A Normalized Edit Distance on Infinite Words

Dana Fisman
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Joshua Grogin
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Gera Weiss
Dept. of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Abstract

We introduce $\overline{\omega}$-NED, an edit distance between infinite words, that is a natural extension of NED, the normalized edit distance between finite words. We show it is a metric on (equivalence classes of) infinite words. We provide a polynomial time algorithm to compute the distance between two ultimately periodic words, and a polynomial time algorithm to compute the distance between two regular ω-languages given by non-deterministic Büchi automata.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Pattern matching; Hardware → Robustness

Keywords and phrases Edit Distance, Infinite Words, Robustness

Digital Object Identifier 10.4230/LIPIcs.CSL.2023.20

Related Version `https://jgrogin.github.io/A_Normalized_Edit_Distance_on_Finite_and_Infinite_Words_thesis.pdf`

Funding This work was supported in part by ISF grants 2714/19 and 2507/21.

1 Introduction

Quantifying distances between words is a field of research that provides tools for measuring semantic differences between sequential objects and sets thereof. In this work, we are interested in quantifying the distances between infinite words and between sets of infinite words. The main motivation that led us to examine this issue is the use of infinite words in formal methods for describing behaviors of reactive systems. In particular, in software verification and other applications, it is common to refer to software systems as state machines and to use automata whose languages are the sets of runs of the systems. Specifically, it is customary to specify requirements by automata representing the sets of allowed/desired runs. We argue that in this context, it is natural to define the distance between two runs as the average amount of "discrepancies" between the two runs and to study algorithmic problems related to those distances. Since we deal with infinite words, a description of a set of words by an automaton usually uses the Büchi acceptance condition, where a word is accepted by an automaton if and only if there is a trajectory in the automaton that reads the word and passes through accepting states infinitely many times. Sets of infinite words that can be described by Büchi automata are called ω-regular languages.

Significance of our contribution. Verification tools, which deal primarily with compliance and non-compliance with requirements defined using temporal logic, provide a yes/no answer but do not quantify the degree of deviation of implementations from the requirements nor the robustness of an implementation. In this work, we present a metric for quantifying distances between infinite words reflecting deviations between runs of implementation and runs specified in the requirements. Specifically, we extend a normalized version of the known

31st EACSL Annual Conference on Computer Science Logic (CSL 2023).
Editors: Bartek Klin and Elaine Pimentel; Article No. 20; pp. 20:1–20:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

edit distance for this purpose and describe algorithms for calculating distances between ultimately periodic words and between ω-regular languages. The distance between two languages L_1 and L_2 is defined as the infimum between any pair of words w_1, w_2 in L_1 and L_2, respectively.

It can be used, for example, to measure the *robustness of an implementation* [1,2,6,16,17]. Consider a system S implementing a specification φ. The system is considered robust if it continues to satisfy the specification, even when errors disrupt its normal behavior. That is, robustness asks what is the minimum number of errors that would render the system to violate the specification. Thus, robustness is taken to be the infimum between any pair of words w, w' in $L(S)$ and $L(\neg\varphi)$, which is exactly the distance between the language of S and the language of $\neg\varphi$.

Relations to other notions of distance between words. Our work is based on the well-known *edit distance* (aka *Levenshtein* distance) [13]. For two finite words $u_1, u_2 \in \Sigma^*$, this distance, denoted $\textsc{ed}(u_1, u_2)$, is defined as the minimum number of edits we need to apply to u_1 to obtain u_2. Here, edit operations are *delete* a letter, *insert* a letter, or *replace* a letter. For instance, $\textsc{ed}(aabcde, abpcg) = 4$ since by deleting the first a, inserting p, replacing d by g and deleting e we get from the first word to the second, and there is no shorter sequence of edit operations transforming the first word to the latter. In the setting of this paper, differences between words model violations of specifications. We, therefore, treat all types of differences equally, i.e., while in some applications, different edit operations may weigh differently, we consider the case of uniform weights for all edit operations.

Since we are interested in infinite words, we consider normalized versions of ED. To see the role of normalization, even for finite terms, note that $\textsc{ed}(a^{98}b^4, a^{100})$ is also four, even though these words are almost identical. Note that the distance remains four even if we replace 100 by 10^6 and 98 by $10^6 - 2$ in which case the words are even more identical. Intuitively, the problem is that we need to count error rates, not just numbers, if we want to compare words of different, even infinite, lengths. To achieve normalization, people have considered dividing ED by the sum, max, or min of the lengths of the words, but these do not satisfy the triangle inequality [5,15]. To bypass this problem, Marzal and Vidal [15] proposed to divide ED by the length of the sequence of operations transforming the first word to the second, *the edit path*, as formally defined in Section 2. Using NED (for *normalized edit distance*) to refer to this function, we get that $\textsc{ned}(aabcde, abpcg) = 4/7$ and $\textsc{ned}(a^{98}b^4, a^{100}) = 4/102$, which better reflects the "average" number of required edit operations, i.e., this captures a notion of "error rates" as needed. In their paper, Marzal and Vidal demonstrated that NED is not necessarily a metric when the weights are not uniform. The question of whether it is a metric when costs are uniform remained unsettled. This led others to propose alternative notions such as the *generalized normalized edit distance* [14] and the *contextual edit distance* [5]. Still, because of its simplicity and based on empirical data that showed that it behaves very close to a metric, it is widely used in applications. In this paper, we rely on a recent work [7] that established that NED is indeed a metric when the weights are uniform (all edit operations cost the same). The paper above also lists properties of NED and the other two normalized edit distances demonstrating that NED is more suitable when considering the edit operations as errors.

We turn now to discuss possibilities for distances between infinite words. The most famous distance function on infinite words is the one on which the Cantor topology is defined, according to which the distance between $w_1, w_2 \in \Sigma^\omega$ decreases exponentially with the length of the longest common prefix [10]. Formally, using CTD to denote this distance function, $\textsc{ctd}(w_1, w_2)$ is zero if $w_1 = w_2$ and otherwise it is $2^{-\min\{i \mid w_1[i] \neq w_2[i]\}}$ where $w[i]$ denotes the ith letter of w starting from 0. The intuition behind CTD and the edit distance functions mentioned above (ED and NED) is very different. We have that $\textsc{ctd}(ab^\omega, a^\omega) = 1/2$ and $\textsc{ctd}(ba^\omega, a^\omega) = 1$ while more edit operations are needed to get from ab^ω to a^ω than from ba^ω (infinitely many operations are required in the former and only one in the latter). Other commonly used distance functions for infinite words are defined using some weight function and some summation function defined on that (see, e.g., [3,4]). This is more similar in spirit to our motivation. Formally, for $w_1, w_2 \in \Sigma^\omega$ we define their weight difference sequence as $\eta(w_1, w_2) = e_1, e_2, \ldots$ where $e_i = 0$ if $w_1[i] = w_2[i]$ and $e_i = W(w_1[i], w_2[i])$ for some weight function $W\colon \Sigma^2 \to \mathbb{R}$ otherwise. In the examples, we assume, for simplicity, that W assigns the number 1 to all pairs of distinct letters. Given an infinite sequence $\eta = e_1, e_2, e_3, \ldots$ with $e_i \in \mathbb{R}$ the common summation functions are defined as follows:

$$\begin{array}{rlrl}
\mathit{Sup}(\eta) & = \sup_{n\in\mathbb{N}} e_n & \mathit{LimSup}(\eta) & = \limsup_{n\to\infty} e_n = \lim_{n\to\infty} \sup_{m\geq n} e_m \\
\mathit{Inf}(\eta) & = \inf_{n\in\mathbb{N}} e_n & \mathit{LimInf}(\eta) & = \liminf_{n\to\infty} e_n = \lim_{n\to\infty} \inf_{m\geq n} e_m \\
\mathit{LimSupAvg}(\eta) & = \limsup_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} e_i & \mathit{Disc}_\lambda(\eta) & = \sum_{n=1}^{\infty} \lambda^n e_n \\
\mathit{LimInfAvg}(\eta) & = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} e_i & &
\end{array}$$

For finite words $\mathit{Sum}, \mathit{Min}, \mathit{Max}, \mathit{Avg}$ and Disc_λ, for $\lambda \in (0,1)$, are also used see, e.g., [6]. Different summation functions are called for in different situations. For instance, *Sup* is used for peak consumption; *LimSupAvg*, *LimInfAvg* for average response time or rate of failures; and Disc_λ when late failures are less important than early ones.

We next demonstrate why *Sup*, *Inf*, *LimSup*, *LimInf*, and Disc_λ do not capture the notion of distance we seek in this paper. Applying *Sup* to $\eta(w_1, w_2)$ would give 1 if $w_1 \neq w_2$ and 0 otherwise. Applying *Inf* to $\eta(w_1, w_2)$ would give 1 only if $w_1[i] \neq w_2[i]$ for every i, i.e., they completely disagree. These two are too coarse to quantify the closeness of implementations to specifications. Applying *LimSup* to $\eta(w_1, w_2)$ would give 1 if there are infinitely many indices i in which $w_1[i] \neq w_2[i]$ and 0 otherwise, and *LimInf* would give 1 if there is i such that $w_1[j] \neq w_2[j]$ for every $j > i$. These are still too coarse for our purpose. The summation Disc_λ disregards what happens ad infinitum, or more precisely, gives later events an exponentially smaller weight making them negligible. Even for words that agree on the suffixes, e.g., $w_1 = a^\omega$, $w_2 = aba^\omega$, $w_3 = aaaba^\omega$ we get that $\mathit{Disc}_{\frac{1}{2}}(\eta(w_1, w_2)) = 1/4$ and $\mathit{Disc}_{\frac{1}{2}}(\eta(w_1, w_3)) = 1/16$ though both w_2 and w_3 have one discrepancy compared to w_1.

The summation functions closest to what we seek are *LimSupAvg* and *LimInfAvg*. Indeed, if we consider the words $w_1 = a^\omega$ and $w_2 = (aaaab)^\omega$, applying the uniform weight function, we obtain the sequence $\eta = (00001)^\omega$ since every fifth letter is different. Thus, $\mathit{LimSupAvg}(\eta) = 1/5$, which is consistent with our intuition on the normalized number of edit operations required to apply to w_1 to obtain w_2. For the words $v_1 = c^{100}a^\omega$ and $v_2 = d^{35}(aaaab)^\omega$ we also get the desired $1/5$ by applying *LimSupAvg* on $\eta(v_1, v_2)$ which is $1^{100}(00001)^\omega$. Indeed, *LimSupAvg* (and *LimInfAvg*) is indifferent to any finite prefix, as we expect the case to be since a finite prefix is always negligible compared to the infinite suffix. Still, *LimSupAvg* and *LimInfAvg* do not always correspond to our intuition of the normalized number of edits required to get from one word to the other. Consider $x_1 = (abc)^\omega$ and $x_2 = (acb)^\omega$. We have that $\eta(x_1, x_2) = (011)^\omega$ so *LimInfAvg* is $2/3$. But if we consider edit operations, we can transform *abcabc* to *acbacb* using an edit path of length eight with two delete operations and two insert operations hence $\textsc{ned}(x_1x_1, x_2x_2)$=4/8=1/2 meaning the actual error rate should be $1/2$ rather than $2/3$. Another issue arises when considering, for example, $y_1 = (abcd)^\omega$ and $y_2 = (bcda)^\omega$. The obtained sequence $\eta(y_1, y_2)$ is $(1)^\omega$ thus *LimSupAvg* and *LimInfAvg* result in 1, meaning the words are farthest apart, while the number of edits required to get from y_1 to y_2 is one, since we can simply drop the first letter of y_1 to get y_2. Since this is one edit out of infinitely many letters we expect the error rate to be 0 in this case.

