPTIME, and the $\mathbf{Del}$-determinization problem is equivalent to the general (*unconstrained*) determinization problem.

The notion of delay has been central in many works on automata with outputs. For instance, it has been a key notion in transducer theory (automata with word outputs) for the determinization and the functionality problems [14], and the decomposition of finite-valued transducers [15]. The notion of delay has been also used in the theory of WA, for instance to give sufficient conditions for determinizability [11] or for the decomposition of finite-valued group automata [7].

Example 1. Let $A = \{a, b\}$ and $k \in \mathbb{N}$. The left automaton of Fig. 1 maps the words $\{a, aa, bb\}$ to k, and the words $\{b, ab, ba\}$ to $-k$. It is $\mathbf{Del}_{2k}$-determinizable by the right automaton of Fig. 1. With words of length 1, the delay of the DWA is 0 with the only accepting run of the left WA. With words of length 2, it may be $2k$. The NWA is not $\mathbf{Del}_j$-determinizable for any $j < 2k$.

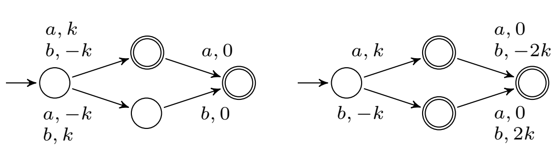

Fig. 1. Another WA (left) and one of its $2k$-delay determinizers (right)

Second, we consider the class $\mathbf{Hom}$ of so-called homomorphic DWA. Intuitively, any DWA $\mathcal{D} \in \mathbf{Hom}(\mathcal{N})$ maintains a close relation with the structure of $\mathcal{N}$: the existence of an homomorphism from $\mathcal{D}$ to $\mathcal{N}$. An alternative definition is that of a 0-*regret game* [16] played on $\mathcal{N}$: Adam chooses input symbols one by one (forming a word $\alpha \in \mathcal{L}_\mathcal{N}$), while Eve reacts by choosing transitions of $\mathcal{N}$, thus constructing a run ϱ of $\mathcal{N}$ on the fly (*i.e.* without knowing the full word α in advance). Eve wins the game if ϱ is accepting and its value is equal to $[\![\mathcal{N}]\!](\alpha)$, *i.e.* ϱ is a maximal accepting run on α. Then, any (finite memory) winning strategy for Eve can be seen as a $\mathbf{Hom}$-determinizer of $\mathcal{N}$ and conversely. This generalizes the notion of good-for-games automata, which do not need to be determinized prior to being used as observers in a game, from the Boolean setting [17] to the quantitative one. In some sense, $\mathbf{Hom}$-determinizable WA are "good for quantitative games": when used as an observer in a quantitative game, Eve's strategy can be applied on the fly instead of determinizing the WA and constructing the synchronized product of the resulting DWA with the game arena. This notion has been first introduced in [18] with motivations coming from the analysis of online algorithms. In [18], it was shown that the $\mathbf{Hom}$-determinization problem is in PTIME.

Example 2. The following WA maps both ab and aa to 0. It is not $\mathbf{Hom}$-determinizable because Eve has to choose whether to go left or right on reading a. If she goes right, then Adam wins by choosing letter b. If she goes left, Adam wins by picking a again. However, it is *almost* $\mathbf{Hom}$-determinizable by the DWA obtained by removing the right part, in the sense that the function realized by this DWA is 1-close from the original one. This motivates approximate determinization.

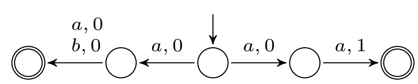

Fig. 2. A WA that is $\mathbf{Del}_0$-determinizable and $(1, \mathbf{Hom})$-determinizable, but not $\mathbf{Hom}$-determinizable

Approximate determinization. Approximate determinization of a WA $\mathcal{N}$ relaxes the determinization problem to determinizers which do not define exactly the same function as $\mathcal{N}$ but approximate it. Precisely, for a class $\mathfrak{C}$ of DWA, $\mathcal{D}$ a DWA and $r \in \mathbb{N}$, we say that $\mathcal{D}$ is an $(r, \mathfrak{C})$-determinizer of $\mathcal{N}$ if (i) $\mathcal{D} \in \mathfrak{C}(\mathcal{N})$, (ii) $\mathcal{L}_\mathcal{D} = \mathcal{L}_\mathcal{N}$ and (iii) for all words $\alpha \in \mathcal{L}_\mathcal{N}$, $|[\![\mathcal{N}]\!](\alpha) - [\![\mathcal{D}]\!](\alpha)| \leq r$. Then, $\mathcal{N}$ is $(r, \mathfrak{C})$-determinizable if it admits some $(r, \mathfrak{C})$-determinizer, and it is *approximately* $\mathfrak{C}$-determinizable if it is $(r, \mathfrak{C})$-determinizable for some r.

As Example 2 shows, there are WA that are approximately $\mathbf{Hom}$-determinizable but not $\mathbf{Hom}$-determinizable, making this notion appealing for the class of homomorphic determinizers. However, there are classes $\mathfrak{C}$ for which a WA is approximately $\mathfrak{C}$-determinizable if and only if it is $\mathfrak{C}$-determinizable, making approximate determinization much less interesting for such classes. This is the case for classes $\mathfrak{C}$ which are *complete for determinization* (Theorem 1), in the sense that any determinizable WA is also $\mathfrak{C}$-determinizable. Obviously, the class $\mathbf{DWA}$ is complete for determinization, but we show it is also the case for the class of bounded-delay determinizers $\mathbf{Del}$ (Theorem 2). Therefore, we study approximate determinization for the class of homomorphic determinizers only. We call such determinizers r-regret determinizers, building on the regret game analogy given above. Indeed, a WA $\mathcal{N}$ is $(r, \mathbf{Hom})$-determinizable if and only if Eve wins the regret game previously defined, with the following modified winning condition: the run that she constructs on the fly must be such that $|[\![\mathcal{N}]\!](\alpha) - [\![\mathcal{D}]\!](\alpha)| \leq r$, for all words $\alpha \in \mathcal{L}_{\mathcal{N}}$ that Adam can play. We say that a WA $\mathcal{N}$ is approximately $\mathbf{Hom}$-determinizable if there exists a $r \in \mathbb{N}$ such that $\mathcal{N}$ is $(r, \mathbf{Hom})$-determinizable.

Contributions. We show that the $\mathbf{Del}_k$-determinization problem problem is EXPTIME-complete, even when k is fixed (Theorems 3 and 4). We also show that the class $\mathbf{Del}$ is complete for determinization, *i.e.* any determinizable WA $\mathcal{N}$ is k-delay determinizable for some k (Theorem 2). This shows that solving the $\mathbf{Del}$-determinization problem would solve the (open) general determinization problem. This also gives a new (complete) semi-algorithm for determinization, which consists in testing for the existence of k-delay determinizers for increasing values k. We exhibit a family of bounded-delay determinizable WA, for delays which depend exponentially on the WA. Despite our efforts, exponential delays are the highest lower bound we have found. Interestingly, finding higher lower bounds would lead to a better understanding of the determinization problem, and proving that one of these lower bound is also an upper bound would immediately give decidability. To decide $\mathbf{Del}_k$-determinization, we provide a reduction to $\mathbf{Hom}$-determinization (*i.e.* 0-regret determinization), which is known to be decidable in polynomial time [18].

We show that the approximate $\mathbf{Hom}$-determinization problem is decidable in exponential time (Theorem 7), a problem which was left open in [18]. This result is based on a non-trivial extension to the quantitative setting of a game tool proposed by Kuperberg and Skrzypczak in [19] for Boolean automata. In particular, our quantitative extension is based on energy games [20] while parity games are sufficient for the Boolean case. If r is given (in binary) the $(r, \mathbf{Hom})$-determinization problem is shown to be EXPTIME-complete (Theorems 5 and 8). The hardness holds even if r is given in unary. In the course of establishing our results, we also show that every WA A that is approximately $\mathbf{Hom}$-determinizable is also *exactly* determinizable but there may not be a homomorphism from a deterministic version of the automaton to the original one (Lemma 22 and Theorem 6). Hence, the decision procedure for approximate $\mathbf{Hom}$-determinizability can also be used as an algorithmically verifiable sufficient condition for determinizability.

Other related works. In transducer theory, a notion close to the notion of k-delay determinizer has been introduced in [21], that of k-delay uniformizers of a transducer. A uniformizer of a transducer T is an (input)-deterministic transducer such that the word-to-word function it defines (seen as a binary relation) is included in the relation defined by T, and any of its accepting runs should be k-delay close from some accepting run of T. While the notion of k-delay uniformizer in transducer theory is close to the notion of k-delay determinizer for WA, the presence of a max operation in WA makes the k-delay determinization problem conceptually harder.

II. Preliminaries

We denote by $\mathbb{Z}$ the set of all integers; by $\mathbb{N}$, the set of all non-negative integers, *i.e.* the natural numbers—including 0; by $\mathbb{N}_{>0}$ the set of all positive integers. Finally, by ε we denote the *empty word* over any alphabet.

Automata. A *(non-deterministic weighted finite) automaton* $\mathcal{N} = (Q, I, A, \Delta, w, F)$ consists of a finite set Q of states, a set $I \subseteq Q$ of initial states, a finite alphabet A of symbols, a transition relation $\Delta \subseteq Q \times A \times Q$, a weight function $w : \Delta \to \mathbb{Z}$, and a set $F \subseteq Q$ of final states. By $w_{\max}$ we denote the maximal absolute value of a transition weight. We say $\mathcal{N}$ is *pair-deterministic* if $|I| = 1$ and for all $(q, a) \in Q \times A$ we have that $(q, a, q_1), (q, a, q_2) \in \Delta$ implies $q_1 = q_2$ or $w(q, a, q_1) \neq w(q, a, q_2)$; *deterministic*, if $|I| = 1$ and $(q, a, q_1), (q, a, q_2) \in \Delta$ implies $q_1 = q_2$, for all $(q, a) \in Q \times A$.

A *run* of $\mathcal{N}$ on a word $a_0 \dots a_{n-1} \in A^*$ is a sequence $\varrho = q_0 a_0 q_1 \dots q_{n-1} a_{n-1} q_n \in (Q \cdot A)^* Q$ such that $(q_i, a_i, q_{i+1}) \in \Delta$ for all $0 \leq i < n$. We say ϱ is *initial* if $q_0 \in I$; *final*, if $q_n \in F$; *accepting*, if it is both initial and final. The automaton $\mathcal{N}$ is said to be *trim* if for all states $q \in Q$, there is a run from a state $q_I \in I$ to q and there is a run from q to some $q_F \in F$. The *value of* ϱ, denoted by $w(\varrho)$, corresponds to the sum of the weights of its transitions: $w(\varrho) := \sum_{i=0}^{n-1} w(q_i, a_i, q_{i+1})$.

