The Per-character Cost of Repairing Word Languages

Michael Benedikt[a], Gabriele Puppis[b], Cristian Riveros[c]

[a] *Department of Computer Science – University of Oxford*
[b] *CNRS / LaBRI – University of Bordeaux*
[c] *Department of Computer Science – Pontificia Universidad Catolica de Chile*

Abstract

We show how to calculate the maximum number of edits per character needed to convert any string in one regular language to a string in another language. Our algorithm makes use of a local determinization procedure applicable to a subclass of distance automata. We then show how to calculate the same property when the editing needs to be done in streaming fashion, by a finite state transducer, using a reduction to mean-payoff games. In this case, we show that the optimal streaming editor can be produced in P.

Keywords: Repair, asymptotic normalized cost, edit distance, distance automata, regular languages.

1. Introduction

Edit distance is a well-studied metric between strings, measuring how many operations are needed to get from one string to another. In this paper we look for natural (asymmetric) analogs for regular languages: how many edits does it require to get from a word in regular language R to a word in regular language T, in the worst case? Our notation is motivated by considering R to be a *restriction* – a constraint that the input is guaranteed to satisfy – and T to be a *target* – a constraint that we want to enforce.

In a prior work [1], we considered the basic question of whether one can get from a word in R to a word in T with a finite, uniformly bounded number

Email addresses: michael.benedikt@cs.ox.ac.uk (Michael Benedikt), gabriele.puppis@labri.fr (Gabriele Puppis), criveros@ing.puc.cl (Cristian Riveros)

of edits. One of the main results of [1] was a characterization of the pairs (R, T) for which such a uniform bound exists.

Example 1. *Consider the regular languages $R = a^* \ b^*$ and $T = a^* \ c \ b^*$. Clearly, any string in R can be converted to a string in T with at most* 1 *edit operation.*

Such a bound, when it exists, shows that the language R is "quite close to being a subset of T" – the gap between strings in R and strings in T is small. However, having a uniform bound on the number of edits is a strong requirement. In this paper we look not at the absolute number of edits required to get from R to T, but rather at the *percentage* of letters that need to be edited.

Example 2. *Consider the languages $R = (a + b)^*$ and $T = (ab)^*$. Roughly, for any pair of consecutive occurrences of the letter a in the input, we will have to perform one edit in order to ensure alternation in the output. In particular, the number of edits required to get from a string in R to a string in T is unbounded. On the other hand, it is clear that, in the worst case (i.e., a^{2n}), one needs to edit approximately half of the letters in order to produce a string in T – in this case, we say that all strings in R can be repaired into T with* normalized cost *at most $\frac{1}{2}$.*

We measure the gap from a restriction language R to a target language T via the worst case, over all strings $u \in R$, of the number of edits needed to bring u into T divided by the length of u. Since we want the definition to be robust to a finite number of outliers, we take the limit of this quantity as the strings are of larger and larger length – this is the *asymptotic* (*normalized*) *cost* in getting from R to T. This gives us a measure of the distortion needed to get from R to T, lying always between 0 and 1.

In our prior work [1] we have given algorithms for determining when the absolute cost of repairing R into T is uniformly bounded, and in this case compute a bound. Similarly, the main result of the present paper is an algorithm that computes the asymptotic cost of repairing R into T. The techniques used for the asymptotic cost analysis are radically different from those used for the bounded repair problem. Specifically, they rely on ideas from the theory of *distance automata* [2], and in particular on an application of determinization of distance automata, closely related to Mohri's determinization procedure [3].

We then turn to the setting where the repair is required to be done in streaming fashion, producing the edits immediately on seeing the input letter. We measure a streaming repair processor by the number of edits per character it requires to get from any string in R to a string in T, again looking at the limit as the string length gets large. We accordingly define the *streaming asymptotic cost* to be the optimal cost of a streaming processor. We show that this quantity can also be calculated effectively, using techniques from mean-payoff games.

Example 3. *Consider* $R = (a+b)\,c^*(a^+ + b^+)$ *and* $T = a\,c^*a^+ + b\,c^*b^+$. *One can get from* R *to* T *by only editing the initial letter: so the asymptotic cost is* 0. *However, a streaming strategy must commit to changing the initial letter or leaving it be: if it makes the "incorrect" choice, it will have to edit an unbounded final segment; thus the streaming asymptotic cost is* 1.

The above two results give us the ability to compare the cost one should pay in editing strings in R to strings in T with an arbitrary processor with the cost when we are restricted to use a streaming processor. If these are the same, it shows that streaming processors that edit strings in R to T can approximate arbitrary processors in worst-case behavior.

In summary our contributions are:

- We present an algorithm for calculating the asymptotic normalized cost of repairing strings in regular language R to strings in regular language T, based on locally determinizing a subclass of distance automata.
- We give an algorithm for calculating the optimal asymptotic normalized cost achieved using a streaming editing algorithm.

Organization. Section 2 gives the preliminaries, while Section 3 defines the basic problems. Section 4 studies the problem of computing the asymptotic cost in the non-streaming case, while Section 5 deals with the streaming case. Section 6 gives conclusions.

2. Preliminaries

Given a word w over an alphabet Σ, we denote by $|w|$ its length and, given two positions $1 \le i \le j \le |w|$, we denote by $w[i]$ (resp., $w[i \dots j]$) the i-th symbol of w (resp., the infix of w starting at position i and ending at position j).

Automata. Non-deterministic finite state automata (shortly, *NFA*) will be represented by tuples of the form $\mathcal{A} = (\Sigma, Q, E, I, F)$, where Σ is a finite alphabet, Q is a finite set of states, $E \subseteq Q \times \Sigma \times Q$ is a transition relation, and $I, F \subseteq Q$ are sets of initial and finite states. The notions of run and accepted word are the usual ones. $\mathscr{L}(\mathcal{A})$ is the language recognized by $\mathcal{A}$. If $\mathcal{A}$ is a deterministic finite state automaton (*DFA*), then we usually denote the unique initial state by q_0 and turn its transition relation E into a partial function δ from $Q \times \Sigma^*$ to Q defined by $\delta(q, \varepsilon) = q$ and $\delta(q, a\ u) = \delta(q', u)$ iff $(q, a, q') \in E$.

For technical reasons, it is convenient to assume that an automaton is *trimmed*, namely, all its states are reachable from some initial states (i.e., they are accessible) and they can reach some final states (i.e., they are co-accessible). It is worth noticing that, since the decision problems we are going to deal with are at least NLogSpace-hard and since states of automata that are not accessible or not co-accessible can be pruned using some simple NLogSpace reachability analysis, this assumption will have no impact on our complexity results.

Since automata can be viewed as directed (labeled) graphs, we inherit the standard definitions and constructions in graph theory. In particular, given an automaton $\mathcal{A} = (\Sigma, Q, E, I, F)$ and a state $q \in Q$, we denote by $\mathcal{C}(q)$ the *strongly connected component* (shortly, SCC) of $\mathcal{A}$ that contains all states mutually reachable from q. We say that a component C of $\mathcal{A}$ is *final* if it can reach a final state (possibly outside C). Given a set C of states of $\mathcal{A}$ (e.g., a SCC), we denote by $\mathcal{A}|C$ the NFA obtained by restricting $\mathcal{A}$ to the set C and by letting the new initial and final states be all and only the states in C (note that if C consists of a single transient state, then the language $\mathscr{L}(\mathcal{A}|C)$ recognized by the subautomaton $\mathcal{A}|C$ is empty). Finally, we denote by $\mathsf{dag}(\mathcal{A})$ the directed acyclic (unlabeled) graph of the SCCs of $\mathcal{A}$ and by $\mathsf{dag}^*(\mathcal{A})$ the graph obtained from the symmetric and transitive closure of the edges of $\mathsf{dag}(\mathcal{A})$.

Transducers. A (real-time sub-sequential) *transducer* is a tuple $\mathcal{S} = (\Sigma, \Delta, Q, \delta, q_0, \Omega)$, where Σ is a finite input alphabet, Δ is a finite output alphabet, Q is a finite set of states, δ is a partial transition function from $Q \times \Sigma$ to $\Delta^* \times Q$, q_0 is an initial state, and Ω is a partial function from Q to Δ^*. For every input word $u = a_1 \dots a_n \in \Sigma^*$, there is at most one run of $\mathcal{S}$ on u of the form

$$q_0 \xrightarrow{a_1/v_1} q_1 \xrightarrow{a_2/v_2} \dots \xrightarrow{a_n/v_n} q_n \xrightarrow{\varepsilon/v_{n+1}}$$

where $\delta(q_i, a_i) = (v_i, q_{i+1})$ for all $0 \leq i < n$ and $\Omega(q_n) = v_{n+1}$. In such a case, we define the *output* of $\mathcal{S}$ on u to be the word $\mathcal{S}(u) = v_1 v_2 \dots v_n v_{n+1}$ (observe that the transducer outputs an additional, possibly empty, word to be added on at the end of the computation).

Transducers as above produce an output word immediately on reading an input character. We will also consider transducers with a bounded amount of "delay". A *k-lookahead* transducer, with $k \in \mathbb{N}$, is as above, but where the transition function δ now has input in $Q \times \Sigma_\perp^{k+1}$, where $\Sigma_\perp = \Sigma \cup \{\perp\}$ and $\perp \notin \Sigma$. Given an input word u and a position $1 \leq i \leq |u|$ in it, we denote by $\vec{u}_i$ the $(k+1)$-character subword of $u \perp^k$ that starts at position i and ends at position $i + k$. The output of a k-lookahead transducer $\mathcal{S}$ on an input u of length n is the unique word $v = v_1 v_2 \dots v_n v_{n+1}$ for which there exists a sequence of states $q_0, ..., q_n$ satisfying $\delta(q_i, \vec{u}_i) = (v_i, q_{i+1})$, for all $1 \leq i \leq n$, and $\Omega(q_n) = v_{n+1}$. Clearly, a 0-lookahead transducer is simply a standard (real-time sub-sequential) transducer.

3. Problem setting

Given two words $u \in \Sigma^*$ and $v \in \Delta^*$, we denote by

$$\mathsf{dist}(u, v)$$

the *Levenshtein distance* (henceforth, *edit distance*) between u and v, which is defined as the length of a shortest sequence s of edit operations (e.g., deleting a single character, modifying a single character, and inserting a single character) that transforms u into v [4].

We are interested in quantifying how difficult it is to edit a word in one language to obtain a word in another language. That is, we have finite alphabets Σ and Δ and regular languages $R \subseteq \Sigma^*$ and $T \subseteq \Delta^*$, called the *restriction* and *target* languages, respectively. We would like to edit any string that is known to belong to the restriction language R into a string in the target language T. We will also consider the special case where $R = \Sigma^*$, which we denote as the *unrestricted case*.

A *repair strategy* for two languages R and T is any function from R to T. For a repair strategy f and a word $u \in R$, we define the (absolute) *cost of f on u*, denoted $\mathsf{cost}(u, f)$, as the edit distance between u and $f(u)$. In [1] we have given a characterization of those pairs (R, T) of languages for which there exist a repair strategy f whose (absolute) cost is finite and uniformly

bounded. In this paper we will not be concerned with the absolute cost of repairing a word, since the worst case of this is often infinite. We consider instead a notion of repair cost between words that looks at the *percentage* of symbols in a word that need to be edited. More precisely, given a repair strategy f for R and T and given a word $u \in R$, we define the *normalized cost of f on u* as the ratio between the cost of f on u and the length of u:

$$\mathsf{ncost}(u, f) \ =^{\text{def}} \ \frac{\mathsf{cost}(u, f)}{|u|}.$$

In order to measure the asymptotic behavior of the normalized cost, we define the *asymptotic cost of f* as the *limit superior* of the normalized cost when the length of words in the restriction language tends to infinity:

$$\mathsf{acost}(R, f) \ =^{\text{def}} \ \lim_{n\to\infty} \sup_{\substack{u\in R \\ |u|\geq n}} \mathsf{ncost}(u, f).$$

Accordingly, we define the *asymptotic repair cost* $\mathsf{acost}(R, T)$ for two languages R and T as the *minimum* of $\mathsf{acost}(R, f)$ taken over all repair strategies f for R and T. Note that $\mathsf{acost}(R, T)$ can be equally defined by

$$\mathsf{acost}(R, T) \ = \ \lim_{n\to\infty} \sup_{\substack{u\in R \\ |u|\geq n}} \min_{v\in T} \frac{\mathsf{dist}(u, v)}{|u|}.$$

Example 2 (continued). *Consider again the languages $R = (a + b)^*$ and $T = (ab)^*$ of Example 2 in the introduction. Recall that, in the worst case (i.e., a^{2n}), one needs to edit approximately half of the letters in order to produce a string in T. This shows that $\mathsf{acost}(R, T) = \frac{1}{2}$.*

Remark 1. *It is easy to see that the asymptotic repair cost $\mathsf{acost}(R, T)$ of any pair of languages ranges over the interval $[0, 1]$ of the real numbers. Indeed, for large words in the restriction language R, one can modify and delete the letters to create shorter words in the target language T, and thus the resulting editing cost is always less than the length of the input word.*

Ideally, we are interested in computing the asymptotic cost $\mathsf{acost}(R, T)$ for any pair of regular languages R and T, provided that this number is rational. We will indeed show that this is the case and describe a procedure that computes the asymptotic cost.

Streaming vs non-streaming. We know from [4] that there is a dynamic programming algorithm that, given a word u and a regular target language T represented by means of a finite state automaton $\mathcal{T}$, computes in time $\mathcal{O}(|u| \cdot |\mathcal{T}|)$ an optimal edit sequence that transforms u into some word in T. In particular, this shows that optimal repair strategies can be described by functions of fairly low complexity.

Sometimes it is desirable to have repair strategies that are in even more limited classes. Perhaps the ideal case is when we can repair R into T with a one-pass algorithm, that is, using a real-time sub-sequential transducer. Recall that a real-time sub-sequential transducer defines a word-to-word function; if this function happens to produce a word in T for every input $u \in R$, then we say that it is a *streaming repair strategy* for R and T. Similarly, we can consider k-lookahead transducers, with $k \in \mathbb{N}$: this type of transducer outputs words on the basis of its current state and an input $(k+1)$-character window that represents a substring of u of the form $u[i] \dots u[i+k]$, where $u[i]$ is either the i-th symbol of w, if $i \leq |u|$, or a dummy symbol $\perp$, if $i > |u|$. Accordingly, we talk about a *k-lookahead streaming repair strategy* for R and T.

Given a k-lookahead streaming edit strategy $\mathcal{S}$ for R and T and given a word $u \in R$, we can define the (absolute) *cost of $\mathcal{S}$ on u* in two ways:

1. letting $q_0 \xrightarrow{a_1/v_1} q_1 \xrightarrow{a_2/v_2} \dots \xrightarrow{a_n/v_n} q_n \xrightarrow{v_{n+1}}$ be the run of $\mathcal{S}$ on u, we define the *aggregate cost of $\mathcal{S}$ on u*, denoted $\mathsf{cost}^{\mathsf{aggr}}_{\mathcal{S}}(u)$, to be the length of the final output v_{n+1} plus the sum, over all indices $1 \leq i \leq n$, of $\mathsf{dist}(a_i, v_i)$, where $\mathsf{dist}(a_i, v_i)$ is 1 if v_i is empty, $|v_i| - 1$ if a_i occurs in v_i, and $|v_i|$ otherwise;

2. considering the transducer $\mathcal{S}$ as a repair strategy, we define the *edit cost of $\mathcal{S}$ on u*, denoted $\mathsf{cost}^{\mathsf{edit}}_{\mathcal{S}}(u)$, to be simply the edit distance between u and the output $\mathcal{S}(u)$.

The first notion of cost considers the distortions performed in producing the input from the output – it is equivalent to considering the transducer as producing edit sequences rather than strings and counting the number of edits produced. The second notion of cost is global and it considers only the output and not its production (clearly, the edit cost never exceeds the aggregate cost). These two models of cost can be very different in general. As an example, consider a transducer $\mathcal{S}$ on the input alphabet $\Sigma = \{a, b\}$ that swaps a's and b's. On the string $u_n = (ab)^n$, the aggregate cost is $2n$ since $\mathcal{S}$

changes each letter, but the edit distance between u and $\mathcal{S}(u)$ (i.e., the edit cost of $\mathcal{S}$ on u in our sense) is only 2.

In the streaming setting, we will mainly focus on the model of aggregate cost, as for the model of edit cost we do not have any interesting result. Formally, we define the *asymptotic* (*normalized aggregate*) *cost* of a k-lookahead streaming strategy $\mathcal{S}$ for R and T, as

$$\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(R,T) \;=^{\text{def}}\; \lim_{n\to\infty} \sup_{\substack{u\in R\\ |u|\geq n}} \frac{\mathsf{cost}^{\mathsf{aggr}}_{\mathcal{S}}(u)}{|u|}$$

where $\mathsf{cost}^{\mathsf{aggr}}_{\mathcal{S}}(u)$ is defined above. Similarly, we define the *asymptotic k-lookahead streaming cost of R and T*, denoted $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$, as the *infimum* of $\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(R,T)$ taken over all k-lookahead streaming repair strategies $\mathcal{S}$ for R and T.

We remark that, a priori, the infimum in the previous definition cannot be replaced by a minimum: it is conceivable that the asymptotic aggregate costs of the k-lookahead streaming repair strategies for R and T are arbitrary close to $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$, but never achieve this value. In fact, in Section 5 we will show that this is not the case, as we can enforce, without loss of generality, a uniform bound to the memory of the k-lookahead streaming repair strategies.

Example 2 (continued). *Consider again the languages $R = (a+b)^*$ and $T = (ab)^*$ of Example 2 and recall that $\mathsf{acost}(R,T) = \frac{1}{2}$. We show that the asymptotic 0-lookahead streaming aggregate cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(R,T)$ is higher than the asymptotic cost in the nonstreaming case. Suppose that $\mathcal{S}$ is a 0-lookahead streaming repair strategy for R and T. One can inductively construct arbitrarily long words $u_n = a_1\, a_2\, \ldots\, a_n \in R$ such that if*

$$q_0 \xrightarrow{a_1/v_1} q_1 \xrightarrow{a_2/v_2} \cdots \xrightarrow{a_n/v_n} q_n$$

is a partial run of $\mathcal{S}$ on u_n, then each next letter a_n is equal to the last letter of the prefix $v_1\, v_2\, \ldots\, v_{n-1}$ of the output of $\mathcal{S}$ (if $v_1 v_2 \ldots v_{n-1}$ is empty, then a_n can be chosen arbitrarily). It is easy to see that the aggregate cost induced by the run of $\mathcal{S}$ on u_n is at least $n-1$, whence $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(R,T) = 1$. However, if we consider the model of edit cost, then we have $\mathsf{acost}^{\mathsf{edit}}_{0-\mathsf{lookahead}}(R,T) = \frac{1}{2}$ (in fact, any streaming strategy for R and T achieves this asymptotic edit cost).

Observe that, in general, the k-lookahead streaming asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$ associated with two languages R and T is a non-increasing function of the lookahead parameter $k \in \mathbb{N}$ and it is bounded from below by the non-streaming asymptotic cost $\mathsf{acost}(R,T)$.

Towards the end of Section 5 we will consider the problem of computing the limit of the streaming asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$ as the lookahead parameter k gets bigger. We are not able to compute the exact value of this limit, nor to prove that $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$ stabilizes for sufficiently large k, as it seems reasonable. However, we will show how to solve a simpler problem, that consists of deciding given two DFA R and T and a rational threshold ν whether there is $k \in \mathbb{N}$ such that $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T) < \nu$.

4. Asymptotic cost in the non-streaming case

In this section, we study the problem of computing the asymptotic cost in the non-streaming setting. We begin with some background on distance automata, which will play a key role in the main characterization result.

4.1. Distance automata computing the edit cost

Intuitively, a distance automaton [2] is a transducer $\mathcal{D}$ that receives as input a finite word u and outputs a corresponding cost $\mathcal{D}(u)$ in $\mathbb{N} \cup \{\infty\}$. Distance automata can be equivalently defined using two different presentations based, respectively, on matrices of transition costs and on transition relations. Here, we adopt the latter type of presentation, which is more convenient for our purposes (e.g., it eases the definition of a run of a distance automaton).

Formally, a *distance automaton* is a tuple $\mathcal{D} = (\Sigma, Q, E, I, F)$, where Q is a finite set of states, $E \subseteq Q \times \Sigma \times \mathbb{N} \times Q$ is a finite transition relation, I and F are some initial and final conditions described by partial functions from Q to $\mathbb{N}$ and representing the costs of beginning and ending a run with certain states. A *run* of $\mathcal{D}$ on a word $u \in \Sigma^*$ is a sequence

$$\gamma \;=\; (q_0, a_1, c_1, q_1)\ (q_1, a_2, c_2, q_2)\ \ldots\ (q_{n-1}, a_n, c_n, q_n)$$

of pairwise adjacent transitions in E that spell the input word $u = a_1 a_2 \ldots a_n$. The *cost* of the run γ is naturally defined by

$$\mathsf{cost}(\gamma) \;=^{\mathrm{def}}\; \sum\nolimits_{1 \le i \le n} c_i.$$

We denote by $\mathcal{D}(u)$ the *minimum* value $I(q_0) + \mathsf{cost}(\gamma) + F(q_n)$ among all states q_0 in the domain $\mathsf{dom}(I)$ of I, all states q_n in the domain $\mathsf{dom}(F)$ of F, and all runs γ of $\mathcal{D}$ on u that start in q_0 and end in q_n. We let $\mathcal{D}(u) = \infty$ if there are no such states q_0 and q_n, or if there is no run from q_0 to q_n.

When considering the edit distance of a word $u \in \Sigma^*$ to a regular language $T \subseteq \Delta^*$, it is fairly natural to express this value in terms of the cost computed by a distance automaton. By default, we assume that the target language T is recognized by an NFA $\mathcal{T} = (\Delta, Q, E, I, F)$. Given two states p, q of $\mathcal{T}$, we let $\mathcal{T}_{p,q}$ be the NFA obtained from $\mathcal{T}$ by letting p be the new initial state and q the new unique final state. The distance automaton that computes the edit distance of a word $u \in \Sigma^*$ to the target language $\mathscr{L}(\mathcal{T})$ is defined as $\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}} = (\Sigma, Q, E^{\mathsf{edit}}, I^{\mathsf{edit}}, F^{\mathsf{edit}})$, where

- E^{edit} is the set of all transitions of the form (p, a, c, q), with $p, q \in Q$, $a \in \Sigma$, q reachable from p in $\mathcal{T}$, and $c = \min\big\{\mathsf{dist}(a, v) \ : \ v \in \mathscr{L}(\mathcal{T}_{p,q})\big\}$,
- I^{edit} is the partial function that maps a state $q \in Q$ to the minimum among the values $\mathsf{dist}(\varepsilon, v)$, with $v \in \bigcup_{p \in I} \mathscr{L}(\mathcal{T}_{p,q})$ (if q is not reachable from some initial state of $\mathcal{T}$, then $I^{\mathsf{edit}}(q)$ is undefined),
- F^{edit} is the partial function that maps a state $p \in Q$ to the minimum among the values $\mathsf{dist}(\varepsilon, v)$, with $v \in \bigcup_{q \in F} \mathscr{L}(\mathcal{T}_{p,q})$ (if p cannot reach a final state of $\mathcal{T}$, then $F^{\mathsf{edit}}(p)$ is undefined).