Desired criteria from an edit distance on infinite words. We want it to be *normalized* in the sense that the distance between two words is in $[0,1]$ and that it reflects the number of edits needed on average to get from one word to the other. In particular, it would be nice if we can find such a metric in which the distance between $(u_1)^\omega$ and $(u_2)^\omega$ would be close to $\textsc{ned}(u_1,u_2)$. We would also like it to return zero for words with a common infinite suffix. Do we want to require that the distance between two words is zero if and only if they have a common suffix? While this makes sense when considering ultimately periodic words, there are more cases where we would like the distance to be zero when we consider arbitrary words. Consider, e.g., $w_1 = a^\omega$ and $w_2 = ba^9ba^{99}ba^{999}b\cdots$. That is, w_2 has a in almost all positions, but b's creep in intervals of powers of 10. We expect the distance between w_1 and w_2 to be 0 since the normalized number of required edits is negligible because the necessity for an edit operation diminishes as the word progresses.

Due to space limitations, some proofs are deferred to the full version; they can also be found in the second author's master thesis [9].

2 Preliminaries

Sequences, sub-sequences, repetitions, projection. We use $[i..j]$ to denote the set $\{i, i+1,\dots,j\}$ for naturals i,j such that $i \leq j$. If $\rho = (r_0, r_1, \dots)$ is a sequence, we use $\rho[i..j]$ for the subsequence $(r_i, r_{i+1},\dots,r_j)$. Similarly, $\rho[i]$ is used to denote r_i, and $\rho[..i]$ (resp. $\rho[i..]$) is used for the prefix (resp. suffix) of ρ ending (resp. starting) at i. If $\rho = (r_0, r_1, \dots r_{l-1})$ is a sequence, we use ρ^k for the k-times repetition of ρ and ρ^ω for the infinite repetition of ρ. Then, if l is the length of ρ, we have that $\rho^\omega[i] = \rho[i \bmod l]$ for every $i \in \mathbb{N}$. Given a tuple $t = \langle a_1, a_2, \dots, a_n\rangle$ we use $\pi_i(t)$ for a_i, namely the projection of t on the i-th coordinate. These notions extend to sets and sequences in the usual manner, thus, e.g., given a sequence $\rho{=}(\langle\sigma_1,\sigma'_1\rangle,\langle\sigma_2,\sigma'_2\rangle\dots)$ we use $\pi_1(\rho)$ for the sequence $(\sigma_1,\sigma_2,\dots)$.

Words, ω-words, ultimately periodic words, rotations. An *alphabet* Σ is a finite non-empty set of symbols. A finite sequence over Σ is a word and an infinite sequence over Σ is an ω-word. We use $|w|$ to denote the length of w. Thus, $|w| = l$ if $w = \sigma_0\sigma_1\dots\sigma_{l-1}$ and $|w| = \omega$ if w is an ω-word. An ω-word w is termed *ultimately periodic* if $w = uv^\omega$ for some $u \in \Sigma^*$ and $v \in \Sigma^+$. An ultimately periodic word $w = u(v)^\omega$ can be finitely represented as the pair (u, v). The set of finite words is denoted Σ^*, the set of infinite words is denoted Σ^ω, their union is denoted Σ^∞. The set of ultimately periodic words over Σ is denoted $\Sigma^{\textsc{up}}$. Let $w \in \Sigma^*$, we use $\mathit{rot}(w)$ for all rotations of w, namely, the set of words uv such that $w = vu$.

Directions and Paths. We consider a set $\mathbb{D} = \{(0,1), (1,0), (1,1)\}$ of three *directions*. The element $(0,1)$ denotes the *south* direction, and is abbreviated as $d_{\textsc{s}}$; the element $(1,0)$ the *east* direction, and is abbreviated as $d_{\textsc{e}}$; and the element $(1,1)$ the *south-east* direction, and is abbreviated as $d_{\textsc{se}}$. A sequence $\rho \in \mathbb{D}^\infty$ is called a *path*.

A South-East Graph, Endpoint. A *south-east graph* is composed of a set $V = [0..n_1]\times[0..n_2]$ of vertices, for some $n_1, n_2 \in \mathbb{N}\cup\{\omega\}$, and a set $E = (E_{\textsc{s}} \cup E_{\textsc{e}} \cup E_{\textsc{se}}) \cap V^2$ of edges, where $E_{\textsc{s}} = \{(\langle i,j\rangle,\langle i,j+1\rangle)\}$, $E_{\textsc{e}} = \{(\langle i,j\rangle,\langle i+1,j\rangle)\}$, $E_{\textsc{se}} = \{(\langle i,j\rangle,\langle i+1,j+1\rangle)\}$. We refer to such a southeast graph as *an* $(n_1\times n_2)$*-south-east graph.* The vertices of the graph can be placed on a $[0..n_1]\times[0..n_2]$ grid. We visualize the top-left position as the $\langle 0,0\rangle$ point. Thus, a path $\rho = (d_0,\dots,d_{l-1})$ starting in $\langle 0,0\rangle$ will end in $\sum_{i=0}^{l-1} d_i$ where summation is coordinate wise. We use $\mathit{endpoint}(\rho)$ to denote the endpoint of path ρ starting at $\langle 0,0\rangle$.

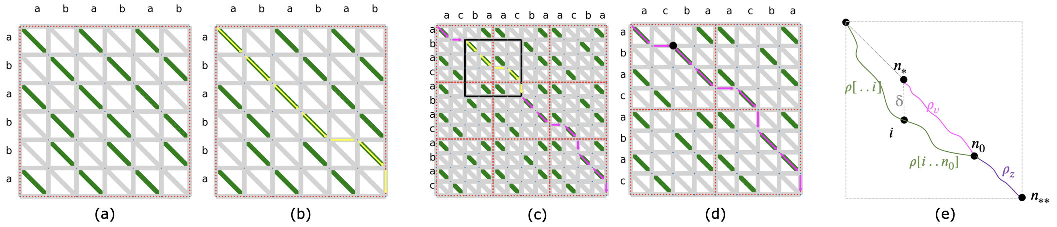

Figure 1 **(a)** the words graph $\mathcal{G} = \mathcal{G}(ababab, ababba)$, gray edges are assigned the weight 1, green edges are assigned the weight 0. **(b)** an edit path ρ in $\mathcal{G}$ (in yellow) where $endpoint(\rho) = (6, 6)$, $wgt(\rho) = 2$, $len(\rho) = 7$, and $cost(\rho) = 2/7$. **(c)** an edit path ρ and points $s = (2, 1)$, $t = (6, 5)$ on $\mathcal{G}((acba)^3, (abac)^3)$ where $\rho[s..t]$ (in yellow) fits a 4-square. **(d)** the path $\rho' = \rho_s \cdot \rho_t$ obtained by removing from ρ the sub-path $\rho[s..t]$ (and replacing it by a black dot) as in the proof of Prop. 9. **(e)** notations for proof of Theorem 30.

Words Graph. Given two words $w_1, w_2 \in \Sigma^\infty$ their corresponding graph $\mathcal{G}(w_1, w_2)$ is the weighted graph (G, θ) where $G = (V, E)$ is the $(|w_1| \times |w_2|)$-south-east graph and the weight of edges $\theta\colon E \to \{0, 1\}$ is defined as follows

$$\theta(e) = \begin{cases} 0 & \text{if } e{=}(v, v') \text{ for } v{=}\langle i, j\rangle, w_1[i]{=}w_2[j], v'{-}v{=}d_{\text{SE}}, \\ 1 & \text{otherwise} \end{cases}$$

That is, the weight of all east edges and south edges is 1 and the weight of a south-east edge starting at $\langle i, j\rangle$ is 0 if the letters $w_1[i]$ and $w_2[j]$ are the same, and 1 otherwise. The word graph $\mathcal{G}(ababab, ababba)$ is given in Figure 1 (a).

Edit Path. A sequence $\rho \in \mathbb{D}^*$ is termed an *edit path* for (w_1, w_2) if $endpoint(\rho) = (|w_1|, |w_2|)$. The *weight* of ρ, denoted $wgt(\rho)$, is the sum of weights of the corresponding edges in $\mathcal{G}(w_1, w_2)$. Formally, if $\rho = (d_0, d_1, \ldots, d_{l-1})$ then the traversed edges are $(e_0, e_1, \ldots, e_{l-1})$ where $e_i = (s_{i-1}, s_i)$, $s_{-1} = \langle 0, 0\rangle$ and $s_i = endpoint(\rho[..i])$ for $0 \leq i < l$. Hence, we can define $wgt(\rho) = \sum_{i=0}^{l-1} \theta(e_i)$. We use the notation $len(\rho)$ to denote the length $|\rho|$ of ρ. The *cost* of ρ, denoted $cost(\rho)$, is defined to be $\frac{wgt(\rho)}{len(\rho)}$. See Figure 1 (b). Intuitively, an edit path for (w_1, w_2) prescribes how to transform w_1 into w_2. In particular, a south direction from $\langle i, j\rangle$ marks that letter $w_1[i]$ is deleted, an east direction from $\langle i, j\rangle$ marks that letter $w_2[j]$ is added, a south-east direction from $\langle i, j\rangle$ marks substitution of $w_1[i]$ by $w_2[j]$, thus it costs nothing if $w_1[i] = w_2[j]$.

Edit Distance and the Normalized Edit Distance. Using the above notations we provide the formal definitions of the *edit distance* and the *normalized edit distance*. The (not normalized) *edit distance* of $u_1, u_2 \in \Sigma^*$, denoted $\text{ED}(u_1, u_2)$, is the minimum weight of an edit path for (u_1, u_2). That is,

$$\text{ED}(u_1, u_2) = \min\{wgt(\rho) \mid \rho \text{ is an edit path for } (u_1, u_2)\}.$$

The *normalized edit distance* of two finite words $u_1, u_2 \in \Sigma^*$, denoted $\text{NED}(u_1, u_2)$, is the minimum cost of an edit path for (u_1, u_2). That is,

$$\text{NED}(w_1, w_2) = \min\{cost(\rho) \mid \rho \text{ is an edit path for } (u_1, u_2)\}.$$

Since we are interested in the NED metric, we say that an edit path ρ for (u_1, u_2) is *optimal* if $\text{NED}(u_1, u_2) = cost(\rho)$.

A Metric Space. A metric space is an ordered pair $(\mathbb{M}, d)$ where $\mathbb{M}$ is a set and $d\colon \mathbb{M}\times\mathbb{M} \to \mathbb{R}$ is a *metric*, i.e., it satisfies the following for all $m_1, m_2, m_3 \in \mathbb{M}$:

1. $d(m_1, m_2) = 0$ iff $m_1 = m_2$;
2. $d(m_1, m_2) = d(m_2, m_1)$;
3. $d(m_1, m_3) \leq d(m_1, m_2) + d(m_2, m_3)$.

The first condition is referred to as *identity of indiscernibles*, the second as *symmetry*, and the third as the *triangle inequality.*

▶ **Theorem 1** ([7]). *(Σ^*, NED) is a metric space.*

3 A Normalized Edit Distance for Infinite Words

Intuitively, it makes sense to define the *normalized edit distance* for two infinite words as the limit of NED of their prefixes. Since the limit may not exist, we define two candidate versions, one using lim inf and one using lim sup as follows.

▶ **Definition 2** ($\overline{\omega}$-NED,$\underline{\omega}$-NED)**.** *Let $w_1, w_2 \in \Sigma^\omega$ be two infinite words. We define two candidate notions of a normalized edit distance, as follows:*

$$\overline{\omega}\text{-}NED(w_1, w_2) \overset{\text{def}}{=\!=} \limsup_{i\to\infty} NED(w_1[..i], w_2[..i])$$

$$\underline{\omega}\text{-}NED(w_1, w_2) \overset{\text{def}}{=\!=} \liminf_{i\to\infty} NED(w_1[..i], w_2[..i])$$

Since for every pair of finite words u_1, u_2, $\text{NED}(u_1, u_2)$ is bounded (between 0 and 1) both $\overline{\omega}\text{-NED}(w_1, w_2)$ and $\underline{\omega}\text{-NED}(w_1, w_2)$ converge for every pair w_1, w_2 of infinite words.