The automaton $\mathcal{N}$ has the (unweighted) *language* $\mathcal{L}_{\mathcal{N}} = \{\alpha \in A^* \mid \text{there is an accepting run of } \mathcal{N} \text{ on } \alpha\}$ and defines a function $[\![\mathcal{N}]\!] : \mathcal{L}_{\mathcal{N}} \to \mathbb{Z}$ as follows $\alpha \mapsto \max\{w(\varrho) \mid \varrho \text{ is an accepting run of } \mathcal{N} \text{ on } \alpha\}$. A run ϱ of $\mathcal{N}$ on α is said to be *maximal* if $w(\varrho) = [\![\mathcal{N}]\!](\alpha)$.

Determinization with delay. Given $k \in \mathbb{N}$ and two automata $\mathcal{N} = (Q, I, A, \Delta, w, F)$ and $\mathcal{N}' = (Q', I', A, \Delta', w', F')$, we say that $\mathcal{N}$ is *k-delay-included* (or k-included, for short) in $\mathcal{N}'$, denoted by $\mathcal{N} \subseteq_k \mathcal{N}'$, if for every accepting run $\varrho = q_0 a_0 \dots a_{n-1} q_n$ of $\mathcal{N}$, there exists an accepting run $\varrho' = q'_0 a_0 \dots a_{n-1} q'_n$ of $\mathcal{N}'$ such that $w'(\varrho') = w(\varrho)$, and for every $1 \leq i \leq n$, $|w'(q'_0 \dots q'_i) - w(q_0 \dots q_i)| \leq k$. For an automaton $\mathcal{N}$, we denote by $\mathbf{Del}_k(\mathcal{N})$ the set $\{\mathcal{D} \in \mathbf{DWA} \mid \mathcal{D} \subseteq_k \mathcal{N}\}$.

An automaton $\mathcal{N}$ is said to be *k-delay determinizable* if there exists an automaton $\mathcal{D} \in \mathbf{Del}_k(\mathcal{N})$ such that $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$. Such an automaton is called a *k-delay determinizer of $\mathcal{N}$*.

Determinization with regret. Given two automata $\mathcal{N} = (Q, I, A, \Delta, w, F)$ and $\mathcal{N}' = (Q', I', A, \Delta', w', F')$, a mapping $\mu : Q \to Q'$ from states in $\mathcal{N}$ to states in $\mathcal{N}'$ is a *homomorphism from $\mathcal{N}$ to $\mathcal{N}'$* if $\mu(I) \subseteq I'$, $\mu(F) \subseteq F'$, $\{(\mu(p), a, \mu(q)) \mid (p, a, q) \in \Delta\} \subseteq \Delta'$, and $w'(\mu(p), a, \mu(q)) = w(p, a, q)$. For an automaton $\mathcal{N}$, we denote by $\mathbf{Hom}(\mathcal{N})$ the set of deterministic automata $\mathcal{D}$ for which there is a homomorphism from $\mathcal{D}$ to $\mathcal{N}$. The following lemma follows directly from the preceding definitions.

Lemma 1. *For all automata $\mathcal{N}$, for all $\mathcal{D} \in \mathbf{Hom}(\mathcal{N})$, we have that $\mathcal{L}_{\mathcal{D}} \subseteq \mathcal{L}_{\mathcal{N}}$ and $[\![\mathcal{D}]\!](\alpha) \leq [\![\mathcal{N}]\!](\alpha)$ for all $\alpha \in \mathcal{L}_{\mathcal{N}}$.*

Given $r \in \mathbb{N}$ and an automaton $\mathcal{N}$, we say $\mathcal{N}$ is *r-regret determinizable* if there is a deterministic automaton $\mathcal{D}$ such that: (i) $\mathcal{D} \in \mathbf{Hom}(\mathcal{N})$, (ii) $\mathcal{L}_{\mathcal{N}} = \mathcal{L}_{\mathcal{D}}$, and (iii) $\sup_{\alpha \in \mathcal{L}_{\mathcal{N}}} |[\![\mathcal{N}]\!](\alpha) - [\![\mathcal{D}]\!](\alpha)| \leq r$. The automaton $\mathcal{D}$ is said to be an *r-regret determinizer of $\mathcal{N}$*. Note that (i) implies we can remove the absolute value in (iii) because of Lemma 1.

Regret games. Given $r \in \mathbb{N}$ and an automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$, an *$r$-regret game* is a two-player turn-based game played on $\mathcal{N}$ by Eve and Adam. To begin, Eve chooses an initial state. Then, the game proceeds in rounds as follows. From the current state q, Adam chooses a symbol $a \in A$ and Eve chooses a new state q'. After a word $\alpha \in \mathcal{L}_{\mathcal{N}}$ has been played by Adam, he may decide to stop the game. At this point Eve loses if the current state is not final or if she has not constructed a valid run of $\mathcal{N}$ on α. Furthermore, she must pay a (*regret*) value equal to $[\![\mathcal{N}]\!](\alpha)$ minus the value of the run she has constructed.

Formally, a *strategy for Adam* is a finite word $\alpha \in A^*$ and a *strategy for Eve* is a function $\sigma : (Q \cdot A)^* \to Q$ from state-symbol sequences to states. Given a word (strategy) $\alpha = a_0 \dots a_{n-1}$, we write $\sigma(\alpha)$ to denote the sequence $q_0 a_0 \dots a_{n-1} q_n$ such that $\sigma(\varepsilon) = q_0$ and $\sigma(q_0 a_0 \dots q_i a_i) = q_{i+1}$ for all $0 \leq i < n$. The *regret of* σ is defined as follows: $\mathbf{reg}^\sigma(\mathcal{N}) := \sup_{\alpha \in \mathcal{L}_{\mathcal{N}}} [\![\mathcal{N}]\!](\alpha) - \mathsf{Val}(\sigma(\alpha))$ where, for all sequences $\varrho \in (Q \cdot A)^* Q$, the function $\mathsf{Val}(\varrho)$ is such that $\varrho \mapsto w(\varrho)$ if ϱ is an accepting run of $\mathcal{N}$ and $\varrho \mapsto -\infty$ otherwise. We say Eve wins the r-regret game played on $\mathcal{N}$ if she has a strategy such that $\mathbf{reg}^\sigma(\mathcal{N}) \leq r$. Such a strategy is said to be *winning* for her in the regret game.

Games & determinization. A *finite-memory* strategy σ for Eve in a regret game played on an automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$ is a strategy that can be encoded as a deterministic *Mealy machine* $\mathcal{M} = (S, s_I, A, \lambda_u, \lambda_o)$ where S is a finite set of (memory) states, s_I is the initial state, $\lambda_u : S \times A \to S$ is the update function and $\lambda_o : S \times (Q \times A \cup \{\varepsilon\}) \to Q$ is the output function. The machine encodes σ in the following sense: $\sigma(\varepsilon) = \lambda_o(s_I, \varepsilon)$ and $\sigma(q_0 a_0 \dots q_n a_n) = \lambda_o(s_n, q_n, a_n)$ where $s_0 = s_I$ and $s_{i+1} = \lambda_u(s_i, q_i, a_i)$ for all $0 \leq i < n$. We then say that $\mathcal{M}$ *realizes* the strategy σ and that σ has *memory* $|S|$. In particular, strategies which have memory 1 are said to be *positional* (or *memoryless*).

A finite-memory strategy σ for Eve in a regret game played on $\mathcal{N}$ defines the deterministic automaton $\mathcal{N}_\sigma$ obtained by taking the *synchronized product* of $\mathcal{N}$ and the finite Mealy machine $(S, s_I, A, \lambda_u, \lambda_o)$ realizing σ. Formally $\mathcal{N}_\sigma$ is the automaton $(Q \times S, (\lambda_o(s_I, \varepsilon), s_I), A, \Delta', w', F \times S)$ where: Δ' is the set of all triples $((q, s), a, (q', s'))$ such that $(q, s) \in Q \times S, a \in A$, $s' = \lambda_u(s, a)$, and $q' = \lambda_o(s, a)$; and w' is such that $((q, s), a, (q', s')) \mapsto w(q, a, q')$.

We remark that, for all $r \in \mathbb{N}$, for all finite-memory strategies σ for Eve such that $\mathbf{reg}^\sigma(\mathcal{N}) \leq r$, we have that $\mathcal{N}_\sigma$ is an r-regret determinizer of $\mathcal{N}$. Indeed, the desired homomorphism from $\mathcal{N}_\sigma$ to $\mathcal{N}$ is the projection on the first dimension of $Q \times S$, *i.e.* $(q, s) \mapsto q$. Furthermore, from any r-regret determinizer $\mathcal{D}$ of $\mathcal{N}$, it is straightforward to define a finite-memory strategy for Eve that is winning for her in the r-regret game.

Lemma 2. *For all $r \in \mathbb{N}$, an automaton $\mathcal{N}$ is r-regret determinizable if and only if there exists a finite-memory strategy σ for Eve such that $\mathbf{reg}^\sigma(\mathcal{N}) \leq r$.*

In [18] it was shown that if there exists a 0-regret strategy for Eve in a regret game, then a 0-regret memoryless strategy for her exists as well. Furthermore, deciding if the latter holds is in PTIME. Hence, by Lemma 2 we obtain the following.

Proposition 3 (From [18])**.** *Determining if a given automaton is 0-regret determinizable is decidable in polynomial time.*

A sufficient condition for determinizability. Given $B \in \mathbb{N}$, we say an automaton $\mathcal{N}$ is B-bounded if it is trim and for every maximal accepting run $\varrho_p = p_0 a_0 p_1 \dots a_{n-1} p_n$ of $\mathcal{N}$, for every $0 \leq i \leq n$, and for every initial run $\varrho_q = q_0 a_0 q_1 \dots a_{i-1} q_i$, we have $w(\varrho_q) - w(p_0 a_0 p_1 \dots a_{i-1} p_i) \leq B$.

We now prove that, given a B-bounded automaton, we are able to build an equivalent deterministic automaton.

Proposition 4. *Let $B \in \mathbb{N}$ and let $\mathcal{N} = (Q, I, A, \Delta, w, F)$ be an automaton. If $\mathcal{N}$ is B-bounded, then there exists a deterministic automaton $\mathcal{D}$ such that $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$, and whose size and maximal weight are polynomial w.r.t. $w_{\max}$ and B, and exponential w.r.t. $|Q|$.*

Proof sketch. The result is proved by exposing the construction of the deterministic automaton $\mathcal{D}$, inspired by the *determinization algorithm* presented in [11]. On each input word $\alpha \in A^*$, $\mathcal{D}$ outputs the value of the maximal initial run ϱ_α of $\mathcal{N}$ on α (respectively the maximal accepting run if $\alpha \in \mathcal{L}_{\mathcal{N}}$), and keeps track of all the other initial runs on α by storing in its state the pairs $(q, w_q) \in Q \times \{-B, \dots, B\}$ such that the maximal initial run on α that ends in q has weight $w(\varrho_\alpha) + w_q$. If for some state q the delay w_q gets lower than $-B$, the B-boundedness assumption allows $\mathcal{D}$ to drop the corresponding runs without modifying the function defined: whenever a run has a delay smaller than $-B$ with respect to ϱ_α, no continuation will ever be maximal. This ensures that our construction always yields a finite automaton, unlike the determinization algorithm, that does not always terminate. □

On complete-for-determinization classes. Given $r \in \mathbb{N}$, a class $\mathfrak{C}$ of DWA, and an automaton $\mathcal{N}$, we say $\mathcal{N}$ is $(r, \mathfrak{C})$-determinizable if there exists $\mathcal{D} \in \mathfrak{C}(\mathcal{N})$ such that: (i) $\mathcal{L}_{\mathcal{N}} = \mathcal{L}_{\mathcal{D}}$, and (ii) $\sup_{\alpha \in \mathcal{L}_{\mathcal{N}}} |[\![\mathcal{N}]\!](\alpha) - [\![\mathcal{D}]\!](\alpha)| \leq r$.