One can easily show that $\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}}$ computes exactly the edit distance of a word $u \in \Sigma^*$ to the regular language $\mathscr{L}(\mathcal{T})$:

Proposition 1. *For every word $u \in \Sigma^*$, we have*

$$\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}}(u) \ = \ \min_{v \in \mathscr{L}(\mathcal{T})} \ \mathsf{dist}(u, v).$$

Proof. Let $\mathcal{T} = (\Delta, Q, E, I, F)$ be an NFA and let $\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}} = (\Sigma, Q, E^{\mathsf{edit}}, I^{\mathsf{edit}}, F^{\mathsf{edit}})$ be the corresponding distance automaton, as defined above. For this proof, it is convenient to introduce the notion of locally optimal run. We say that a run $\gamma = (q_1, a_1, c_1, q_2) \ \dots \ (q_n, a_n, c_n, q_{n+1})$ of $\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}}$ is *locally optimal* if it has the minimum cost among all runs of $\mathcal{D}^{\mathsf{edit}}_{\mathcal{T}}$ on the same word $u = a_1 \dots a_n$ that start in q_1 and end in q_{n+1}. Note that there can exist several locally optimal runs on the same word u with different costs (and, of course, with different beginning and ending states). We have the following characterization of the minimum cost:

Claim 1. *For every word $u \in \Sigma^*$ and every locally optimal run γ of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ on u that starts in p and ends in q, we have that*

$$\mathsf{cost}(\gamma) \;=\; \min_{v \in \mathscr{L}(\mathcal{T}_{p,q})} \mathsf{dist}(u, v).$$

Proof. The proof of the above claim is by induction on the length of the word u. If $u = \varepsilon$, then the claim follows easily. As for the inductive step, let us assume that the claim holds for u and let us prove it for $u\ a$, with $a \in \Sigma$. Let γ be a locally optimal run of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ on $u\ a$ that starts in p and ends in q. Moreover, for every state $r \in Q$, let γ_r be a locally optimal run of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ on u from p to r. Since γ is locally optimal, we have that its cost is the minimum among the cost of a run γ_r, with $r \in Q$, plus the cost of an a-labeled transition from r to q. From the inductive hypothesis, we have that

$$\mathsf{cost}(\gamma_r) \;=\; \min_{v \in \mathscr{L}(\mathcal{T}_{p,r})} \mathsf{dist}(u, v).$$

This shows that

$$\begin{aligned} \mathsf{cost}(\gamma) \;&=\; \min_{(r,a,c,q) \in E^{\mathsf{edit}}} \mathsf{cost}(\gamma_r) + c \\ &=\; \min_{(r,a,c,q) \in E^{\mathsf{edit}}} \; \min_{v \in \mathscr{L}(\mathcal{T}_{p,r})} \mathsf{dist}(u, v) + c. \end{aligned}$$

We now look at the definition of the transition relation E^{edit}. It contains all quadruples (r, a, c, q) such that $c = \min\left\{\mathsf{dist}(a, w) \;:\; w \in \mathscr{L}(\mathcal{T}_{r,q})\right\}$. This shows that

$$\begin{aligned} \mathsf{cost}(\gamma) \;&=\; \min_{r \in Q} \; \min_{v \in \mathscr{L}(\mathcal{T}_{p,r})} \; \min_{w \in \mathscr{L}(\mathcal{T}_{r,q})} \mathsf{dist}(u, v) + \mathsf{dist}(a, w) \\ &=\; \min_{v\, w \in \mathscr{L}(\mathcal{T}_{p,q})} \mathsf{dist}(u\ a,\ v\ w). \end{aligned}$$

□

To complete the proof of the proposition, it is sufficient to recall that for every word $u \in \Sigma^*$, $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}(u)$ is the minimum among the values $\mathsf{cost}(\gamma) + I^{\mathsf{edit}}(p) + F^{\mathsf{edit}}(q)$, for all states $p \in \mathsf{dom}(I^{\mathsf{edit}})$ and $q \in \mathsf{dom}(F^{\mathsf{edit}})$ and all (locally optimal) runs γ of $\mathcal{D}^{\mathsf{edit}}$ on u that start from p and end in q. We also recall that $I^{\mathsf{edit}}(p) = \min\left\{\mathsf{dist}(\varepsilon, v) \;:\; p' \in I,\ v \in \mathscr{L}(\mathcal{T}_{p',p})\right\}$ and $F^{\mathsf{edit}}(q) = \min\left\{\mathsf{dist}(\varepsilon, v) \;:\; q' \in F,\ v \in \mathscr{L}(\mathcal{T}_{q,q'})\right\}$. This implies that $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}(u)$ is the minimum among the values

$$\mathsf{dist}(\varepsilon, v_I) + \mathsf{dist}(u, v) + \mathsf{dist}(\varepsilon, v_F) \;=\; \mathsf{dist}(u,\ v_I\ v\ v_F),$$

where $v_I \in \bigcup_{p' \in I} \mathscr{L}(\mathcal{T}_{p',p})$, $v \in \mathscr{L}(\mathcal{T}_{p,q})$, $v_F \in \bigcup_{q' \in F} \mathscr{L}(\mathcal{T}_{q,q'})$, (hence $v_I\ v\ v_F \in \mathscr{L}(\mathcal{T})$), and $p, q \in Q$. This concludes the proof of the proposition. □

4.2. Shortcut property and determinizable components

Distance automata of the form $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ are a proper sub-class of all distance automata. In particular, they satisfy the *shortcut property*, formalized just below. Given a symbol $a \in \Sigma$ and two states p, q of a distance automaton $\mathcal{D}$, we write $p \xrightarrow{a} q$ to denote the existence in $\mathcal{D}$ of a transition (p, a, c, q) with some cost $c \in \mathbb{N}$.

Definition 1. *A distance automaton $\mathcal{D}$ satisfies the* shortcut property *if for all symbols a, b and all states p, q, r, $p \xrightarrow{a} q \xrightarrow{b} r$ implies $p \xrightarrow{a} r$ and $p \xrightarrow{b} r$.*

Lemma 1. *For every DFA $\mathcal{T}$, $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ satisfies the shortcut property.*

Proof. The proof follows almost immediately from the definition of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$. Let us consider two consecutive transitions (p, a, c, q) and (q, b, c', r) in $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$. We know from the definition of the transition relation of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$ that there exist some words $v \in \mathscr{L}(\mathcal{T}_{p,q})$ and $w \in \mathscr{L}(\mathcal{T}_{q,r})$. It follows that $v\ w \in \mathscr{L}(\mathcal{T}_{p,r})$. Again from the definition of $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$, we derive the existence of a transition (p, a, c'', r), for some $c'' \leq \mathsf{dist}(a, v\ w)$. Similarly, it follows that (p, b, c''', r) is a transition in $\mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}$, for some $c''' \leq \mathsf{dist}(b, v\ w)$. □

As with NFA, we call a *strongly connected component* (SCC) of a distance automaton $\mathcal{D}$ any maximal set of mutually reachable states. Given a SCC C of $\mathcal{D}$, we denote by $\mathcal{D}|C$ the sub-automaton obtained from $\mathcal{D}$ by restricting the set of states and transitions to C and by letting the initial and final conditions map any state of C to 0. Note that the transition graph of $\mathcal{D}|C$ is a clique when $\mathcal{D}$ satisfies the shortcut property.

A crucial property entailed by the shortcut property is the following one. Consider two runs γ and γ' of $\mathcal{D}|C$ that spell the same word u, but end in different states q and q'. If γ and γ' have optimal cost among all runs on $\mathcal{D}|C$ on u that end in q and q' respectively, then one can show that the difference in cost between γ and γ' is uniformly bounded by a constant. This implies that we can determinize $\mathcal{D}|C$ by using a subset construction, maintaining the difference between the optimal cost of reaching each state q and the overall optimal cost (the same idea underlies Mohri's determinization procedure [3]). Since this difference is always uniformly bounded by a constant, we obtain a finite state deterministic distance automaton:

Proposition 2. *For every distance automaton $\mathcal{D}$ that satisfies the shortcut property and every SCC C of $\mathcal{D}$, there is a deterministic distance automaton $\mathsf{det}(\mathcal{D}|C)$ that is equivalent to $\mathcal{D}|C$, namely, such that, for all words u,*

$$\mathsf{det}(\mathcal{D}|C)(u) \;=\; \mathcal{D}|C(u).$$

In addition, one can construct $\mathsf{det}(\mathcal{D}|C)$ so as to satisfy the following property: if $u_1 = u^{k_1}$ and $u_2 = u^{k_2}$ are two repetitions of the same word and ρ_1 and ρ_2 are runs of $\mathsf{det}(\mathcal{D}|C)$ on u_1 and u_2, respectively, that form cycles, then

$$\frac{\mathsf{cost}(\rho_1)}{k_1} \;=\; \frac{\mathsf{cost}(\rho_2)}{k_2}.$$

Proof. Let $\mathcal{D} = (\Sigma, Q, E, I, F)$ be a distance automaton satisfying the shortcut property and let C be a SCC of it. As a preliminary remark, we observe that, by definition, $\mathcal{D}|C = (\Sigma, C, E', I', F')$, where E' is obtained from E by restricting the set of states to C and $I'(q) = F'(q) = 0$ for all $q \in C$. Below, we consider runs of the distance automaton $\mathcal{D}|C$ on a given word u that end in a given state q and have the minimum cost among all runs of the same type. We call these runs (u, q)-*optimal* (note that these are similar to the locally optimal runs used in the proof of Proposition 1, with the only exception that the starting state is not fixed). We also say that a run is u-*optimal* if it is (u, q)-optimal for some state $q \in C$.

The basic idea underlying the determinization of the sub-automaton $\mathcal{D}|C$ stems from the following property:

Claim 2. *Given a word $u \in \Sigma^*$, the costs of any two u-optimal runs of $\mathcal{D}|C$ differ for at most c_{max}, where c_{max} is the maximum cost that appears in the transitions of $\mathcal{D}|C$.*

Proof. Let us fix a word $u = a_1 \dots a_n$ and let us consider two u-optimal runs $\gamma = (q_1, a_1, c_1, q_2) \dots (q_n, a_n, c_n, q_{n+1})$ and $\gamma' = (q'_1, a_1, c'_1, q'_2) \dots (q'_n, a_n, c'_n, q'_{n+1})$ of $\mathcal{D}|C$ on it. Observe that the states q_{n+1} and q'_{n+1} belong to the same SCC C of $\mathcal{D}$ and, in particular, q_{n+1} is reachable from q'_{n+1}, namely,

$$q'_{n+1} \;\xrightarrow{v}\; q_{n+1}$$

where $\xrightarrow{v}$ denotes the natural extension of the transition relation $\xrightarrow{a}$ from symbols to words (i.e., $p \xrightarrow{v} q$ iff $v = \varepsilon$ and $p = q$, or $v = v' \cdot a$, $p \xrightarrow{v'} r$, $r \xrightarrow{a} q$, for some $v' \in \Sigma^*$ and some $r \in C$). Using a basic induction on

$|v|$ and the shortcut property, one can prove that $\mathcal{D}$ contains a transition of the form $(q'_n, a_n, c'', q_{n+1})$, for some $c'' \in \{0, \ldots, c_{\mathsf{max}}\}$. This implies that the following is also a valid run of $\mathcal{D}|C$ on u:

$$\gamma'' \;=\; (q'_1, a_1, c'_1, q'_2)\ (q'_2, a_2, c'_2, q'_3)\ \ldots\ (q'_{n-1}, a_{n-1}, c'_1, q'_n)\ (q'_n, a_n, c'', q_{n+1})$$

Using the u-optimality of γ, we derive

$$\mathsf{cost}(\gamma) \;\leq\; \mathsf{cost}(\gamma'') \;\leq\; \mathsf{cost}(\gamma') \;+\; c_{\mathsf{max}}.$$

By symmetric arguments, one derives the inequality $\mathsf{cost}(\gamma') \leq \mathsf{cost}(\gamma) + c_{\mathsf{max}}$. □

We now construct a deterministic distance automaton $\mathsf{det}(\mathcal{D}|C)$ that turns out to be equivalent to $\mathcal{D}|C$. Intuitively, $\mathsf{det}(\mathcal{D}|C)$ parses an input word u and it outputs the minimal cost of a u-optimal run, keeping track, at the same time, of the differences between this cost and the costs of the (u, q)-optimal runs, for any $q \in C$ (these differences are called *residual costs* and, in view of the previous claim, are uniformly bounded). We formally define the deterministic distance automaton $\mathsf{det}(\mathcal{D}|C)$ equivalent to $\mathcal{D}|C$ as the tuple $(\Sigma, Q', \delta, \bar{r}_0, F')$, where

- Q' is the set of vectors with entries indexed by states in C and values ranging over the finite set $\{0, \ldots, c_{\mathsf{max}}\}$, where c_{max} is the maximum cost that appears in the transitions of $\mathcal{D}|C$ (intuitively, these vectors represent the residual costs of (u, q)-locally optimal runs, for each state $q \in C$ and for some fixed word u);
- δ is the partial function from $Q' \times \Sigma$ to $\mathbb{N} \times Q'$ defined by $\delta(\bar{r}, a) = (c, \bar{r}')$, where $c = \min\{\bar{r}[p] + c' \;:\; p \in C,\ (p, a, c', q) \in E\}$ and $\bar{r}'[q] = \min\{\bar{r}[p] + c' - c \;:\; p \in C,\ (p, a, c', q) \in E\}$;
- $\bar{r}_0$ is the initial vector defined by $\bar{r}_0[q] = 0$ for all $q \in C$;
- F' is the constant function that maps any vector $\bar{r} \in Q'$ to 0 (note that there always exist $q \in C$ for which the corresponding residual $\bar{r}[q]$ in $\bar{r}$ is 0).

We show that $\mathsf{det}(\mathcal{D}|C)$ is equivalent to $\mathcal{D}|C$, namely, that $\mathsf{det}(\mathcal{D}|C)(u) = \mathcal{D}|C(u)$ for all $u \in \Sigma^*$. Let us consider a word $u = a_1 \ldots a_n$. By exploiting a simple induction on the length of u, one can prove that

1. there exists a run of $\mathsf{det}(\mathcal{D}|C)$ on u if, and only if, there exists a run of $\mathcal{D}|C$ on u, -

2. if $\rho = (\bar{r}_1, a_1, c_1, \bar{r}_2)\dots(\bar{r}_n, a_n, c_n, \bar{r}_{n+1})$ is the unique run of $\mathsf{det}(\mathcal{D}|C)$ on $u = a_1 \dots a_n$ starting from state $\bar{r}_1$ (recall that $\mathsf{det}(\mathcal{D}|C)$ is deterministic), then the cost of ρ is equal to the cost of a u-optimal run γ augmented with the residual $\bar{r}_1[p]$, where p is the initial state of γ. That is:

$$\mathsf{cost}(\rho) \;=\; \mathsf{cost}(\gamma) \;+\; \bar{r}_1[p]. \tag{1}$$

We omit the formal proof of the above properties and we observe that they immediately imply that $\mathsf{det}(\mathcal{D}|C)(u) = \mathcal{D}|C(u)$ for all words $u \in \Sigma^*$.

Towards a conclusion, we can use equation (1) to prove the additional property concerning the cycles in $\mathsf{det}(\mathcal{D}|C)$. Consider two repetitions of the same word, that is, $u_1 = u^{k_1}$ and $u_2 = u^{k_2}$ and suppose that ρ_1 and ρ_2 are runs of $\mathsf{det}(\mathcal{D}|C)$ on u_1 and u_2 that form cycles (note that the two runs do not need to start from the same state). We denote by c_{max} be the maximal cost in $\mathcal{D}|C$ and, for every $n \in \mathbb{N}$, we let $\gamma^{(n)}$ be a u^n-optimal run of $\mathcal{D}|C$. We now consider the costs of suitable repetitions of the cycles ρ_1 and ρ_2:

$$\begin{aligned}
k_2 \cdot n \cdot \mathsf{cost}(\rho_1) \;=\; \mathsf{cost}(\rho_1^{k_2 \cdot n}) \;&\leq\; \mathsf{cost}\big(\gamma^{(k_1 \cdot k_2 \cdot n)}\big) \;+\; c_{\mathsf{max}} && \text{(by (1))} \\
&\leq\; \mathsf{cost}(\rho_2^{k_1 \cdot n}) \;+\; 2 \cdot c_{\mathsf{max}} && \text{(by (1))} \\
&\leq\; k_1 \cdot n \cdot \mathsf{cost}(\rho_2) \;+\; 2 \cdot c_{\mathsf{max}}.
\end{aligned}$$

As the above inequality holds for all natural numbers n, we conclude that $\frac{\mathsf{cost}(\rho_1)}{k_1} \leq \frac{\mathsf{cost}(\rho_2)}{k_2}$. Finally, a symmetric argument shows that $\frac{\mathsf{cost}(\rho_1)}{k_1} \geq \frac{\mathsf{cost}(\rho_2)}{k_2}$. □

Hereafter, given a distance automaton $\mathcal{D}$ satisfying the shortcut property and a SCC C in it, we denote by $\mathsf{det}(\mathcal{D}|C)$ the deterministic distance automaton that satisfies Proposition 2. A close inspection to the proof of the above proposition shows that $\mathsf{det}(\mathcal{D}|C)$ can be constructed in exponential time from $\mathcal{D}$ and C.

Example 4. *Consider the distance automaton $\mathcal{D}$ of Figure 1, which computes the edit distance of any word to the target language $T = (ab + b)^*\, a^*$. As $\mathcal{D}$ satisfies the shortcut property and consists of two SCCs C_1 and C_2, the two sub-automata $\mathcal{D}|C_1$ and $\mathcal{D}|C_2$ can be turned into equivalent deterministic distance automata $\mathsf{det}(\mathcal{D}|C_1)$ and $\mathsf{det}(\mathcal{D}|C_2)$, depicted to the right of Figure 1.*

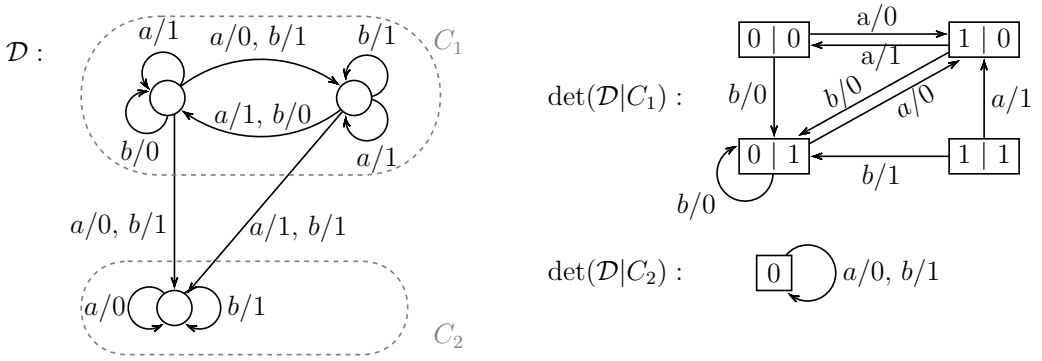

Figure 1: A distance automaton with two SCCs and its determinized sub-automata.

We remark that the above result does not imply that the entire distance automaton $\mathcal{D}$ is determinizable. Consider, for instance, a distance automaton $\mathcal{D}$ that computes the edit distance of a word u to the target language $T = a^* + b^*$. This distance is given by the minimum between the number of occurrences of a and the number of occurrences of b and hence any deterministic device that computes $\mathsf{dist}(u, \mathscr{L}(\mathcal{T}))$ must use unbounded memory.

4.3. Asymptotic cost in the unrestricted case

We now look at a special case of the asymptotic cost problem, where the source restriction is trivial. Thanks to Proposition 1 and Lemma 1, we can reduce the problem of computing the asymptotic repair cost $\mathsf{acost}(\Sigma^*, \mathscr{L}(\mathcal{T}))$ in the unrestricted case to the problem of computing the *asymptotic cost* of a distance automaton $\mathcal{D}$ satisfying the shortcut property:

$$\mathsf{acost}(\mathcal{D}) \ =^{\text{def}} \ \lim_{n\to\infty} \sup_{\substack{u\in\Sigma^* \\ |u|\geq n}} \frac{\mathcal{D}(u)}{|u|}.$$

This section is devoted to provide an effective characterization of the asymptotic cost $\mathsf{acost}(\mathcal{D})$ that will imply that the value is rational and computable from $\mathcal{D}$.

Before turning to the characterization, we prove that, in general, it is not possible to compute the asymptotic cost $\mathsf{acost}(\mathcal{D})$ for an arbitrary distance automaton $\mathcal{D}$:

Proposition 3. *The problem of deciding, given an arbitrary distance automaton $\mathcal{D}$, whether or not $\mathsf{acost}(\mathcal{D}) \leq \frac{1}{2}$ is undecidable.*

Proof. We use the undecidability of the $\frac{1}{2}$-*threshold problem* for normalized costs induced by distance automata [5], which consists of deciding, given a distance automaton $\mathcal{D}$, whether $\frac{\mathcal{D}(u)}{|u|} \leq \frac{1}{2}$ holds for all words $u \in \Sigma^*$.