▶ **Example 3.**

a. $\overline{\omega}\text{-NED}(a^\omega,\ (aaaab)^\omega) = 1/5$
$\underline{\omega}\text{-NED}(a^\omega,\ (aaaab)^\omega) = 1/5$

b. $\overline{\omega}\text{-NED}(a^\omega,\ a \cdot b^1 \cdot a \cdot b^2 \cdot a \cdot b^3 \cdot a \cdot b^4 \cdots) = 1$
$\underline{\omega}\text{-NED}(a^\omega,\ a \cdot b^1 \cdot a \cdot b^2 \cdot a \cdot b^3 \cdot a \cdot b^4 \cdots) = 1$

c. $\overline{\omega}\text{-NED}(a^\omega,\ a^1 \cdot b^1 \cdot a^2 \cdot b^2 \cdot a^4 \cdot b^4 \cdots) = 1/2$
$\underline{\omega}\text{-NED}(a^\omega,\ a^1 \cdot b^1 \cdot a^2 \cdot b^2 \cdot a^4 \cdot b^4 \cdots) = 1/3$

Example 3 (c) shows that $\overline{\omega}$-NED and $\underline{\omega}$-NED, in general, may converge to different numbers. Since our motivation is to quantify errors, the worst-case view reflected by $\overline{\omega}$-NED seems more appropriate. Moreover, as we show next, $\overline{\omega}$-NED satisfies the triangle inequality while $\underline{\omega}$-NED does not.

▶ **Proposition 4.** *$\overline{\omega}$-NED satisfies the triangle inequality and $\underline{\omega}$-NED does not.*

Proof. We show that $\overline{\omega}$-NED satisfies the triangle inequality by proving a more general claim stating that given a function $d\colon \Sigma^* \times \Sigma^* {\to} \mathbb{R}_+$ satisfying the triangle inequality then the function $\overline{\omega}\text{-}d\colon \Sigma^\omega \times \Sigma^\omega {\to} \mathbb{R}_+$ where $\overline{\omega}\text{-}d(w_1, w_2)$ is defined by $\limsup_{i\to\infty} d(w_1[..i], w_2[..i])$ satisfies the triangle inequality as well. To see why this holds, let $w_1, w_2, w_3 \in \Sigma^\omega$. We have

$$\begin{aligned}
\overline{\omega}\text{-}d(w_1, w_3) &= \limsup_{i\to\infty} d(w_1[..i], w_3[..i]) \\
&\leq \limsup_{i\to\infty} \left(d(w_1[..i], w_2[..i]) + d(w_2[..i], w_3[..i])\right) \\
&\leq \limsup_{i\to\infty} d(w_1[..i], w_2[..i]) + \limsup_{i\to\infty} d(w_2[..i], w_3[..i]) \\
&= \overline{\omega}\text{-}d(w_1, w_2) + \overline{\omega}\text{-}d(w_2, w_3)
\end{aligned}$$

where the first inequality holds since d satisfies the triangle inequality and the second inequality is a property of sum of lim sup of non-negative sequences.

To see that $\underline{\omega}$-NED does not satisfy the triangle inequality, take $w_1 = a^\omega$, $w_3 = b^\omega$ and $w_2 = u_1 v_1 u_2 v_2 \cdots$, where $u_1 = a$, $v_1 = bb$, $u_2 = aaa$ and $v_{i+1} = b^{2|u_i|}$ for $i \geq 1$ and $u_{i+1} = a^{2|v_i|}$ for $i \geq 2$. Thus $w_2 = ab^2a^3b^6a^{12}b^{24}a^{48}b^{96}\ldots$. Note that after reading u_i the number of a's is twice the number of b's, and two-thirds of the total number of letters (and the number of b's is a third of the total number of letters), and likewise after reading v_i the number of b's is twice the number of a's, and two-thirds of the total number of letters (and the number of a's is a third of the total number of letters). We get that $\underline{\omega}\text{-NED}(w_1, w_2) = 1/3 = \underline{\omega}\text{-NED}(w_2, w_3)$, which contradicts the triangle inequality since $\underline{\omega}\text{-NED}(w_1, w_3) = 1 > 1/3 + 1/3 = \underline{\omega}\text{-NED}(w_1, w_2) + \underline{\omega}\text{-NED}(w_2, w_3)$. ◀

In order for $\overline{\omega}$-NED to be a metric it also needs to satisfy symmetry (which clearly it does) and the condition of identity of indiscernibles. The following example shows that $\overline{\omega}$-NED does not satisfy identity of indiscernibles: $\overline{\omega}\text{-NED}((ab)^\omega, (ba)^\omega) = 0$ though these are non-identical words. However, as per the discussion in the introduction, we do want to allow zero distance between such words. Specifically, we want the metric to be defined on equivalence classes of words so that the distance between two words is zero if and only if they are in the same equivalence class. To define this equivalence relation, let us revisit the examples where a distance of zero is expected. The first examples considered words that have a common suffix. Indeed, in such pairs of words, the number of required operations is finite (can be used to eliminate both prefixes), and thus negligible compared to the length of infinite words. The last example was $w_1 = a^\omega$ and $w_2 = ba^9ba^{99}ba^{999}b\cdots$. In this example, we view the number of edits as negligible since the necessity for edit operations diminishes as the work progresses. In this example, the number of required edits decreases exponentially. Should this be a requirement? What if it decreases quadratically or logarithmically? Observe that we can create words where the number of edits from a^ω decreases as slowly as desired by considering $w = (ab)^{n_1}(aab)^{n_2}(aaab)^{n_3}\cdots$. The larger $n_1, n_2, n_3, \ldots$ are, the slower the number of edits decreases. Still, in all such words, it diminishes over the infinite word and thus we expect the difference from a^ω to be zero.

We therefore define two infinite words w_1, w_2 to be *almost equal*, denoted $w_1 \equiv w_2$, if $\lim_{i\to\infty} \text{NED}(w_1[..i], w_2[..i]) = 0$. Note that words that have a common suffix are almost equal according to this definition, as are the other discussed examples. With respect to the equivalence classes of $\equiv$, identity of indiscernibles holds for $\overline{\omega}$-NED as formally stated in Prop. 5.

▶ **Proposition 5.** *$\overline{\omega}\text{-NED}(w_1, w_2) = 0$ iff $w_1 \equiv w_2$.*

Proof. If $w_1 \equiv w_2$ then $\lim_{i\to\infty} \text{NED}(w_1[..i], w_2[..i]) = 0$. Thus $\limsup_{i\to\infty} \text{NED}(w_1[..i], w_2[..i]) = 0$. Therefore, by definition $\overline{\omega}\text{-NED}(w_1, w_2) = 0$.

If $\overline{\omega}\text{-NED}(w_1, w_2) = 0$ then $\limsup_{i\to\infty} d(w_1[..i], w_2[..i]) = 0$. Since $\text{NED}(v_1, v_2)$ is bounded between 0 and 1 for any $v_1, v_2 \in \Sigma^*$, it follows that $\liminf_{i\to\infty} d(w_1[..i], w_2[..i]) = 0$. Hence $\lim_{i\to\infty} \text{NED}(w_1[..i], w_2[..i]) = 0$ implying $w_1 \equiv w_2$. ◀

Thus $\overline{\omega}$-NED satisfies the three conditions of being a metric on the space $\Sigma^\omega/\equiv$.

▶ **Theorem 6.** *$(\Sigma^\omega/\equiv,\ \overline{\omega}\text{-NED})$ is a metric space.*

We therefore henceforth focus on $\overline{\omega}$-NED.

4 The Case of Ultimately Periodic Words

We turn to discuss *ultimately periodic words.* The interest in ultimately periodic words stems from the fact that (a) they allow a finite representation of an infinite word, so we can ask whether we can compute $\overline{\omega}$-NED for such words; (b) deterministic finite state machines generate ultimately periodic words and; (c) two regular ω-languages are equivalent iff they agree on the set of ultimately periodic words.

We next show that $\overline{\omega}$-NED for ultimately periodic words can be computed using NED on the best rotations of the periodic parts:

▶ **Definition 7** (best rotation)**.** *Let $u_1, u_2 \in \Sigma^+$. Let n be the least common multiple of $|u_1|$ and $|u_2|$. Let $u'_1 = u_1^\omega[..n]$ and $u'_2 = u_2^\omega[..n]$. We say that (v_1, v_2) is a* best rotation *for (u_1, u_2) if $v_1 \in rot(u'_1)$, $v_2 \in rot(u'_2)$ and for every $v'_1 \in rot(u'_1)$ and $v'_2 \in rot(u'_2)$ it holds that $\textsc{ned}(v_1, v_2) \leq \textsc{ned}(v'_1, v'_2)$. If (v_1, v_2) is a best rotation for some (u_1, u_2) we say that (v_1, v_2) is a* best rotation pair*. The* size *of such a best rotation is defined to be n.*

▶ **Theorem 8** ($\overline{\omega}$-NED for ultimately periodic words)**.** *Let $w_1 = z_1 u_1^\omega$ and $w_2 = z_2 u_2^\omega$. Let (v_1, v_2) be a best rotation for (u_1, u_2). Then $\overline{\omega}\text{-}\textsc{ned}(w_1, w_2) = \textsc{ned}(v_1, v_2)$.*

To prove Theorem 8, it is tempting to consider using the South-East graph with wrap-around as a game graph towards a reduction to 1-player MeanPayoff games, but unfortunately, it does not work. The problem is that such a reduction allows finding an optimal path that does not correspond to prefixes of the same length, while $\overline{\omega}$-NED is defined as $\limsup_{i\to\infty}(\textsc{ned}(w_1[..i], w_2[..i])$ thus insists on the same length of prefixes. Consider $w_1 = (aaab)^\omega$ and $w_2 = (aab)^\omega$. The LCM is 12, and $\overline{\omega}\text{-}\textsc{ned}(w_1, w_2) = 4/14$ where transformation of $(aaab)^3$ to $(aab)^4$ is via $aaabaaabaa\_a\_b \mapsto aa\_baa\_baabaab$. However, the reduction returns $1/4$, via the path deleting every first a in a "block" of w_1, essentially returning $\limsup_{i\to\infty}(\textsc{ned}(w_1[..3i], w_2[..4i])$ rather than $\limsup_{i\to\infty}(\textsc{ned}(w_1[..i], w_2[..i])$.

We, therefore, continue with a series of propositions that lead to a proof of Theorem 8. The first proposition concerns the cost of prefixes that are multiples of n (the least common multiple of the two periods).

▶ **Proposition 9** (The cost of prefixes which are multiplications of the best rotation)**.** *Let $u_1, u_2 \in \Sigma^+$ be a best rotation pair of size n. Let ρ be an optimal edit path for (u_1^i, u_2^i) for some $i > 0$. Let ρ_* be an optimal edit path for (u_1, u_2). Then $cost(\rho) = cost(\rho_*)$.*

The proof of Prop. 9 builds on the following proposition, claiming that when looking at ρ on the words graph $\mathcal{G}(u_1^i, u_2^i)$, then some infix of ρ fits an $n \times n$ square, as formally stated in Prop. 10, and illustrated in Figure 1 (c).

▶ **Proposition 10** (Fitting an n-square)**.** *Let $u_1, u_2 \in \Sigma^+$ be a best rotation pair of size n. Let ρ be an optimal edit path for (u_1^i, u_2^i) for some $i > 1$. There exists s and t, $0 \leq s < t < |\rho|$, such that $endpoint(\rho[s..t]) = \langle n, n\rangle$.*

The proof of Prop. 10 uses the intermediate value theorem over how much a $k \times k$ square drifts from the main diagonal. The proof can be found in the full version of the paper.

Now, using Prop. 10, we can show that the cost of an optimal edit path ρ for (u_1^i, u_2^i) is the same as the cost of ρ_*, an optimal edit path for (u_1, u_2).