We will now confirm our claim from the introduction: approximate determinization is not interesting for some classes.

Proposition 5. *Let $\mathcal{N} = (Q, I, A, \Delta, w, F)$ be a trim automaton such that the range of $[\![\mathcal{N}]\!]$ is included into $\{-B, \dots, B\}$, for some $B \in \mathbb{N}$. Then $\mathcal{N}$ is determinizable.*

Proof. Let $\varrho_p = p_0 a_0 p_1 \dots a_{n-1} p_n$ be a maximal accepting run of $\mathcal{N}$ and let $\varrho_q = q_0 a_0 q_1 \dots a_{i-1} q_i$ be an initial run of length $i \leq n$. We define $\varrho_p^i = p_0 a_0 p_1 \dots a_{i-1} p_i$ for $i \leq n$.

By assumption, we have $w(\varrho_p) \geq -B$. By trimness assumption, the state q_i can reach a final state and we have $w(\varrho_q) - |Q|w_{\max} \leq B$ otherwise there would be an accepting run of value greater than B. Similarly, since state p_i can be reached from an initial state, we have $-|Q|w_{\max}+a \leq B$, with $a = w(p_i a_i \dots a_{n-1} p_n) = w(\varrho_p) - w(\varrho_p^i)$. By combining the three constraints, we obtain: $w(\varrho_q) - |Q|w_{\max} - |Q|w_{\max} + a - w(\varrho_p) \leq 3B$ which, once rearranged, yields: $w(\varrho_q) - w(\varrho_p^i) \leq 3B + 2|Q|w_{\max}$. Thus, $\mathcal{N}$ is $(3B + 2|Q|w_{\max})$-bounded and determinizable (by Proposition 4). $\square$

Recall that a class $\mathfrak{C}$ of DWA is complete for determinization if any determinizable automaton is also $\mathfrak{C}$-determinizable.

Theorem 1. *Given a complete-for-determinization class $\mathfrak{C}$ of DWA, an automaton $\mathcal{N}$ is $(r, \mathfrak{C})$-determinizable, for some $r \in \mathbb{N}$, if and only if it is $\mathfrak{C}$-determinizable.*

Proof. If $\mathcal{N}$ is determinizable, then in particular it is $(r, \mathfrak{C})$-determinizable for any r. Conversely, let us assume that $\mathcal{D}$ is an $(r, \mathfrak{C})$-determinizer of $\mathcal{N}$, for some r.

Then one can construct an automaton $\mathcal{M}$ such that $[\![\mathcal{M}]\!] = [\![\mathcal{N}]\!] - [\![\mathcal{D}]\!]$ by taking the product of $\mathcal{N}$ and $\mathcal{D}$ with transitions weighted by the difference of the weights of $\mathcal{N}$ and $\mathcal{D}$. Since $\mathcal{D}$ is r-close to $\mathcal{N}$, the range of $\mathcal{M}$ is included in the set $\{-r, \dots, r\}$. This means, according to Propositions 5, that $\mathcal{M}$ (once trimmed) is determinizable and that one can construct a deterministic automaton realizing $[\![\mathcal{D}]\!] + [\![\mathcal{M}]\!] = [\![\mathcal{N}]\!]$. Since $\mathfrak{C}$ is complete for determinization, the result follows. $\square$

III. DECIDING k-DELAY DETERMINIZABILITY

In this section we prove that deciding k-delay determinizability is EXPTIME-complete. First, however, we show that **Del**-determinization is complete for determinization: if a given automaton is determinizable, then there is a k such that it is k-delay determinizable as well. Hence, exposing an upper bound for k would lead to an algorithm for the general determinizability problem. We also give a family of automata for which an exponential delay is required.

A. Completeness for determinization

Theorem 2. *If an automaton $\mathcal{N}$ is determinizable, then there exists $k \in \mathbb{N}$ such that $\mathcal{N}$ is k-delay determinizable.*

Proof sketch. We proceed by contradiction. Suppose $\mathcal{N} = (Q, I, A, \Delta, w, F)$ is determinizable. Denote by $\mathcal{D} = (Q', \{q'_I\}, A, \Delta', w', F')$ a deterministic automaton such that $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$. Let us assume, towards a contradiction, that for all $k \in \mathbb{N}$ there is no deterministic automaton $\mathcal{E}$ such that $\mathcal{E} \subseteq_k \mathcal{N}$ and $[\![\mathcal{D}]\!] = [\![\mathcal{E}]\!]$. In particular, we have that $\mathcal{D} \not\subseteq_\chi \mathcal{N}$ for $\chi := |Q||Q'|(w_{\max} + w'_{\max})$. This means that there is a word $\alpha = a_0 \dots a_{n-1} \in \mathcal{L}_\mathcal{N}$ such that for a maximal accepting run $\varrho = q_0 a_0 \dots a_{n-1} q_n$ of $\mathcal{N}$ on α it holds that

$$|w(q_0 a_0 \dots a_{\ell-1} q_\ell) - w'(q'_0 a_0 \dots a_{\ell-1} q'_\ell)| > \chi \qquad (1)$$

for some $0 \leq \ell \leq n$ and $q'_0 a_0 \dots a_{n-1} q'_n$ the unique initial run of $\mathcal{D}$ on α. We consider the implications of the two inequalities arising from the above equation.

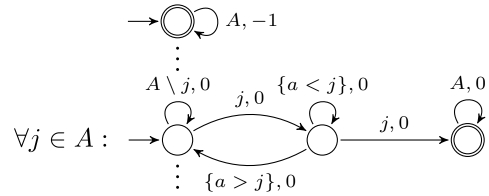

Fig. 3. Automaton $\mathcal{N}$ realizing function f which outputs the (negative) length of the word if it has no j-pair.

Suppose that $w(q_0 a_0 \dots a_{\ell-1} q_\ell) - w'(q'_0 a_0 \dots a_{\ell-1} q'_\ell) > \chi$. Then, at least one final state q_n is reachable in $\mathcal{N}$ from q_ℓ, and the shortest path to it consists of at most $|Q|$ transitions. Since $\chi \geq |Q|(w_{\max} + w'_{\max})$, $\mathcal{D}$ cannot realize the same function as $\mathcal{N}$, which contradicts our hypothesis.

Suppose that $w'(q'_0 a_0 \dots a_{\ell-1} q'_\ell) - w(q_0 a_0 \dots a_{\ell-1} q_\ell) > \chi$. Using the fact that $\chi = |Q||Q'|(w_{\max} + w'_{\max})$, we expose a loop that can be pumped down to present a word mapped to different values by $\mathcal{D}$ and $\mathcal{N}$. $\square$

Although we do not have an upper bound on the k needed for a determinizable automaton to be k-delay determinizable, we are able to provide an exponentially large lower bound.

Proposition 6. *Given an automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$, a delay k as big as $2^{\mathcal{O}(|Q|)}$ might be needed for it to be k-delay determinizable.*

To prove the above proposition we will make use of the language of words with a *j-pair* [22].

Words with a j-pair. Consider the alphabet $A = \{1, \dots, n\}$. Let $\alpha = a_0 a_1 \dots \in A^*$ and $j \in A$. A *j-pair* is a pair of positions $i_1 < i_2$ such that $a_{i_1} = a_{i_2} = j$ and $a_k \leq j$, for all $i_1 \leq k \leq i_2$.

Lemma 7. *For all $j \in A$: (i) for all $\alpha \in A^*$, if α contains no j-pair, then $|\alpha| < 2^n$; (ii) for all $j \in A$, there exists $\alpha \in A^*$ such that $|\alpha| = 2^n - 1$ and α contains no j-pair.*

Proof. A proof of the first claim is given by Klein and Zimmermann in [22, Theorem 1].

To show the second claim holds as well, we can inductively construct a word with the desired property. As the base case, consider $\alpha_1 = 1$. Thus, for some i, there is α_i which contains no j-pair, contains no letter bigger than i, and is of length $2^i - 1$. For the inductive step, we let $\alpha_i = \alpha_{i-1} i \alpha_{i-1}$. It is easy to verify that the properties hold once more. $\square$

We will now focus on the function $f : A^* \to \mathbb{Z}$, which maps a word α to 0 if it contains a j-pair and to $-|\alpha|$ otherwise. Fig. 3 depicts the automaton $\mathcal{N}$ realizing f with $3n+1$ states. Using Lemma 7, it is easy to show that $\mathcal{N}$ is B-bounded and thus, by Proposition 4 and Theorem 2, both determinizable and bounded-delay determinizable. Proposition 6 then follows from the following result.

Lemma 8. *Any determinizer of automaton $\mathcal{N}$ (see Fig. 3), which realizes the function f, has a delay of at least $2^n - 1$.*

Proof. Consider a word α of length $2^n - 1$ containing no j-pair—which exists according to Lemma 7. Further consider an arbitrary k-determinizer $\mathcal{D}$ for $\mathcal{N}$. We remark that $[\![\mathcal{D}]\!](\alpha) = [\![\mathcal{N}]\!](\alpha) = 1 - 2^n$, since both automata realize f and α does not contain a j-pair. It follows from Lemma 7 that $\alpha \cdot a$ contains a j-pair (that is, for all $a \in A$). Hence, for all $a \in A$, we have $[\![\mathcal{N}]\!](\alpha \cdot a) = 0$. Furthermore, by construction of $\mathcal{N}$, for all maximal accepting runs $q_0 a_0 \dots a_{|\alpha|+1} q_{|\alpha|+2}$ of $\mathcal{N}$ on $\alpha \cdot a$ we have $w(q_0 \dots q_i) = 0$ for all $1 \le i \le |\alpha| + 2$. In particular, for $i = |\alpha| + 1$, we have $[\![\mathcal{D}]\!](\alpha) - w(q_0 \dots q_i) = 2^n - 1$. $\square$

B. Upper bound

We now argue that 0-delay determinizability is in EXPTIME. Then, we show how to reduce (in exponential time) k-delay determinizability to 0-delay determinizability. We claim that the composition of the two algorithms remains singly exponential.

Proposition 9. *Deciding the* 0*-delay problem for a given automaton is in* EXPTIME*.*

The result will follow from Propositions 3 and 12. Before we state and prove Proposition 12 we need some intermediate definitions and lemmas. The following properties of k-inclusion, which follow directly from the definition, will be useful later.