Let us consider a distance automaton $\mathcal{D} = (\Sigma, Q, E, I, F)$. We compute a variant of the Kleene closure of $\mathcal{D}$, denoted $\mathcal{D}^*_\#$, by introducing a fresh symbol $\# \notin \Sigma$ and by adding 0-cost #-labeled transitions from all states $p \in \mathsf{dom}(F)$ to all states $q \in \mathsf{dom}(I)$. The new automaton $\mathcal{D}^*_\#$ satisfies the following property:

$$\forall m \in \mathbb{N},\ u_1, \ldots, u_m \in \Sigma^*. \qquad \mathcal{D}^*_\#(u_1 \# \ldots \# u_m) \ = \ \sum\nolimits_{1 \leq i \leq m} \mathcal{D}(u_i).$$

Clearly, this implies that $\mathsf{acost}(\mathcal{D}^*_\#) \ \geq \ \sup\limits_{u \in \Sigma^*} \frac{\mathcal{D}(u)}{|u|}$. As for the converse inequality, we consider a family of words $u^{(n)} = u_1^{(n)} \# \ldots \# u_{m_n}^{(n)}$ of length n such that $\lim\limits_{n \to \infty} \dfrac{\mathcal{D}^*_\#(u^{(n)})}{n} \ = \ \mathsf{acost}(\mathcal{D}^*_\#)$ and we observe that

$$\frac{\mathcal{D}^*_\#(u^{(n)})}{n} \ = \ \frac{\sum_{1 \leq i \leq m_n} \mathcal{D}(u_i^{(n)})}{\sum_{1 \leq i \leq m_n} |u_i^{(n)}| + m_n - 1} \ \leq \ \sup_{u \in \Sigma^*} \frac{\mathcal{D}(u)}{|u|}.$$

In particular, the above inequalities imply that $\mathsf{acost}(\mathcal{D}^*_\#) \ = \ \sup\limits_{u \in \Sigma^*} \dfrac{\mathcal{D}(u)}{|u|}$ and hence they reduce the $\frac{1}{2}$-threshold problem for $\mathcal{D}$ to the problem of deciding whether $\mathsf{acost}(\mathcal{D}^*_\#) \leq \frac{1}{2}$. From previous remarks about the undecidability of the $\frac{1}{2}$-threshold problem, it follows that it is not possible to compute the asymptotic cost for generic distance automata. □

The above proof gives a reduction from the $\frac{1}{2}$-threshold problem for the *normalized* cost of a distance automaton $\mathcal{D}$ to the $\frac{1}{2}$-threshold problem for the *asymptotic normalized* cost of a distance automaton $\mathcal{D}^*_\#$. It is worth remarking that this reduction does not preserve the shortcut property. This means that, even though it is possible to compute the asymptotic normalized cost for the sub-class of distance automata satisfying the shortcut property, the decidability of the analogous $\frac{1}{2}$-threshold problem for the normalized cost cannot be immediately derived from that. The problem of computing the normalized cost for a distance automaton satisfying the shortcut property remains, to our knowledge, open.

An approximate variant of the threshold problem for distance automata was solved in [6] by an algorithm that, given any rational number $\epsilon > 0$, tells

apart the case $\frac{\mathcal{D}(u)}{|u|} \leq (1-\epsilon)\cdot\frac{1}{2}$ from the case $\frac{\mathcal{D}(u)}{|u|} \geq \frac{1}{2}$ (in the reaming cases the algorithm may return any output).

Next we explain how the shortcut property helps in computing the asymptotic cost. One can show that the problem of computing $\mathsf{acost}(\mathcal{D})$ for a distance automaton $\mathcal{D}$ that is *deterministic* is reducible to the problem of computing normalized costs of simple cycles. Formally, a *simple cycle* is a run that is a cycle (i.e., that starts and ends in the same state) but that does not contain proper sub-cycles. It is then easy to show that for a deterministic distance automaton $\mathcal{D}$, $\mathsf{acost}(\mathcal{D})$ coincides with the maximum of $\frac{\mathsf{cost}(L)}{|L|}$ among all simple cycles L of $\mathcal{D}$, where $\mathsf{cost}(L)$ denotes the cost of the simple cycle L and $|L|$ its length (i.e., number of transitions in it). Thus by Proposition 2, calculation with simple cycles suffices to compute the asymptotic cost of any distance automaton satisfying the shortcut property and having a *single* SCC.

We consider now the more general case of a distance automaton $\mathcal{D}$ satisfying the shortcut property and having many SCCs, say $C_1, \ldots, C_k$. The situation in this case is slightly more complicated, as $\mathsf{acost}(\mathcal{D})$ cannot be expressed as a function of $\mathsf{acost}(\mathcal{D}|C_1), \ldots, \mathsf{acost}(\mathcal{D}|C_k)$. For this we define $\bar{\mathcal{D}}$ as the deterministic *multi-distance* automaton obtained from the synchronous product of $\mathsf{det}(\mathcal{D}|C_1)$, ..., $\mathsf{det}(\mathcal{D}|C_k)$ and we denote by $L_1, \ldots, L_m$ the simple cycles of $\bar{\mathcal{D}}$. Moreover, given $1 \leq i \leq m$ and $1 \leq j \leq k$, we denote by $\mathsf{cost}_j(L_i)$ the cost of the projection of the simple cycle L_i into the j-th component of $\bar{\mathcal{D}}$. Assuming that $\mathcal{D}$ is *trimmed*, namely, all its states are reachable from some states in $\mathsf{dom}(I)$ and they can reach some states in $\mathsf{dom}(F)$, we can characterize the asymptotic cost of $\mathcal{D}$ as follows:

Theorem 1. *For every (trimmed) distance automaton $\mathcal{D}$ satisfying the shortcut property,*

$$\mathsf{acost}(\mathcal{D}) \;=\; \max_{\alpha_1,\ldots,\alpha_m \geq 0} \; \min_{1\leq j\leq k} \; \frac{\sum_{1\leq i\leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1\leq i\leq m} \alpha_i \cdot |L_i|} \tag{2}$$

where $C_1, \ldots, C_k$ are the SCCs of the distance automaton $\mathcal{D}$, $L_1, \ldots, L_m$ are the simple cycles of the multi-distance automaton $\bar{\mathcal{D}} = \mathsf{det}(\mathcal{D}|C_1) \times \ldots \times \mathsf{det}(\mathcal{D}|C_k)$, and $\mathsf{cost}_j(L_i)$ is the cost of the projection of the simple cycle L_i into the j-th component of $\bar{\mathcal{D}}$.

The idea underlying the above characterization is that the asymptotic cost $\mathsf{acost}(\mathcal{D})$ is achieved by repetitions of simple cycles in the multi-distance

automaton $\bar{\mathcal{D}}$. The parameters $\alpha_1, \ldots, \alpha_m$ represent a correlation between the numbers of repetitions of the various simple cycles and the index j represents the SCC of $\mathcal{D}$ that optimizes the normalized cost of these repetitions. Before turning to the proof of this characterization, we illustrate by means of an example the rationale behind the use of cycles in $\bar{\mathcal{D}}$.

Example 5. *Consider again the distance automaton $\mathcal{D}$ of Figure 1, with the two SCCs C_1 and C_2. The determinized sub-automaton $\mathsf{det}(\mathcal{D}|C_1)$ has four different simple cycles: one spelling aa with cost 1, one spelling ab with cost 0, one spelling b with cost 0, and one spelling aab with cost 1. Similarly, the determinized sub-automaton $\mathsf{det}(\mathcal{D}|C_2)$ has two simple cycles: one spelling a with cost 0, and the other spelling b with cost 1. Hence $(aa)^n$ is a family of words achieving a worst-case asymptotic cost of $\lim \frac{n}{2n} = \frac{1}{2}$ for the sub-automaton $\mathcal{D}|C_1$, and b^n is a family of words achieving a worst-case asymptotic cost of $\lim \frac{n}{n} = 1$ for the sub-automaton $\mathcal{D}|C_2$. However, a^{2n} is not a worst-case for $\mathcal{D}|C_2$ (as it can be repaired with asymptotic cost 0) and, symmetrically, b^n is not a worst-case for $\mathcal{D}|C_1$. This means that the asymptotic cost for $\mathcal{D}$ must be achieved by a suitable combination of both families of words. To find the correct combination that witnesses the asymptotic cost for $\mathcal{D}$ it is convenient to consider cycles in the multi-distance automaton $\bar{\mathcal{D}} = \mathsf{det}(\mathcal{D}|C_1) \times \mathsf{detdet}(\mathcal{D}|C_2)$ and linear combinations of their costs in the components C_1 and C_2. For the considered example, we notice that the simple cycle in $\bar{\mathcal{D}}$ that spells repetitions of the form $(aab)^n$ achieves maximal normalized cost in both components, that is, $\frac{1}{3}$, which indeed coincides with the worst-case asymptotic cost $\mathsf{acost}(\mathcal{D})$.*

The proof of Theorem 1 consists of establishing two inequalities, which are given by Lemma 3 and Lemma 4 below.

For the first inequality, we argue that all words can be approximated in cost by repetitions of simple cycles, and that the cost of parsing these words is at most the cost of a "homogeneous run", i.e., a run lying entirely inside a single component of $\mathcal{D}$. The first part of the proof relies on the following property, which is also present in [7]. We state it in a graph-theoretic setting, in such a way that it can be later reused in several proofs. In particular, we consider a directed graph $\mathcal{G}$, which can be understood as the multi-distance automaton $\bar{\mathcal{D}}$, and a path ρ in it, that is, a sequence of edges of the form $e_1\ e_2\ \ldots\ e_n$, where the target vertex of each edge e_i coincides with source vertex of the next edge e_{i+1}. Intuitively, this property state in the following

lemma eases the calculation of the cost within a SCC C_j of a run ρ of $\bar{\mathcal{D}}$ that does not show any particular 'cyclic' structure.

Lemma 2 (Simple cycle decomposition [7])**.** *Let $\mathcal{G}$ be a finite graph and let $L_1, \ldots, L_m$ be all the simple cycles in it. Given a path ρ in $\mathcal{G}$, one can find a partition the domain of ρ into (possibly non-convex) subsets $X_0, X_1, \ldots, X_m$ such that*

1. *$|X_0| \leq K$, where K is the number of vertices of $\mathcal{G}$,*
2. *for all $1 \leq i \leq m$, the sub-sequence $\rho|X_i$ is a repetition of L_i.*

Proof. We find the sets $X_0, X_1, \ldots, X_m$ by exploiting an induction. At the beginning we define $X_{0,0}$ to be the entire domain of the path ρ and $X_{0,i} = \varnothing$ for all $1 \leq i \leq m$. At each induction step on $n \in \mathbb{N}$, we subtract a suitable convex subset Y_n from $X_{n,0}$ and we add it to one of the subsets $X_{n,i}$, with $1 \leq i \leq m$. More precisely, if $|X_{n,0}| \leq K$, where K is the number of vertices of $\mathcal{G}$, then we terminate the induction with the current sets $X_{n,0}, X_{n,1}, \ldots, X_{n,m}$. Otherwise, we continue the induction by specifying the sets $X_{n+1,0}$, $X_{n+1,1}$, …, $X_{n+1,m}$ in terms of $X_{n,0}$, $X_{n,1}$, …, $X_{n,m}$ as follows. We first claim that there is an interval Y_n contained in $X_{n,0}$ for which the sub-sequence $\rho|Y_n$ is an occurrence of a simple cycle L_i, for some $1 \leq i \leq m$. Indeed, since the length of the sub-sequence $\rho|X_{n,0}$ exceeds the number K of states of $\mathcal{G}$, we know that $\rho|X_{n,0}$ contains two repeated occurrences of the same state, and hence a cycle L. In its turn, the cycle L must contain an occurrence of a simple cycle among $L_1, \ldots, L_m$ (this follows from the fact that the containment relation between cycles is a well-founded partial order). We choose such an occurrence of the simple cycle L_i in $\rho|X_{n,0}$ and we denote by Y_n the set of the positions in $X_{n,0}$ that carry the chosen occurrence. Accordingly, we define

- $X_{n+1,0} = X_{n,0} \setminus Y_n$,
- $X_{n+1,i} = X_{n,i} \cup Y_n$,
- $X_{n+1,i'} = X_{n,i'}$ for all indices $1 \leq i' \leq m$ different from i.

If n is the last step of the induction, then we define $X_i = X_{n,i}$ for all $0 \leq i \leq m$. Note that we have $|X_0| = |X_{n,0}| \leq K$. Moreover, it is easy to verify (e.g., by induction on n) that each sub-sequence $\rho|X_{n,i}$ (and hence, in particular, the sub-sequence $\rho|X_i$) is a repetition of the corresponding simple cycle L_i. This concludes the proof of the claim. □

Hereafter, for the sake of brevity, we tacitly assume that $C_1, \ldots, C_k$ are the SCCs of the distance automaton $\mathcal{D}$ and $L_1, \ldots, L_m$ are the simple cycles of the multi-distance automaton $\bar{\mathcal{D}} = \mathsf{det}(\mathcal{D}|C_1) \times \ldots \times \mathsf{det}(\mathcal{D}|C_k)$. Moreover, we say that a run $\gamma = (q_0, a_1, c_1, q_1) \ \ldots \ (q_{n-1}, a_n, c_n, q_n)$ of $\mathcal{D} = (\Sigma, Q, E, I, F)$ is *successful* if it starts in a state $q_0 \in \mathsf{dom}(I)$ and it ends in a state $q_n \in \mathsf{dom}(F)$.

Lemma 3. *For every distance automaton $\mathcal{D}$ satisfying the shortcut property,*

$$\mathsf{acost}(\mathcal{D}) \ \leq \ \max_{\alpha_1,\ldots,\alpha_m \geq 0} \ \min_{1 \leq j \leq k} \ \frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}.$$

Proof. Let $\left(u^{(n)}\right)_{n \in \mathbb{N}}$ be a family of words over the alphabet Σ such that

$$\mathsf{acost}(\mathcal{D}) \ = \ \limsup_{n \to \infty} \ \frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|}.$$

Without loss of generality, we can assume that the limit of the sequence $\frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|}$, for arbitrarily large numbers n, exists, and hence it coincides with $\mathsf{acost}(\mathcal{D})$. Indeed, if this were not the case, we could restrict ourselves to a proper sub-family $\left(u^{(n)}\right)_{n \in N}$ of words, where N is a an infinite subset of the natural numbers, in such a way that the sequence $\frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|}$ converges for n ranging over N. Assuming that $\lim_{n \to \infty} \frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|}$ is defined will allow us to further restrict, if necessary, to sub-families of words without compromising the above equality. To prove the lemma, it is sufficient to find some parameters $\alpha_1, ..., \alpha_m \geq 0$ that satisfy the following inequality:

$$\limsup_{n \to \infty} \ \frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|} \ \leq \ \min_{1 \leq j \leq k} \ \frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}. \tag{3}$$

Let us fix $n \in \mathbb{N}$ and denote by $\rho^{(n)}$ the (unique) successful run of $\bar{\mathcal{D}}$ on the word $u^{(n)}$, and by $\rho_j^{(n)}$ the projection of it into the j-th component C_j, for any $1 \leq j \leq k$.

First, we compare the cost $\mathcal{D}(u^{(n)})$ with the cost $\mathcal{D}|C_j(u^{(n)})$ for each SCC C_j. Consider a successful run $\gamma^{(n)}$ of $\mathcal{D}$ on $u^{(n)}$ which starts in some state p and ends in some state q and that minimizes the value $\mathsf{cost}(\gamma^{(n)}) + I(p) + F(q)$. Clearly, we have $\mathcal{D}(u^{(n)}) \leq \mathsf{cost}(\gamma^{(n)}) + I_{\mathsf{max}} + F_{\mathsf{max}}$, where I_{max} is the maximum value taken by the initial condition of $\mathcal{D}$ and F_{max} is the maximum value taken by the final condition of $\mathcal{D}$. Consider now a run $\gamma_j^{(n)}$ of $\mathcal{D}|C_j$ on

the same word $u^{(n)}$, but entirely inside the SCC C_j, which starts in p' and ends in q' and that minimizes the relative cost. Since the initial and final conditions of $\mathcal{D}|C_j$ map every state to 0, we have $\mathcal{D}|C_j(u^{(n)}) = \mathsf{cost}(\gamma_j^{(n)})$. Moreover, since $\mathcal{D}$ is trimmed, we know that p' is reachable from p and q is reachable from q'. Thus, using the shortcut property, one can easily verify that $\mathsf{cost}(\gamma^{(n)}) \leq \mathsf{cost}(\gamma_j^{(n)}) + 2c_{\mathsf{max}}$, where c_{max} is the maximum cost that appears in the transitions of $\mathcal{D}$ (the additive constant $2c_{\mathsf{max}}$ accounts for the cost discount in considering a run of $\mathcal{D}|C_j$ rather than a run of $\mathcal{D}$). This shows that

$$\begin{aligned}
\mathcal{D}(u^{(n)}) &\leq \mathsf{cost}(\gamma^{(n)}) + I_{\mathsf{max}} + F_{\mathsf{max}} \\
&\leq \min_{1 \leq j \leq k} \mathsf{cost}(\gamma_j^{(n)}) + 2c_{\mathsf{max}} + I_{\mathsf{max}} + F_{\mathsf{max}} \\
&= \min_{1 \leq j \leq k} \mathcal{D}|C_j(u^{(n)}) + 2c_{\mathsf{max}} + I_{\mathsf{max}} + F_{\mathsf{max}}
\end{aligned}$$

From Proposition 2, we also know that $\mathcal{D}|C_j(u^{(n)}) = \mathsf{det}(\mathcal{D}|C_j)(u^{(n)})$. Moreover, since $\mathsf{det}(\mathcal{D}|C_j)$ is a deterministic distance automaton, the projection $\rho_j^{(n)}$ of $\rho^{(n)}$ can be viewed as the unique run of $\mathsf{det}(\mathcal{D}|C_j)$ on $u^{(n)}$. We thus obtain

$$\mathcal{D}|C_j(u^{(n)}) = \mathsf{det}(\mathcal{D}|C_j)(u^{(n)}) = \mathsf{cost}(\rho_j^{(n)}).$$

Below, we explicitly compute the cost of each run $\rho_j^{(n)}$ using the costs of the simple cycles L_i in the component C_j of $\mathcal{D}$. The problem is that the run $\rho^{(n)}$ may not contain factors consisting of entire repetitions of these simple cycles. We overcome this problem by viewing the multi-distance automaton $\bar{\mathcal{D}}$ as a finite graph and the run $\rho^{(n)}$ of $\bar{\mathcal{D}}$ as a path in it. The simple cycle decomposition lemma (Lemma 2) implies the existence of a partition of the domain of $\rho^{(n)}$ into (possibly non-convex) subsets $X_0^{(n)}, X_1^{(n)}, \ldots, X_m^{(n)}$ such that

1. $|X_0^{(n)}|$ is uniformly bounded by the number K of states of $\bar{\mathcal{D}}$,
2. for all $1 \leq i \leq m$, the sub-sequence $\rho^{(n)}|X_i^{(n)}$ is a repetition of the simple cycle L_i of $\bar{\mathcal{D}}$.

For every index $1 \leq i \leq m$, we denote by $\mathsf{occ}_i^{(n)}$ the number of repetitions of the simple cycle L_i in the sub-sequence $\rho^{(n)}|X_i^{(n)}$ (by a slight abuse of terminology we say that these are also 'repetitions' in the run $\rho^{(n)}$). We are

now ready to bound the cost of $\rho^{(n)}$ in the component C_j in terms of the costs of the 'repetitions' of each simple cycle L_i in $\rho^{(n)}$.

The first, straightforward, inequality is as follows (recall that the sets $X_0^{(n)}$, $X_1^{(n)}$, ..., $X_m^{(n)}$ form a partition of the domain of $\rho^{(n)}$ and $|X_0^{(n)}| \leq K$):

$$\begin{aligned} \mathsf{cost}(\rho_j^{(n)}) &= \textstyle\sum_{1\leq i\leq m} \mathsf{cost}(\rho_j^{(n)}|X_i^{(n)}) + \mathsf{cost}(\rho_j^{(n)}|X_0^{(n)}) \\ &\leq \textstyle\sum_{1\leq i\leq m} \mathsf{cost}(\rho_j^{(n)}|X_i^{(n)}) + K \cdot c'_{\mathsf{max}} \end{aligned}$$

where c'_{max} is the maximum cost that appears in the transitions of $\bar{\mathcal{D}}$. Moreover, it easily follows from the fact that the sub-sequence $\rho^{(n)}|X_i^{(n)}$ is an $\mathsf{occ}_i^{(n)}$-fold repetition of the simple cycle L_i, that

$$\mathsf{cost}(\rho_j^{(n)}|X_i^{(n)}) = \mathsf{occ}_i^{(n)} \cdot \mathsf{cost}_j(L_i).$$

Now, in order to find the parameters $\alpha_1, ..., \alpha_m \geq 0$ that satisfy Equation (3), we consider the asymptotic behaviour of the sequence $\frac{\mathsf{occ}_i^{(n)}}{n}$. Without loss of generality, we can assume that the limit of $\frac{\mathsf{occ}_i^{(n)}}{n}$ for arbitrarily large numbers $n \in \mathbb{N}$ exists. Indeed, we can always find an infinite set N of natural numbers such that the sequence $\frac{\mathsf{occ}_i^{(n)}}{n}$ converges for n ranging over N. Note that restricting to the corresponding sub-family of words $u^{(n)}$ and runs $\gamma^{(n)}$ and $\rho^{(n)}$, for $n \in N$, does not affect the previously established equalities (in particular, $\mathsf{acost}(\mathcal{D}) = \lim_{n\to\infty} \frac{\mathsf{cost}(\gamma^{(n)})}{|u^{(n)}|}$). For the sake of simplicity, we shall not explicitly mention the set N hereafter and we use, for instance, $\lim_{n\to\infty} f(n)$ to denote the limit of a certain function f for arbitrarily large numbers $n \in N$. Accordingly, for every index $1 \leq i \leq m$, we define

$$\alpha_i \stackrel{\text{def}}{=} \lim_{n\to\infty} \frac{\mathsf{occ}_i^{(n)}}{n}.$$

Clearly, we have that

$$\mathsf{occ}_i^{(n)} \cdot \mathsf{cost}_j(L_i) = \big(n\cdot\alpha_i + \mathsf{occ}_i^{(n)} - n\cdot\alpha_i\big)\cdot \mathsf{cost}_j(L_i) = n\cdot\alpha_i\cdot\mathsf{cost}_j(L_i) + g_{i,j}(n)$$

where $g_{i,j}(n) \stackrel{\text{def}}{=} (\mathsf{occ}_i^{(n)} - n\cdot\alpha_i)\cdot\mathsf{cost}_j(L_i)$ is a function whose limit tends to 0 (hence $g_{i,j} \in \mathcal{O}(1)$, using the 'big-$\mathcal{O}$' notation).