Proof of Prop. 9. Clearly, since $(\rho_*)^i$ is an edit path for (u_1^i, u_2^i) and ρ is defined to be optimal for (u_1^i, u_2^i) it must hold that $cost(\rho) \leq cost((\rho_*)^i) = cost(\rho_*)$.

The proof that $\mathit{cost}(\rho) \geq \mathit{cost}(\rho_*)$ is by induction on i. For $i = 1$, the path ρ clearly cannot cost less than the path ρ_* which is optimal for this dimension.

Consider $i > 1$. By Prop. 10 there exists $0 < s < n - 1$ and $t > s$ such that $\mathit{endpoint}(\rho[s..t]) = \langle n, n\rangle$. Let $\rho_s = \rho[..s-1]$, $\rho_t = \rho[t+1..]$. Consider the path $\rho' = \rho_s \cdot \rho_t$ obtained by removing the sub-path of ρ from s to t (as illustrated in Figure 1 (c) and (d)). It is an edit path for (u_1^{i-1}, u_2^{i-1}). Thus, by the induction hypothesis, we have that $\mathit{cost}(\rho') \geq \mathit{cost}(\rho_*)$. Since $\mathit{endpoint}(\rho[s..t]) = \langle n, n\rangle$ we also have $\mathit{cost}(\rho[s..t]) \geq \mathit{cost}(\rho_*)$. Note that the cost of ρ can be computed using its sub-paths $\rho[s..t]$ and ρ' which combines ρ_s and ρ_t. Let $l' = \mathit{len}(\rho')$ and $d' = \mathit{wgt}(\rho')$. Similarly, let $l_{st} = \mathit{len}(\rho[s..t])$ and $d_{st} = \mathit{wgt}(\rho[s..t])$. Last, let $l_* = \mathit{len}(\rho_*)$ and $d_* = \mathit{wgt}(\rho_*)$. We get that

$$\mathit{cost}(\rho) = \frac{\mathrm{wgt}(\rho[s..t])+\mathrm{wgt}(\rho')}{\mathrm{len}(\rho[s..t])+\mathrm{len}(\rho')} = \frac{d_{st}+d'}{l_{st}+l'} \geq \min\left\{\frac{d_{st}}{l_{st}}, \frac{d'}{l'}\right\} \geq \min\left\{\frac{d_*}{l_*}, \frac{d_*}{l_*}\right\} = \frac{d_*}{l_*} = \mathit{cost}(\rho_*)$$

where the first inequality holds by Fact 11 (proven in the full version). ◀

▶ **Fact 11.** Let $a_1, \ldots, a_n > 0$, $b_1, \ldots, b_n > 0$. Then $\frac{\sum_{1\leq i\leq n} a_i}{\sum_{1\leq i\leq n} b_i} \geq \min\limits_{1\leq i\leq n} \left\{\frac{a_i}{b_i}\right\}$.

Next, we bound from above and below, the cost of prefixes that are not a multiplication of n.

▶ **Proposition 12** (cost of prefixes which are not multiplications of n)**.** *Let $u_1, u_2 \in \Sigma^+$ be a best rotation pair of size n. Let $m = i \cdot n + j$ for some $i > 0$ and $0 < j < n$ and let ρ be an optimal edit path for $(u_1^\omega[..m], u_2^\omega[..m])$. Let ρ_* be an optimal path for (u_1, u_2), and assume $d_* = \mathit{wgt}(\rho_*)$ and $l_* = \mathit{len}(\rho_*)$. Then*

$$\frac{d_*}{l_*} - \frac{2(n-j)}{i\cdot n+j} \quad\leq\quad \mathit{cost}(\rho) \quad\leq\quad \frac{d_*+\frac{2j}{i}}{l_*+\frac{2j}{i}} \ .$$

Proof. For the left inequality, consider the path $\rho' = \rho \cdot (d_{\mathrm{S}})^{n-j} \cdot (d_{\mathrm{E}})^{n-j}$. That is, the path obtained from ρ by extending it with $(n - j)$ south and $(n - j)$ east steps. Note that $\mathit{endpoint}(\rho') = \langle(i+1)n, (i+1)n\rangle$. It follows from Prop. 9 that the cost of ρ' is at least $\frac{d_*}{l_*}$. Thus we have $\mathit{cost}(\rho') = \frac{\mathrm{wgt}(\rho)+2(n-j)}{\mathrm{len}(\rho)+2(n-j)} \geq \frac{d_*}{l_*}$. From here we get:

$$\mathit{wgt}(\rho)l_* \quad\geq\; d_* \cdot \mathit{len}(\rho) + d_* \cdot 2(n-j) - l_* \cdot 2(n-j) \quad\geq\quad d_* \cdot \mathit{len}(\rho) - l_* \cdot 2(n-j)$$

dividing both sides by $l_* \cdot \mathit{len}(\rho)$ gives us $\frac{\mathrm{wgt}(\rho)}{\mathrm{len}(\rho)} \geq \frac{d_*}{l_*} - \frac{2(n-j)}{\mathrm{len}(\rho)} \geq \frac{d_*}{l_*} - \frac{2(n-j)}{i\cdot n+j}$ where the last inequality follows from the fact that $\mathit{len}(\rho) \geq i \cdot n + j$.

For the right inequality, consider the path $\rho' = (\rho_*)^i \cdot (d_{\mathrm{S}})^j \cdot (d_{\mathrm{E}})^j$. That is, the path obtained from ρ_*^i by extending it with j south and j east steps. Note that $\mathit{endpoint}(\rho') = \langle i\cdot n+j, i\cdot n+j\rangle$. Since ρ is optimal we get: $\mathit{cost}(\rho) \leq \mathit{cost}(\rho') = \frac{\mathrm{wgt}(\rho_*^i)+2j}{\mathrm{len}(\rho_*^i)+2j} = \frac{i\cdot d_*+2j}{i\cdot l_*+2j} = \frac{d_*+2j/i}{l_*+2j/i}$. ◀

We are now ready to prove Theorem 8.

Proof of Theorem 8. Assume that $w_1 = z_1 u_1^\omega$ and $w_2 = z_2 u_2^\omega$. Let n be the gcd of $|u_1|$ and $|u_2|$ and let (v_1, v_2) be a best rotation of (u_1, u_2). Since $z_i u_i^\omega$, u_i^ω and v_i^ω are almost equal, for $i \in \{1, 2\}$, by Theorem 6, $\overline{\omega}$-$\textsc{ned}(w_1, w_2) = \overline{\omega}$-$\textsc{ned}(v_1^\omega, v_2^\omega)$. Let ρ^* be an optimal path for v_1, v_2. Let d_* be its weight and l_* its length. Consider $\textsc{ned}(v_1^\omega[..i], v_2^\omega[..i])$. If i is a multiple of n then by Prop. 9 we get that $\textsc{ned}(v_1^\omega[..i], v_2^\omega[..i]) = d_*/l_*$. If i is not a multiple of n then by Prop. 12 we have that $\textsc{ned}(v_1^\omega[..i], v_2^\omega[..i])$ approaches d_*/l_* as i grows. Thus, $\lim_{i\to\infty} \textsc{ned}(v_1^\omega[..i], v_2^\omega[..i]) = d_*/l_* = \textsc{ned}(v_1, v_2)$. ◀

5 Computing $\overline{\omega}$-NED for Languages of Infinite Words

We define the distance between two languages as the infimum of distances between two words in the respective languages (as is common in metrics when extending the distance between pair of elements in the space to pair of sets in the space). That is, $\text{NED}(L_1, L_2) = \inf_{w_1 \in L_1, w_2 \in L_2} \text{NED}(w_1, w_2)$ and $\overline{\omega}\text{-NED}(L_1, L_2) = \inf_{w_1 \in L_1, w_2 \in L_2} \overline{\omega}\text{-NED}(w_1, w_2)$. In this section we tackle the problem of computing $\overline{\omega}$-NED for regular languages of infinite words. Two questions that should be answered first, are (a) how to compute $\overline{\omega}$-NED for two ultimately periodic words (which we discuss in Subsection 5.1), and (b) how to compute NED for languages of finite words (which we discuss in Subsection 5.2). After discussing these, in Subsection 5.3, we tackle the problem of computing $\overline{\omega}$-NED for regular languages of infinite words.

5.1 Computing $\overline{\omega}$-NED for ultimately periodic words

We summarize first how computation of NED for finite words can be done.

Computing NED for words. Computation of NED for two finite words can be done in PTIME using a dynamic programming algorithm as follows [15]. Let w_1, w_2 be words of lengths m and n, respectively. Any edit path of size k satisfies $\max\{m, n\} \leq k \leq m + n$. Following the definition,

$$D(i, j, k) = \min \left\{ wgt(\rho) \;\middle|\; \rho \text{ is an edit path for } (w_1[..i-1], w_2[..j-1]) \text{ and } len(\rho) = k \right\}$$

we obtain $\text{NED}(w_1, w_2) = \min_{max(m,n) \leq k \leq m+n} \frac{1}{k} D(m, n, k)$ and it can be computed in $O(m \cdot n \cdot \min\{m, n\})$, since for k there are $m + n - \max\{m, n\}$ entries in D.

Computing $\overline{\omega}$-NED for ultimately periodic words. Let $w_1 = z_1 u_1^\omega$ and $w_2 = z_2 u_2^\omega$. Let $v_1 = u_1^\omega[..m]$ and $v_2 = u_2^\omega[..m]$ where m is the least common multiple of $|u_1|$ and $|u_2|$. It follows from Theorem 8 that $\overline{\omega}\text{-NED}(w_1, w_2)$ equals NED of a best rotation (v_1', v_2') of (v_1, v_2). We note that to look for a best rotations it suffices to check only rotations of either v_1 or v_2.

▷ Claim 13. Let $v_1, v_2 \in \Sigma^*$ such that $|v_1| = |v_2|$. There exists a best rotation (v_1', v_2') of (v_1, v_2) where $v_2' = v_2$.

Therefore $\overline{\omega}\text{-NED}(w_1, w_2) = \min_{v_1' \in rot(v_1)} \text{NED}(v_1', v_2)$; and it can be computed in $O(m^4)$. Hence:

▶ **Theorem 14.** *Let $w_1, w_2 \in \Sigma^{UP}$. Then $\overline{\omega}\text{-NED}(w_1, w_2)$ can be computed in PTIME.*

5.2 Computing NED for regular languages

It is known that the infimum of the mean of a weighted graph can be computed in PTIME [6].

▶ **Lemma 15** ([6]). *Let $G = (V, E, \theta \colon E \to \mathbb{Q}_{\geq 0})$ be a weighted graph, and $V_I \subseteq V$, $V_F \subseteq F$ source and target vertices. The infimum of the mean weights of paths from V_I to V_F can be computed in PTIME.*

We can use this result to compute NED of regular languages, by building a weighted graph that corresponds to the product of two NFAs, where instead of allowing only transitions where both NFAs read the same letter, we also allow transitions where they read different letters, and transitions where one reads a letter and the other one does not (instead it reads ε). Transitions where both read a letter correspond to *replace*, where only the first reads a letter to *delete* and where only the second reads a letter to *insert*.

▶ **Definition 16** (Edit Distance Graph of two NFAs). *For $i \in \{1,2\}$ let $\mathcal{N}_i = (\Sigma, Q_i, s_i, \delta_i, F_i)$ be an NFA with no ε-moves. The* edit-distance graph of $\mathcal{N}_1$ and $\mathcal{N}_2$ *is a labeled weighted graph, denoted $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ which is defined as follows $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2) = (V, L, E, \theta)$ where $V = Q_1 \times Q_2$ is the set of vertices; the set of labels is $\Sigma_\varepsilon \times \Sigma_\varepsilon$ where $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$; the set of edges $E \subseteq V \times L \times V$ is given by*

$$E = \left\{ (\langle q_1, q_2\rangle, \langle \sigma_1, \sigma_2\rangle, \langle q_1', q_2'\rangle) \;\middle|\; \begin{array}{l} \textit{either } \sigma_1 \neq \varepsilon \textit{ or } \sigma_2 \neq \varepsilon \\ \textit{and if } \sigma_i \neq \varepsilon \textit{ then } q_i' \in \delta_i(q_i, \sigma_i) \textit{ otherwise } q_i' = q_i \end{array} \right\}$$

and the weight function θ associates a weight with edge $(u, l, u') \in E$ solely based on l as follows: $\theta(u, l, u') = \theta_{\text{ED}}(l)$ where if $\sigma_1 = \sigma_2$ then $\theta_{\text{ED}}(\langle\sigma_1, \sigma_2\rangle) = 0$ otherwise $\theta_{\text{ED}}(\langle\sigma_1, \sigma_2\rangle) = 1$.