Lemma 10. *For all automata $\mathcal{N}$, $\mathcal{N}'$, and $\mathcal{N}''$, for all $k, k' \in \mathbb{N}$, the following hold:*

1) *if $\mathcal{N} \subseteq_k \mathcal{N}'$ and $k \le k'$, then $\mathcal{N} \subseteq_{k'} \mathcal{N}'$;*
2) *if $\mathcal{N} \subseteq_k \mathcal{N}'$ and $\mathcal{N}' \subseteq_{k'} \mathcal{N}''$, then $\mathcal{N} \subseteq_{k+k'} \mathcal{N}''$;*
3) *if $\mathcal{N} \subseteq_k \mathcal{N}'$, then $\mathcal{L}_{\mathcal{N}} \subseteq \mathcal{L}_{\mathcal{N}'}$, and for every $\alpha \in \mathcal{L}_{\mathcal{N}}$, $[\![\mathcal{N}]\!](\alpha) \le [\![\mathcal{N}']\!](\alpha)$.*

We now show how to decide 0-delay determinizability by reduction to 0-regret determinizability. Let us first convince the reader that 0-regret determinizability implies 0-delay determinizability.

Proposition 11. *If an automaton $\mathcal{N}$ is* 0*-regret determinizable, then it is* 0*-delay determinizable.*

Proof. We have, from Lemma 2, that Eve has a finite-memory winning strategy σ in the 0-regret game played on $\mathcal{N}$. Then, by definition of the regret game, $\mathcal{L}_{\mathcal{N}} = \mathcal{L}_{\mathcal{N}_\sigma}$, and for every $\alpha \in \mathcal{L}_{\mathcal{N}}$, $[\![\mathcal{N}]\!](\alpha) - [\![\mathcal{N}_\sigma]\!](\alpha) \le 0$, hence $[\![\mathcal{N}]\!] = [\![\mathcal{N}_\sigma]\!]$. Moreover, as Eve chooses a run in $\mathcal{N}$, we have $\mathcal{N}_\sigma \subseteq_0 \mathcal{N}$. Therefore $\mathcal{N}_\sigma$ is a 0-delay determinizer of $\mathcal{N}$. $\square$

The converse of the above result does not hold in general (see Fig. 2). Nonetheless, it holds when the automaton is pair-deterministic. We now show that, under this hypothesis, an automaton is 0-regret determinizable if and only if the automaton is 0-delay determinizable.

Proposition 12. *A pair-deterministic automaton $\mathcal{N}$ is* 0*-delay determinizable if and only if it is* 0*-regret determinizable.*

Proof sketch. If $\mathcal{N}$ is 0-regret determinizable, then $\mathcal{N}$ is 0-delay determinizable by Proposition 11. Now suppose that $\mathcal{N}$ is 0-delay determinizable, and let $\mathcal{D}$ be a 0-delay determinizer of $\mathcal{N}$. For every initial run $\varrho_\alpha = p_0 a_0 p_1 \dots a_{n-1} p_n$ of $\mathcal{D}$ on input $\alpha = a_0 \dots a_{n-1}$, there exists exactly one initial run $\varrho'_\alpha = p'_0 a_0 p'_1 \dots a_{n-1} p'_n$ of $\mathcal{N}$ such that for every $1 \le i \le n$, $w(p'_0 \dots p'_i) = w'(p_0 \dots p_i)$. The existence of ϱ'_α is guaranteed by the fact that $\mathcal{D}$ is a 0-delay determinizer of $\mathcal{N}$, and, since $\mathcal{N}$ is pair-deterministic, such a run is unique. Then the strategy for Eve in the 0-regret game played on $\mathcal{N}$ obtained by following, given an input word α, the run ϱ'_α of $\mathcal{N}$, is winning. $\square$

We observe that any automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$ can be transformed into a pair-deterministic automaton $\mathcal{P}(\mathcal{N})$ with at most an exponential blow-up in the state-space. Intuitively, we merge all the states from the original automaton which can be reached by reading $a \in A$ and taking a transition with weight $x \in \mathbb{Z}$. This is a generalization of the classical subset construction used to determinize unweighted automata. Critically, the construction is such that $\mathcal{P}(\mathcal{N}) \subseteq_0 \mathcal{N}$ and $\mathcal{N} \subseteq_0 \mathcal{P}(\mathcal{N})$. The next result then follows immediately from the latter property and from Lemma 10 item 2.

Proposition 13. *An automaton $\mathcal{N}$ is* 0*-delay determinizable if and only if $\mathcal{P}(\mathcal{N})$ is* 0*-delay determinizable if and only if $\mathcal{P}(\mathcal{N})$ is* 0*-regret determinizable.*

We now show how to extend the above techniques to the general case of k-delay.

Theorem 3. *Deciding the k-delay problem for a given automaton is in* EXPTIME*.*

Given an automaton $\mathcal{N}$ and $k \in \mathbb{N}$, we will construct a new automaton $\delta_k(\mathcal{N})$ that will encode delays (up to k) in its state space. In this new automaton, for every state-delay pair (p, i) and for every transition $(p, a, q) \in \Delta$, we will have an a-labelled transition to (q, j) with weight $i + w(p, a, q) - j$ for all $-k \le j \le k$. Intuitively, i is the amount of delay the automaton currently has, and to get to a point where the delay becomes j via transition (p, a, q) a weight of $i + w(p, a, q) - j$ must be outputted. We will then show that the resulting automaton is 0-delay determinizable if and only if the original automaton is k-delay determinizable.

k-delay construction. Let $\mathcal{N} = (Q, I, A, \Delta, w, F)$ be an automaton. Let $\delta_k(\mathcal{N}) = (Q', I', A, \Delta', w', F')$ be the automaton defined as follows.

- $Q' = Q \times \{-k, \dots, k\}$;
- $I' = I \times \{0\}$;
- $\Delta' = \{((p, i), a, (q, j)) \mid (p, a, q) \in \Delta\}$;
- $w' : \Delta' \to \mathbb{Z}$, $((p, i), a, (q, j)) \mapsto i + w(p, a, q) - j$;
- $F' = F \times \{0\}$.

Lemma 14. *The k-delay construction satisfies the following properties.*

1) *$\delta_k(\mathcal{N}) \subseteq_k \mathcal{N}$;*
2) *for all automata $\mathcal{M}$ s.t. $\mathcal{M} \subseteq_k \mathcal{N}$, $\mathcal{M} \subseteq_0 \delta_k(\mathcal{N})$;*
3) *$[\![\delta_k(\mathcal{N})]\!] = [\![\mathcal{N}]\!]$.*

Proof.
1) Let $(q_0, i_0) a_0 (q_1, i_1) \dots a_{n-1} (q_n, i_n)$ be an accepting run of $\delta_k(\mathcal{N})$. Then $q_0 a_0 q_1 \dots a_{n-1} q_n$ is an accepting run of $\mathcal{N}$,

and for every $0 \leq j < n$,

$$\left|\sum_{\ell=0}^{j} w'((q_\ell, i_\ell), a_\ell, (q_{\ell+1}, i_{\ell+1})) - \sum_{\ell=0}^{j} w(q_\ell, a_\ell, q_{\ell+1})\right|$$
$$= \left|\sum_{\ell=0}^{j} (i_\ell + w(q_\ell, a_\ell, q_{\ell+1}) - i_{\ell+1} - w(q_\ell, a_\ell, q_{\ell+1}))\right|$$
$$= |i_0 - i_j| = |i_j| \leq k \text{ (since } i_0 = 0).$$

2) Let $\mathcal{M} = (Q'', I'', A, \Delta'', w'', F'')$ be an automaton such that $\mathcal{M} \subseteq_k \mathcal{N}$. For every accepting run $p_0 a_0 \dots a_{n-1} p_n$ of $\mathcal{M}$, there exists an accepting run $q_0 a_0 \dots a_{n-1} q_n$ of $\mathcal{N}$ such that for every $0 \leq j < n$

$$\sum_{l=0}^{j} (w(q_l, a_l, q_{l+1}) - w''(p_l, a_l, p_{l+1})) \in \{-k, \dots, k\}.$$

Let i_j denote the above value. Then $(q_0, i_0) a_0 \dots a_{n-1} (q_n, i_n)$ is an accepting run of $\delta_k(\mathcal{N})$, and for every $0 \leq j < n$,

$$\begin{aligned} & w'((q_j, i_j), a_j, (q_{j+1}, i_{j+1})) \\ &= i_j + w(q_j, a_j, q_{j+1}) - i_{j+1} = w''(p_j, a_j, p_{j+1}). \end{aligned}$$

3) This follows from the first property, the second one in the particular case $\mathcal{M} = \mathcal{N}$, and Lemma 10 item 3. ☐

The next result follows immediately from the preceding Lemma and Lemma 10 item 2.

Proposition 15. *An automaton $\mathcal{N}$ is k-delay determinizable if and only if $\delta_k(\mathcal{N})$ is 0-delay determinizable.*

The above result raises the question of whether, for all k, 0-delay determinization can be reduced to k-delay determinization. We give a positive answer to this question in the form of Lemma 16 in Section III-C.

We now proceed with the proof of Theorem 3.

Proof of Theorem 3. Membership in 2EXPTIME follows from Proposition 15 and Proposition 9. We now observe that the subset construction used to decide 0-delay determinizability need only be applied on the first component of the state space resulting from the use of the delay construction. In other words, once both constructions are applied, a state will correspond to a function $f : Q \to \{-k, \dots, k\} \cup \{\perp\}$, where $q \mapsto \perp$ signifies that q is not in the subset. The size of the resulting state space is then $2^{\mathcal{O}(|Q| \log_2 k)}$. Thus, the composition of these two constructions yields only a single exponential. ☐

C. Lower bound

We reduce the 0-delay uniformization problem for synchronous transducers to that of deciding whether a given automaton is k-delay determinizable (for any fixed $k \in \mathbb{N}$). As the former problem is known to be EXPTIME-complete (see [21]), this implies the latter is EXPTIME-hard.

Theorem 4. *Deciding the k-delay problem for a given automaton is* EXPTIME*-hard, even for fixed $k \in \mathbb{N}$.*

For convenience, we will first prove that the 0-delay problem reduces to the k-delay problem for any fixed k. We then show the former is EXPTIME-hard.

Lemma 16. *The 0-delay problem reduces in logarithmic space to the k-delay problem, for any fixed $k \in \mathbb{N}$.*

Let us fix some $k \in \mathbb{N}$. Given the automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$ we denote by $x \cdot \mathcal{N}$ the automaton $(Q, I, A, \Delta, x \cdot w, F)$, where $x \cdot w$ is such that $d \mapsto x \cdot w(d)$ for all $d \in \Delta$. Lemma 16 is a direct consequence of the following.