Putting all together, we get an upper bound to the cost of the run $\gamma^{(n)}$:

$$\mathcal{D}(u^{(n)}) \leq \min_{1\leq j\leq k} \mathcal{D}|C_j(u^{(n)}) + 2c_{\mathsf{max}} + I_{\mathsf{max}} + F_{\mathsf{max}}$$

$$
\begin{aligned}
&= \min_{1\le j\le k} \mathsf{cost}(\rho_j^{(n)}) \;+\; 2c_{\max} + I_{\max} + F_{\max} \\
&\le \min_{1\le j\le k} \left(\sum_{1\le i\le m} \mathsf{cost}(\rho_j^{(n)} | X_i^{(n)}) + K \cdot c'_{\max} \right) \;+\; 2c_{\max} + I_{\max} + F_{\max} \\
&= \min_{1\le j\le k} \sum_{1\le i\le m} \left(\mathsf{occ}_i^{(n)} \cdot \mathsf{cost}_j(L_i) \right) \;+\; 2c_{\max} + I_{\max} + F_{\max} + K \cdot c'_{\max} \\
&= \min_{1\le j\le k} \sum_{1\le i\le m} \left(n \cdot \alpha_i \cdot \mathsf{cost}_j(L_i) \;+\; g_{i,j}(n) \right) \;+\; 2c_{\max} + I_{\max} + F_{\max} + K \cdot c'_{\max} \\
&= n \cdot \min_{1\le j\le k} \textstyle\sum_{1\le i\le m} \alpha_i \cdot \mathsf{cost}_j(L_i) \;+\; \mathcal{O}(1).
\end{aligned}
$$

Similarly, we obtain a lower bound to the length of the word $u^{(n)}$:

$$
\begin{aligned}
|u^{(n)}| &\ge \textstyle\sum_{1\le i\le m} u^{(n)} | X_i^{(n)} \\
&= \textstyle\sum_{1\le i\le m} \mathsf{occ}_i^{(n)} \cdot |L_i| \\
&= n \cdot \textstyle\sum_{1\le i\le m} \alpha_i \cdot |L_i| \;-\; \mathcal{O}(1).
\end{aligned}
$$

Towards a conclusion, we prove Equation (3) as follows:

$$
\begin{aligned}
\limsup_{n\to\infty} \frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|} &\le \limsup_{n\to\infty} \frac{n \cdot \min_{1\le j\le k} \sum_{1\le i\le m} \alpha_i \cdot \mathsf{cost}_j(L_i) \;+\; \mathcal{O}(1)}{n \cdot \sum_{1\le i\le m} \alpha_i \cdot |L_i| \;-\; \mathcal{O}(1)} \\
&= \min_{1\le j\le k} \frac{\sum_{1\le i\le m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1\le i\le m} \alpha_i \cdot |L_i|}.
\end{aligned}
$$

□

For the converse inequality, we present a large family of words for which the optimal runs are nearly homogeneous (in the sense that they lie almost entirely inside a single component of $\mathcal{D}$). The words will consist of nested repetitions of simple cycles in such a way that any optimal run stabilizes in the same component.

Lemma 4. *For every distance automaton $\mathcal{D}$ satisfying the shortcut property,*

$$
\mathsf{acost}(\mathcal{D}) \;\ge\; \max_{\alpha_1,\ldots,\alpha_m \ge 0} \; \min_{1\le j\le k} \; \frac{\sum_{1\le i\le m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1\le i\le m} \alpha_i \cdot |L_i|}.
$$

Proof. Let us fix arbitrarily some parameters $\alpha_1, \ldots, \alpha_m \geq 0$. We prove the above inequality by constructing a family of 'cyclic' words $u^{(n)}$ that depend on $\alpha_1, \ldots, \alpha_m$ and n and such that the normalized cost of any run of $\mathcal{D}$ on $u^{(n)}$ dominates, in the limit, the cost $\frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}$. For the moment, we assume that all the states of the cycles $L_1, \ldots, L_m$ are mutually reachable in $\bar{\mathcal{D}}$. Towards the end of the proof, we will show how to drop this assumption.

We fix (i) a run σ_0 of $\bar{\mathcal{D}}$ that starts from the initial state of $\bar{\mathcal{D}}$ and ends in the first/last state of L_1, (ii) a run σ_m of $\bar{\mathcal{D}}$ that starts from the first/last state of L_m and ends in the first/last state of L_1, and (iii) for all $1 \leq i < m$, a run σ_i of $\bar{\mathcal{D}}$ that starts from the first/last state of L_i and ends in the first/last state of L_{i+1}. Without loss of generality, we can assume that the lengths of the runs $\sigma_0, \sigma_1, \ldots, \sigma_m$ do not exceed the number K of states of $\bar{\mathcal{D}}$. Now, we think of each simple cycle L_i as a run of the multi-distance automaton $\bar{\mathcal{D}}$ that starts and ends in the same state and we construct, for every natural number n, a 'cyclic' run $\rho^{(n)}$ of $\bar{\mathcal{D}}$ as follows:

$$\rho^{(n)} \quad =^{\text{def}} \quad \sigma_0 \ (\rho^{(n)}_{\mathsf{cycles}})^n$$

$$\text{where} \qquad \rho^{(n)}_{\mathsf{cycles}} \quad =^{\text{def}} \quad L_1^{\lceil n \cdot \alpha_1 \rceil} \ \sigma_1 \ L_2^{\lceil n \cdot \alpha_2 \rceil} \ \ldots \ \sigma_{m-1} \ L_m^{\lceil n \cdot \alpha_m \rceil} \ \sigma_m.$$

Accordingly, we denote by $u^{(n)}$ the word spelled out by the run $\rho^{(n)}$.

To prove the claim of the lemma, it is sufficient to prove that the following inequality holds for all $n \in \mathbb{N}$ and for all choices of runs $\gamma^{(n)}$ of $\mathcal{D}$ on $u^{(n)}$:

$$\limsup_{n \to \infty} \ \frac{\mathsf{cost}(\gamma^{(n)})}{|u^{(n)}|} \ \geq \ \min_{1 \leq j \leq k} \ \frac{\sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|}. \tag{4}$$

Let us now fix further a run $\gamma^{(n)}$ of $\mathcal{D}$ on $u^{(n)}$ for each $n \in \mathbb{N}$. We introduce some additional notation. Given $n \in \mathbb{N}$, $1 \leq i \leq m$, and $1 \leq j \leq k$, we define:

- $X_j^{(n)}$ to be the set of positions of $\gamma^{(n)}$ that carry occurrences of transitions whose states belong to the same SCC C_j (note that $X_j^{(n)}$ is an interval);
- $Y_j^{(n)}$ to be the maximal subset of $X_j^{(n)}$ such that the sub-run $\rho^{(n)}|Y_j^{(n)}$ is a repetition of the block $\rho^{(n)}_{\mathsf{cycles}}$ (note that $Y_j^{(n)}$ is also an interval);
- $Z_{j,i}^{(n)}$ to be the maximal subset of $Y_j^{(n)}$ such that $\rho^{(n)}|Z_{j,i}^{(n)}$ is a repetition of the simple cycle L_i (note that the sets $Z_{j,1}^{(n)}$, ..., $Z_{j,m}^{(n)}$ form a partition of $Y_j^{(n)}$ and they contain possibly non-contiguous positions).

- $\mathsf{occ}_j^{(n)}$ to be the number of repetitions of $\rho_{\mathsf{cycles}}^{(n)}$ in the sub-run $\rho^{(n)}|Y_j^{(n)}$, namely, $\mathsf{occ}_j^{(n)} = \frac{|Y_j^{(n)}|}{|\rho_{\mathsf{cycles}}^{(n)}|}$ (note that this implies $\rho^{(n)}|Z_{j,i}^{(n)} = L_i^{\lceil n\cdot\alpha_i\rceil\cdot\mathsf{occ}_j^{(n)}}$).

The first inequality is straightforward (the sets $Z_{j,i}^{(n)}$ are pairwise disjoint):

$$\mathsf{cost}(\gamma^{(n)}) \;\geq\; \sum_{1\leq j\leq k}\sum_{1\leq i\leq m} \mathsf{cost}(\gamma^{(n)}|Z_{j,i}^{(n)}).$$

Given an index $1 \leq j \leq k$, we also denote by $\rho_j^{(n)}$ the projection of $\rho^{(n)}$ into the j-th component (we can think of it as a run of the deterministic distance automaton $\mathsf{det}(\mathcal{D}|C_j)$ on $u^{(n)}$). Below, we fix some indices $1 \leq i \leq m$ and $1 \leq j \leq k$ and we compare the cost of each sub-sequence $\gamma^{(n)}|Z_{j,i}^{(n)}$ with the cost of the corresponding sub-run $\rho_j^{(n)}|Z_{j,i}^{(n)}$ of $\mathcal{D}|C_j$.

We observe that $\gamma^{(n)}|Z_{j,i}^{(n)}$ is not necessarily a run of $\mathcal{D}|C_j$, since the set $Z_{j,i}^{(n)}$ is not an interval of the domain of $\gamma^{(n)}$ (we can still compute its cost though). We first turn $\gamma^{(n)}|Z_{j,i}^{(n)}$ into a run $\tilde{\gamma}_{j,i}^{(n)}$ of $\mathcal{D}|C_j$ on $u^{(n)}|Z_{j,i}^{(n)}$ of similar cost, as follows. Suppose that there exist two positions $x < y$ in $Z_{j,i}^{(n)}$ such that $z \notin Z_{j,i}^{(n)}$ for all $x < z < y$ and the corresponding transitions $\gamma^{(n)}[x] = (p_x, a_x, c_x, q_x)$ and $\gamma^{(n)}[y] = (p_y, a_y, c_y, q_y)$, which are consecutive in $\gamma^{(n)}|Z_{j,i}^{(n)}$, do not match (i.e., $q_x \neq p_y$). We call such a pair (x, y) of positions a *gap* of $Z_{j,i}^{(n)}$. The states q_x and p_y belong to the same SCC C_j of $\mathcal{D}$ and hence it follows from the shortcut property that $\mathcal{D}|C_j$ contains a transition of the form (p_x, a_x, c'_x, p_y), for some $c'_x \in \mathbb{N}$, connecting p_x to p_y. The described operation of connecting states through shortcuts can be applied to every gap (x, y) of $Z_{j,i}^{(n)}$, thus resulting in a correct run $\tilde{\gamma}_{j,i}^{(n)}$ of $\mathcal{D}|C_j$ on the sub-word $u^{(n)}|Z_{j,i}^{(n)}$. We observe two crucial properties about this construction. First, the number of required operations is at most $\mathsf{occ}_j^{(n)}$ (this follows from the fact that the gaps of $Z_{j,i}^{(n)}$ can only appear between the positions that correspond to two non-consecutive occurrences of $L_i^{\lceil n\cdot\alpha_i\rceil}$ in $\rho^{(n)}$ and from the fact that there exist at most $\mathsf{occ}_j^{(n)}$ such occurrences). Second, the difference in cost that results from one application of this operation never exceeds the maximum cost c_{max} of the transitions in $\mathcal{D}$. In view of these properties, we have

$$\mathsf{cost}(\gamma^{(n)}|Z_{j,i}^{(n)}) \;\geq\; \mathsf{cost}(\tilde{\gamma}_{j,i}^{(n)}) \;-\; c_{\mathsf{max}}\cdot\mathsf{occ}_j^{(n)}.$$

From the fact that $\tilde{\gamma}_{j,i}^{(n)}$ is a correct run of $\mathcal{D}|C_j$ on $u^{(n)}|Z_{j,i}^{(n)}$ and the initial and final conditions of the sub-automaton $\mathcal{D}|C_j$ map every state in C_j to the cost 0, we derive

$$\mathsf{cost}(\tilde{\gamma}_{j,i}^{(n)}) \;\geq\; \mathcal{D}|C_j(u^{(n)}|Z_{j,i}^{(n)}).$$

Proposition 2 then implies that

$$\mathcal{D}|C_j(u^{(n)}|Z_{j,i}^{(n)}) \;=\; \mathsf{det}(\mathcal{D}|C_j)(u^{(n)}|Z_{j,i}^{(n)}).$$

Now, consider the j-th projection $\rho_j^{(n)}|Z_{j,i}^{(n)}$ of the sub-run $\rho^{(n)}|Z_{j,i}^{(n)}$ of the deterministic multi-distance $\bar{\mathcal{D}}$. By construction, $\rho_j^{(n)}|Z_{j,i}^{(n)}$ is a run of $\mathsf{det}(\mathcal{D}|C_j)$ on the sub-word $u^{(n)}|Z_{j,i}^{(n)}$. Note that $\rho_j^{(n)}|Z_{j,i}^{(n)}$ can start from a state that is different from the initial state of $\mathsf{det}(\mathcal{D}|C_j)$ and hence it is not guaranteed to have optimal cost. However, the first state of $\rho_j^{(n)}|Z_{j,i}^{(n)}$ is reachable from the initial state of $\mathsf{det}(\mathcal{D}|C_j)$ by a path $\tau_{j,i}^{(n)}$ of length at most K, where K is the number of states of $\bar{\mathcal{D}}$. For the sake of brevity, we denote by $x_{j,i}^{(n)}$ be the word spelled out by $\tau_{j,i}^{(n)}$ and by $z_{j,i}^{(n)}$ the sub-word $u^{(n)}|Z_{j,i}^{(n)}$ (which is spelled out by $\rho^{(n)}|Z_{j,i}^{(n)}$). Clearly, we have $\mathsf{cost}(\rho_j^{(n)}|Z_{j,i}^{(n)}) \leq \mathsf{cost}(\tau_{j,i}^{(n)}) + \mathsf{cost}(\rho_j^{(n)}|Z_{j,i}^{(n)}) = \mathsf{det}(\mathcal{D}|C_j)(x_{j,i}^{(n)}\ z_{j,i}^{(n)}) = \mathcal{D}|C_j(x_{j,i}^{(n)}\ z_{j,i}^{(n)})$. Let us now consider two optimal runs $\alpha_{j,i}^{(n)}$ and $\beta_{j,i}^{(n)}$ of $\mathcal{D}|C_j$ on $x_{j,i}^{(n)}$ and $z_{j,i}^{(n)}$, respectively. Clearly, we have $\mathcal{D}|C_j(x_{j,i}^{(n)}) = \mathsf{cost}(\alpha_{j,i}^{(n)})$ and $\mathcal{D}|C_j(z_{j,i}^{(n)}) = \mathsf{cost}(\beta_{j,i}^{(n)})$. Moreover, since $\mathcal{D}|C_j$ satisfies the shortcut property, we have that the juxtaposition of the two runs $\alpha_{j,i}^{(n)}$ and $\beta_{j,i}^{(n)}$ of $\mathcal{D}|C_j$ can be turned into a valid run $\lambda_{j,i}^{(n)}$ of $\mathcal{D}|C_j$ on $x_{j,i}^{(n)}\ z_{j,i}^{(n)}$, having cost at most $\mathsf{cost}(\alpha_{j,i}^{(n)}) + \mathsf{cost}(\beta_{j,i}^{(n)}) + c_{\mathsf{max}}$, where c_{max} is the maximum cost of the transitions in $\mathcal{D}$. This shows that $\mathcal{D}|C_j(x_{j,i}^{(n)}\ z_{j,i}^{(n)}) \leq \mathsf{cost}(\lambda_{j,i}^{(n)}) \leq \mathsf{cost}(\alpha_{j,i}^{(n)}) + \mathsf{cost}(\beta_{j,i}^{(n)}) + c_{\mathsf{max}} = \mathcal{D}|C_j(x_{j,i}^{(n)}) + \mathcal{D}|C_j(z_{j,i}^{(n)}) + c_{\mathsf{max}} \leq K \cdot c_{\mathsf{max}} + \mathsf{det}(\mathcal{D}|C_j)(z_{j,i}^{(n)}) + c_{\mathsf{max}}$. Overall, this shows that

$$\mathsf{det}(\mathcal{D}|C_j)(u^{(n)}|Z_{j,i}^{(n)}) \;\geq\; \mathsf{cost}(\rho_j^{(n)}|Z_{j,i}^{(n)}) \;-\; (K+1)\cdot c_{\mathsf{max}}.$$

Now, we explicitly compute the cost of $\rho_j^{(n)}|Z_{j,i}^{(n)}$ as follows. Since $\rho_j^{(n)}|Z_{j,i}^{(n)}$ is an $(\lceil n\cdot\alpha_i\rceil \cdot \mathsf{occ}_j^{(n)})$-fold repetition of the simple cycle L_i, we have

$$\mathsf{cost}(\rho_j^{(n)}|Z_{j,i}^{(n)}) \;=\; \lceil n\cdot\alpha_i\rceil \cdot \mathsf{occ}_j^{(n)} \cdot \mathsf{cost}_j(L_i).$$

Another crucial step amounts to showing that the sum of the numbers $\mathsf{occ}_j^{(n)}$ over all indices $1 \leq j \leq k$ is almost equal (i.e., equal up to a constant) to the total number n of repetitions of the block $\rho_{\mathsf{cycles}}^{(n)}$ in $\rho^{(n)}$. The first inequality follows trivially by construction:

$$\sum_{1 \leq j \leq k} \mathsf{occ}_j^{(n)} \;\leq\; n.$$

As for the converse equality, we recall that each set $Y_j^{(n)}$ is defined as the maximal set of positions of $\gamma^{(n)}$ that contain states from the SCC C_j and such that the sub-run $\rho^{(n)}|Y_j^{(n)}$ is a repetition of the block $\rho_{\mathsf{cycles}}^{(n)}$. This implies that there exist at most k occurrences of the block $\rho_{\mathsf{cycles}}^{(n)}$ in $\rho^{(n)}$ that are not entirely covered by some set $Y_j^{(n)}$, for any $1 \leq j \leq k$. From this and from the definition of $\mathsf{occ}_j^{(n)}$, we derive the following inequalities:

$$\sum_{1 \leq j \leq k} \mathsf{occ}_j^{(n)} = \sum_{1 \leq j \leq k} \frac{|Y_j^{(n)}|}{|\rho_{\mathsf{cycles}}^{(n)}|} \geq \frac{|\rho^{(n)}| - k \cdot |\rho_{\mathsf{cycles}}^{(n)}|}{|\rho_{\mathsf{cycles}}^{(n)}|} \geq \frac{n \cdot |\rho_{\mathsf{cycles}}^{(n)}| - k \cdot |\rho_{\mathsf{cycles}}^{(n)}|}{|\rho_{\mathsf{cycles}}^{(n)}|} \geq n - k.$$

Putting together all the inequalities (and using some basic rewriting), we obtain a lower bound to the cost of the run $\gamma^{(n)}$:

$$\begin{aligned}
\mathsf{cost}(\gamma^{(n)}) \;&\geq\; \sum_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \mathsf{cost}(\gamma^{(n)}|Z_{j,i}^{(n)}) \\
&\geq\; \sum_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \mathsf{cost}(\tilde{\gamma}_{j,i}^{(n)}) \;-\; m \cdot c_{\mathsf{max}} \cdot \sum_{1 \leq j \leq k} \mathsf{occ}_j^{(n)} \\
&\geq\; \sum_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \mathsf{det}(\mathcal{D}|C_j)(u^{(n)}|Z_{j,i}^{(n)}) \;-\; m \cdot c_{\mathsf{max}} \cdot \sum_{1 \leq j \leq k} \mathsf{occ}_j^{(n)} \\
&\geq\; \sum_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \Big(\mathsf{cost}(\rho_j^{(n)}|Z_{j,i}^{(n)}) - (K+1) \cdot c_{\mathsf{max}}\Big) \;-\; m \cdot c_{\mathsf{max}} \cdot n \\
&\geq\; \sum_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \Big(\lceil n \cdot \alpha_i \rceil \cdot \mathsf{occ}_j^{(n)} \cdot \mathsf{cost}_j(L_i)\Big) \;-\; m \cdot c_{\mathsf{max}} \cdot \big(k \cdot (K+1) + n\big) \\
&\geq\; \sum_{1 \leq j \leq k} \Big(\mathsf{occ}_j^{(n)} \cdot n \cdot \sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)\Big) \;-\; m \cdot c_{\mathsf{max}} \cdot \big(k \cdot (K+1) + n\big)
\end{aligned}$$

$$\geq\ n \cdot \Big(\sum_{1 \leq j \leq k} \mathsf{occ}_j^{(n)} \Big) \cdot \min_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \Big(\alpha_i \cdot \mathsf{cost}_j(L_i) \Big)\ -\ m \cdot c_{\mathsf{max}} \cdot \big(k \cdot (K+1) + n \big)$$

$$\geq\ n \cdot \big(n - k \big) \cdot \min_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \Big(\alpha_i \cdot \mathsf{cost}_j(L_i) \Big)\ -\ m \cdot c_{\mathsf{max}} \cdot \big(k \cdot (K+1) + n \big)$$

$$=\ n^2 \cdot \min_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)\ -\ \mathcal{O}(n). \qquad (*)$$

Similarly, we easily obtain an upper bound to the length of the word $u^{(n)}$:

$$|u^{(n)}| = n \cdot \sum_{1 \leq i \leq m} \Big(\lceil n \cdot \alpha_i \rceil \cdot |L_i| \Big)\ +\ |\sigma_0| + n \cdot \sum_{1 \leq i \leq m} |\sigma_i|$$

$$\leq n^2 \cdot \sum_{1 \leq i \leq m} \Big(\alpha_i \cdot |L_i| \Big)\ +\ n \cdot \sum_{1 \leq i \leq m} |L_i| + K + n \cdot m \cdot K$$

$$= n^2 \cdot \sum_{1 \leq i \leq m} \alpha_i \cdot |L_i|\ +\ \mathcal{O}(n). \qquad (**)$$

Pairing the above equations will allow us to prove the claim of the lemma in the case where all states of the cycles $L_1, \ldots, L_m$ are mutually reachable in $\bar{\mathcal{D}}$ (recall that we made such an assumption at the beginning of the proof).

Since, in general, this assumption may not hold, in order to complete the proof of the lemma we need to show how to derive Equation (4) where some cycles among $L_1, \ldots, L_m$ cannot be reached from the others cycles.