Clearly, any path in $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ corresponds to an edit path of the respective words and vice versa.

▷ Claim 17. $\rho_{\text{ED}} = ((u_0, l_1, u_1),\ (u_1, l_2, u_2),\ \ldots, (u_{k-1}, l_k, u_k))$ for $l_i = (\sigma_i, \sigma_i')$ is a path in $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ if and only if $\rho = (d_1, d_2, \ldots, d_k)$ is a path in the words graph $\mathcal{G}(w_1, w_2)$ where $w_1 = \sigma_1\sigma_2 \cdots \sigma_k$, $w_2 = \sigma_1'\sigma_2' \cdots \sigma_k'$ and $d_i = d_{\text{S}}$ if $\sigma_i = \varepsilon$, $d_i = d_{\text{E}}$ if $\sigma_i' = \varepsilon$, and $d_i = d_{\text{SE}}$ otherwise.

Note that in addition, all edges but those corresponding to pairs of identical letters cost 1. Thus, the sum of weights of a path ρ_{ED} in $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ corresponds to $\mathit{wgt}(\rho)$ and the length of ρ_{ED} to $\mathit{len}(\rho)$. Therefore, the infimum of the mean path in $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ corresponds exactly to the desired NED value. Hence, the NED distance between two regular languages given by NFAs $\mathcal{N}_1$ and $\mathcal{N}_2$ can be reduced to computing the infimum of the mean cycle in $\mathcal{G}_{\text{ED}}(\mathcal{N}_1,\mathcal{N}_2)$ from $V_I = \{(s_1, s_2)\}$ to $V_F = F_1 \times F_2$, and following Lemma 15 it can be computed in PTIME.

▶ **Theorem 18.** *The NED distance between two regular languages given by NFAs $\mathcal{N}_1$ and $\mathcal{N}_2$ can be computed in PTIME.*

5.3 Computing $\overline{\omega}$-NED for Regular ω-Languages

The gist of the proof of Lemma 15 is important for the development of the algorithm for infinite words. The idea is that the infimum of the mean path in a weighted graph is obtained either on a simple path, or on a path with a simple cycle, by repeating the cycle over and over. Thus, one can use Karp's dynamic programming algorithm to compute the minimal mean value amongst simple paths and cycles, that works in PTIME [12].

We would like to lift these ideas to compute $\overline{\omega}$-NED between two regular ω-languages, given by non-deterministic Büchi automata, henceforth NBA.[1] To cope with the fact that we work with Büchi automata, we need to insist that the path has a cycle, and when projected on either component, visits an accepting state somewhere along the cycle (this will guarantee that the corresponding runs of both NBAs accept). Unfortunately, this alone does not suffice.

The problem is that if we find such a path with a cycle, which indeed corresponds to traversing infixes of the corresponding words, it might not correspond to infixes of the same length, as required by the definition of $\overline{\omega}$-NED. This is since the edit-distance graph has edges corresponding to deletes and inserts, and such edges process letters only on one of the given NBA, not simultaneously on both. We should consider only ultimately periodic paths,

[1] For lack of space we do not include the standard definition of Büchi automata and refer the unfamiliar reader to [8, Chapter 1.3].

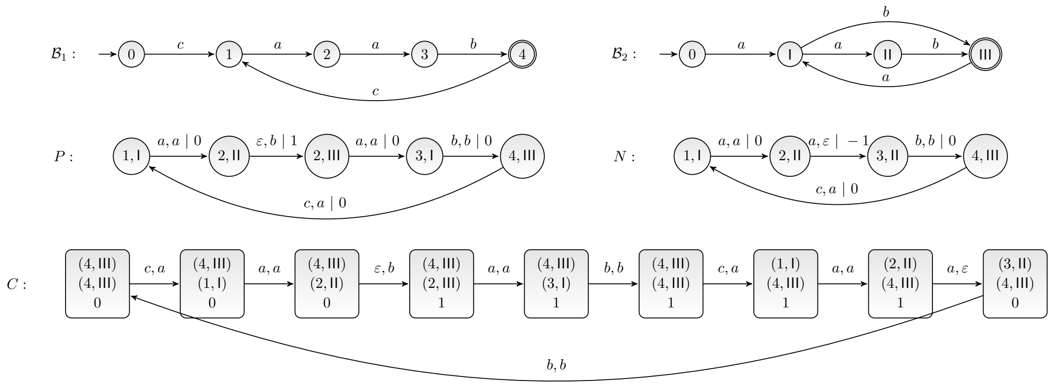

Figure 2 First row: an NBA $\mathcal{B}_1$ for $L_1 = \{caab\}^\omega$, and an NBA $\mathcal{B}_2$ for $L_2 = \{aab, ab\}^\omega$. Second row: a positive cycle P and a negative cycle N in $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$. Recall that edge labels are of the form $\sigma_1, \sigma_2 \mid bal(\sigma_1, \sigma_2)$ (the weight of the edge is not indicated; it can be inferred from the σ_1, σ_2 component of the label). Third row: an optimal cycle C in the balance t-counter graph $\mathcal{G}^t_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$ (achieving the cost $\frac{3}{9}$ via words $(caab)^\omega \in L_1$ and $(aabaabab)^\omega \in L_2$) whose balance index never goes above 1 or below 0 (i.e. here $t = 1$ suffices).

henceforth *lasso paths*, with the same number of delete and insert operations, as these are the paths that correspond to prefixes of the same length. To this aim, we add an additional annotation to edges: the *balance*.

▶ **Definition 19** (Edit Distance Graph of two NBAs). *Let $\mathcal{B}_i = (\Sigma, Q_i, s_i, \delta_i, F_i)$ be an NBA for $i \in \{1, 2\}$. The* edit-distance graph *of $\mathcal{B}_1$ and $\mathcal{B}_2$ is a labeled weighted graph, denoted $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2) = (V, L', E, \theta)$. It is defined similarly to the edit-distance graph for NFAs with one difference: the set of labels L' carries an additional annotation, the* balance. *Formally, $L' = L \times B$ where $L = \Sigma_\varepsilon \times \Sigma_\varepsilon$ is defined as for NFAs, $B = \{-1, 0, 1\}$, and label l is replaced by $(l, \text{bal}(l))$ where $\text{bal}(l) = 1$ if $l \in \{\varepsilon\} \times \Sigma$, $\text{bal}(l) = -1$ if $l \in \Sigma \times \{\varepsilon\}$, and otherwise $\text{bal}(l) = 0$. For a path $\rho = v_0 \xrightarrow{l_1, b_1} v_1 \xrightarrow{l_2, b_2} \cdots \xrightarrow{l_t, b_t} v_t$ we use $\text{bal}(\rho)$ for $\sum_{i=1}^t b_i$.*

Figure 2, first row shows two NBAs $\mathcal{B}_1, \mathcal{B}_2$. It does not depict the entire graph $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$ which has 5×4 states. Instead, the second row, depicts two cycles of $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$: cycle P that is positively balanced, and cycle N that is negatively balanced. We will explain the third row in the sequel.

With this definition, we are looking for the infimum mean lasso path whose balance is zero and contains a cycle that visits both components F_1 and F_2 at least once. Note that in a lasso path, it is the weight of the cycle that matters, since by repeating the cycle as much as desired, the weight of the path until the cycle begins becomes negligible. Observe that the infimum mean balanced lasso path need not be simple, i.e., it can involve two or more cycles. E.g., if we have a cycle with balance -2, another cycle with balance -3 and a cycle with balance 7 that share a vertex, the non-simple cycle repeating the -2 balanced cycle twice, and the other two cycles once is balanced. In this example since the cycles have a shared vertex, there is a cycle corresponding to the concatenation of one cycle twice with the other two cycles. But, even if they do not share a vertex, that would be fine, because, again, by repeating the cycles a large number of times the paths that connect them become negligible (though these parts as well are repeated infinitely often). The next section shows that it suffices to consider lasso paths containing at most two simple cycles – either one (that is zero balanced) or two (one that is negatively balanced and one that is positively balanced).

5.4 Enough to consider two cycles

Consider the graph $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$. Let $\mathcal{C} = \mathcal{C}_= \uplus \mathcal{C}_+ \uplus \mathcal{C}_-$ where $\mathcal{C}_= = \{\alpha \mid \alpha \text{ is a simple cycle and } bal(\alpha) = 0\}$, $\mathcal{C}_- = \{\alpha \mid \alpha \text{ is a simple cycle and } bal(\alpha) < 0\}$, $\mathcal{C}_+ = \{\alpha \mid \alpha \text{ is a simple cycle and } bal(\alpha) > 0\}$. That is, $\mathcal{C}_=$ represents cycles that are zero balanced, $\mathcal{C}_+$ represents cycles that are positively balanced (have more inserts than deletes), and $\mathcal{C}_-$ those that are negatively balanced. We define for a set of cycles $\alpha_1, \ldots, \alpha_n$, their value as follows.

▶ **Definition 20** (Value of a set of cycles). *Given a set of cycles $\{\alpha_1, \ldots, \alpha_n\}$, we use the following notations for $1 \leq i \leq n$: $w_i = \theta(\alpha_i)$, $l_i = |\alpha_i|$, $b_i = bal(\alpha_i)$. We define*

$$val(\alpha_1, \ldots, \alpha_n) = \min \left\{ \frac{\sum_{i=1}^n t_i \cdot w_i}{\sum_{i=1}^n t_i \cdot l_i} \;\middle|\; \sum_{i=1}^n t_i \cdot b_i = 0, \quad t_i \geq 0 \textit{ for } 1 \leq i \leq n, \quad \exists j.\ t_j > 0 \right\}$$

That is, $val(\alpha_1, \ldots, \alpha_n)$ computes the minimum among the NED value of the (imaginary) path obtained by concatenating all cycles, where cycle α_i is repeated t_i times, such that the balance of the constructed path is zero.

When we have one negatively balanced and one positively balanced cycles, their *val* can be easily computed, as follows.

▶ **Observation 21.** *Let $\alpha_- \in \mathcal{C}_-, \alpha_+ \in \mathcal{C}_+$. Assume $w_\bullet = \theta(\alpha_\bullet)$, $b_\bullet = bal(\alpha_\bullet)$, $l_\bullet = len(\alpha_\bullet)$ for $\bullet \in \{-, +\}$. Then $val(\alpha_+, \alpha_-) = \frac{b_+ \cdot w_- \;-\; b_- \cdot w_+}{b_+ \cdot l_- \;-\; b_- \cdot l_+}$.*

The following proposition states that from a set involving no zero balanced cycles, it is possible to choose just one positively balanced cycle and one negatively balanced cycle and the cost of the resulting path would not be worse than the one touring many cycles.

▶ **Proposition 22** (Two cycles suffice). *Let $\alpha_1, \ldots, \alpha_m \in \mathcal{C}_-$, $\alpha_{m+1}, \ldots, \alpha_{m+k} \in \mathcal{C}_+$ for some $m, k \geq 1$. There exists $\alpha_+ \in \mathcal{C}_+$ and $\alpha_- \in \mathcal{C}_-$ with $val(\alpha_1, \ldots, \alpha_{m+k}) = val(\alpha_+, \alpha_-)$.*

The proof makes use of the following fact (proven in the full version of the paper).