Lemma 17. *For every automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$, the following statements are equivalent.*

1) *$\mathcal{N}$ is 0-delay determinizable;*
2) *$(4k+1) \cdot \mathcal{N}$ is 0-delay determinizable;*
3) *$(4k+1) \cdot \mathcal{N}$ is k-delay determinizable.*

Proof sketch. Given a 0-delay determinizer $\mathcal{D}$ of $\mathcal{N}$, the automaton $(4k+1)\cdot\mathcal{D}$ is easily seen to be a 0-delay determinizer of $(4k+1)\cdot\mathcal{N}$. Hence, the first statement implies the second one. Moreover, as a direct consequence of Lemma 10 item 1, the second statement implies the third one. To complete the proof, we argue that if $(4k+1)\cdot\mathcal{N}$ is k-delay determinizable, then $\mathcal{N}$ is 0-delay determinizable. Let $\mathcal{D}' = (Q', I', A, \Delta', w', F')$ be a k-delay determinizer of $(4k+1)\cdot\mathcal{N}$. Let γ be the function mapping every integer x to the unique integer $\gamma(x)$ satisfying $|(4k+1)\gamma(x) - x| \leq 2k$, and let $\gamma(\mathcal{D}')$ denote the deterministic automaton $(Q', I', A, \Delta', \gamma \circ w', F')$. One can then show that $\gamma(\mathcal{D}')$ is a 0-delay determinizer of $\mathcal{N}$. ☐

We can now show that the k-delay problem is EXPTIME-hard by arguing that the 0-delay problem is EXPTIME-hard. Let us introduce some notation regarding transducers.

Transducers. A *(synchronous) transducer* $\mathcal{T}$ from an input alphabet A_I to an output alphabet A_O is an unweighted automaton $(Q, I, A_I \times A_O, \Delta, F)$. We denote the *domain* of $\mathcal{T}$ by $\mathsf{dom}(\mathcal{T}) := \{a_0 \dots a_{n-1} \in A_I^* \mid (a_0, b_0) \dots (a_{n-1}, b_{n-1}) \in (A_I \times A_O)^*\}$. The transducer $\mathcal{T}$ is said to be *input-deterministic* if for all $p \in Q$, for all $a \in A_I$, there exist at most one state-output pair $(q, b) \in Q \times A_O$ such that $(p, (a, b), q) \in \Delta$. A transducer $\mathcal{U}$ from A_I to A_O is a *0-delay uniformizer* of $\mathcal{T}$ if (i) $\mathcal{U}$ is input-deterministic, (ii) $\mathcal{L}_\mathcal{U} \subseteq \mathcal{L}_\mathcal{T}$, and (iii) $\mathsf{dom}(\mathcal{U}) = \mathsf{dom}(\mathcal{T})$. If such a transducer exists, we say $\mathcal{T}$ is 0-delay uniformizable. Given a transducer, to determine whether it is 0-delay uniformizable is an EXPTIME-hard problem [21].

Intuitively, a transducer induces a relation from input words to output words. We construct an automaton that replaces the output alphabet by unique positive integer identifiers. For convenience, we also make sure the constructed automaton defines a function which maps every word in its language to 0.

From transducers to weighted automata. Given a transducer $\mathcal{T} = (Q, I, A_I \times A_O, \Delta, F)$ with $A_0 = \{1, \dots M\}$, we construct a weighted automaton $\mathcal{N}_\mathcal{T} = (Q', I, A_I \cup \{\#\}, \Delta', w, F)$ as follows.

- $Q' = Q \cup Q \times A_O$
- $\Delta' = \{(p, a, (q, m)), ((q, m), \#, q) \mid (p, (a, m), q) \in \Delta\}$
- $w : \Delta' \to \mathbb{Z}$, $(p, a, (q, m)) \mapsto m$ and $((q, m), \#, q) \mapsto -m$.

Lemma 18. *The translation from transducers to weighted automata satisfies the following properties.*

1) *$q_0(a_0, m_0) \dots (a_{n-1}, m_{n-1})q_n$ is a run of $\mathcal{T}$ if and only if $q_0 a_0 (q_0, m_0)\# \dots (q_{n-1}, m_{n-1})\# q_n$ is a run of $\mathcal{N}_\mathcal{T}$. Moreover, for all $0 \leq i \leq n$,*
   - *$w(q_0 \dots q_i) = 0$, and $w(q_0 \dots (q_i, m_i)) = m_i$;*
2) *$\mathcal{L}_{\mathcal{N}_\mathcal{T}} = \{a_0\# \dots \# a_n\# \mid a_0 \dots a_n \in \mathsf{dom}(\mathcal{T})\}$;*
3) *$[\![\mathcal{N}_\mathcal{T}]\!](\alpha) = 0$ for all $\alpha \in \mathcal{L}_{\mathcal{N}_\mathcal{T}}$.*

Proof. The first item follows by construction of the automaton $\mathcal{N}_\mathcal{T}$. Items 2 and 3 are direct consequences of item 1. □

We are now ready to show the 0-delay uniformization problem reduces in polynomial time to the 0-delay determinization problem. To do so, we show that any 0-delay uniformizer of a transducer $\mathcal{T}$ can be transformed into a 0-delay determinizer of $\mathcal{N}_\mathcal{T}$, and vice versa.

Lemma 19. *Deciding the 0-delay problem for a given automaton is* EXPTIME*-hard.*

Proof sketch. Given a transducer $\mathcal{T} = (Q, I, A_I \times A_O, \Delta, F)$ with $A_0 = \{1, \dots, M\}$, we construct $\mathcal{N}_\mathcal{T} = (Q', I, A_I \cup \{\#\}, \Delta', w, F)$. Suppose $\mathcal{U} = (S, \{s_0\}, A_I \times A_O, R, G)$ is a 0-delay uniformizer of $\mathcal{T}$. Using Lemma 18, it is easy to show that $\mathcal{N}_\mathcal{U}$ is a 0-delay determinizer of $\mathcal{N}_\mathcal{T}$.

Conversely, if we assume $\mathcal{D} = (S, \{s_0\}, A_I \cup \{\#\}, R, \mu, G)$ is a 0-delay determinizer of $\mathcal{N}_\mathcal{T}$. Let $\mathcal{U}$ be the transducer $(S, \{s_0\}, A_I \times A_O, R', G)$ where $R' = \{(p, (a, m), s)) \mid (p, a, q), (q, \#, s) \in R \wedge \mu(p, a, q) = -\mu(q, \#, s) = m\}$. To conclude the proof of the claim, we again use Lemma 18 to argue that $\mathcal{U}$ is a 0-delay uniformizer of $\mathcal{T}$. □

IV. Deciding r-regret determinizability

In this section we argue that the r-regret problem is EXPTIME-complete. It will be convenient to **suppose all automata we work with are trim**. This is no loss of generality with regard to r-regret determinizability, *i.e.* an automaton $\mathcal{N}$ is r-regret determinizable if and only if its trim version, $\mathcal{N}'$, is r-regret determinizable. Clearly, an r-regret determinizer of $\mathcal{N}'$ is also an r-regret determinizer of $\mathcal{N}$. Also, it is easy to show that the trim version $\mathcal{D}'$ of an r-regret determinizer $\mathcal{D}$ of $\mathcal{N}$ must also be an r-regret determinizer of $\mathcal{N}'$. Furthermore, any automaton can be trimmed in polynomial time.

A. Upper bound

We will now give an exponential time algorithm to determine whether a given automaton is r-regret determinizable, for a given r. The algorithm is based on a quantitative version of the *Joker game* introduced by Kuperberg and Skrzypczak to study the determinization of good-for-games automata [19]. More precisely, the Joker game will correspond to a generalization of the classical energy games [23].

The algorithm is as follows: construct an energy game with resets (which we call the Joker game) based on the given automaton and decide if Eve wins it; if this is not the case, then for all $r \in \mathbb{N}$ the automaton is not r-regret determinizable; otherwise, using the winning strategy for Eve in the Joker game, construct a deterministic automaton $\mathcal{D}$ realizing the same function as $\mathcal{N}$ and use it to decide if $\mathcal{N}$ is r-regret determinizable. The last step of the algorithm is the simplest. Given a deterministic version of the original automaton, one can use it as a "monitor" and reduce the r-regret determinizability problem to deciding the winner in an energy game.

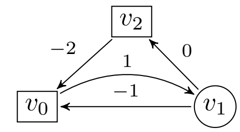

Fig. 4. Energy game with reset edges $E_\varrho = \{(v_1, v_2)\}$ where Eve wins from v_0 with initial credit $c_0 = 2$

Theorem 5. *Deciding the r-regret problem for a given automaton is in* EXPTIME.

Energy games with resets. An *energy game with resets* (EGR for short) is an infinite-duration two-player turn-based game played by Eve and Adam on a directed weighted graph. Formally, an EGR $\mathcal{G} = (V, V_\exists, E, E_\varrho, w)$ consists of: a set V of vertices, a set $V_\exists \subseteq V$ of vertices of Eve—the set $V_\forall := (V \setminus V_\exists)$ of vertices thus belongs to Adam, a set $E \subseteq V \times V$ of directed edges, a set $E_\varrho \subseteq E$ of *reset edges* such that $E_\varrho \subseteq V_\forall \times V$, and a weight function $w : E \to \mathbb{Z}$. (Observe that if $E_\varrho = \varnothing$, we obtain the classical energy games without resets [23].) Pictorially, we represent Eve vertices by squares and Adam vertices by circles. We denote by $w_{\max}$ the value $\max_{e \in E} |w(e)|$. Intuitively, from the current vertex u, the player who owns u (*i.e.* Eve if $u \in V_\exists$, and Adam otherwise) chooses an edge $(u, v) \in E$ and the *play* moves to v. We formalize the notions of strategy and play below.

A strategy for Eve (respectively, Adam) in $\mathcal{G}$ is a mapping $\sigma : V^* \cdot V_\exists \to V$ (respectively, $\tau : V^* \cdot V_\forall \to V$) such that $\sigma(v_0 \dots v_n) = v_{n+1}$ ($\tau(v_0 \dots v_n) = v_{n+1}$) implies $(v_n, v_{n+1}) \in E$. As in regret games, a strategy σ for either player is one which can be encoded as a deterministic Mealy machine $(S, s_I, \lambda_u, \lambda_o)$ with update function $\lambda_u : S \times V \to S$ and output function $\lambda_o : S \times V \to V$. The machine encodes σ in the following sense: $\sigma(v_0 \dots v_n) = \lambda_u(s_n, v_n)$ where $s_0 = s_I$ and $s_{i+1} = \lambda_u(s_i, v_i)$ for all $0 \leq i < n$. As usual, the memory of a finite-memory strategy refers to the size of the Mealy machine realizing it.

A play in $\mathcal{G}$ from $v \in V$ corresponds to an infinite path in the underlying directed graph (V, E). That is, a sequence $\pi = v_0 v_1 \dots$ such that $(v_i, v_{i+1}) \in E$ for all $i \in \mathbb{N}$. Since an EGR is played for an infinite duration, we will henceforth assume they are played on digraphs with no sinks: *i.e.* for all $u \in V$, there exists $v \in V$ such that $(u, v) \in E$. We say a play $\pi = v_0 v_1 \dots$ is consistent with a strategy σ for Eve (respectively, τ for Adam) if it holds that $v_i \in V_\exists$ implies $\sigma(v_0 \dots v_i) = v_{i+1}$ ($v_i \notin V_\forall$ implies $\tau(v_0 \dots v_i) = v_{i+1}$). Given a strategy σ for Eve and a strategy τ for Adam, and a vertex $v \in V$ there is a unique play $\pi^v_{\sigma\tau}$ compatible with both σ and τ from v.