Let $v_1, \ldots, v_m$ be the words spelled by the cycles $L_1, \ldots, L_m$ in $\bar{\mathcal{D}}$. First of all, notice that $\mathsf{det}(\mathcal{D}|C_1), \ldots, \mathsf{det}(\mathcal{D}|C_k)$ are complete automata, namely, in these automata, every state has one outgoing transition labelled with each letter of the alphabet. This means that $\bar{\mathcal{D}}$ is also complete. In particular, we can find a bottom strongly-connected component of $\bar{\mathcal{D}}$ that contains a set of mutually reachable cycles $L'_1, \ldots, L'_m$, where each cycle L'_i spells a repetition of the word v_i, say $v_i^{l_i}$ for some positive number l_i. Now, set $l = l_1 \cdot \ldots \cdot l_m$ and define the cycles $L''_i = (L'_i)^{\frac{l}{l_i}}$. Clearly, the states of the cycles $L''_1, \ldots, L''_m$ are mutually reachable. Furthermore, we can derive both Equation (*) and Equation (**) exactly as we did before. From these equations we easily get

$$\limsup_{n \to \infty} \frac{\mathsf{cost}(\gamma^{(n)})}{|u^{(n)}|} \quad \geq \quad \limsup_{n \to \infty} \frac{n^2 \cdot \min_{1 \leq j \leq k} \sum_{1 \leq i \leq m} \alpha_i \cdot \mathsf{cost}_j(L''_i)\ -\ \mathcal{O}(n)}{n^2 \cdot \sum_{1 \leq i \leq m} \alpha_i \cdot |L''_i|\ +\ \mathcal{O}(n)}$$

Moreover, since each cycle L_i'' spells the l_i-fold repetition of the word spelled by L_i, we derive from the second claim of Proposition 2 the following equality:

$$\mathsf{cost}(L_i'') \;=\; l \cdot \mathsf{cost}_j(L_i).$$

Finally, by replacing $\mathsf{cost}(L_i'')$ with $l \cdot \mathsf{cost}_j(L_i)$ in the above equation, we derive the desired Equation (4):

$$\limsup_{n\to\infty} \frac{\mathsf{cost}(\gamma^{(n)})}{|u^{(n)}|} \;\geq\; \limsup_{n\to\infty} \frac{n^2 \cdot \min\limits_{1\leq j\leq k} \sum_{1\leq i\leq m} \alpha_i \cdot l \cdot \mathsf{cost}_j(L_i) \;-\; \mathcal{O}(n)}{n^2 \cdot \sum_{1\leq i\leq m} \alpha_i \cdot l \cdot |L_i| \;+\; \mathcal{O}(n)}$$

$$= \min_{1\leq j\leq k} \frac{\sum_{1\leq i\leq m} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1\leq i\leq m} \alpha_i \cdot |L_i|}.$$

This completes the proof of Lemma 4. □

By virtue of Lemma 3 and Lemma 4, the proof of Equation (2) from Theorem 1 becomes trivial.

We make a few remarks related to the effectiveness of the characterization. First of all, we observe that the right handside term of Equation (2) can be rewritten as the following instance of a linear programming problem:

$$\texttt{maximize} \quad y \quad \texttt{subject to} \quad \begin{array}{ll} \sum_{1\leq i\leq m} c_{i,j} \cdot x_i \geq y & \forall 1 \leq j \leq k \\ \sum_{1\leq i\leq m} x_i \leq 1, \quad x_i \geq 0 & \forall 1 \leq i \leq m. \end{array}$$

where, for every $1 \leq i \leq m$ and every $1 \leq j \leq k$, $c_{i,j} = \frac{\mathsf{cost}_j(L_i)}{|L_i|}$. Intuitively, the variables $x_1, \ldots, x_m$ represent the values $\alpha_1 \cdot |L_1|$, …, $\alpha_m \cdot |L_m|$ normalized in such a way that they sum up to 1, and the variable y represents an under-approximation of the value of the right handside term of the equation. It is also known [8] that the optimal choices for the parameters $x_1, \ldots, x_m, y$ can be found at the 'corners' of the $(m+1)$-dimensional polyhedron that results from the intersection of the finitely many half-spaces defined by the above linear inequalities. This explains why we put $\max_{\alpha_1,\ldots,\alpha_m\geq 0}$ instead of $\sup_{\alpha_1,\ldots,\alpha_m\geq 0}$ in Equation (2). Moreover, it also implies that the asymptotic cost $\mathsf{acost}(\mathcal{D})$ is a rational number.

Regarding the complexity of the problem of computing $\mathsf{acost}(\mathcal{D})$, we observe that (i) the size $|\bar{\mathcal{D}}|$ of the multi-distance automaton $\bar{\mathcal{D}}$ is exponential in $|\mathcal{D}|$, (ii) each simple cycle L_i has length at most linear in $|\bar{\mathcal{D}}|$, (iii) the number m of all simple cycles of $\bar{\mathcal{D}}$ is exponential in $|\bar{\mathcal{D}}|$, and (iv) each constant

$c_{i,j} = \frac{\mathsf{cost}_j(L_i)}{|L_i|}$ can be computed in time polynomial in $|\bar{\mathcal{D}}|$ and $|L_i|$. Overall, the problem of computing the asymptotic cost of $\mathcal{D}$ is reduced, in time doubly exponential, to an instance of a linear programming problem. The latter problem is known to be in P [9], which proves that $\mathsf{acost}(\mathcal{D})$ can be computed in doubly exponential time.

If we consider the threshold problem for the asymptotic cost, that is, the problem of deciding whether $\mathsf{acost}(\mathcal{D}) \leq \nu$ for a given a distance automaton $\mathcal{D}$ satisfying the shortcut property and a given rational number ν, then the complexity can be lowered to coNExp. Indeed, one observes that the cost of the projection into the j-th component of a simple cycle L of $\bar{\mathcal{D}}$ is at most $|\bar{\mathcal{D}}| \cdot c_{\mathsf{max}}$, where c_{max} is the maximum cost that appears in the transitions of $\bar{\mathcal{D}}$. This implies that there exist at most $M = |\bar{\mathcal{D}}|^{k+1} \cdot c_{\mathsf{max}}^k$ (i.e., exponentially in $|\mathcal{D}|$) distinct tuples $(c_1, \ldots, c_k, l)$ such that $c_j = \mathsf{cost}_j(L)$ and $l = |L|$ for some simple cycle L of $\bar{\mathcal{D}}$. Equation (2) can be then rewritten as

$$\mathsf{acost}(\mathcal{D}) \;=\; \max_{\substack{\alpha_1,\ldots,\alpha_M \geq 0 \\ L_1,\ldots,L_M \text{ simple cycles of } \bar{\mathcal{D}}}} \; \min_{1 \leq j \leq k} \; \frac{\sum_{1 \leq i \leq M} \alpha_i \cdot \mathsf{cost}_j(L_i)}{\sum_{1 \leq i \leq M} \alpha_i \cdot |L_i|}.$$

and hence $\mathsf{acost}(\mathcal{D}) \leq \nu$ holds iff for all M-tuples of simple cycles $L_1, \ldots, L_M$ of $\bar{\mathcal{D}}$, with $M = |\bar{\mathcal{D}}|^2 \cdot c_{\mathsf{max}} \cdot k$, the system of the following linear inequalities in the variables $\alpha_1, \ldots, \alpha_M$ is *unsatisfiable*:

$$\begin{array}{ll} \alpha_1 \;\geq\; 0 & \sum_{1 \leq i \leq M} \left(\mathsf{cost}_1(L_i) - \nu \cdot |L_i|\right) \cdot \alpha_i \;>\; 0 \\ \ldots & \ldots \\ \alpha_M \;\geq\; 0 & \sum_{1 \leq i \leq M} \left(\mathsf{cost}_k(L_i) - \nu \cdot |L_i|\right) \cdot \alpha_i \;>\; 0. \end{array}$$

Since satisfiability of systems of linear equations is decidable in polynomial time, this gives a coNExp algorithm that decides whether $\mathsf{acost}(\mathcal{D}) \leq \nu$. As a consequence, we have that the complexity of the threshold problem for the asymptotic repair cost for a universal restriction language and a target language represented by a NFA is between PSpace and coNExp (note that the PSpace lower bound follows from a reduction from the universality problem for NFA but it holds also for target languages represented by DFA):

Proposition 4. *The problem of deciding, given an alphabet Σ, an NFA $\mathcal{T}$, and a rational number ν, whether $\mathsf{acost}(\Sigma^*, \mathscr{L}(\mathcal{T})) \leq \nu$ is in* coNExp *and it is* PSpace*-hard already when $\mathcal{T}$ is a DFA and $\nu = \frac{1}{2}$.*

Proof. The coNExp upper bound of the considered threshold problem follows directly from the previous arguments. Below, we prove that the analogous problem that involves target languages represented by DFA is already PSpace-hard. The proof is by reduction of the universality problem for NFA. Let us fix an NFA $\mathcal{A}$ over the alphabet Σ.

First of all, we construct an intermediate NFA $\mathcal{T}$ that recognizes the language $(\mathscr{L}(\mathcal{A})\ \{\#\})^*$, where $\#$ is a fresh symbol not belonging to Σ. Let $\Delta = \Sigma \uplus \{\#\}$ and recall that $\mathcal{A}$ recognizes the universal language Σ^* if and only if Δ^* is repairable into $\mathscr{L}(\mathcal{T})$ with uniformly bounded cost. In fact, it is easy to see that the following stronger property holds:

$$\begin{aligned} \mathscr{L}(\mathcal{A}) = \Sigma^* \quad &\text{iff} \quad \mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T})) < \infty \\ &\text{iff} \quad \mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T})) = 0. \end{aligned} \tag{$\star$}$$

Note that this already implies that the threshold problem for the asymptotic cost of a target NFA is PSpace-hard. Below, we transform the NFA $\mathcal{T}$ into a DFA $\mathcal{T}'$ such that

$$\mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T})) < \infty \quad \text{implies} \quad \mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) \leq \tfrac{1}{2} \tag{a}$$

$$\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T})) > 0 \quad \text{implies} \quad \mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) > \tfrac{1}{2} \tag{b}$$

(note that pairing the implications (a) and (b) with the equivalences in $(\star)$ gives the desired reduction).

Intuitively, the DFA $\mathcal{T}'$ is defined in such a way that it accepts all and only the successful runs of the NFA $\mathcal{T}$. Precisely, if $\mathcal{T} = (\Delta, Q, E, I, F)$, then we define $\mathcal{T}' = (\Delta', Q', E', q_0', F')$, where

- $\Delta' = \Delta \uplus Q$;
- $Q' = Q \uplus (Q \times \Delta) \uplus \{q_0'\}$, where q_0' is a new state;
- E' consists of all transitions of the form:
  1. $\big(q_0', q, q\big)$, with $q \in I$ (namely, at the beginning $\mathcal{T}'$ reads an initial state q of $\mathcal{T}$ as input symbol and accordingly moves from its initial state q_0 to q),
  2. $\big(q, b, (q, b)\big)$, with $q \in Q$, $b \in \Delta$
     (namely, on input symbol $b \in \Delta$, $\mathcal{T}'$ moves deterministically from any state q to the state (q, b)),

3. $\big((q,b),q',q'\big)$, with $(q,b,q') \in E$
(namely, $\mathcal{T}'$ moves deterministically from state (q,b) to state q' on input q', provided that (q,b,q') is a valid transition of $\mathcal{T}$);

- $F' = F$.

Clearly, the automaton $\mathcal{T}'$ is deterministic. Moreover, it is easy to see that the language recognized by $\mathcal{T}'$ consists of all and only the encodings of the successful runs of $\mathcal{T}$ (here we represent a run of $\mathcal{T}$ as a sequence of states from Q interleaved with letters from Δ).

Let us prove the first implication (a). Suppose that $\mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T})) < \infty$, namely, that Δ^* can be repaired into $\mathscr{L}(\mathcal{T})$ with uniformly bounded cost. Let $u = a_1\ a_2\ \ldots\ a_{2n-1}\ a_{2n}$ be a word of even length over Δ, let v be a word that belongs to $\mathscr{L}(\mathcal{T})$, and let v' be the encoding of a successful run of $\mathcal{T}$ on v (hence $v' \in \mathscr{L}(\mathcal{T}')$). We denote by u_{even} the sub-sequence obtained from u by selecting the symbols at the even positions, i.e., $u_{\mathsf{even}} = a_2\ a_4\ \ldots\ a_{2n}$, and we analyze its edit distance from v. We know form the definition of edit distance that the word v can be factorized into n $(= |u_{\mathsf{even}}|)$ possibly empty words $v_1, v_2, \ldots, v_n$ such that

$$\mathsf{dist}(u_{\mathsf{even}}, v) \ = \ \textstyle\sum_{1 \le i \le n} \ \mathsf{dist}(a_{2i}, v_i).$$

The factorization $v_1, v_2, \ldots, v_n$ of v induces a corresponding factorization $v'_1, v'_2, \ldots, v'_n, v'_{n+1}$ of v', where $|v'_i| = 2 \cdot |v_i|$ for all $1 \le i \le n$ and $|v'_{n+1}| = 1$. We now observe that for all $1 \le i \le n$,

$$\mathsf{dist}(a_{2i-1}\ a_{2i},\ v'_i) \ \le \ \mathsf{dist}(a_{2i}, v_i) \ + \ \max\{|v_i|, 1\}$$

and hence

$$\begin{aligned}
\mathsf{dist}(u, v') \ &\le \ \textstyle\sum_{1 \le i \le n} \ \mathsf{dist}(a_{2i-1}\ a_{2i},\ v'_i) \ + \ 1 \\
&\le \ \textstyle\sum_{1 \le i \le n} \ \big(\mathsf{dist}(a_{2i}, v_i) + \max\{|v_i|, 1\}\big) \ + \ 1 \\
&\le \ \mathsf{dist}(u_{\mathsf{even}}, v) \ + \ \max\{|u_{\mathsf{even}}|, |v|\} \ + \ 1.
\end{aligned}$$

Recall that the minimum of $\mathsf{dist}(u_{\mathsf{even}}, v)$ over all $v \in \mathscr{L}(\mathcal{T})$ is uniformly bounded by $\mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T}))$, and it is realized by some word $v \in \mathscr{L}(\mathcal{T})$ that has length $|v| \le |u_{\mathsf{even}}| + \mathsf{dist}(u_{\mathsf{even}}, v) \le |u_{\mathsf{even}}| + \mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T}))$. We thus derive

$$\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) \ = \ \lim_{n \to \infty} \ \sup_{\substack{u \in (\Delta\Delta)^* \\ |u| \ge n}} \ \min_{v' \in \mathscr{L}(\mathcal{T}')} \ \frac{\mathsf{dist}(u, v')}{|u|}$$

$$\leq \lim_{n\to\infty} \sup_{\substack{u\in(\Delta\Delta)^* \\ |u|\geq n}} \min_{v\in\mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u_{\mathsf{even}}, v) \;+\; \max\{|u_{\mathsf{even}}|, |v|\} \;+\; 1}{|u|}$$

$$\leq \lim_{n\to\infty} \sup_{\substack{u\in(\Delta\Delta)^* \\ |u|\geq n}} \frac{2\cdot \mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T})) \;+\; |u_{\mathsf{even}}| \;+\; 1}{|u|}$$

$$\leq \frac{1}{2}.$$

This proves the first implication (a).

As for the second implication (b), we consider a generic pair of words $u \in \Delta^*$ and $v \in \mathscr{L}(\mathcal{T})$. We denote by v' the encoding of a successful run of $\mathcal{T}$ on v (hence $v' \in \mathscr{L}(\mathcal{T}')$) and by u^{double} the word obtained from u by repeating each letter twice (i.e., $u^{\mathsf{double}} = u(1)\, u(1)\, u(2)\, u(2)\, \ldots\, u(|u|)\, u(|u|)$). Below we show that

$$\mathsf{dist}(u^{\mathsf{double}}, v') \;\geq\; \mathsf{dist}(u, v) + |u| + 1$$

(we will then argue that this inequality entails the implication (b)).

For the sake of brevity, we write v' as $q_0\; b_1\; q_1\; \ldots\; q_{n-1}\; b_n\; q_n$. We know from the definition of edit distance that the word u^{double} can be factorized into $n+1$ words u_1^{double}, u_2^{double}, …, $u_{n+1}^{\mathsf{double}}$ such that

$$\mathsf{dist}(u^{\mathsf{double}}, v') \;=\; \textstyle\sum_{1\leq i\leq n} \mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i) \;+\; \mathsf{dist}(u_{n+1}^{\mathsf{double}}, q_n).$$

Without loss of generality we can assume that the first factor u_1^{double} has even length. Indeed, suppose that this is not the case and let u_i^{double} be the first factor after u_1^{double} that is non-empty. Observe that the first symbol of u_i^{double} coincides with the last symbol of u_1^{double}. Now, we consider a new factorization of u^{double} that is obtained from the previous one by removing the first symbol from the i-th factor u_i^{double} and by adding it at the end of the first factor u_1^{double}. Let us write the new factorization as $\tilde{u}_1^{\mathsf{double}}$, u_2^{double}, …, $u_{i-1}^{\mathsf{double}}$, $\tilde{u}_i^{\mathsf{double}}$, $u_{i+1}^{\mathsf{double}}$, …, $u_{n+1}^{\mathsf{double}}$ (notice that only the first and the i-th factor are changed). It is easy to see that if b_1 occurs in $\tilde{u}_1^{\mathsf{double}}$ (or, equally, in u_1^{double}), then $\mathsf{dist}(\tilde{u}_1^{\mathsf{double}}, q_0b_1) = |\tilde{u}_1^{\mathsf{double}}| - 1 = |u_1^{\mathsf{double}}| = \mathsf{dist}(u_1^{\mathsf{double}}, q_0b_1) + 1$, otherwise $\mathsf{dist}(\tilde{u}_1^{\mathsf{double}}, q_0b_1) = |\tilde{u}_1^{\mathsf{double}}| = |u_1^{\mathsf{double}}| + 1 = \mathsf{dist}(u_1^{\mathsf{double}}, q_0b_1) + 1$. Similarly, if b_i occurs in $\tilde{u}_i^{\mathsf{double}}$ (or, equally, in u_i^{double}), then $\mathsf{dist}(\tilde{u}_i^{\mathsf{double}}, q_{i-1}b_i) = |\tilde{u}_i^{\mathsf{double}}| - 1 = |u_i^{\mathsf{double}}| - 2 = \mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i) - 1$, otherwise $\mathsf{dist}(\tilde{u}_1^{\mathsf{double}}, q_0b_1) = |\tilde{u}_i^{\mathsf{double}}| = |u_i^{\mathsf{double}}| - 1 = \mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i) - 1$. In all cases we have $\mathsf{dist}(\tilde{u}_1^{\mathsf{double}}, q_0b_1) + \mathsf{dist}(\tilde{u}_i^{\mathsf{double}}, q_{i-1}b_i) = \mathsf{dist}(u_1^{\mathsf{double}}, q_0b_1) + \mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i)$. This means that

we could have equally considered an alternative factorization of u^{double} that begins with a factor of even length.

Following similar arguments, we can assume, without loss of generality, that all factors but the last one have even length. This allows us to define a corresponding factorization of u as $u_1, u_2, \ldots, u_n$, where each u_i is the subsequence of u_i^{double} obtained by selecting only the symbols at the odd positions – note that $u = u_1\, u_2\, \ldots\, u_n$. Thanks to the above definitions, we have that for all $1 \leq i \leq n$,

$$\mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i) = \begin{cases} 2 \cdot \mathsf{dist}(u_i, b_i) & \text{if } b_i \text{ does not occur in } u_i, \\ 2 \cdot \mathsf{dist}(u_i, b_i) + 1 & \text{if } b_i \text{ occurs in } u_i, \end{cases}$$

and hence, letting I be the set of indices $i \in \{1, \ldots, n\}$ such that b_i occurs in u_i, we obtain

$$\begin{aligned} \mathsf{dist}(u^{\mathsf{double}}, v') &= \sum_{1 \leq i \leq n} \mathsf{dist}(u_i^{\mathsf{double}}, q_{i-1}b_i) + \mathsf{dist}(u_{n+1}^{\mathsf{double}}, q_n) \\ &= \sum_{1 \leq i \leq n} 2 \cdot \mathsf{dist}(u_i, b_i) + |I| + \max\left\{1, |u_{n+1}^{\mathsf{double}}|\right\} \\ &= \mathsf{dist}(u, v) + \sum_{1 \leq i \leq n} \mathsf{dist}(u_i, b_i) + |I| + \max\left\{1, |u_{n+1}^{\mathsf{double}}|\right\} \\ &= \mathsf{dist}(u, v) + \sum_{1 \leq i \leq n} |u_i| + \max\left\{1, |u_{n+1}^{\mathsf{double}}|\right\} \\ &\geq \mathsf{dist}(u, v) + |u|. \end{aligned}$$

We thus conclude that

$$\begin{aligned} \mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) &\geq \lim_{n \to \infty} \sup_{\substack{u \in \Delta^* \\ |u^{\mathsf{double}}| \geq n}} \min_{v' \in \mathscr{L}(\mathcal{T}')} \frac{\mathsf{dist}(u^{\mathsf{double}}, v')}{|u^{\mathsf{double}}|} \\ &\geq \lim_{n \to \infty} \sup_{\substack{u \in \Delta^* \\ |u^{\mathsf{double}}| \geq n}} \min_{v \in \mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u, v) + |u|}{2 \cdot |u|} \\ &= \frac{\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T})) + 1}{2} \end{aligned}$$

which immediately entails the implication (b).

Finally, observe that, as the NFA $\mathcal{T}$ satisfies property $(\star)$ and the DFA $\mathcal{T}'$ satisfies both implications (a) and (b), we have that $\mathscr{L}(\mathcal{A}) = \Sigma^*$ implies $\mathsf{cost}(\Delta^*, \mathscr{L}(\mathcal{T})) < \infty$, and hence $\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) \leq \frac{1}{2}$; conversely, $\mathscr{L}(\mathcal{A}) \neq$

Σ^* implies $\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T})) > 0$, and hence $\mathsf{acost}(\Delta^*, \mathscr{L}(\mathcal{T}')) > \frac{1}{2}$. This reduces the universality problem for the NFA $\mathcal{A}$ to a threshold problem for the asymptotic repair cost of a DFA $\mathcal{T}'$. □

4.4. Asymptotic cost in the general case

Here we show how to generalize the characterization of the asymptotic cost in the unrestricted case to our original repair problem, which involves the presence of both a restriction and a target language. We first modify the definition of asymptotic cost for a distance automaton to include the presence of a restriction language $\mathscr{L}(\mathcal{R})$ recognized by an NFA $\mathcal{R}$:

$$\mathsf{acost}(\mathcal{R}, \mathcal{D}) \ =^{\text{def}} \lim_{n\to\infty} \sup_{\substack{u\in\mathscr{L}(\mathcal{R}) \\ |u|\geq n}} \frac{\mathcal{D}(u)}{|u|}.$$

Thanks to Proposition 1, we have that the asymptotic cost $\mathsf{acost}(R, T)$ for two regular languages R and T recognized by NFA $\mathcal{R}$ and $\mathcal{T}$ is equal to $\mathsf{acost}(\mathcal{R}, \mathcal{D}^{\mathsf{edit}}_{\mathcal{T}})$.