▶ **Fact 23.** Given two sequences of positive numbers $\vec{a} = (a_1, \ldots, a_m)$ and $\vec{b} = (b_1, \ldots, b_k)$ satisfying $\sum_{i=1}^m a_i = \sum_{j=1}^k b_j$ there exists a matrix $P \in [0,1]^{m \times k}$ such that $\sum_{i=1}^m p_{ij} = 1$ for every $1 \leq j \leq k$ and $\vec{a} = P \cdot \vec{b}$. In other words, for all $1 \leq i \leq m$ we have $a_i = \sum_{j=1}^k p_{ij} \cdot b_j$.

Proof of Prop. 22. Assume $t_1 \ldots t_{m+k} \in \mathbb{N}$ are an optimal solution for $val(\alpha_1, \ldots, \alpha_{m+k})$. Following Def. 20 the t_is need to satisfy the following constraint:

$$0 \neq \sum_{i=1}^m t_i \cdot (-b_i) = \sum_{j=m+1}^{m+k} t_j \cdot b_j \tag{5.1}$$

By Fact 23 we can write Equation 5.1 as the following m linear combinations, one for each $1 \leq i \leq m$.

$$t_i \cdot (-b_i) = \sum_{j=m+1}^{m+k} \alpha_{ji} \cdot t_j \cdot b_j$$

such that $\alpha_{ji} \in [0,1]$ for all i, j; and $\sum_{i=1}^m \alpha_{ji} = 1$ for every j.

Following this observation and due to the optimal solution we get that $val(\alpha_1, \dots, \alpha_{m+k}) =$

$$\begin{aligned}
&= \frac{\sum\limits_{i=1}^{m+k} t_i \cdot w_i}{\sum\limits_{i=1}^{m+k} t_i \cdot l_i} = \frac{\sum\limits_{j=m+1}^{m+k} \sum\limits_{i=1}^{m} t_j \cdot \alpha_{ji} \cdot (w_j - \frac{b_j}{b_i} \cdot w_i)}{\sum\limits_{j=m+1}^{m+k} \sum\limits_{i=1}^{m} t_j \cdot \alpha_{ji} \cdot (l_j - \frac{b_j}{b_i} \cdot l_i)} \\
&\geq \min_{m+1 \leq j \leq m+k, t_j \neq 0} \left\{ \frac{\sum\limits_{i=1}^{m} t_j \cdot \alpha_{ji} \cdot (w_j - \frac{b_j}{b_i} \cdot w_i)}{\sum\limits_{i=1}^{m} t_j \cdot \alpha_{ji} \cdot (l_j - \frac{b_j}{b_i} \cdot l_i)} \right\} \\
&\geq \min_{m+1 \leq j \leq m+k, t_j \neq 0} \left\{ \min_{1 \leq i \leq m, \alpha_{ji} \neq 0} \left\{ \frac{t_j \cdot \alpha_{ji} \cdot (w_j - \frac{b_j}{b_i} \cdot w_i)}{t_j \cdot \alpha_{ji} \cdot (l_j - \frac{b_j}{b_i} \cdot l_i)} \right\} \right\} \\
&= \min_{\substack{m+1 \leq j \leq m+k \\ 1 \leq i \leq m,\ \alpha_{ji}, t_j \neq 0}} \left\{ \frac{t_j \cdot \alpha_{ji} \cdot (w_j - \frac{b_j}{b_i} \cdot w_i)}{t_j \cdot \alpha_{ji} \cdot (l_j - \frac{b_j}{b_i} \cdot l_i)} \right\} \\
&= \min_{\substack{m+1 \leq j \leq m+k \\ 1 \leq i \leq m,\ \alpha_{ji}, t_j \neq 0}} \left\{ \frac{b_i \cdot w_j - b_j \cdot w_i}{b_i \cdot l_j - b_j \cdot l_i} \right\}
\end{aligned}$$

◀

Henceforth, we use the following notations

$$\mu_{=} = \min_{\alpha \in \mathcal{C}_{=}} \{val(\alpha)\}, \quad \mu_{+-} = \min_{\alpha_+ \in \mathcal{C}_+, \alpha_- \in \mathcal{C}_-} \{val(\alpha_+, \alpha_-)\},$$

$$\mu_* = \min_{\{\alpha_1, \dots, \alpha_n\} \subseteq \mathcal{C}} \{val(\alpha_1, \dots, \alpha_k)\}.$$

With these notations, we can conclude from Prop. 22 that μ_*, the best value achieved for an arbitrary set of cycles, is no better than the best value achieved for one cycle or two.

▶ **Corollary 24.** $\mu_* = \min\{\mu_{=}, \mu_{+-}\}$.

In Subsection 5.5 we show how μ_* can be computed. Then, in Subsection 5.6 we show that there exists words w_1, w_2 achieving μ_* and that no pair of words can achieve a value better than μ_*, i.e. that $\overline{\omega}$-NED(L_1, L_2) is indeed μ_*.

5.5 Computing μ_*

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be complete NBAs for ω-regular languages L_1, L_2. We want to calculate $\overline{\omega}$-NED(L_1, L_2). We create $\mathcal{G}_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$, the edit distance graph of the NBAs (from Def. 19). By Claim 17 a path in $\mathcal{G}_{\text{ED}}$ represents an edit path from a word in L_1 to a word in L_2. For the path to be accepted by both NBAs it needs to visit an accepting state of $\mathcal{B}_i$ infinitely often for $i \in \{1, 2\}$. Thus we are interested in maximal strongly connected components (MSCC) that have at least one state from F_1 and at least one state from F_2. We can hence remove all vertices that do not reach such an MSCC. For simplicity we assume that our graph has one such MSCC (if this is not the case, we apply our algorithm to each such MSCC separately).

We need the following construction to make sure pairs of cycles are balanced and that we evaluate all pairs of cycles:

▶ **Definition 25** (The balance t-counter graph). *Let $\mathcal{G}_{ED}(\mathcal{B}_1, \mathcal{B}_2) = (V, L \times B, E, \theta)$ be the edit distance graph of the NBAs, and let $n = |V|$. Recall that $L = \Sigma_\varepsilon \times \Sigma_\varepsilon$ and $B = \{-1, 0, 1\}$. For threshold $t \in \mathbb{N}$ we define $\mathcal{G}^t_{\text{ED}}(\mathcal{B}_1, \mathcal{B}_2)$ as the labeled weighted graph $(V_t, L_t, E_t, \theta_t)$ where*

$V_t = V \times V \times [-t,t]$; the labeling function L_t omits the balance label from edges, i.e. $L_t = L$; the weight function θ_t associates with edge (v,l,v') for $v,v' \in V_t$ and $l \in L$ the weight $\theta_{\text{ED}}(l)$ (as in Def. 16); and the edges are $E_t = E'_t \cap (V_t \times L_t \times V_t)$ where

$$\begin{array}{rcl} E'_t & = & \{\ (\langle u,v,i\rangle, l, \langle u',v,j\rangle) \mid\ (u,\langle l,b\rangle,u') \in E,\ \ b=j-i\ \ \} \ \ \cup \\ & & \{\ (\langle u,v,i\rangle, l, \langle u,v',j\rangle) \mid\ (v,\langle l,b\rangle,v') \in E,\ \ b=j-i\ \ \} \end{array}$$

Figure 2, third row shows a balanced cycle C of $\mathcal{G}^t_{\text{ED}}(\mathcal{B}_1,\mathcal{B}_2)$ for $t \geq 1$.

We continue by claiming that μ_* can be computed on the balanced graph, $\mathcal{G}^t_{\text{ED}}$ for some t. Later, we will bound the size of the required t. Let μ_t be the minimal simple cycle in $\mathcal{G}^t_{\text{ED}}$.

▶ **Lemma 26.** *There exist $t \in \mathbb{N}$ such that $\mu_t = \mu_*$.*

Proof. Recall that by Cor. 24 $\mu_* = \min\{\mu_=, \mu_{+-}\}$. For the first direction ($\geq$) let $C = ((u_1,v_1,b_1),(u_2,v_2,b_2),\ldots,(u_r,v_r,b_r),(u_1,v_1,b_1))$ be a simple cycle in $\mathcal{G}_t$ such that $\theta(C)/|C| = \mu_t$. By projecting the path C onto each coordinate we get two closed walks in $\mathcal{G}^t_{\text{ED}}$: $C_1 = (u_1,\ldots,u'_r,u_1)$ and $C_2 = (v_1,\ldots,v''_r,v_1)$. Since we have two closed walks we can decompose them into simple cycles $\alpha_1,\ldots\alpha_{m+k+l}$. We can partition them into sets according to their balances. Let $\{\alpha_1,\ldots,\alpha_m\} \subseteq \mathcal{C}_-$, $\{\alpha_{m+1},\ldots,\ \alpha_{m+k}\} \subseteq \mathcal{C}_+$, and $\{\alpha_{m+k+1},\ldots,\alpha_{m+k+l}\} \subseteq \mathcal{C}_=$, and let $t_1,\ldots,t_{m+k+l}$ denote their respective number of repetitions in the path C. The claim now follows from Cor. 24.

For the second direction ($\leq$), we show that we can find paths corresponding to $\mu_=$ and μ_{+-} of $\mathcal{G}_{\text{ED}}$ in $\mathcal{G}^t_{\text{ED}}$.

Case of $\mu_=$. Let $c_=$ be a zero balanced simple cycle in $\mathcal{G}_{\text{ED}}$ achieving $\mu_=$. Assume

$$c_= \quad = \quad v_1 \xrightarrow{l_1,b_1} v_2 \xrightarrow{l_2,b_2} \cdots \xrightarrow{l_{k-1},b_{k-1}} v_k \xrightarrow{l_k,b_k} v_1$$

where $\sum_{i=1}^{k} b_i = 0$ and $\frac{\theta(c_=)}{|c_=|} = \mu_=$. We observe that for big enough t and for every $r \in [-n,n]$ the following cycle c^r_t is in $\mathcal{G}^t_{\text{ED}}$ and it achieves the same value.

$$c^r_t = (v_1,v_1,r) \xrightarrow{l_1} (v_2,v_1,r{+}b_1) \xrightarrow{l_2} \cdots \xrightarrow{l_{k-1}} (v_k,v_1,r{+}b_k) \xrightarrow{l_k} (v_1,v_1,r)$$

Since $c_=$ is reachable from the initial state and the balance of a simple path is never larger than the length of the path there exists an $r \in [-n,n]$ such that (v_1,v_1,r) is reachable. Therefore, taking $t > n+n$ suffices.

Case of μ_{+-}. Now, let c_- and c_+ be a negatively and positively balanced simple cycles, resp., in $\mathcal{G}_{\text{ED}}$ achieving μ_{+-}. Assume

$$c_- = u_1 \xrightarrow{l_1,b_1} u_2 \xrightarrow{l_2,b_2} \cdots \xrightarrow{l_{m-1},b_{m-1}} u_m \xrightarrow{l_m,b_m} u_1$$

and

$$c_+ = v_1 \xrightarrow{l'_1,b'_1} v_2 \xrightarrow{l'_2,b'_2} \cdots \xrightarrow{l'_{k-1},b'_{k-1}} v_k \xrightarrow{l'_k,b'_k} v_1$$

where $b_- = \sum_{i=1}^{m} b_i < 0$, $b_+ = \sum_{j=1}^{k} b'_j > 0$, $\frac{\theta(c_-)}{|c_-|} = \mu_-$ and $\frac{\theta(c_+)}{|c_+|} = \mu_+$.