Given a *finite path* φ in $\mathcal{G}$, *i.e.* a sequence $v_0 \dots v_n$ such that $(v_i, v_{i+1}) \in E$ for all $0 \leq i < n$, and an initial credit

$c_0 \in \mathbb{N}$, we define the *energy level* of φ as $\mathsf{EL}_{c_0}(\varphi) := c_0 + \sum_{j=i_0}^{n-1} w(v_j, v_{j+1})$ where $0 \le i_0 < n$ is the minimal index such that $(v_\ell, v_{\ell+1}) \notin E_\varrho$ for all $i_0 < \ell < n$.

We say Eve wins the EGR from a vertex $v \in V$ with initial credit c_0 if she has a strategy σ such that, for all strategies τ for Adam, for all finite prefixes φ of $\pi^v_{\sigma\tau}$ we have $\mathsf{EL}_{c_0}(\varphi) \ge 0$. Adam wins the EGR from v with initial credit c_0 if and only if Eve does not win it.

Example 3. Consider the EGR shown in Fig. 4. In this game, Eve wins from v_0 with initial credit 2. Indeed, whenever Adam plays from v_1 to v_0 the energy level drops by 1 but is then increased by 1 when the play returns to v_1; when he plays from v_1 to v_2 the energy level is first reset to 2 and then drops to 0 when the play reaches v_0. Clearly then, Adam cannot force a negative energy level. However, if E_ϱ were empty, then Eve would lose the game regardless of the initial credit.

The following properties of energy games (both, with or without resets), which include *positional determinacy*, will be useful in the sequel. A game is positionally determined if: for all instances of the game, from all vertices, either Eve has a positional strategy which is winning for her against any strategy for Adam, or Adam has a positional strategy which is winning for him against any strategy for Eve.

Proposition 20. *For any energy game (both, with or without resets) $\mathcal{G} = (V, V_\exists, E, E_\varrho, w)$ the following hold.*

1) *The game is positionally determined if $c_0 \ge |V| w_{\max}$.*
2) *For all $v \in V$, Eve wins from $v \in V$ with initial credit $|V| w_{\max}$ if and only if there exists $c_0 \in \mathbb{N}$ such that she wins from $v \in V$ with initial credit c_0.*
3) *Determining if there exists $c_0 \in \mathbb{N}$ such that Eve wins from $v \in V$ with initial credit c_0 is decidable in time polynomial in $|V|$, $|E|$, and $w_{\max}$.*

Proof sketch. All three properties are known to hold for energy games without resets (see, *e.g.* [23], [24]).

For EGRs the arguments to show these properties hold are almost identical to those used in [24]. We first define a finite version of the game which is stopped after the first cycle is formed and in which the winner is determined based on properties of that cycle. If we let Eve win if and only if the cycle has non-negative sum of weights or it contains a reset, then we can show she wins this *First Cycle Game* [25] if and only if she wins the EGR with initial credit $|V| w_{\max}$. Furthermore, using a result from [25] we obtain that positional strategies suffice for both players in both games, *i.e.* the games are positionally determined.

The second property follows from the relationship between the EGR and the first cycle game we construct. More precisely, we show that winning strategies for both players transfer between the games. In the first cycle game, Adam wins if he can force cycles which have a negative sum of weights. Hence, if Eve does not win the EGR with initial credit $|V| w_{\max}$, then by determinacy, Adam wins the first cycle game, and his strategy—when played on the original EGR—ensures only negatively-weighted cycles are formed, which in turn means that he wins the EGR with any initial credit.

Finally, to obtain an algorithm, we reduce the problem of deciding if Eve wins the EGR from $v \in V$ with a given initial credit c_0 to her winning a *safety game* [26] played on an unweighted digraph where the states keep track of the energy level (up to a maximum of $|V| w_{\max}$). $\square$

Energy games will be our main tool for the rest of this section. They allow us to claim that, given an automaton $\mathcal{N}$ and a deterministic automaton $\mathcal{D}$ which defines the same function, we can decide r-regret determinizability.

Proposition 21. *Given an automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$ and $\mathcal{D} = (Q', \{q'_I\}, A, \Delta', w', F')$ such that $\mathcal{D}$ is deterministic and $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$, the r-regret problem for $\mathcal{N}$ is decidable in time polynomial in $|Q|$, $|Q'|$, $|A|$, $w_{\max}$, and $w'_{\max}$.*

Proof sketch. We construct an energy game without resets which simulates the regret game played on $\mathcal{N}$ while using $\mathcal{D}$ to compare the weights of transitions chosen by Eve to those of the maximal run of $\mathcal{N}$. Intuitively, Eve chooses an initial state in $\mathcal{N}$, then Adam chooses a symbol, and Eve responds with a transition $t \in \Delta$ in $\mathcal{N}$. Finally, the state of $\mathcal{D}$ is deterministically updated via transition t'. The weight of the whole round is $w(t) - w'(t')$. We also make sure Eve loses if in the regret game she reaches a non-final state and the run of $\mathcal{D}$ is at a final state; or if she reaches a final state with too low an energy level (implying a large regret). Formally, the energy game without resets is $\mathcal{G} = (V, V_\exists, E, \varnothing, \mu)$ where:

- $V = Q^2 \cup Q^3 \times A \cup \{\top, \bot, \bot^r_1, \bot^r_2\}$;
- $V_\exists = Q^3 \times A$;
- E contains edges to simulate transitions of $\mathcal{N}$ and $\mathcal{D}$, *i.e.* $\{((p,q),(p,q,q',a)) \mid (q,a,q') \in \Delta'\} \cup \{((p,q,q',a),(p',q')) \mid (p,a,p') \in \Delta\}$, edges required to verify Eve does not reach a non-final state when $\mathcal{D}$ accepts, *i.e.* $\{((p,q),\bot) \mid p \notin F \wedge q \in F'\} \cup \{(\bot,\bot)\}$, edges to make sure the regret is at most r, *i.e.* $\{((p,q),\bot^r_1) \mid p \in F \wedge q \in F'\} \cup \{(\bot^r_1,\bot^r_2),(\bot^r_2,\bot^r_1)\}$, and edges to punish one of the players if an automaton blocks, *i.e.* $\{((p,q),\top) \mid \neg\exists (q,a,q') \in \Delta'\} \cup \{((p,q,q',a),\bot) \mid \neg\exists (p,a,p') \in \Delta\} \cup \{(\top,\top)\}$;
- $\mu : E \to \mathbb{Z}$ is such that
  - $((p,q,q',a),(p',q')) \mapsto w(p,a,p') - w'(q,a,q')$,
  - $(\bot,\bot) \mapsto -1$,
  - $((p,q),\bot^r_1) \mapsto 1 - |Q'|(w_{\max} + w'_{\max})$,
  - $(\bot^r_1,\bot^r_2) \mapsto -1$, $(\bot^r_2,\bot^r_1) \mapsto 1$,
  - $(\top,\top) \mapsto 1$, and
  - $e \mapsto 0$ for all other $e \in E$.

We then claim that for some $p_I \in I$, Eve wins the energy game without resets $\mathcal{G}$ from (p_I, q'_I) with initial credit $r + |Q'|(w_{\max} + w'_{\max})$ if and only if $\mathcal{N}$ is r-regret determinizable. The result then follows from the fact $\mathcal{G}$ is of size polynomial w.r.t. $\mathcal{D}$ and $\mathcal{N}$, and the application of the algorithm (see Proposition 20) to determine the winner of $\mathcal{G}$. $\square$

The Joker game. The Joker game (JG) is a game played by Eve and Adam on an automaton (Q, I, A, Δ, w, F). It is played as follows: Eve chooses as initial state $p \in I$ and Adam an initial state $q \in I$ and the initial configuration becomes $(p,q) \in I^2$. From the current configuration $(p,q) \in Q^2$ **(Step i):** Adam chooses a symbol $a \in A$, **(Step ii):** then Eve chooses a transition $(p,a,p') \in \Delta$, and **(Step iii):** Adam can **(Step iii.a):** choose a transition $(q,a,q') \in \Delta$ or **(Step iii.b):** play joker and choose a transition $(p,a,q') \in \Delta$. The new configuration is then (p',q'). The weight assigned to each round corresponds to the weight of the transition chosen by Eve minus the weight of that chosen by Adam. If Adam played joker, then the sum of weights is reset before adding the weight of the configuration change. Additionally, if Eve moves to a non-final state and Adam moves to a final state, or if Eve can no longer extend the run she is constructing, then **(Step $\star$):** we ensure Eve loses the game.

We formalize the JG played on $\mathcal{N} = (Q, I, A, \Delta, w, F)$ as an EGR $(V, V_\exists, E, E_\varrho, \mu)$ with $V = V_\exists \cup V_\forall$, $E = \bigcup_{1 \le i \le 3} E_{\forall_i} \cup E_\exists \cup E_\varrho \cup \{(\bot,\bot)\}$ where:

- $V_\exists = Q^2 \times A \cup \{\bot\}$;
- $V_\forall = Q^2 \cup Q^3 \times A$;
- **(Step i):** $E_{\forall_1} = \{((p,q),(p,q,a)) \mid (p,q) \in Q^2, (q,a,q') \in \Delta\}$;
- **(Step ii):** $E_\exists = \{((p,q,a),(p,q,p',a)) \mid (p,q,a) \in V_\exists, (p,a,p') \in \Delta\}$;
- **(Step iii.a):** $E_{\forall_2} = \{((p,q,p',a),(p',q')) \mid (p,q,p',a) \in Q^3 \times A, (q,a,q') \in \Delta\}$;
- **(Step iii.b):** $E_\varrho = \{((p,q,p',a),(p',p'')) \mid (p,q,p',a) \in Q^3 \times A, (p,a,p'') \in \Delta\}$;
- **(Step $\star$):** $E_{\forall_3} = \{((p,q),\bot) \mid p \notin F \wedge q \in F$ or $\exists a \in A, \forall p' \in Q : (p,a,p') \notin \Delta\} \cup \{(\bot,\bot)\}$; and
- μ is such that
  - $(\bot,\bot) \mapsto -1$,
  - $e \mapsto w(p,a,p') - w(q,a,q')$ for all $e = ((p,q,p',a),(p',q')) \in E_{\forall_2}$,
  - $e \mapsto w(p,a,p') - w(p,a,p'')$ for all $e = ((p,q,p',a),(p,p'')) \in E_\varrho$,
  - and $e \mapsto 0$ for all other $e \in E$.

It is easy to verify that there are no sinks in the EGR.

Winning the Joker game. We say Eve wins the JG played on (Q, I, A, Δ, w, F) if there is $p \in I$ such that, for all $q \in I$, she wins from (p,q) with initial credit $|V|\mu_{\max}$ (where $\mu_{\max} := \max_{e \in E} |\mu(e)|$). Proposition 20 tells us that, if Eve wins with some initial credit, then she also wins with initial credit $|V|\mu_{\max}$.

We now establish a relationship between r-regret determinization and the JG.