As usual, given a distance automaton $\mathcal{D}$ satisfying the shortcut property, we denote by $\bar{\mathcal{D}}$ the multi-distance automaton $\mathsf{det}(\mathcal{D}|C_1) \times \ldots \times \mathsf{det}(\mathcal{D}|C_k)$, where $C_1, \ldots, C_k$ are all the SCCs of $\mathcal{D}$. Moreover, given an NFA $\mathcal{R}$ and a SCC B of it, we consider the synchronized product $\bar{\mathcal{D}} \times (\mathcal{R}|B)$ of the multi-distance automaton $\bar{\mathcal{D}}$ and the sub-automaton $\mathcal{R}|B$, which is obtained from $\mathcal{R}$ by restricting the set of states to B (it does not matter which state is chosen to be initial/final in $\mathcal{R}|B$). We then denote by $L^B_1, \ldots, L^B_{m^B}$ all the simple cycles of $\bar{\mathcal{D}} \times (\mathcal{R}|B)$. Finally, given a simple cycle L^B_i of $\bar{\mathcal{D}} \times (\mathcal{R}|B)$ and a SCC C of $\mathcal{D}$, we denote by $\mathsf{cost}_C(L^B_i)$ the cost of the projection of L^B_i into the component C of $\bar{\mathcal{D}} \times (\mathcal{R}|B)$. The generalized characterization result is as follows:

Theorem 2. *For every (trimmed) NFA $\mathcal{R}$ and every (trimmed) distance automaton $\mathcal{D}$ satisfying the shortcut property,*

$$\mathsf{acost}(\mathcal{R}, \mathcal{D}) \ = \max_{\substack{\tau=B_1\ldots B_h \ \in \ \mathsf{dag}(\mathcal{R}) \\ \alpha^{B_1}_1, \ \ldots, \ \alpha^{B_1}_{m^{B_1}} \geq 0 \\ \ldots\ldots \\ \alpha^{B_h}_1, \ \ldots, \ \alpha^{B_h}_{m^{B_h}} \geq 0}} \min_{\pi=C_1\ldots C_h \ \in \ \mathsf{dag}(\mathcal{D})} \frac{\sum\limits_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha^{B_l}_i \cdot \mathsf{cost}_{C_l}(L^{B_l}_i)}{\sum\limits_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha^{B_l}_i \cdot |L^{B_l}_i|} \tag{5}$$

Proof sketch. The proof is very similar to the proof of Theorem 1. In particular, we prove two inequalities between the asymptotic cost $\mathsf{acost}(\mathcal{R},\mathcal{D})$ and the right handside expression.

Let us consider first the inequality

$$\mathsf{acost}(\mathcal{R},\mathcal{D}) \;\leq\; \max_{\substack{\tau=B_1\dots B_h \,\in\, \mathsf{dag}(\mathcal{R}) \\ \alpha_1^{B_1},\,\dots,\,\alpha_{m^{B_1}}^{B_1}\geq 0 \\ \dots\dots \\ \alpha_1^{B_h},\,\dots,\,\alpha_{m^{B_h}}^{B_h}\geq 0}} \;\min_{\pi=C_1\dots C_h \,\in\, \mathsf{dag}(\mathcal{D})} \frac{\sum\limits_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha_i^{B_l}\cdot \mathsf{cost}_{C_l}(L_i^{B_l})}{\sum\limits_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha_i^{B_l}\cdot |L_i^{B_l}|}$$

In order to prove this inequality, one needs to fix, as in Lemma 3, a family of words $(u^{(n)})_{n\in\mathbb{N}}$ from the restriction language $\mathscr{L}(\mathcal{R})$ such that

$$\mathsf{acost}(\mathcal{D}) \;=\; \limsup_{n\to\infty}\; \frac{\mathcal{D}(u^{(n)})}{|u^{(n)}|}.$$

By possibly restricting to sub-families of words, one can replace lim sup by lim in the above equation.

One new ingredient is the following. Without loss of generality, we can also assume that all words $u^{(n)}$ induce successful runs on the NFA $\mathcal{R}$ following the same path of SCCs of $\mathcal{R}$. More precisely, we denote by $\sigma^{(n)}$ some successful run of $\mathcal{R}$ on $u^{(n)}$, and by $\tau^{(n)}$ the path in $\mathsf{dag}(\mathcal{R})$ that consists of the sequence of SCCs visited by $\sigma^{(n)}$. Then, since there are only a finite number of paths in $\mathsf{dag}(\mathcal{R})$, we can restrict ourselves to suitable sub-families of words and runs in such a way that all paths $\tau^{(n)}$ are the same. We denote them simply by $\tau = B_1 \dots B_h$.

The proof then continues as follows. We partition the domain of each run $\sigma^{(n)}$ of the NFA $\mathcal{R}$ into some intervals $Y_1^{(n)},\dots,Y_h^{(n)}$ (recall that h is the number of SCCs in the path τ) in such a way that each sub-sequence $\sigma^{(n)}|Y_l^{(n)}$, for $1\leq l\leq h$, is a run of the sub-automaton $\mathcal{R}|B_l$ on the sub-word $u^{(n)}|Y_l^{(n)}$ (in fact the sets $Y_1^{(n)},\dots,Y_h^{(n)}$ do not form a partition of the entire domain of $\sigma^{(n)}$, since there can be transitions crossing different SCCs in $\mathcal{R}$; however, the number of these transitions is at most the number of SCCs in $\mathcal{R}$, and thus their cost is negligible for n that tends to ∞). One then considers the (unique) successful run $\rho^{(n)}$ of $\bar{\mathcal{D}}$ on the word $u^{(n)}$. Given an index $1\leq l\leq h$ and a SCC C of $\mathcal{D}$, we denote by $\rho_{l,C}^{(n)}$ the projection of

the sub-run $\rho^{(n)}|Y_l^{(n)}$ into the component C. Every sequence $\rho_{l,C}^{(n)}$ can be viewed as a run of $\mathsf{det}(\mathcal{D}|C)$ on the sub-word $u^{(n)}|Y_l^{(n)}$. This run has cost almost equal (up to additive constants) to the cost of some optimal run $\gamma_{l,C}^{(n)}$ of $\mathcal{D}|C$ on $u^{(n)}|Y_l^{(n)}$. Moreover, given a path $\pi = C_1 \dots C_h$ in $\mathsf{dag}(\mathcal{D})$, one can construct a run $\gamma_\pi^{(n)}$ of $\mathcal{D}$ on $u^{(n)}$ by 'concatenating' the runs $\gamma_{1,C_1}^{(n)}, \dots, \gamma_{h,C_h}^{(n)}$ (this requires the use of the shortcut property to correct the possible pairs of consecutive transitions that have unmatched states). This shows that

$$\begin{aligned}
\mathcal{D}(u^{(n)}) &\leq \min_{\pi=C_1\dots C_h \,\in\, \mathsf{dag}(\mathcal{D})} \mathsf{cost}(\gamma_\pi^{(n)}) \;+\; \mathcal{O}(1) \\
&= \min_{\pi=C_1\dots C_h \,\in\, \mathsf{dag}(\mathcal{D})} \textstyle\sum_{1\leq l\leq h} \mathsf{cost}(\gamma_{l,C_l}^{(n)}) \;+\; \mathcal{O}(1) \\
&= \min_{\pi=C_1\dots C_h \,\in\, \mathsf{dag}(\mathcal{D})} \textstyle\sum_{1\leq l\leq h} \mathsf{cost}(\rho_{l,C_l}^{(n)}) \;+\; \mathcal{O}(1).
\end{aligned}$$

Given the above inequality, the rest of the proof is similar to that of Lemma 3, namely, we decompose each run $\rho_{l,C_l}^{(n)}$ into simple cycles and we approximate its cost up to additive constants.

We now turn to the converse inequality:

$$\mathsf{acost}(\mathcal{R},\mathcal{D}) \;\geq\; \max_{\substack{\tau=B_1\dots B_h \,\in\, \mathsf{dag}(\mathcal{R}) \\ \alpha_1^{B_1},\,\dots,\,\alpha_{m^{B_1}}^{B_1}\geq 0 \\ \dots\dots \\ \alpha_1^{B_h},\,\dots,\,\alpha_{m^{B_h}}^{B_h}\geq 0}} \;\min_{\pi=C_1\dots C_h \,\in\, \mathsf{dag}(\mathcal{D})} \frac{\displaystyle\sum_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha_i^{B_l}\cdot \mathsf{cost}_{C_l}(L_i^{B_l})}{\displaystyle\sum_{\substack{1\leq l\leq h \\ 1\leq i\leq m^{B_l}}} \alpha_i^{B_l}\cdot |L_i^{B_l}|}$$

As in the proof of Lemma 4, the first step is to fix a path $\tau = B_1 \dots B_h$ in $\mathsf{dag}(\mathcal{R})$ and some parameters $\alpha_i^{B_l} \geq 0$ for each $1 \leq l \leq h$ and each $1 \leq i \leq m^{B_l}$, where m^{B_l} denotes the number of simple cycles of the automaton $\bar{\mathcal{D}}\times(\mathcal{R}|B_l)$.

One proves the inequality by defining the following family of runs of $\bar{\mathcal{D}}\times\mathcal{R}$:

$$\rho^{(n)} \;=^{\text{def}}\; \sigma_0 \left(\rho_{1,\mathsf{cycles}}^{(n)}\right)^n \;\dots\; \sigma_{h-1} \left(\rho_{h,\mathsf{cycles}}^{(n)}\right)^n \sigma_h$$

where, for all $1 \leq l \leq h$,

$$\rho_{l,\mathsf{cycles}}^{(n)} \;=^{\text{def}}\; (L_1^{B_l})^{\lceil n\cdot\alpha_1^{B_l}\rceil}\, \sigma_{l,1}\, (L_2^{B_l})^{\lceil n\cdot\alpha_2^{B_l}\rceil} \;\dots\; \sigma_{l,m^{B_l}-1}\, (L_{m^{B_k}}^{B_l})^{\lceil n\cdot\alpha_{m^{B_l}}^{B_l}\rceil}\, \sigma_{l,m^{B_l}}$$

and $\sigma_0, \sigma_1, \dots, \sigma_h, \sigma_{l,1}, \dots, \sigma_{l,m^{B_l}}$ are suitable runs of $\bar{\mathcal{D}}$ of bounded length that connect the various simple cycles $L_i^{B_l}$. Accordingly, one defines $u^{(n)}$ to

be the word spelled out by the run $\rho^{(n)}$. Observe that, by construction (and under the assumption that the automaton $\mathcal{R}$ is trimmed), this word belongs to the language recognized by the NFA $\mathcal{R}$.

It is not difficult then to generalize the arguments used in Lemma 4 (we thus omit the details). □

Using arguments similar to the complexity analysis of the unrestricted case, we obtain a coNExp algorithm that decides whether the asymptotic repair cost $\mathsf{acost}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$ $(= \mathsf{acost}(\mathcal{R}, \mathcal{D}_{\mathcal{T}}^{\mathsf{edit}}))$ associated with two NFA $\mathcal{R}$ and $\mathcal{T}$ is less than or equal to a certain threshold $\nu \in \mathbb{Q}$:

Corollary 1. *The problem of deciding, given two NFA $\mathcal{R}$ and $\mathcal{T}$ and a rational number ν, whether $\mathsf{acost}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) \leq \nu$ is in* coNExp.

5. Asymptotic cost in the streaming case

Here we characterize the asymptotic repair (aggregate) cost in the streaming setting in terms of the value of a mean-payoff game [7].

5.1. Mean-payoff games

A *mean-payoff game* is an infinite, turn-based game played over an arena $\mathcal{A} = (V, E, v_0)$, where V is the union of two disjoint finite sets of vertices, V_{Adam} (owned by player Adam) and V_{Eve} (owned by player Eve), $E \subseteq V \times \mathbb{N} \times V$ is a finite set of weighted edges, and $v_0 \in V$ is an initial vertex. The game starts at v_0 and, at each round, the player who owns the current vertex v moves along an edge $(v, c, v') \in E$. The reward for Adam (resp., the penalty for Eve) in an infinite play $\pi = (v_0, c_1, v_1)\ (v_1, c_2, v_2)\ \ldots$ is given by the value $\nu^{\pi}_{\mathsf{Adam}}$ (resp., ν^{π}_{Eve}), where

$$\nu^{\pi}_{\mathsf{Adam}} =^{\text{def}} \liminf_{n\to\infty} \frac{\sum_{i=1}^{n} c_i}{n} \qquad \nu^{\pi}_{\mathsf{Eve}} =^{\text{def}} \limsup_{n\to\infty} \frac{\sum_{i=1}^{n} c_i}{n}.$$

Intuitively, Adam wants to maximize his reward $\nu^{\pi}_{\mathsf{Adam}}$ while Eve wants to minimize her penalty ν^{π}_{Eve}.

It is known from [7] that, in any mean-payoff game, the best reward that can be enforced by Adam coincides with the least penalty that can be enforced by Eve, and, furthermore, these values can be achieved by positional strategies:

Theorem 3 (Ehrenfeucht and Mycielski [7])**.** *We can associate with each mean-payoff game $\mathcal{A}$ a value $\nu_{\mathcal{A}}$ such that Adam (resp., Eve) has a positional strategy that guarantees $\nu^{\pi}_{\mathsf{Adam}} \geq \nu_{\mathcal{A}}$ (resp., $\nu^{\pi}_{\mathsf{Eve}} \leq \nu_{\mathcal{A}}$) for all plays π that respect his (resp., her) strategy.*

In view of the above theorem, we can denote by $\nu_{\mathcal{A}}$ the value of a mean-payoff game over the arena $\mathcal{A}$ and we can restrict ourselves to positional strategies for both Adam and Eve. We will represet a positional strategy for Adam (resp., Eve) as a function from Adam's vertices (resp., Eve's vertices) to outgoing edges.

5.2. Characterization of asymptotic streaming cost

Let R and T be the languages recognized by two (trimmed) DFA $\mathcal{R} = (\Sigma, Q, \delta, q_0, F)$ and $\mathcal{T} = (\Delta, Q', \delta', r_0, F')$, respectively. To compute the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(R, T)$ for streaming (0-lookahead) repair strategies we construct the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$, where Adam's vertices are pairs of the form (q, r), with $q \in Q$ and $r \in Q'$, and Eve's vertices are pairs of the form (q, r, a), with $q \in Q$, $r \in Q'$, and $a \in \Sigma$. The edges of the arena are triples of the form $\big((q, r), 0, (q', r, a)\big)$, where $q' = \delta(q, a)$, or of the form $\big((q, r, a), c, (q, r')\big)$, where $r' \in Q'$ and $c = \min\big\{\mathsf{dist}(a, v) \ : \ v \in \mathscr{L}(\mathcal{T}_{r,r'})\big\}$. (recall that $\mathcal{T}_{r,r'}$ is the DFA obtained from $\mathcal{T}$ by letting r be the initial state and r' be the unique final state). The initial vertex of the arena is the pair (q_0, r_0) (so Adam moves first). Observe that the final states of $\mathcal{R}$ and $\mathcal{T}$ do not play any relevant role in this definition: this is because $\mathcal{R}$ and $\mathcal{T}$ are assumed to be trimmed and the costs of moving from non-final states to final states are irrelevant for the asymptotic behavior. Furthermore, note that the game alternates between Adam and Eve, and only the second player can incur positive costs.

Remark 2. *In order to avoid that players get stuck at some vertices of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ that have no outgoing edges, we tacitly assume that all states of the DFA $\mathcal{R}$ and $\mathcal{T}$ can reach non-transient states (i.e., states contained in some cycles). In particular, as the automata are also trimmed, we have that for all states $q \in Q$ and $r \in Q'$, both languages $\mathscr{L}(\mathcal{R}_{q,F})$ and $\mathscr{L}(\mathcal{T}_{r,F'})$ contain infinitely many words. Note that it is safe to make this assumption when considering the streaming asymptotic cost, as this cost is preserved when we remove the states that can only reach transient states.*

Below, we show that the value of the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$, multiplied by 2, coincides with the asymptotic aggregate cost in the streaming setting.

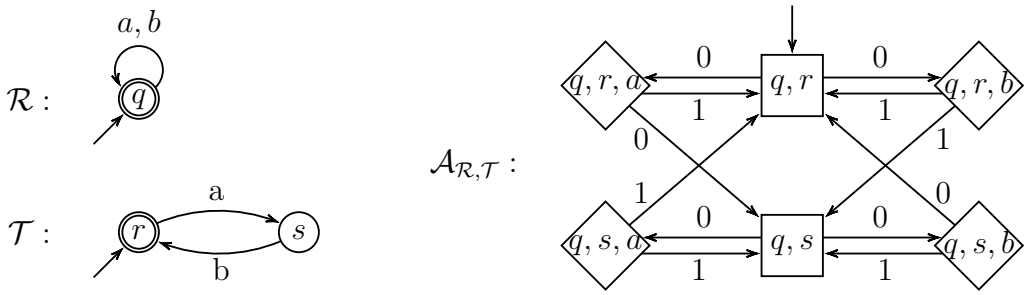

Figure 2: Two DFA and the arena for the associated mean-payoff game.

Theorem 4. *For all (trimmed) DFA $\mathcal{R}$ and $\mathcal{T}$, we have*

$$\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) \;=\; 2 \cdot \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$$

where $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ is the value of the mean-payoff game over the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$. Moreover, $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$ is rational, it can be computed in polynomial time, and it is achieved by a single streaming edit strategy for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$ – which can also be computed in P.

Example 6. *Consider the restriction and target languages $R = (a + b)^*$ and $T = (ab)^*$, whose automata $\mathcal{R}$ and $\mathcal{T}$ and mean-payoff arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ are shown in Figure 2 (diamond nodes are owned by Eve and square nodes are owned by Adam). One can easily see that an optimal positional strategy for Adam is to play $(q, r) \xrightarrow{\text{Adam}} (q, r, b)$ and $(q, s) \xrightarrow{\text{Adam}} (q, s, a)$. With this optimal strategy we get that the value $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ of the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ is equal to $\frac{1}{2}$ and thus $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(R, T) = 1$. This value contrasts with the non-streaming asymptotic cost between R and T, which is equal to $\frac{1}{2}$.*

Even if it seems natural that the value of the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ determines the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$, the proof of the Theorem 4 is not trivial. Indeed, the mean-payoff game corresponds directly to a version of the streaming repair problem where the input to the repair strategy is a sequence of prefixes of a *single infinite word* spelled by a run of $\mathcal{R}$. The core of the proof is to show a correspondence between the this infinitary version of the streaming repair problem and the original problem as stated in Section 3. This is done by proving two inequalities. In one case (Lemma 5) we show that ($\star$) for an optimal strategy $\mathcal{S}$ of Eve in the mean-payoff game, one can construct a streaming repair strategy $\mathcal{S}'$ for $\mathcal{R}$

and $\mathcal{T}$ such that $\frac{\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}'}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))}{2}$ does not exceed the penalty for Eve induced by her strategy $\mathcal{S}$. Intuitively, the repair strategy $\mathcal{S}'$ mimics Eve's strategy $\mathcal{S}$ until the string terminates, at which point it performs additional insertions to get to a final state. For the other direction (Lemma 6) we consider a streaming repair strategy $\mathcal{S}'$ for $\mathcal{R}$ and $\mathcal{T}$ with asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}'}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$ and we show that no strategy $\mathcal{S}$ for Adam can guarantee a reward of more than $\frac{\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}'}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))}{2}$. By the result from [7] mentioned above, this shows that Eve can enforce a penalty less than or equal to this amount. The limit on Adam's ability is shown by combating his strategy $\mathcal{S}$ using the repair strategy $\mathcal{S}'$. Putting these two directions together, we see that the optimal streaming repair strategy is produced by first computing Eve's optimal strategy, and then applying the transformation $(\star)$ described above; one can then argue that this strategy can be computed in polynomial time from $\mathcal{R}$ and $\mathcal{T}$.

Lemma 5. *For all (trimmed) DFA $\mathcal{R}$ and $\mathcal{T}$ and all streaming repair strategies $\mathcal{S}$ for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$, we have $2 \cdot \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \leq \mathsf{acost}^{aggr}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$.*

Proof. Let us fix two DFA $\mathcal{R} = (\Sigma, Q, \delta, q_0, F)$ and $\mathcal{T} = (\Delta, Q', \delta', r_0, F')$ and a transducer $\mathcal{S} = (\Sigma, \Delta, Q'', \delta'', s_0, \Omega)$ that implements a streaming repair strategy for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$. Let us also fix an optimal positional strategy $f : Q \times Q' \to E$ for Adam, where E is the set of edges of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$.

On the basis of the transducer $\mathcal{S}$ and Adam's positional strategy f, we inductively construct (i) an infinite play π on $\mathcal{A}_{\mathcal{R},\mathcal{T}}$, (ii) an infinite word $u \in \Sigma^\omega$, and (iii) an infinite run ρ of $\mathcal{S}$ on u, as follows. The first edge of the play π is given by Adam's move $f(v_0) = (v_0, 0, v_1)$, where $v_0 = (q_0, r_0)$. Accordingly, the first symbol of the word u is the symbol a_1 that is contained in the vertex v_1 (note that $v_1 \in Q \times Q' \times \Sigma$). The run ρ of $\mathcal{S}$ at the beginning is the empty sequence. As for the induction step, we first extend ρ and π, using the transducer $\mathcal{S}$, and then we extend π and u, using Adam's strategy again. Formally:

- Given a prefix $(v_0, 0, v_1)\ (v_1, c_1, v_2)\ \ldots\ (v_{2n}, 0, v_{2n+1})$ of π that ends in a vertex $v_{2n+1} = (q_{n+1}, r_n, a_{n+1})$ owned by Eve and given the corresponding prefix $a_1\ \ldots\ a_{n+1}$ of u, we extend the prefix of ρ from $s_0 \xrightarrow{a_1/w_1} \ldots \xrightarrow{a_n/w_n} s_n$ to $s_0 \xrightarrow{a_1/w_1} \ldots \xrightarrow{a_n/w_n} s_n \xrightarrow{a_{n+1}/w_{n+1}} s_{n+1}$, where $\delta''(s_n, a_{n+1}) = (w_{n+1}, s_{n+1})$. Accordingly, we extend the prefix of π by adding the edge $(v_{2n+1}, c_{n+1}, v_{2n+2})$, where $v_{2n+2} = (q_{n+1}, r_{n+1})$, r_{n+1} is

the state of $\mathcal{T}$ reached from r_n after consuming the word w_{n+1}, and $c_{n+1} = \min\{\mathsf{dist}(a_{n+1}, w) \ : \ w \in \mathscr{L}(\mathcal{T}_{r_n, r_{n+1}})\}$.