In $\mathcal{G}^t_{\text{ED}}$ for big enough t we can find a cycle c^t_r corresponding to repeating b_+ times the cycle c_-, and then repeating b_- times the cycle c_+. It will have the following form

$$\begin{array}{llll}
(u_1, v_1, r_{1,1}), & (u_2, v_1, r_{1,2}), & \ldots, & (u_m, v_1, r_{1,m}), \\
(u_1, v_1, r_{2,1}), & (u_2, v_1, r_{2,2}), & \ldots, & (u_m, v_1, r_{2,m}), \\
 & & \ldots & \\
(u_1, v_1, r_{b_+,1}), & (u_2, v_1, r_{b_+,2}), & \ldots, & (u_m, v_1, r_{b_+,m}), \\
(u_1, v_1, r'_{1,1}), & (u_1, v_2, r'_{1,2}), & \ldots, & (u_1, v_k, r'_{1,k}), \\
(u_1, v_1, r'_{2,1}), & (u_1, v_2, r'_{2,2}), & \ldots, & (u_1, v_k, r'_{2,k}), \\
 & & \ldots & \\
(u_1, v_1, r'_{b_+,1}), & (u_1, v_2, r'_{b_+,2}), & \ldots, & (u_1, v_k, r'_{b_-,k}),
\end{array}$$

where if $r_{1,1} = r$ then $r_{i,j} = r + i(b_-) + \sum_{k=1}^{j} b_i$ and $r'_{i,j} = r + (b_+)(b_-) + i(b_+) + \sum_{k=1}^{j} b'_i$. Hence $r'_{b_-,k} = r + (b_+)(b_-) + (b_-)(b_+) = r$ and c^t_r is a balanced cycle in $\mathcal{G}^t_{\text{ED}}$. Since c_- is reachable from the initial state and the balance of a path is never larger than the length of the path there exists $r \in [-n, n]$ such that (u_1, v_1, r) is reachable. Since both $-b_-, b_+ < n$ we have that $t > n + n^2$ suffices. ◀

The next proposition shows that we can bound t better than $n + n^2$, more precisely, that taking $t = n$ suffices. The idea of the proof is that in $\mathcal{G}^t_{\text{ED}}$ it is possible to traverse part of the negative cycle, then move to traverse part of the positive cycle and continue traversing the negative cycle from where we left off. Alternating between portions of the cycle, we can ensure the balance is never more than n or less than $-n$.

▶ **Proposition 27.** $\mu_n = \mu_*$.

Proof. We use the same idea as in the previous proof, except instead of touring the first cycle and then the second cycle, we alternate between them. Consider the cycles c_+ and c_- repeated indefinitely. Then since the accumulated balance is unbounded and *discretely-continuous* in the sense that it changes in jumps of $\{-1, 0, 1\}$, it follows from the discrete version of the intermediate value theorem [11] that we will eventually encounter all values. We can thus choose the indices $0 = i_0 < i_1 < i_2 \ldots$ on the path for which the accumulated balance reaches exactly $\lceil \frac{n}{2} \rceil \cdot j$ for increasing j's: $\mathit{bal}(c^\omega_+[0..i_j]) = \lceil \frac{n}{2} \rceil \cdot j$ for all $j > 0$. This means that $\mathit{bal}(c^\omega_+[i_{j-1} \,..\, i_j - 1]) = \lceil \frac{n}{2} \rceil$. Similarly there exist indices $i'_1, i'_2 \ldots$ such that $\mathit{bal}(c^\omega_-[i'_{j-1} \,..\, i'_j - 1]) = -\lceil \frac{n}{2} \rceil$ for all $j > 0$. We can therefore alternate between the cycles by progressing in the positive cycle until we reach $\min\{i_j, -b_-|c_+|\}$ then progress on the negative cycle until we reach $\min\{i'_j, b_+|c_-|\}$ and so on. We can keep alternating between them until we reach a balance of 0 (after $-b_-$ of the positive cycle and b_+ of the negative cycle). Since both b_+ and $-b_-$ are at most n we have that at any given point our total balance will be in the range of $[-n, n]$. Therefore $t = n$ suffices. ◀

This gives us that computing μ_* is polynomial in the size of the given automata. Specifically, for $t = n$, the graph $\mathcal{G}^t_{\text{ED}}$ is $2n + 1$ times the size of $\mathcal{G}_{\text{ED}}$ which is polynomial in the size of the automata. The overall computation of μ_* is in PTIME since (i) computing minimal cycles can be done in PTIME (ii) $\mu_* = \mu_n$ and (iii) μ_n is the minimal mean cycle in $\mathcal{G}^n_{\text{ED}}$.

5.6 Proving $\overline{\omega}$-NED$(L_1, L_2) = \mu_*$

Next, we would like to show that $\overline{\omega}$-NED$(L_1, L_2) = \mu_*$. Prop. 28 shows that for every ultimately periodic words $w_1 \in L_1$ and $w_2 \in L_2$ we have $\overline{\omega}$-NED$(w_1, w_2) \geq \mu_*$. Theorem 30 shows that this is the case also for arbitrary words $w_1 \in L_1$ and $w_2 \in L_2$.

On the other hand, we show that μ_* can be achieved by respective words $w_1 \in L_1$ and $w_2 \in L_2$. Prop. 29 shows that we can get arbitrarily close to μ_* in the sense that for every $\varepsilon > 0$ we can find ultimately periodic words w_1, w_2 such that $|\overline{\omega}\text{-NED}(w_1, w_2) - \mu_*| < \varepsilon$.

▶ **Proposition 28** (Lower-bound). *For $w_1 \in L_1 \cap \Sigma^{UP}$, $w_2 \in L_2 \cap \Sigma^{UP}$, $\overline{\omega}$-NED$(w_1, w_2) \geq \mu_*$.*

Proof. Let (v_1, v_2) be a best rotation pair for the periods of w_1, w_2 and let $\rho \in \mathbb{D}^*$ be an optimal edit path for (v_1, v_2). We show that there exists a cycle c in $\mathcal{G}^t_{\text{ED}}$ that is accepting in both NBAs and satisfies $\theta(c)/|c| = \text{NED}(v_1, v_2)$.

For $i \in \{1, 2\}$ let $\beta_i \cdot \gamma_i^\omega$ be an accepting lasso run of $\mathcal{B}_i$ on w_i where γ_i reads the same multiple m of v_i. Then ρ^m is a best edit path for v_1^m and v_2^m. To simplify the notations, we use v_i and ρ instead of v_i^m and ρ^m, i.e., assume w.l.o.g. $|\gamma_i| = |v_i| = k$. Assume $\beta_1 = q_1 q_2 \ldots q_{\ell_1}$, $\beta_2 = q'_1 q'_2 \ldots q'_{\ell_2}$, $\gamma_1 = p_1 p_2 \ldots p_k$, $\gamma_2 = p'_1 p'_2 \ldots p'_k$. Consider the paths $\beta = (q_1, q'_1), (q_2, q'_1), \ldots, (q_{\ell_1}, q'_1)$, $(q_{\ell_1}, q'_2), (q_{\ell_1}, q'_3) \ldots (q_{\ell_1}, q'_{\ell_2})$ and $\gamma = (p_{i_1}, p'_{j_1}), \ldots, (p_{i_k}, p'_{j_k})$ where $i_1 = 1$, $j_1 = 1$ and

$$i_r = \begin{cases} i_{r-1} & \text{if } \rho[r] = d_{\text{E}} \text{ or } i_r = |\gamma_1| \\ i_{r-1}+1 & \text{otherwise} \end{cases} \qquad j_r = \begin{cases} j_{r-1} & \text{if } \rho[r] = d_{\text{S}} \text{ or } j_r = |\gamma_2| \\ j_{r-1}+1 & \text{otherwise} \end{cases}$$

Since $|\gamma_1| = |\gamma_2|$ we have that $|\rho| = |\gamma|$ and γ is a reachable cycle with value: $\theta(\gamma)/|\gamma| = \mathit{wgt}(\rho)/\mathit{len}(\rho) = \mathit{cost}(\rho) \geq \mu_*$. ◀

▶ **Proposition 29** (Upper-bound). *For all $\varepsilon > 0$ there exists $w_1 \in L_1 \cap \Sigma^{UP}, w_2 \in L_2 \cap \Sigma^{UP}$ such that $|\overline{\omega}$-NED$(w_1, w_2) - \mu_*| < \varepsilon$.*

Proof. We show there are valid edit paths corresponding to both $\mu_=$ and μ_{+-}. Let $\varepsilon > 0$.

Case of $\mu_=$: Let $c_=$ be a simple cycle in $\mathcal{G}_{\text{ED}}$ with $\mathit{val}(c_=) = \mu_=$ and $\mathit{bal}(c_=) = 0$. By our assumption on the MSCC, there exists a cycle c traversing an accepting state from both F_1 and F_2. W.l.o.g. it intersects $c_=$. We claim that we can assume $\mathit{bal}(c) = 0$. Assume w.l.o.g. $\mathit{bal}(c) = b > 0$. We can find a negatively balanced cycle c_- that starts at the intersection of c and $c_=$ by tracing only edges corresponding to $\mathcal{B}_1$ and staying put on a fixed state of $\mathcal{B}_2$ (for details see Lemma 36 in the full version). Then $c' = (c)^{(b-)}(c_-)^{(b)}$ is a zero balanced cycle in $\mathcal{G}_{\text{ED}}$ visiting both F_1 and F_2.

Let ρ_u be a path from the initial state to $c_=$. Let ρ_v be the cycle $((c_=)^{n_\varepsilon} \cdot c)$. Consider $\rho = \rho_u \cdot (\rho_v)^\omega$. Recall that a path in $\mathcal{G}_{\text{ED}}$ is an element of $(V \times L' \times V)^\infty$ where $V = Q_1 \times Q_2$ and $L' = \Sigma_\varepsilon \times \Sigma_\varepsilon \times \{-1, 0, 1\}$. Given a path ρ in $\mathcal{G}_{\text{ED}}$, we use $\mathit{labels}(\rho)$ for $\pi_2(\rho) \in (L')^\infty$. For $i \in \{1, 2\}$, let $u_i = \pi_i(\mathit{labels}(\rho_u))$, $v_i = \pi_i(\mathit{labels}(\rho_v))$ and $w_i = u_i(v_i)^\omega$. Then $w_i \in L_i$ since the corresponding runs of $\mathcal{B}_i$ visit an accepting state in F_i infinitely often.

We turn to show that $\overline{\omega}\text{-NED}(w_1, w_2) - \mu_= < \varepsilon$. Note that since ρ_v is balanced $|v_1| = |v_2|$. Following Theorem 8, $\overline{\omega}\text{-NED}(w_1, w_2) \leq \text{NED}(v_1, v_2)$. Thus, for a large enough n_ε:

$$\overline{\omega}\text{-NED}(w_1, w_2) - \mu_= \;\leq\; \text{NED}(v_1, v_2) - \mu_= \;=\; \frac{\theta(\rho_v)}{|\rho_v|} - \mu_= \;=\; \frac{n_\varepsilon \theta(c_=) + \theta(c)}{n_\varepsilon |c_=| + |c|} - \frac{\theta(c_=)}{|c_=|} \;\leq\; \frac{\theta(c)}{n_\varepsilon |c_=|} < \varepsilon$$

Case of μ_{+-}: In a similar fashion we have the simple cycles c_+, c_- $\mathcal{G}_{\text{ED}}$ with $\mathit{val}(c_+) = \mu_+$, $\mathit{val}(c_-) = \mu_-$ and $\mathit{bal}(c_+) = b_+ > 0, \mathit{bal}(c_-) = b_- < 0$. Since c_+, c_- are in the same SCC, we have a cycle $c = p_1 p_2$ where p_1 is from the first state of c_+ to the first state of c_- and p_2 comes back to the first state in c_+ while containing accepting states from both F_1 and F_2. We can assume that $\mathit{bal}(c) = b$ is 0 since if this is not the case we can extract from c a positive and a negative cycle (see Lemma 36 in the full version), and achieve a balance of zero by repeating each number of times that corresponds to the balance of the other.