Lemma 22. *If an automaton $\mathcal{N} = (Q, I, A, \Delta, w, F)$ is r-regret determinizable, for some $r \in \mathbb{N}$, then Eve wins the JG played on $\mathcal{N}$.*

Proof. We will actually prove the contrapositive holds. Suppose Eve does not win the JG. By determinacy of EGRs (Proposition 20 item 1) we know that Adam, for all $p_0 \in I$, has a strategy τ to force from some $(p_0,q_0) \in I^2$ a play which eventually witnesses a negative energy level. Furthermore, he can do so for any initial credit (Proposition 20 item 2). Let us now assume, towards a contradiction, that Eve wins the r-regret game with a strategy σ such that $\sigma(\varepsilon) = p_0$. Since σ is winning for her in the regret game, then for all $\alpha \in A^*$, $\sigma(\alpha)$ is an initial run of $\mathcal{N}$. Hence σ can be converted into a strategy for Eve in the JG by ignoring the transitions chosen by Adam and following σ when Adam chooses a symbol $a \in A$. If Eve follows σ to play in the JG against τ, then there exists $q_0 \in I$ such that $\pi_{\sigma\tau}^{(p_0,q_0)}$ eventually witnesses a negative energy level even if the initial credit is $r + 2|Q|w_{\max}$ (because τ is winning for Adam in the JG with any initial credit). Moreover, $\pi_{\sigma\tau}^{(p_0,q_0)}$ never reaches the vertex $\bot$, since $\sigma(\alpha)$ is an initial run of $\mathcal{N}$ for all $\alpha \in A^*$. If we let $\varphi = (p_0,q_0)(p_0,q_0,a_0)(p_0,q_0,p_1,a_0)\dots(p_n,q_n)$ be the first prefix that witnesses a negative energy level with initial credit $r + 2|Q|w_{\max}$, and $0 \le i_0 < n$ be the minimal index such that no reset occurs for all $i_0 < \ell < n$, then $\varrho = p_0a_0p_1a_1\dots p_n$ and $\varrho' = p_0a_0\dots p_{i_0}a_{i_0}q_{i_0+1}\dots q_n$ are two runs in $\mathcal{N}$ such that $w(\varrho') > w(\varrho) + r + 2|Q|w_{\max}$. Since $\mathcal{N}$ is trim, there is a final run $q_na_n\dots a_{m-1}q_m$ such that $m - n \le |Q|$. Hence, we have that $[\![\mathcal{N}]\!](a_0\dots a_{m-1}) - \mathsf{Val}(\sigma(a_0\dots a_{m-1})) > r$, which contradicts the fact that σ is winning for Eve in the regret game. It follows that there cannot be a winning strategy for Eve in the r-regret game. □

From the above results we have that if we construct the JG for the given automaton and Eve does not win the JG, then the automaton cannot be r-regret determinizable (no matter the value of r). We now study the case when Eve does win.

Using the JG to determinize an automaton. Let $\mathcal{N} = (Q, I, A, \Delta, w, F)$ be an automaton. We will assume that Eve wins the JG played on $\mathcal{N}$. Denote by $W^{\mathrm{JG}} \subseteq Q^2$ the *winning region* of Eve. That is, W^{JG} is the set of all $(p,q) \in Q^2$ such that Eve wins the EGR from (p,q) with initial credit $|V|\mu_{\max}$. Also, let us write Q^{JG} for the projection of W^{JG} on its first component. Moreover, for every $(p,q) \in W^{\mathrm{JG}}$, let $\mathsf{Cr}(p,q)$ denote the minimal integer $c \in \mathbb{N}$ such that Eve wins the JG from (p,q) with initial credit c.

We will now prove some properties of the sets W^{JG} and Q^{JG}. First, the relation W^{JG} is transitive.

Lemma 23. *For all $p,q,t \in Q$, if $(p,q),(q,t) \in W^{\mathrm{JG}}$ then $(p,t) \in W^{\mathrm{JG}}$.*

Proof sketch. For every $(p,q) \in W^{\mathrm{JG}}$, let $\sigma_{(p,q)}$ denote a winning strategy for Eve in the JG played from (p,q) with initial credit $\mathsf{Cr}(p,q)$. We define a strategy σ for Eve in the JG as follows. For $(p,q),(q,t) \in W^{\mathrm{JG}}$, let $q_{p,t} \in Q$ denote the state such that $\mathsf{Cr}(p,q_{p,t}) + \mathsf{Cr}(q_{p,t},t)$ is minimal. For every $(p,t,a) \in Q^2 \times A$, if $(p,q),(p,t) \in W^{\mathrm{JG}}$, we then set $\sigma((p,t,a)) = \sigma_{(p,q_{p,t})}((p,q_{p,t})(p,q_{p,t},a))$. We then claim that for every $(p,q)(q,t) \in W^{\mathrm{JG}}$, the strategy σ is winning for Eve in the EGR starting from (p,t) with initial credit $\mathsf{Cr}(p,q_{p,t}) + \mathsf{Cr}(q_{p,t},t)$. □

Another property which will be useful in the sequel is that, all the a-successors of a state $p \in Q^{\mathrm{JG}}$ are related (by W^{JG}) to the a-successor chosen by a winning strategy for Eve.

Lemma 24. *For all $(p,q) \in W^{\mathrm{JG}}$ and $a \in A$, let σ^{JG} be a winning strategy for Eve in the JG from (p,q) with initial credit $c \in \mathbb{N}$, and let $(p,q,p',a) = \sigma^{\mathrm{JG}}((p,q)(p,q,a))$. Then, for all $(t,a,p'') \in \Delta$ such that $t \in \{p,q\}$, it holds that $(p',p'') \in W^{\mathrm{JG}}$, and $\mathsf{Cr}(p',p'') \le c + w(p,a,p') - w(t,a,p'')$.*

Proof. Observe that from any $(p,q) \in W^{\mathrm{JG}}$ in the JG, after Adam has chosen a letter $a \in A$ and Eve a transition $(p,a,p') \in \Delta$, Adam could play joker and choose any transition $(p,a,p'') \in \Delta$ or (without playing joker) choose any transition (q,a,p''). Hence, for any winning strategy σ^{JG} for Eve in the JG played from (p,q) with initial credit c such that $\sigma^{\mathrm{JG}}((p,q)(p,q,a)) = (p,q,p',a)$, for any $(t,a,p'') \in \Delta$ such that $t \in \{p,q\}$, reaching (p',p'') is consistent with σ^{JG}. It follows that σ^{JG} must be winning for Eve from (p',p'') with initial credit $c_1 = c + w(p,a,p') - w(t,a,p'')$. If $c_1 \le |V|\mu_{\max}$, we are done. Otherwise, by Proposition 20, there is a strategy σ' winning for Eve from (p',p'') with initial credit $|V|\mu_{\max}$. From the definition of W^{JG} we get that $(p',p'') \in W^{\mathrm{JG}}$ as required, and the result follows. ◻

Corollary 25. *If there are winning strategies $\sigma_1^{\mathrm{JG}}, \sigma_2^{\mathrm{JG}}$ for Eve in the JG with initial credit $|V|\mu_{\max}$ from $(p,q_1),(p,q_2)$, respectively, such that $\sigma_1^{\mathrm{JG}}((p,q_1)(p,q_1,a)) = (p,q_1,p_1,a)$ and $\sigma_2^{\mathrm{JG}}((p,q_2)(p,q_2,a)) = (p,q_2,p_2,a)$ for some $a \in A$, then $(p_1,p_2),(p_2,p_1) \in W^{\mathrm{JG}}$.*

Finally, we note that by following a winning strategy for Eve in the JG, we are sure all alternative runs of an automaton are related (by W^{JG}) to the run built by Eve.

Lemma 26. *For all play prefixes $(p_0,q_0)(p_0,q_0,a_0)\ldots(p_{n-1},q_{n-1},p_n,a_{n-1})(p_n,q_n)$ consistent with a winning strategy for Eve in the JG from (p_0,q_0) with initial credit $|V|\mu_{\max}$, for all runs $p_0 a_0 p'_1 a_1 \ldots a_{n-1} p'_n$ of $\mathcal{N}$ on $a_0 \ldots a_{n-1}$ we have that $(p_n,p'_n) \in W^{\mathrm{JG}}$.*

Proof. Let σ_1^{JG} denote the winning strategy from the claim.

First, it is easy to show by induction that $(p_i,q_i) \in W^{\mathrm{JG}}$ for all $0 \le i \le n$. Intuitively, using the fact that σ_1^{JG} is winning for Eve with initial credit $|V|\mu_{\max}$ from (p_0,q_0) we get that for any (p_i,q_i) the strategy σ_1^{JG} is winning for her with some initial credit. Then, by Proposition 20, there is another strategy σ' that is winning from (p_i,q_i) with initial credit $|V|\mu_{\max}$.

We will now argue, by induction, that $(p_i,p'_i) \in W^{\mathrm{JG}}$ for all $0 < i \le n$. For the base case, it should be clear that $(p_1,p'_1) \in W^{\mathrm{JG}}$. This follows from Lemma 24. Hence, we can assume the claim holds for some $0 < i < n$. By definition of W^{JG} we have that Eve has a winning strategy σ_2^{JG} in the JG from (p_i,p'_i) with initial credit $|V|\mu_{\max}$. It follows from Corollary 25 that $(p_{i+1},t) \in W^{\mathrm{JG}}$ where $(p_i,q_i,p_{i+1},a_i) = \sigma_1^{\mathrm{JG}}((p_0,q_0)\ldots(p_i,q_i,a_i))$ and $(p_i,q_i,t,a_i) = \sigma_2^{\mathrm{JG}}((p_0,p_0)\ldots(p_i,p'_i,a_i))$. Using Lemma 24 we get that $(t,p'_{i+1}) \in W^{\mathrm{JG}}$. Now, by transitivity of W^{JG} (see Lemma 23) we conclude that $(p_{i+1},p'_{i+1}) \in W^{\mathrm{JG}}$. The claim then follows by induction. ◻

We now prove that if Eve wins the JG played on $\mathcal{N}$, then the automaton $\mathcal{N}$ is determinizable. In order to do so, we first prove that $\mathcal{N}$ is $2|V|\mu_{\max}$-bounded.