- Similarly, given a prefix $(v_0, 0, v_1)\ (v_1, c_1, v_2)\ \ldots\ (v_{2n+1}, c_{n+1}, v_{2n+2})$ of π that ends in a vertex $v_{2n+2} = (q_{n+1}, r_{n+1})$ owned by Adam, we extend it using Adam's positional strategy f, namely, by adding the edge $f(v_{2n+2}) = (v_{2n+2}, 0, v_{2n+3})$. Accordingly, we extend the prefix of u from $a_1\ \ldots\ a_{n+1}$ to $a_1\ \ldots\ a_{n+1}\ a_{n+2}$, where a_{n+2} is the symbol contained in the vertex v_{2n+3}.

It is easy to check that the above definitions lead to an infinite play

$$\pi \ = \ (v_0, 0, v_1)\ (v_1, c_1, v_2)\ (v_2, 0, v_3)\ \ldots$$

over the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$, and an infinite run

$$\rho \ = \ s_0 \ \xrightarrow{a_1/w_1} \ s_1 \ \xrightarrow{a_2/w_2} \ \ldots$$

of the transducer $\mathcal{S}$ on the word $u = a_1\ a_2\ \ldots$ such that, for every $n \in \mathbb{N}$, $c_n \leq \mathsf{dist}(a_n, w_n)$. The play π clearly respects Adam's optimal strategy and hence, by Theorem 3, we have $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \leq \nu^{\pi}_{\mathsf{Adam}}$. Moreover, observe that every prefix $a_1 \ldots a_n$ of the infinite word u can be extended to a word $a_1\ \ldots\ a_n\ w'_n$ that belongs to the restriction language $\mathscr{L}(\mathcal{R})$, where w'_n has length at most $|Q|$. By applying the various definitions and some basic rewriting, we easily obtain:

$$\begin{aligned}
2 \cdot \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \ &\leq \ 2 \cdot \nu^{\pi}_{\mathsf{Adam}} \\
&= \ 2 \cdot \liminf_{n\to\infty} \ \frac{\sum_{i=1}^{n} c_i}{2 \cdot n} \\
&\leq \ \liminf_{n\to\infty} \ \frac{\sum_{i=1}^{n} \mathsf{dist}(a_n, w_n)}{n} \\
&\leq \ \limsup_{n\to\infty} \ \frac{\sum_{i=1}^{n} \mathsf{dist}(a_n, w_n)}{n} \\
&= \ \limsup_{n\to\infty} \ \frac{\mathsf{cost}^{\mathsf{aggr}}(a_1\ \ldots\ a_n\ w'_n,\ \mathcal{S})}{n + |w'_n|} \\
&\leq \ \mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})).
\end{aligned}$$

□

Lemma 6. *For all (trimmed) DFA $\mathcal{R}$ and $\mathcal{T}$, there is a transducer $\mathcal{S}$, whose states are pairs of states of $\mathcal{R}$ and $\mathcal{T}$, that implements a streaming repair strategy for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$ and such that $\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \leq 2 \cdot \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$.*

Proof. We fix two DFA $\mathcal{R} = (\Sigma, Q, \delta, q_0, F)$ and $\mathcal{T} = (\Delta, Q', \delta', r_0, F')$ and an optimal positional strategy $g : Q \times Q' \times \Sigma \to E$ for Eve, where E is the set of edges of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$. We then construct from that a transducer $\mathcal{S} = (\Sigma, \Delta, V_{\mathsf{Adam}}, \delta'', v_0, \Omega)$ as follows:

- $V_{\mathsf{Adam}} = Q \times Q'$ is the set of vertices of $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ owned by Adam;
- δ'' is the function that maps any pair $(v, a) \in V_{\mathsf{Adam}} \times \Sigma$, with $v = (q, r)$, to the (unique) pair $(w, v') \in \Delta^* \times V_{\mathsf{Adam}}$, with $v' = (q', r')$, that satisfies
  1. $g(v_a) = (v_a, c, v')$, with $v_a = (\delta(q, a), r, a)$ (note that $v_a \in V_{\mathsf{Eve}}$),
  2. $w \in \mathscr{L}(\mathcal{T}_{r,r'})$, with $\mathsf{dist}(a, w) = c$ (note that since (v_a, c, v') is an edge in $\mathcal{A}_{\mathcal{R},\mathcal{T}}$, there exist such a word w).
- $v_0 = (q_0, r_0)$ is the initial vertex of $\mathcal{A}_{\mathcal{R},\mathcal{T}}$,
- Ω is the function that maps any vertex $v = (q, r) \in V_{\mathsf{Adam}}$ to a word w from the language $\bigcup_{r' \in F'} \mathscr{L}(\mathcal{T}_{r,r'})$ (since $\mathcal{T}$ is pruned, there always exists such a word).

Observe that $\mathcal{S}$ implements a streaming strategy for repairing $\mathscr{L}(\mathcal{R})$ into $\mathscr{L}(\mathcal{T})$ (it basically differs from Eve's strategy only in the use of the final output function Ω, which guarantees that the edited words belong to the target language $\mathscr{L}(\mathcal{T})$). Moreover, by definition, the states of the transducer $\mathcal{S}$ range over the set $V_{\mathsf{Adam}} = Q \times Q'$.

Let us now consider a family of words $\left(u^{(n)}\right)_{n \in \mathbb{N}}$ from the restriction language $\mathscr{L}(\mathcal{R})$ such that

$$\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;=\; \limsup_{n \to \infty} \; \frac{\mathsf{cost}^{\mathsf{aggr}}_{\mathcal{S}}(u^{(n)})}{|u^{(n)}|}.$$

Moreover, let $L_1, \ldots, L_m$ be all the simple cycles of the transition graph of $\mathcal{S}$ and let $\rho^{(n)}$ be the run of $\mathcal{S}$ on the word $u^{(n)}$. We use the simple cycle decomposition Lemma 2 from Section 4 to find a partition of the domain of $\rho^{(n)}$ into (possibly non-convex) subsets $X_0^{(n)}, X_1^{(n)}, \ldots, X_m^{(n)}$ such that

1. $|X_0^{(n)}|$ is uniformly bounded by the number K of states of $\mathcal{S}$,

2. for all $1 \leq i \leq m$, the sub-sequence $\rho^{(n)}|X_i^{(n)}$ is a repetition of the simple cycle L_i of $\mathcal{S}$.

As usual, we denote by $\mathsf{occ}_i^{(n)}$ the number of repetitions of the simple cycle L_i in the sub-sequence $\rho^{(n)}|X_i^{(n)}$, namely, we let $\mathsf{occ}_i^{(n)} = \frac{|X_i^{(n)}|}{|L_i|}$. We have that

$$\limsup_{n\to\infty} \frac{\mathsf{cost}_{\mathcal{S}}^{\mathsf{aggr}}(u^{(n)})}{|u^{(n)}|} \leq \limsup_{n\to\infty} \frac{\sum_{1\leq i\leq m} \mathsf{occ}_i^{(n)} \cdot \mathsf{cost}(L_i) + K \cdot c_{\mathsf{max}}}{\sum_{1\leq i\leq m} \mathsf{occ}_i^{(n)} \cdot |L_i| + K}$$

where $\mathsf{cost}(L_i)$ denotes the sum of the costs of the transitions in the simple cycle L_i and c_{max} denotes the maximum cost of a transition of $\mathcal{S}$. Note that the additive terms $K \cdot c_{\mathsf{max}}$ and K above can be ignored when considering the limit for n tending to infinity. Moreover, it is easy to see that the following inequality holds

$$\limsup_{n\to\infty} \frac{\sum_{1\leq i\leq m} \mathsf{occ}_i^{(n)} \cdot \mathsf{cost}(L_i)}{\sum_{1\leq i\leq m} \mathsf{occ}_i^{(n)} \cdot |L_i|} \leq \max_{1\leq i\leq m} \frac{\mathsf{cost}(L_i)}{|L_i|}.$$

For the sake of brevity, we denote by L some simple cycle among $L_1, \ldots, L_m$ that maximizes the ratio $\frac{\mathsf{cost}(L)}{|L|}$, namely, such that

$$\max_{1\leq i\leq m} \frac{\mathsf{cost}(L_i)}{|L_i|} = \frac{\mathsf{cost}(L)}{|L|}.$$

We now construct a strategy for Adam in the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ by following the simple cycle L. We denote by u_L be the word that forms the input of the simple cycle L and by u_0 any word that makes the DFA $\mathcal{R}$ move from its initial state q_0 to the state that appears in the first/last position of L (recall that the states of $\mathcal{S}$ are pairs of states from $\mathcal{R}$ and $\mathcal{T}$). Clearly, the infinite word $u_0\, u_L^\omega$ induces an infinite run inside the automaton $\mathcal{R}$. Adam's strategy will follow precisely this infinite word, choosing at each round n the edge in $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ that corresponds to the correct transition that consumes the n-th symbol of $u_0\, u_L^\omega$.

Pairing Adam's strategy given above with Eve's strategy, we obtain an infinite 'cyclic' play $\pi = (v_0, 0, v_1)\,(v_1, c_1, v_2)\,\ldots$ over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$. By construction,

the average cost incurred by Eve in following the play π' coincides with $\frac{\mathsf{cost}(L)}{2\cdot|L|}$, namely,

$$\frac{\mathsf{cost}(L)}{2\cdot|L|} \;=\; \nu^{\pi}_{\mathsf{Eve}}$$

Finally, recall that Eve's strategy was assumed to be optimal and hence, by Theorem 3,

$$\nu^{\pi}_{\mathsf{Eve}} \;\leq\; \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}.$$

Summing up, we just proved that

$$\begin{aligned}
\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;&=\; \limsup_{n\to\infty}\; \frac{\mathsf{cost}^{\mathsf{aggr}}_{\mathcal{S}}(u^{(n)})}{|u^{(n)}|}\\
&\leq\; \limsup_{n\to\infty}\; \frac{\sum_{1\leq i\leq m}\; \mathsf{occ}_i^{(n)}\cdot\mathsf{cost}(L_i)}{\sum_{1\leq i\leq m}\; \mathsf{occ}_i^{(n)}\cdot|L_i|}\\
&\leq\; \max_{1\leq i\leq m}\; \frac{\mathsf{cost}(L_i)}{|L_i|}\\
&=\; 2\cdot\nu^{\pi}_{\mathsf{Eve}}\\
&\leq\; 2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}.
\end{aligned}$$

□

We now turn to the proof of Theorem 4.

Proof of Theorem 4. Let $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ be the value of the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$. We know from Lemma 5 that for every streaming repair strategy $\mathcal{S}$ for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$, $2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \;\leq\; \mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$. Since the asymptotic repair cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$ in the streaming case is defined as the infimum of $\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$ over all streaming repair strategies $\mathcal{S}$, we have

$$2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \;\leq\; \mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})).$$

Conversely, we know from Lemma 6 that there is a streaming repair strategy $\mathcal{S}$ for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$ such that $\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;\leq\; 2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ and hence, since $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \leq \mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T}))$, we have

$$\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;\leq\; 2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}.$$

We have just shown that $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;=\; 2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$.

Now, recall that from the results in [10] the value $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ is rational and it can be computed by a deterministic procedure that runs in time $\mathcal{O}(|V|^2 \cdot |E| \cdot c_{\mathsf{max}})$, where V is the set of vertices of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ (hence $|V| \leq |Q| \cdot |Q'| \cdot (|\Sigma| + 1)$), E is the set of edges of $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ (hence $|E| \leq |V|^2$), and c_{max} is the maximum weight of an edge in $\mathcal{A}_{\mathcal{R},\mathcal{T}}$. Since c_{max} never exceeds the number $|Q'|$ of states of the target automaton $\mathcal{T}$, this gives a polynomial time procedure for computing the value $\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}}$ of the mean-payoff game over $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ (and hence the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$).

It remains to show that the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$ is achieved by a single streaming edit strategy for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$ whose states range over $Q \times Q'$. This is proven again using the previous two lemmas. For every streaming edit strategy $\mathcal{S}$ for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$, there is a streaming edit strategy $\mathcal{S}'$ for the same languages whose states range over $Q \times Q'$ (let us call such a strategy *positional*) and such that

$$\mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}'}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) \;\leq\; 2 \cdot \nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \;\leq\; \mathsf{acost}^{\mathsf{aggr}}_{\mathcal{S}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})).$$

Without loss of generality, we can also assume that, at each step, the transducer $\mathcal{S}'$ outputs a word of length at most $|Q'|^2$, that is, for every symbol $a \in \Sigma$ and every state s of $\mathcal{S}'$, if $s \xrightarrow{a/w} s'$ is a transition of $\mathcal{S}'$, then $|w| \leq |Q'|^2$. It is safe to make this assumption because the replacement in any transition $s \xrightarrow{a/w} s'$ of w by w', where w' minimizes $\mathsf{dist}(a, w'')$ among all words w'' that are Myhill-Nerode equivalent to w (i.e., $w \in \mathscr{L}(\mathcal{T}_{r,r'})$ iff $w'' \in \mathscr{L}(\mathcal{T}_{r,r'})$ for all $r, r' \in Q'$) can only result in a discount of the overall aggregate cost incurred by the streaming repair strategy $\mathcal{S}$.

The above arguments show that we can equivalently calculate the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$ as the infimum over all positional streaming repair strategies $\mathcal{S}'$ for $\mathscr{L}(\mathcal{R})$ and $\mathscr{L}(\mathcal{T})$ that, at each step, output words of length at most $|Q'|^2$. Since there are only finitely many such strategies, we conclude that the asymptotic cost $\mathsf{acost}^{\mathsf{aggr}}_{0-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T}))$ is achieved by a single positional streaming edit strategy $\mathcal{S}'$. □

5.3. Asymptotic streaming cost with lookahead

We conclude the section by mentioning some natural generalizations of Theorem 4 related to streaming repair strategies with lookahead.

First of all, we observe that in order to compute the asymptotic cost of an optimal streaming repair strategy with k-lookahead, where $k \in \mathbb{N}$ is a given parameter, it is sufficient to modify the definition of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ in such

a way that Adam plays $(k+1)$-character windows representing substrings of an infinite word. This requires extending the set of vertices of the arena $\mathcal{A}_{\mathcal{R},\mathcal{T}}$ from $(Q \times Q') \cup (Q \times Q' \times \Sigma)$ to $(Q \times Q' \times \Sigma^k) \cup (Q \times Q' \times \Sigma^{k+1})$ and letting the game start from any vertex of the form (p_0, q_0, u_0), where p_0 is the initial state of $\mathcal{R}$, q_0 is the initial state of $\mathcal{T}$, and $u_0 \in \Sigma^k$. We denote by $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$ the new arena and by $\nu_{\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}}$ the value of the mean-payoff game associated with it. Following the same arguments of the proof of Theorem 4, one shows that

$$\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) \;=\; 2 \cdot \nu_{\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}}.$$

We also know that streaming repair strategies with longer lookahead outperform those with shorter lookahead, that is, $\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(R,T)$ is a non-increasing function of $k \in \mathbb{N}$.

Now, it becomes natural to ask whether one can compute the inferior of the asymptotic costs for *all possible streaming strategies with finite (unbounded) lookahead*, and whether this value can be achieved using a fixed amount of lookahead that only depends on the restriction and target languages. For instance, a similar result for quantitative games has been proven in [11]. As we are not able to answer these questions, we address in the following a simpler *threshold problem* for the streaming asymptotic cost:

Theorem 5. *Given two DFA $\mathcal{R}$ and $\mathcal{T}$ and a rational threshold ν, one can decide in double exponential time whether there is $k \in \mathbb{N}$ such that*

$$\mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) \;<\; \nu.$$

The proof of Theorem 5 is based on a reduction of the game-theoretic version of the streaming repair problem (i.e., a mean-payoff game with arbitrary lookahead) to a suitable regular infinite game that is similar to the type of games considered in [11]. Below, we recall the two types of games we are dealing with. The first type of game is a mean-payoff game played by Adam and Eve over an arena of the form $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$, where $\mathcal{R}$ and $\mathcal{T}$ are two DFA and k is a lookahead parameter. We have already shown that the value of this game characterizes the k-lookahead streaming asymptotic cost for $\mathcal{R}$ and $\mathcal{T}$. The second type of game is a qualitative game between two players, Input and Output, who act according to the following rules. Player Input moves first by choosing 2 elements q_0, q_1 from a fixed finite set Q; player Output responds by choosing a single element r_0 from another finite set Q'. At the next round, player Input chooses another element $q_2 \in Q$, and player Output

responds by choosing $r_1 \in Q'$. The game continues in this way by alternating between the two players. The resulting play is an infinite sequence

$$w \;=\; \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{r_2}\ldots.$$

The winner is determined by a given regular ω-language $L \subseteq (Q \times Q')^\omega$, that is, player Input wins the game over L if he can enforce infinite plays $w \in L$. Notice that, in the above game, player Output has a slight advantage in that his moves are one step behind those of player Input. We call this type of game a 1-*lookahead regular game*.

Part of the proof of Theorem 5 requires also a bit of reasoning on *non-streaming* asymptotic costs. The following lemma discloses a technical property that will be used in the sequel.

Lemma 7. *Given a threshold $\nu \in \mathbb{Q}$ and two DFA $\mathcal{R}$ and $\mathcal{T}$ such that $\mathsf{acost}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) < \nu$, there is a number $\mathsf{maxlength}^{\nu}_{\mathcal{R},\mathcal{T}}$ such that, for all words $u \in \mathscr{L}(\mathcal{R})$,*

$$|u| \;>\; \mathsf{maxlength}^{\nu}_{\mathcal{R},\mathcal{T}} \qquad \textit{implies} \qquad \min_{v \in \mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u,v)}{|u|} \;<\; \nu.$$

Proof. We remark that the following proof is not constructive, so it does not provides any effective means of computing the number $\mathsf{maxlength}^{\nu}_{\mathcal{R},\mathcal{T}}$ from $\mathcal{R}$, $\mathcal{T}$, and ν. As a matter of fact, computing such a number would enable us to compute the least amount of lookahead k that satisfies the claim of Theorem 5.

Let $\mathcal{R}$ and $\mathcal{T}$ be two DFA such that $\mathsf{acost}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) < \nu$. It is easy to see that there exist only finitely many words $u \in \mathscr{L}(\mathcal{R})$ that are at normalized distance from $\mathscr{L}(\mathcal{T})$ at least ν – indeed, if this were not the case, then we would have that the limit superior of $\min_{v \in \mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u,v)}{|u|}$ for arbitrarily long words $u \in \mathscr{L}(\mathcal{R})$ would be at least ν, thus contradicting $\mathsf{acost}(\mathscr{L}(\mathcal{R}), \mathscr{L}(\mathcal{T})) < \nu$. This enables the use of max in the following definition:

$$\mathsf{maxlength}^{\nu}_{\mathcal{R},\mathcal{T}} \;=^{\text{def}}\; \max\left\{ |u| \;:\; u \in \mathscr{L}(\mathcal{R}),\; \min_{v \in \mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u,v)}{|u|} \geq \nu \right\}.$$

Thanks to the above definition, we have that for all words $u \in \mathscr{L}(\mathcal{R})$, $|u| > \mathsf{maxlength}^{\nu}_{\mathcal{R},\mathcal{T}}$ implies $\min_{v \in \mathscr{L}(\mathcal{T})} \frac{\mathsf{dist}(u,v)}{|u|} < \nu$. □

The following lemma reduces the threshold problem for the arbitrary-lookahead streaming asymptotic cost to the problem of deciding the winner of a 1-lookahead regular game.

Lemma 8. *Given two DFA $\mathcal{R}$ and $\mathcal{T}$ and a rational number ν, one can compute in double exponential time a parity automaton $\mathcal{A}_{\mathcal{R},\mathcal{T},\nu}$ of size polynomial in $|\mathcal{R}| \cdot |\mathcal{T}|$ that recognizes a regular ω-language L such that:*

1. *if Input wins the 1-lookahead regular game on L, then $2 \cdot \nu_{\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}} \geq \nu$, for all $k \in \mathbb{N}$,*

2. *if Output wins the 1-lookahead regular game on L, then $2 \cdot \nu_{\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}} < \nu$, for all $k \geq k_{\mathsf{max}}$, where k_{max} depends only on $\mathcal{R}$, $\mathcal{T}$, and ν.*

Proof. We first define the language L on the basis of the two DFA $\mathcal{R} = (\Sigma, Q, \delta, q_0, F)$ and $\mathcal{T} = (\Delta, Q', \delta', q'_0, F')$ and the rational number ν. The alphabet of L is the product $Q \times Q'$ of the state spaces of $\mathcal{R}$ and $\mathcal{T}$. L contains all infinite sequences

$$w = \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{r_2}\dots$$

such that

1. q_0 is the initial state of $\mathcal{R}$,

2. for all $i \in \mathbb{N}$, $|\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})| = \infty$, namely, $\mathcal{R}$ consumes arbitrarily long words from state q_i to state q_{i+1};

3. at least one of the following conditions holds:

   a) r_0 is not the initial state of $\mathcal{T}$,

   b) there is $i \in \mathbb{N}$ such that $\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}}) = \varnothing$,

   c) for all but finitely many $i \in \mathbb{N}$, $\mathsf{acost}(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}), \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})) \geq \nu$, namely, there are arbitrarily long words $u_i \in \mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$ having normalized distance from $\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})$ at least ν.

Note that L is a boolean combination of safety, reachability, and liveness properties, and thus it is a regular language. It is also easy to construct a parity automaton $\mathcal{A}_{\mathcal{R},\mathcal{T},\nu}$ that has approximately $\mathcal{O}(|Q \times Q'|)$ states and

that recognizes L. We omit the formal definition of $\mathcal{A}_{\mathcal{R},\mathcal{T},\nu}$ and we only observe that in order to compute $\mathcal{A}_{\mathcal{R},\mathcal{T},\nu}$, one needs to solve a number of threshold problems for the non-streaming asymptotic costs associated with the languages $\mathscr{L}(\mathcal{R}_{q,q'})$ and $\mathscr{L}(\mathcal{T}_{r,r'})$: this can be done in doble exponential time using the procedure described in Section 4.

We also make the following crucial observation. In the definition of the language L, we could have equally rewritten Condition 3.c) as

3. c') *for infinitely many* $i \in \mathbb{N}$, $\mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}), \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})\big) \geq \nu$.