Since c_+ is reachable from the initial state we have a path ρ_u from the initial state to c_+. Let ρ_v be the cycle $(c_+)^{-b_-\cdot n_\varepsilon} \cdot p_1 \cdot (c_-)^{b_+\cdot n_\varepsilon} \cdot p_2$. For $i \in \{1,2\}$, let $u_i = \pi_i(\mathit{labels}(\rho_u))$, $v_i = \pi_i(\mathit{labels}(\rho_v))$ and $w_i = u_i(v_i)^\omega$. Then $w_i \in L_i$ since the corresponding runs of $\mathcal{B}_i$ visit an accepting state in F_i infinitely often. Thus, for large enough n_ε,

$$\begin{aligned}
\overline{\omega}\text{-}\textsc{ned}(w_1, w_2) - \mu_{+-} &= \overline{\omega}\text{-}\textsc{ned}(v_1^\omega, v_2^\omega) - \mu_{+-} \\
&\leq \frac{\theta(\rho_v)}{|\rho_v|} - \mu_{+-} \\
&= \frac{n_\varepsilon\cdot(-b_-\cdot\theta(c_+)+b_+\cdot\theta(c_-))+\theta(c)}{n_\varepsilon\cdot(-b_-\cdot|c_+|+b_+\cdot|c_-|)+|c|} - \frac{-b_-\cdot|c_+|+b_+\cdot|c_-|}{-b_-\cdot|c_+|)+b_+\cdot|c_-|} \\
&\leq \frac{\theta(c)}{n_\varepsilon\cdot(-b_-\cdot|c_+|+b_+\cdot|c_-|)} < \varepsilon.
\end{aligned}$$

◀

▶ **Theorem 30** (Ultimately periodic words suffice). $\overline{\omega}\text{-}\mathit{NED}(L_1, L_2) = \overline{\omega}\text{-}\mathit{NED}(L_1 \cap \Sigma^{UP}, L_2 \cap \Sigma^{UP})$

Proof. Clearly $\overline{\omega}\text{-}\textsc{ned}(L_1, L_2) \leq \overline{\omega}\text{-}\textsc{ned}(L_1 \cap \Sigma^{\text{UP}}, L_2 \cap \Sigma^{\text{UP}})$. For the other direction, it suffices to show that for all $\varepsilon > 0$ and any $w_1 \in L_1, w_2 \in L_2$ there exists $w_1' \in L_1 \cap \Sigma^{\text{UP}}, w_2' \in L_2 \cap \Sigma^{\text{UP}}$ such that $|\overline{\omega}\text{-}\textsc{ned}(w_1', w_2') - d| < \varepsilon$ where $d = \overline{\omega}\text{-}\textsc{ned}(w_1, w_2)$. Let r_1, r_2 be accepting runs of w_1, w_2 in their respective NBAs, $\mathcal{B}_1$ and $\mathcal{B}_2$. Let q_1 and q_2 be such that $r_1[i] = q_1$ and $r_2[i] = q_2$ for infinitely many is. Such a pair exists by the pigeonhole principle. Let n_* be the first index such that $r_1[n_*] = q_1$ and $r_2[n_*] = q_2$.

Since the number of states in both automata is finite, there is a $C \in O(|\mathcal{G}_{\text{ED}}|^2)$ such that if there is a balanced path between some pair of states in $\mathcal{G}_{\text{ED}}$ then there is one of length at most C. Let $n_0 > \max\{n_*,\ 2C/\varepsilon\}$ be such that $|\textsc{ned}(w_1[..n_0], w_2[..n_0]) - d| < \varepsilon/2$. Let ρ be the corresponding edit path. Let $n_{**} \geq n_0$ be the first index after n_0 such that $r_1[n_{**}] = q_1$ and $r_2[n_{**}] = q_2$ and accepting states of both NBAs have been visited along the way. Let $u_i = w_i[..n_*]$, $v_i = w_i[n_*{+}1..n_0]$. Let ρ_z be a path from $(r_1[n_0], r_2[n_0])$ to $(r_1[n_{**}], r_2[n_{**}])$ and z_i the words obtained by projecting ρ_z on the NBAs $\mathcal{B}_i$ for $i \in \{1,2\}$. Note that the length of ρ_z is bounded by C and thus also its weight. Let ρ_v be an optimal edit path for v_1, v_2. Let i be the smallest integer such that both coordinates of $\mathit{endpoint}(\rho[..i])$ are greater or equal to n_* (see Figure 1 (e)). Assume w.l.o.g. that $\mathit{endpoint}(\rho[..i]) = \langle n_*, n_*{+}\delta\rangle$ where $\delta \geq 0$. Because $\langle n_*, n_*\rangle$ is on the diagonal of $\mathcal{G}(v_1z_1, v_2z_2)$, the number of south edges in ρ is at least δ, thus $\mathit{wgt}(\rho[..i]) \geq \delta$. Also $\mathit{wgt}(\rho_v) \leq \delta + \mathit{wgt}(\rho[i{+}1..n_0])$ and a similar inequality holds for the respective lengths. Further $n_* + \delta \geq \mathit{len}(\rho[..i]) \geq \mathit{wgt}(\rho[i..])$. Thus

$$\begin{aligned}
\overline{\omega}\text{-}\textsc{ned}(u_1(v_1z_1)^\omega, u_2(v_2z_2)^\omega) - d \quad &\leq \quad \textsc{ned}(v_1z_1, v_2z_2) - d \\
&\leq \quad \frac{\mathit{wgt}(\rho_v) + \mathit{wgt}(\rho_z)}{\mathit{len}(\rho_v) + \mathit{len}(\rho_z)} - d \\
&\leq \quad \frac{\delta + \mathit{wgt}(\rho[i{+}1..n_0]) + \mathit{wgt}(\rho_z)}{\delta + \mathit{len}(\rho[i{+}1..n_0]) + \mathit{len}(\rho_z)} - d \\
&\leq \quad \frac{n_* + \delta + \mathit{wgt}(\rho[i{+}1..n_0]) + \mathit{wgt}(\rho_z)}{n_* + \delta + \mathit{len}(\rho[i{+}1..n_0]) + \mathit{len}(\rho_z)} - d \\
&\leq \quad \frac{\mathit{wgt}(\rho[..i]) + \mathit{wgt}(\rho[i{+}1..n_0]) + \mathit{wgt}(\rho_z)}{\mathit{len}(\rho[..i]) + \mathit{len}(\rho[i{+}1..n_0]) + \mathit{len}(\rho_z)} - d \\
&\leq \quad \frac{\mathit{wgt}(\rho) + \mathit{wgt}(\rho_z)}{\mathit{len}(\rho) + \mathit{len}(\rho_z)} - d \\
&\leq \quad \frac{\mathit{wgt}(\rho) + C}{\mathit{len}(\rho)} - d\frac{\mathit{wgt}(\rho)}{\mathit{len}(\rho)} - d + \frac{C}{n_0} \leq \varepsilon
\end{aligned}$$

The first inequality follows from Theorem 8. We use inequality because the concerned periods may not be the best rotation. The second follows since $\rho \cdot \rho_z$ is an edit path for (v_1z_1, v_2z_2). For the third inequality, recall that (n_*, n_*) is on the diagonal of $\mathcal{G}(v_1z_1, v_2z_2,)$

and Because the number of south edges in ρ is at least δ, $wgt(\rho[..i]) \geq \delta$. The third inequality follows since the optimal path from (n_*, n_*) to (n_{**}, n_{**}) is better than going δ steps to the south and then following ρ to (n_0, n_0) and then following ρ_z. The rest follows by applying arithmetic on our assumptions and noticing that if $\hat{\rho}$ is an edit path for w_1, w_2 then $cost(\hat{\rho}) \geq cost(\rho)$ by Fact 11. ◀

▶ **Corollary 31.** $\overline{\omega}$-NED$(L_1, L_2) = \mu_*$

▶ Remark 32 (Distance on Muller and Parity automata). In the full version we show that this construction can be extended to work with Muller and Parity automata.

6 Conclusion and Future Work

We have shown that a natural extension of the normalized edit distance (NED) from words to infinite words is a metric and explained its advantages over other measures of distance for infinite words. We have shown that it can be computed in PTIME, when the words are ultimately periodic. We have further shown that the distance between two regular ω-languages given by non-deterministic Büchi automata can be computed in PTIME.

While our choice of ignoring finite prefixes has been justified, in some cases one would like to distinguish for instance $z_1 = a^\omega$ and $z_2 = bbba^\omega$ and $z_3 = b^{100}a^\omega$ and require that the distance between z_1 and z_2 be smaller than the distance between z_1 and z_3. In particular, if the words have a common suffix (as is the case for z_1, z_2, z_3), we would like a measure that reflects the normalized number of edit operations required on the prefix. This can potentially be done by relaxing the requirement for a metric, and working with generalized metric spaces. But it also requires a formal definition of when the prefix ends. Challenges in achieving this and several suggestions are discussed in the 2nd author's master thesis [9].

References

1 Roderick Bloem, Krishnendu Chatterjee, Karin Greimel, Thomas A. Henzinger, Georg Hofferek, Barbara Jobstmann, Bettina Könighofer, and Robert Könighofer. Synthesizing robust systems. *Acta Informatica*, 51(3-4):193–220, 2014. doi:10.1007/s00236-013-0191-5.

2 Roderick Bloem, Krishnendu Chatterjee, Karin Greimel, Thomas A. Henzinger, and Barbara Jobstmann. Specification-centered robustness. In *Industrial Embedded Systems (SIES), 2011 6th IEEE International Symposium on, SIES 2011. Vasteras, Sweden, June 15-17, 2011*, pages 176–185, 2011. doi:10.1109/SIES.2011.5953660.

3 Pavol Cerný, Thomas A. Henzinger, and Arjun Radhakrishna. Simulation distances. *Theor. Comput. Sci.*, 413(1):21–35, 2012.

4 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4):23:1–23:38, 2010. doi:10.1145/1805950.1805953.

5 Colin de la Higuera and Luisa Micó. A contextual normalised edit distance. In *Proceedings of the 24th International Conference on Data Engineering Workshops, ICDE 2008, April 7-12, 2008, Cancún, Mexico*, pages 354–361. IEEE Computer Society, 2008.

6 Emmanuel Filiot, Nicolas Mazzocchi, Jean-François Raskin, Sriram Sankaranarayanan, and Ashutosh Trivedi. Weighted transducers for robustness verification. In *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, pages 17:1–17:21, 2020.

7 Dana Fisman, Joshua Grogin, Oded Margalit, and Gera Weiss. The normalized edit distance with uniform operation costs is a metric. In *33rd Annual Symposium on Combinatorial Pattern Matching (CPM)*, 2022. To appear (meantime available on arxiv). arXiv:2201.06115.

8 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002. doi:10.1007/3-540-36387-4.

9 Joshua Grogin. A normalized edit distance on finite and infinite words, Master Thesis, Ben-Gurion University of the Negev, March 2022. URL: https://jgrogin.github.io/A_Normalized_Edit_Distance_on_Finite_and_Infinite_Words_thesis.pdf.

10 H. J. Hoogeboom and G. Rozenberg. *Infinitary languages: Basic theory and applications to concurrent systems*, pages 266–342. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986. doi:10.1007/BFb0027043.

11 Richard Johnsonbaugh. A Discrete Intermediate Value Theorem. https://www.maa.org/sites/default/files/0746834259610.di020780.02p0372v.pdf, 1998. The College Mathematical Journal.

12 Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discret. Math.*, 23(3):309–311, 1978. doi:10.1016/0012-365X(78)90011-0.

13 Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

14 Yujian Li and Bi Liu. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, 2007.

15 Andrés Marzal and Enrique Vidal. Computation of normalized edit distance and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(9):926–932, 1993.

16 Daniel Neider, Alexander Weinert, and Martin Zimmermann. Robust, expressive, and quantitative linear temporal logics: Pick any two for free. In *Proceedings Tenth International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2019, Bordeaux, France, 2-3rd September 2019*, pages 1–16, 2019. doi:10.4204/EPTCS.305.1.

17 Paulo Tabuada and Daniel Neider. Robust linear temporal logic. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 – September 1, 2016, Marseille, France*, pages 10:1–10:21, 2016. doi:10.4230/LIPIcs.CSL.2016.10.