Proposition 27. *Let $\mathcal{N} = (Q,I,A,\Delta,w,F)$ be such that Eve wins the JG played on $\mathcal{N}$. Then $\mathcal{N}$ is $2|V|\mu_{\max}$-bounded.*

Proof. Let $\varrho_p = p_0 a_0 p_1 \ldots a_{n-1} p_n$ be a maximal accepting run of $\mathcal{N}$, let $i \in \{0,\ldots,n\}$, and let $\varrho_q = q_0 a_0 q_1 \ldots a_{i-1} q_i$ be an initial run. Let us prove that $w(\varrho_q) - w(p_0 a_0 p_1 \ldots a_{i-1} p_i) \le 2|V|\mu_{\max}$. Let $\varrho_{p,1}$ denote the run $p_0 a_0 p_1 \ldots a_{i-1} p_i$, let $\varrho_{p,2}$ denote the run $p_i a_i p_{i+1} \ldots a_{n-1} p_n$. First, let σ_1^{JG} be a winning strategy for Eve in the JG from (p'_0,q_0) (for some $p'_0 \in I$) with initial credit $|V|\mu_{\max}$. Let $\varphi_q = (p'_0,q_0)(p'_0,q_0,a_0)(p'_0,q_0,p'_1,a_0)(p'_1,q_1)\ldots(p'_i,q_i)$ be the play prefix consistent with σ_1^{JG} that corresponds to Adam playing the word $a_0 \ldots a_{i-1}$ and choosing the states from the run ϱ_q. Since σ_1^{JG} is winning, Adam cannot enforce a negative energy level, in other words $\mathsf{EL}_{|V|\mu_{\max}}(\varphi_q) \ge 0$, hence: $w(\varrho_q) - w(p'_0 a_0 p'_1 \ldots a_{i-1} p'_i) \le |V|\mu_{\max}$. Second, by Lemma 26, $(p'_i,p_i) \in W^{\mathrm{JG}}$, hence Eve has a winning strategy σ_2^{JG} in the JG starting from (p'_i,p_i) with initial credit $|V|\mu_{\max}$. Let $\varphi_{p,2} = (p'_i,p_i)(p'_i,p_i,a_i)(p'_i,p_i,p'_{i+1},a_i)(p'_{i+1},p_{i+1})\ldots(p'_n,p_n)$ be the play prefix consistent with σ_2^{JG} that corresponds to Adam playing the word $a_i \ldots a_n$ and choosing the states from the run $\varrho_{p,2}$. Since σ_2^{JG} is winning, p'_n is a final state and (for the same reason as above) we have $w(\varrho_{p,2}) - w(p'_i a_i p'_{i+1} \ldots a_{n-1} p'_n) \le |V|\mu_{\max}$. Finally, since ϱ_p is maximal by hypothesis, $w(p'_0 a_0 p'_1 \ldots a_{n-1} p'_n) \le w(\varrho_p)$. Since $w(\varrho_p) = w(\varrho_{p,1}) + w(\varrho_{p,2})$, the result follows. ◻

Since, by definition of the JG, both $|V|$ and $\mu_{\max}$ are polynomial w.r.t. $|Q|$ and $w_{\max}$, using Proposition 4 gives us the following.

Theorem 6. *Given an automaton $\mathcal{N} = (Q,I,A,\Delta,w,F)$, if Eve wins the JG played on $\mathcal{N}$, then there exists a deterministic automaton $\mathcal{D}$ such that $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$, and whose size and maximal weight are polynomial w.r.t. $w_{\max}$, and exponential w.r.t. $|Q|$.*

With the results above, we are now in position to prove an EXPTIME upper bound for the r-regret problem.

Proof of Theorem 5. Given an automaton $\mathcal{N}$ and $r \in \mathbb{N}$, we first determine whether Eve wins the JG played on $\mathcal{N} = (Q,I,A,\Delta,w,F)$. To do so, we determine the winner of the corresponding EGR from all $(p,q) \in I^2$ with initial credit $|V|\mu_{\max}$. We can then, in polynomial time, decide if there exists $p \in I$ such that, for all $q \in I$, Eve can win the EGR from (p,q). If the latter does not hold, then by contrapositive of Lemma 22, $\mathcal{N}$ is not r-regret determinizable. Otherwise, we construct $\mathcal{D}$ such that $[\![\mathcal{D}]\!] = [\![\mathcal{N}]\!]$ and $\mathcal{D}$ is deterministic using Theorem 6. Finally, we use $\mathcal{D}$ to decide if Eve wins the r-regret game using Proposition 21. Since $\mathcal{D}$ is of size exponential w.r.t. to $\mathcal{N}$ but its maximal weight is polynomial w.r.t. $w_{\max}$, the resulting energy game without resets can be solved in exponential time by Proposition 20. ◻

As a corollary, we obtain that the existential version of the r-regret problem is also decidable. More precisely, using the techniques we have just presented, we are able to decide the question: does there exist $r \in \mathbb{N}$ such that a given automaton $\mathcal{N}$ is r-regret determinizable? The algorithm to decide the latter question is almost identical to the one we give for the r-regret problem. The only difference lies in the last step, that is, the energy game without resets constructed from the deterministic version of the automaton that one can obtain from the JG. Instead of using a function of r as initial credit, we ask if Eve wins the energy game with initial credit $|V|\mu_{\max}$—we also remove the gadget using vertices $\bot_1^r, \bot_2^r$ which ensure a regret of at most r.

Theorem 7. *Given an automaton, deciding whether there exists $r \in \mathbb{N}$ such that it is r-regret determinizable is in* EXPTIME.

B. Lower bound

In this section we argue that the complexity of the algorithm we described in the previous section is optimal. More precisely, the r-regret problem is EXPTIME-hard even if the regret threshold r is fixed.

Theorem 8. *Deciding the r-regret problem for a given automaton is* EXPTIME*-hard, even for fixed $r \in \mathbb{N}_{>0}$.*

Observe that, in regret games, Eve may need to keep track of all runs of the given automaton on the word α which is being "spelled" by Adam. Indeed, if she has so far constructed the run ϱ and Adam chooses symbol a next, then her choice of transition to extend ϱ may depend on the set of states at which alternative runs of the automaton on α end. The set of all such configurations is exponential.

Our proof of the r-regret problem being EXPTIME-hard makes sure that Eve has to keep track of a set of states as mentioned above. Then, we encode configurations of a binary counter into the sets of states so that the set of states at which Eve *believes* alternative runs could be at, represent a valuation of the binary counter. Finally, we give gadgets which simulate addition of constants to the current valuation of the counter. These ingredients allow us to simulate Countdown games [27] using regret games. As the former kind of games are EXPTIME-hard, the result follows. The same reduction has been used to show that regret minimization against *eloquent adversaries* in several *quantitative synthesis games* is EXPTIME-hard [16].

V. Further research directions

When the regret r is given, the r-regret determinization problem is EXPTIME-complete. When r is not given, the problem is in EXPTIME but we did not find any lower bound other than PTIME-hardness. Characterizing the precise complexity of this problem is open.

The latter is related to the following question. From our decision procedure for solving the existential regret problem, it appears that if a WA is r-regret determinizable for some r, it is also r'-regret determinizable for some r' that depends exponentially on the WA. So far, we have not found any family of WA that exhibit exponential regret behaviour, and the best lower bound we have is quadratic in the size of the WA.

Finally, we would like to investigate the notions of delay- and regret-determinization for other measures, such as discounted sum [2] or ratio [3]. These notions also make sense for other problems, such as comparison and equality of weighted automata (which are undecidable for max-plus automata), and disambiguation [10].

References

[1] M. Droste, W. Kuich, and H. Vogler, *Handbook of Weighted Automata*, 1st ed. Springer Publishing Company, Incorporated, 2009.
[2] K. Chatterjee, L. Doyen, and T. A. Henzinger, "Quantitative languages," *ACM Trans. Comput. Log.*, vol. 11, no. 4, 2010.
[3] E. Filiot, R. Gentilini, and J.-F. Raskin, "Quantitative languages defined by functional automata," in *CONCUR*, ser. LNCS, vol. 7454, 2012.
[4] D. Krob, "The equality problem for rational series with multiplicities in the tropical semiring is undecidable," *LNCS*, vol. 623, pp. 101–??, 1992.
[5] L. Daviaud, P. Guillon, and G. Merlet, "Comparison of max-plus automata and joint spectral radius of tropical matrices," *arXiv preprint arXiv:1612.02647*, 2016.
[6] S. Almagor, U. Boker, and O. Kupferman, "What's decidable about weighted automata?" in *ATVA*, ser. LNCS, vol. 6996, 2011, pp. 482–491.
[7] E. Filiot, R. Gentilini, and J.-F. Raskin, "Finite-valued weighted automata," in *FSTTCS*, 2014, pp. 133–145.
[8] K. Chatterjee, L. D. 0001, H. Edelsbrunner, T. A. Henzinger, and P. Rannou, "Mean-payoff automaton expressions," in *CONCUR*, ser. LNCS, vol. 6269. Springer, 2010, pp. 269–283.
[9] M. Droste and P. Gastin, "Weighted automata and weighted logics," *TCS*, vol. 380(1-2):69–86, 2007.
[10] D. Kirsten and S. Lombardy, "Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata," in *STACS*, ser. LIPIcs, vol. 3. Schloss Dagstuhl, 2009, pp. 589–600.
[11] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, no. 4, 1997.
[12] B. Aminof, O. Kupferman, and R. Lampert, "Rigorous approximated determinization of weighted automata," *TCS*, vol. 480(104–117), 2013.
[13] T. Colcombet and L. Daviaud, "Approximate comparison of functions computed by distance automata," *TCS*, vol. 58(4):579–613, 2016.
[14] M.-P. Béal, O. Carton, C. Prieur, and J. Sakarovitch, "Squaring transducers: an efficient procedure for deciding functionality and sequentiality," *TCS*, vol. 292, no. 1, pp. 45–63, 2003.
[15] J. Sakarovitch and R. de Souza, "On the decomposition of k-valued rational relations," in *STACS*, ser. LIPIcs, vol. 1, 2008, pp. 621–632.
[16] P. Hunter, G. A. Pérez, and J.-F. Raskin, "Reactive synthesis without regret," *Acta Informatica*, pp. 1–37, 2016.
[17] T. A. Henzinger and N. Piterman, "Solving games without determinization," in *CSL*, 2006, pp. 395–410.
[18] B. Aminof, O. Kupferman, and R. Lampert, "Reasoning about online algorithms with weighted automata," *ACM Trans. on Algorithms*, 2010.
[19] D. Kuperberg and M. Skrzypczak, "On determinisation of good-for-games automata," in *ICALP*, ser. LNCS, vol. 9135, 2015, pp. 299–310.
[20] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba, "Infinite runs in weighted timed automata with energy constraints," in *FORMATS*, ser. LNCS 5215. Springer, 2008, pp. 33–47.
[21] E. Filiot, I. Jecker, C. Löding, and S. Winter, "On equivalence and uniformisation problems for finite transducers," in *ICALP*, ser. LIPIcs, vol. 55. Schloss Dagstuhl, 2016, pp. 125:1–125:14.
[22] F. Klein and M. Zimmermann, "How much lookahead is needed to win infinite games?" in *ICALP*, ser. LNCS, vol. 9135, 2015, pp. 452–463.
[23] A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga, "Resource interfaces," in *EMSOFT*, ser. LNCS, vol. 2855, 2003.
[24] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J.-F. Raskin, "Faster algorithms for mean-payoff games," *FMSD*, vol. 38(2):97-118, 2011.
[25] B. Aminof and S. Rubin, "First cycle games," in *Proceedings of Strategic Reasoning*, 2014, pp. 83–90.
[26] K. R. Apt and E. Grädel, *Lectures in game theory for computer scientists*. Cambridge University Press, 2011.
[27] M. Jurdzinski, J. Sproston, and F. Laroussinie, "Model checking probabilistic timed automata with one or two clocks," *LMCS*, vol. 4, no. 3, 2008.