Indeed, it is clear that any infinite play that satisfies Condition 3.c) also satisfies Condition 3.c'). As for the converse implication, consider an infinite play $w = \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{r_2}\ldots$ that satisfies both Condition 2. and Condition 3.c'), but not Condition 3.b). Since w satisfies Condition 3.c'), by the Pigeonhole Principle there exist two pairs $(\tilde{q},\tilde{r}), (\tilde{q}',\tilde{r}') \in Q \times Q'$ that occur consecutively and infinitely often in w and such that $\mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{\tilde{q},\tilde{q}'}), \mathscr{L}(\mathcal{T}_{\tilde{r},\tilde{r}'})\big) \geq \nu$. To derive Condition 3.c) it is sufficient to show that for all pairs $(q,r),(q',r') \in Q \times Q'$ that occur infinitely often in w, $\mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q,q'}), \mathscr{L}(\mathcal{T}_{r,r'})\big) \geq \nu$ holds. Let $(q,r),(q',r')$ be two such pairs. Observe that w contains a substring of the form

$$\tbinom{q}{r}\ldots\tbinom{\tilde{q}}{\tilde{r}}\tbinom{\tilde{q}'}{\tilde{r}'}\ldots\tbinom{q'}{r'}\ldots\tbinom{q}{r}\ldots\tbinom{\tilde{q}}{\tilde{r}}\tbinom{\tilde{q}'}{\tilde{r}'}\ldots\tbinom{q'}{r'}.$$

Notice that (i) $\mathscr{L}(\mathcal{R}_{q,q'}) \supseteq \mathscr{L}(\mathcal{R}_{q,\tilde{q}})\ \mathscr{L}(\mathcal{R}_{\tilde{q},\tilde{q}'})\ \mathscr{L}(\mathcal{R}_{\tilde{q}',q'})$, (ii) both $\mathscr{L}(\mathcal{R}_{q,\tilde{q}})$ and $\mathscr{L}(\mathcal{R}_{\tilde{q}',q'})$ are non-empty (this follows from the fact that w satisfies Condition 2.), and (iii) both $\mathscr{L}(\mathcal{T}_{\tilde{r},r})$ and $\mathscr{L}(\mathcal{T}_{r',\tilde{r}'})$ are non-empty (this follows from the fact that w does not satisfy Condition 3.b)). From these properties we easily derive

$$\begin{array}{rcl}\mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q,q'}), \mathscr{L}(\mathcal{T}_{r,r'})\big) & = & \mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q,q'}), \mathscr{L}(\mathcal{T}_{\tilde{r},\tilde{r}'})\big) \\ & \geq & \mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{\tilde{q},\tilde{q}'}), \mathscr{L}(\mathcal{T}_{\tilde{r},\tilde{r}'})\big) \\ & \geq & \nu.\end{array}$$

This shows that Conditions 3.c) and 3.c') are interchangeable when they are used in the definition of the winning condition L.

Below, we prove a correspondence between the outcomes of the mean-payoff games over the arenas $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$, for all $k \in \mathbb{N}$, and the winners of regular games on L. The correspondence can be described as follows. If player Input wins the 1-lookahead regular game by satisfying Conditions 1., 2., and 3.c) (this is the interesting case), then, in the k-lookahead streaming

repair game, player Adam can choose, from some point onwards, arbitrarily long words $u_i \in \mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$ with normalized distance from $\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})$ at least ν – as the lengths of these words increase, the mean-payoff value of the resulting play gets closer to the average of the normalized distances, and thus eventually stabilizes to a value greater than or equal to ν (this happens no matter how large is the lookahead parameter k). Conversely, if player Output wins the 1-lookahead regular game, then he must be able to enforce a play $w = \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{r_2}\dots$ that violates Conditions 3.a), 3.b), and 3.c'). In particular, from some point onwards, only finitely many words $u_i \in \mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$ have normalized distance from $\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})$ greater than ν – in this case we will show that Eve can use a sufficient amount of lookahead to enforce a play in the streaming repair game with value at most ν.

Let us assume that player Input wins the 1-lookahead regular game on L and let us fix a lookahead parameter k for the streaming repair game. Using Input's winning strategy, we have to derive a strategy for Adam, for each lookahead parameter $k \in \mathbb{N}$, that induces a mean-payoff value greater than ν over the arena $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$. We fix $k \in \mathbb{N}$ and we assume that at the beginning player Input has chooses two consecutive elements $q_0, q_1 \in Q$, with q_0 initial state of $\mathcal{R}$. After Output's response, which we assume to be the initial state r_0 of $\mathcal{T}$, Input chooses a third element $q_2 \in Q$. Thus, the partial play constructed so far is

$$\binom{q_0}{r_0}\binom{q_1}{}\binom{q_2}{}.$$

We know that both languages $\mathscr{L}(\mathcal{R}_{q_0,q_1})$ and $\mathscr{L}(\mathcal{R}_{q_1,q_2})$ contain arbitrarily long words, so we can use them to construct the first moves of Adam in the streaming repair game. Precisely, we fix two words $u_1 \in \mathscr{L}(\mathcal{R}_{q_0,q_1})$ and $u_2 \in \mathscr{L}(\mathcal{R}_{q_1,q_2})$, with $|u_1| \geq 1$ and $|u_2| \geq |u_1|+k$, and we denote by $a_1, \dots, a_{|u_1|+k}$ the first $|u_1| + k$ symbols of the juxtaposition $u_1\ u_2$. We define the first $|u_1|$ moves of Adam's strategy as follows:

- at the 1st round, Adam moves from vertex $(q_0, r_0, a_1 \dots a_k)$ to vertex $(\delta(q_0, a_1), r_0, a_1 \dots a_{k+1})$;
- at the 2nd round, after Eve has moved from $(\delta(q_0, a_1), r_0, a_1 \dots a_{k+1})$ to some vertex $(\delta(q_0, a_1), \delta(r_0, v_1), a_2 \dots a_{k+1})$, for some $v_1 \in \Delta^*$, Adam moves to the next vertex $(\delta(q_0, a_1a_2), \delta(r_0, v_1), a_2 \dots a_{k+2})$;
- in general, at the i-th round, with $1 \leq i \leq |u_1|$, Adam moves from any vertex of the form $(\delta(q_0, a_1 \dots a_{i-1}), \delta(r_0, v_1 \dots v_{i-1}), a_i \dots a_{k+i-1})$ to the vertex $(\delta(q_0, a_1 \dots a_i), \delta(r_0, v_1 \dots v_{i-1}), a_i \dots a_{k+i})$.

Now, recall that $q_1 = \delta(a_1 \dots a_{|u_1|})$ and hence, after the first $|u_1|$ rounds, the streaming repair game must reach a vertex of the form $(q_1, r_1, a_{|u_1|+1} \dots a_{|u_1|+k})$, for some $r_1 = \delta(r_0, v_1 \dots v_{|u_1|})$ and some prefix $a_{|u_1|+1} \dots a_{|u_1|+k}$ of $u_2 \in \mathscr{L}(\mathcal{R}_{q_1,q_2})$. Moreover, the partial cost incurred by Eve so far is

$$\sum_{1 \le i \le |u_1|} \mathsf{dist}(a_i, v_i) \;\ge\; \mathsf{dist}(u_1, v_1 \dots v_{|u_1|}) \;\ge \min_{v \in \mathscr{L}(\mathcal{T}_{r_0,r_1})} \mathsf{dist}(u_1, v).$$

The definition of Adam's strategy for the subsequent rounds follows similar arguments. Specifically, we assume that the current partial play of the regular game is

$$\tbinom{q_0}{r_0} \dots \tbinom{q_i}{r_i} \tbinom{q_{i+1}}{}$$

and that the current position of the streaming repair game is $(q_i, r_i, a_{i,1} \dots a_{i,k})$, where $a_{i,1} \dots a_{i,k}$ is a prefix of some word $u_{i+1} \in \mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$. We then look at the next move induced by Input's strategy, which adds a new state q_{i+2} to the partial play, and we choose another word $u_{i+2} \in \mathscr{L}(\mathcal{R}_{q_{i+1},q_{i+2}})$ of length $|u_{i+2}| \ge |u_{i+1}| + k$. Accordingly, we define the moves of Adam's strategy for the next $|u_{i+1}|$ rounds using the first $|u_{i+1}| + k$ letters in the juxtaposition $u_{i+1}\, u_{i+2}$.

Now, consider a play π of the mean-payoff game that results from Adam's strategy. We know form the previous arguments that the play π can be factorized into an infinite sequence $\pi_0, \pi_1, \pi_2, \dots$ of sub-plays of the form

$$\underbrace{(q_0, r_0, a_1 \dots a_k) \dots}_{\pi_0} \; \underbrace{(q_1, r_1, a_{n_1+1} \dots a_{n_1+k}) \dots}_{\pi_1} \; \underbrace{(q_2, r_2, a_{n_2+1} \dots a_{n_2+k}) \dots}_{\pi_2} \; \dots$$

where $w = \tbinom{q_0}{r_0} \tbinom{q_1}{r_1} \tbinom{q_2}{r_2} \dots$ is a corresponding play in the regular game that follows Input's winning strategy. Moreover, the repair cost incurred in each sub-play π_i is at least

$$\min_{v \in \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})} \mathsf{dist}(u_i, v)$$

where u_i is any word from $\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$ of length $\frac{|\pi_i|}{2}$. Without loss of generality, we can further assume that Adam has chosen the words $u_0, u_1, u_2, \dots$ in such a way that the following additional property is satisfied:

$$\liminf_{i \to \infty} \min_{v \in \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})} \frac{\mathsf{dist}(u_i, v)}{|u_i|} \;=\; \liminf_{i \to \infty} \mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}), \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})\big).$$

Recall that r_0 is the initial state of $\mathcal{T}$ (so Condition 3.a) is violated) and each language $\mathscr{L}(\mathcal{T}_{r_{i-1},r_i})$ is non-empty (so Condition 3.b) is violated). As the play is won by player Input, Condition 3.c) must hold. This implies

$$\liminf_{i\to\infty} \ \mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}), \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})\big) \ \geq \ \nu.$$

Putting all together, we obtain that Adam can enforce a play $\pi = \pi_0\,\pi_1\,\pi_2\,\ldots$ with mean-payoff value greater than or equal to $\frac{\nu}{2}$:

$$\begin{array}{rcl} 2\cdot\nu^{\pi}_{\mathsf{Adam}} & \geq & \liminf\limits_{i\to\infty} \ \min\limits_{v\in\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})} \mathsf{dist}(u_i,v) \\ & = & \liminf\limits_{i\to\infty} \ \mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}), \mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})\big) \\ & \geq & \nu. \end{array}$$

This proves that $2\cdot\nu_{\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}} \ \geq \ 2\cdot\nu^{\pi}_{\mathsf{Adam}} \ \geq \ \nu$.

As for the converse direction, we assume that player Output wins the 1-lookahead regular game on L. We will use Output's winning strategy to construct a strategy for Eve that guarantees a value less than ν in the mean-payoff game over the arena $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$, where k is a *sufficiently large* number. In order to establish what 'sufficiently large' means, we use Lemma 7 to devise the existence of a natural number $\ell > |Q|$ such that, for all states $q, q' \in Q$ and $r, r' \in Q'$ and all words $u \in \mathscr{L}(\mathcal{R}_{q,q'})$ of length at least ℓ,

$$\mathsf{acost}(\mathscr{L}(\mathcal{R}_{q,q'}), \mathscr{L}(\mathcal{T}_{r,r'})) \ < \ \nu \quad \text{implies} \quad \min_{v\in\mathscr{L}(\mathcal{T}_{r,r'})} \frac{\mathsf{dist}(u,v)}{|u|} \ < \ \nu \qquad (\dagger)$$

(technically speaking, the number ℓ can be defined as the maximum among the number of states in $\mathcal{R}$ plus 1 and the numbers $\mathsf{maxlength}^{\nu}_{\mathcal{R}_{q,q'},\mathcal{T}_{r,r'}}$ that are obtained from Lemma 7 when considering all possible DFA $\mathcal{R}_{q,q'}$, $\mathcal{T}_{r,r'}$ such that $\mathsf{acost}(\mathscr{L}(\mathcal{R}_{q,q'}), \mathscr{L}(\mathcal{T}_{r,r'})) < \nu$).
Accordingly, we define $k_{\mathsf{max}} = 2\ell$ and we fix a lookahead parameter $k \geq k_{\mathsf{max}}$ for the rest of this proof.

We now turn to the definition of Eve's strategy for the mean-payoff game over $\mathcal{A}^{k-\mathsf{lookahead}}_{\mathcal{R},\mathcal{T}}$. Roughly, the idea is look at each move of player Output and construct from it ℓ consecutive moves of Eve. Suppose that, at the beginning, Adam chooses an edge of the form

$$(q_0, r_0, a_1 \ldots a_k) \ \xrightarrow{\mathsf{Adam}} \ (\delta(q_0,a_1), r_0, a_1 \ldots a_{k+1}),$$

with q_0 initial state of $\mathcal{R}$, r_0 initial state of $\mathcal{T}$, and $a_1, \ldots, a_{k+1} \in \Sigma$. We define $q_1 = \delta(q_0, a_1 \ldots a_\ell)$ and $q_2 = \delta(q_1, a_{\ell+1} \ldots a_{2\ell})$, and we look at the move

induced by Output's winning strategy in the corresponding partial play of the 1-lookahead regular game:

$$\binom{q_0}{r_0}\binom{q_1}{\ }\binom{q_2}{\ } \quad \xrightarrow{\text{Output}} \quad \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{\ }.$$

We then choose some words $v_1, \ldots, v_\ell \in \Delta^*$ whose juxtaposition $v_1 \ldots v_\ell$ belongs to $\mathscr{L}(\mathcal{T}_{r_0,r_1})$ and for which the aggregate distance $\sum_{1\le j\le \ell} \mathsf{dist}(a_j, v_j)$ is minimized (in particular, this aggregate distance must be equal to the distance of $a_1 \ldots a_\ell$ from $\mathscr{L}(\mathcal{T}_{r_0,r_1})$). Accordingly, we define the first ℓ moves of Eve's strategy as follows:

- at the 1st round, Eve moves from vertex $(\delta(q_0, a_1), r_0, a_1 \ldots a_{k+1})$ to vertex $(\delta(q_0, a_1), \delta(r_0, v_1), a_2 \ldots a_{k+1})$;
- at the 2nd round, after Adam has moved from $(\delta(q_0, a_1), \delta(r_0, v_1), a_2 \ldots a_{k+1})$ to some vertex $(\delta(q_0, a_1a_2), \delta(r_0, v_1), a_2 \ldots a_{k+2})$, Eve moves to the next vertex $(\delta(q_0, a_1a_2), \delta(r_0, v_1v_2), a_3 \ldots a_{k+2})$;
- in general, at the j-th round, with $1 \le j \le l$, Eve moves from any vertex of the form $(\delta(q_0, a_1 \ldots a_j), \delta(r_0, v_1 \ldots v_{j-1}), a_i \ldots a_{j+k})$ to the vertex $(\delta(q_0, a_1 \ldots a_j), \delta(r_0, v_1 \ldots v_j), a_{j+1} \ldots a_{j+k})$.

Note that after the first ℓ rounds, a vertex of the form $(q_1, r_1, a_{\ell+1} \ldots a_{\ell+k})$ The strategy for Eve for the subsequent rounds is defined in a similar way, that is, by translating Adam's moves to corresponding moves of player Input and using Output's response to construct blocks of ℓ consecutive moves of Eve.

Below, we prove that the defined strategy for Eve guarantees a mean-payoff value less than $\frac{\nu}{2}$. Let π be an infinite play of the mean-payoff game induced by Eve's strategy and let us focus on the sequence of vertices x_1, x_2, x_3, ... that are reached at the beginnings of rounds 1, $\ell+1$, $2\ell+1$, ...:

$$\pi \;=\; \underbrace{(q_0, r_0, a_1 \ldots a_k)}_{x_1} \;\ldots\; \underbrace{(q_1, r_1, a_{\ell+1} \ldots a_{\ell+k})}_{x_2} \;\ldots\; \underbrace{(q_2, r_2, a_{2\ell+1} \ldots a_{2\ell+k})}_{x_3} \;\ldots.$$

By construction, at each round $i\ell+1$, with $i \in \mathbb{N}$, the ℓ-character prefix $a_{i\ell+1} \ldots a_{i\ell+k}$ of the word that appears at vertex $x_{i\ell}$ belongs to the language $\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$. Moreover, the 2ℓ consecutive moves that are taken alternatively by Adam and Eve at rounds $i\ell+1, \ldots, (i+1)\ell$ induce a cost

$$c_i \;=\; \sum_{1\le j\le \ell} \mathsf{dist}(a_{i\ell+j}, v_{i\ell+j}) \;=\; \min_{v\in\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})} \mathsf{dist}(a_{i\ell+1} \ldots a_{(i+1)\ell}, v).$$

Now, consider the corresponding play in the 1-lookahead regular game:

$$w \;=\; \binom{q_0}{r_0}\binom{q_1}{r_1}\binom{q_2}{r_2}\ldots.$$

Note that w follows Output's winning strategy, so it cannot belong to the language L. We know that state q_0 is initial in $\mathcal{R}$ and hence w satisfies Condition 1. above. We claim that w satisfies Condition 2. as well. Indeed, for all $i \in \mathbb{N}$, we have that the word $a_{i\ell+1}\ldots a_{(i+1)\ell}$ belongs to the language $\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})$ and has length greater than $|Q|$ (recall that $\ell > |Q|$). As we tacitly assumed that all states in $\mathcal{R}$ can reach non-transient states (cf. Remark 2 in Section 5.2) it follows that $\mathcal{R}_{q_i,q_{i+1}}$ visits some same state twice when parsing $a_{i\ell+1}\ldots a_{(i+1)\ell}$, and hence $|\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}})| = \infty$. As w satisfies both Condition 1. and Condition 2. and $w \notin L$, we know that w must violate Condition 3.c'). Therefore, for all but finitely many $i \in \mathbb{N}$, we have

$$\mathsf{acost}\big(\mathscr{L}(\mathcal{R}_{q_i,q_{i+1}}),\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})\big) \;<\; \nu.$$

We now recall the definition of ℓ and, in particular, the fact that it satisfies Property (†) above, namely, for all states $q,q' \in Q$ and $r,r' \in Q'$ and all words $u \in \mathscr{L}(\mathcal{R}_{q,q'})$ of length at least ℓ, $\mathsf{acost}(\mathscr{L}(\mathcal{R}_{q,q'}),\mathscr{L}(\mathcal{T}_{r,r'})) < \nu$ implies $\min\limits_{v\in\mathscr{L}(\mathcal{T}_{r,r'})} \frac{\mathsf{dist}(u,v)}{|u|} < \nu$. This means that for all but finitely many $i \in \mathbb{N}$,

$$\min_{v\in\mathscr{L}(\mathcal{T}_{r_i,r_{i+1}})} \frac{\mathsf{dist}(a_{i\ell+1}\ldots a_{(i+1)\ell},v)}{\ell} \;<\; \nu.$$

Putting all together, we have that Eve's strategy bounds the mean-payoff value of the game by

$$\nu^{\pi}_{\mathsf{Eve}} \;=\; \limsup_{n\to\infty} \frac{\sum_{i=1}^{n} c_i}{2n\ell} \;<\; \frac{\nu}{2}.$$

and hence $2\cdot\nu_{\mathcal{A}_{\mathcal{R},\mathcal{T}}} \;\leq\; 2\cdot\nu^{\pi}_{\mathsf{Eve}} \;<\; \nu$. □

We conclude with the proof of Theorem 5.

Proof of Theorem 5. Thanks to Lemma 8, the problem of deciding whether

$$\exists k \in \mathbb{N}.\quad \mathsf{acost}^{\mathsf{aggr}}_{k-\mathsf{lookahead}}(\mathscr{L}(\mathcal{R}),\mathscr{L}(\mathcal{T})) \;<\; \nu$$

is immediately reduced to the problem of deciding whether player Output wins the 1-lookahead regular game on L, where L is a suitable regular ω-language computable from $\mathcal{R}$, $\mathcal{T}$, and ν in double exponential time. □

6. Conclusions

We have addressed the problem of computing the asymptotic cost of repairing regular languages in the non-streaming and streaming settings. It is surprising that the asymptotic cost in both settings is rational and computable.

In the non-streaming setting, we proved that the threshold problem for the asymptotic cost is between PSpace and coNExp. We leave as an open problem the unclosed gap between our lower and upper bounds.

In the streaming setting, where a finite lookahead is given, we derive optimal online algorithms for editing one language into another, which are quite distinct from traditional edit distance algorithms based on dynamic programming.

We also began an investigation of the best repair cost for arbitrary finite lookup. We leave open the problem of computing the infimum of the asymptotic costs of all such edit strategies, giving here only a decision procedure for the strict threshold problem.

Acknowledgements. We thank Benjamin Aminoff and the anonymous reviewers of ICALP and TCS for their great help with earlier versions of this manuscript. Benedikt, Puppis, and Riveros are supported by EP/G004021/1, the Engineering and Physical Sciences Research Council UK.

References

[1] M. Benedikt, G. Puppis, C. Riveros, Regular repair of specifications, in: LICS, 2011, pp. 335–344.

[2] I. Simon, Recognizable sets with multiplicities in the tropical semiring, in: MFCS, Vol. 324, 1988, pp. 107–120.

[3] M. Mohri, Finite-state transducers in language and speech processing, J. of Comp. Ling. 23 (2) (1997) 269–311.

[4] R. Wagner, Order-n correction for regular languages, CACM 17 (5) (1974) 265–268.

[5] D. Krob, The equality problem for rational series with multiplicities in the tropical semiring is undecidable, in: ICALP, 1992, pp. 101–112.

[6] T. Colcombet, L. Daviaud, Approximate comparison of distance automata, in: STACS, Vol. 20 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, pp. 574–585.

[7] A. Ehrenfeucht, J. Mycielski, Positional strategies for mean payoff games, J. of Game Theory 8 (1979) 109–113.

[8] J. Matouek, B. Gärtner, Understanding and Using Linear Programming, Springer, 2006.

[9] N. Karmarkar, A new polynomial-time algorithm for linear programming, in: STOC, 1984, pp. 302–311.

[10] U. Zwick, M. Paterson, The complexity of mean payoff games on graphs, Theor. Comput. Sci. 158 (1996) 343–359.

[11] M. Holtmann, L. Kaiser, W. Thomas, Degrees of lookahead in regular infinite games, in: FOSSACS, Vol. 6014 of LNCS, Springer, 2010, pp. 252–266.