Asymptotic Hausdorff and Language Similarity

Dana Fisman
Institute for the Theory of Computing, Stein Faculty of Computer and Information Science, Ben Gurion University, Beer Sheva, Israel

Gal Meirom
Institute for the Theory of Computing, Stein Faculty of Computer and Information Science, Ben Gurion University, Beer Sheva, Israel

Abstract

We introduce the *Asymptotic Hausdorff* lifting, denoted $\mathbb{AH}_d$, a general method for lifting an element-level metric d to a (pseudo-) metric on sets, that captures asymptotic similarity in infinite domains equipped with a notion of size. The construction is designed to be insensitive to finite deviations and to avoid the limitations of classical Hausdorff-based approaches, which are often overly sensitive to outliers and fail to reflect asymptotic behavior.

Formal languages provide a central motivating instance of this framework, where elements are words and sets are languages. When applied to normalized edit distances, the Asymptotic Hausdorff lifting yields metric-valued distances between languages that reflect asymptotic edit behavior while preserving metric structure. We study the equivalence classes of regular languages induced by $\mathbb{AH}_d$ for normalized edit distances d, and characterize their asymptotic essence. Focusing in particular on the normalized edit distance of Marzal and Vidal, ned, we investigate the computation of $\mathbb{AH}_{\mathsf{ned}}$ for regular languages and for bounded context-free languages.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Design and analysis of algorithms

Keywords and phrases Automata theory, formal Languages, Metric Spaces, Language similarity, Edit Distance, asymptotic Analysis

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.178

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version *Full Version*: `https://arxiv.org/abs/2605.09668`

Funding *Gal Meirom*: Supported by ISF grant 2507/21 and Frankel Center for Computer Science, BGU.

Acknowledgements We thank Dror Fried, Guy Ofek, Omer Shachar and Gera Weiss for helpful comments on an early draft of this paper.

1 Introduction

Various applications in formal methods call for a notion of similarity between languages. Such a need arises, for example, in applications of program repair, robustness quantification, and grammatical inference. In all these settings, similarity between two languages X and Y is grounded in a notion of edit operations required to transform words $x \in X$ into words $y \in Y$. More generally, edit distance itself has been extensively studied in areas such as error-correcting codes, parsing theory, speech recognition, and molecular biology, highlighting its broad relevance.

In the context of robustness, Filliot et al. [9] study the computation of $\inf_{x\in X} \inf_{y\in Y} \mathsf{d}(x, y)$, where $\mathsf{d}(x, y)$ is a cost function for editing x into y, implemented by a given weighted transducer. Similarity functions of this form are also studied by Mohri [19], Samanta et al. [23], and Henzinger et al. [14]. The latter also considers the dual quantity $\sup_{x\in X} \sup_{y\in Y} \mathsf{d}(x, y)$.


In repair applications, Benedikt et al. [2, 4] consider the standard edit operations of insertion, substitution, and deletion (with uniform costs) required to transform a string in X into a string in Y. In [2], they ask whether one can transform any word in X into a word in Y using a bounded number of edits. For example, if $X = a^*c^*$ and $Y = a^*bc^*$, then at most one edit operation (inserting a b) is required to transform any string in X into a string in Y. In contrast, if $X = a^*$ and $Y = (ab)^*$, then there is no bound on the number of edits required to transform a word in X into a word in Y. They note that requiring a uniform bound on the number of edits is a strong restriction, and in subsequent work [4] they therefore study the *percentage* of letters that need to be edited. In the latter example, the expected value is $\frac{1}{2}$, since words in X are of the form a^n and such a word requires $\frac{n}{2}$ edit operations to transform into a closest word in Y.

In general, one expects such similarity notions to induce a metric or a pseudo-metric. Some applications, such as repair, are inherently asymmetric; in such cases, one may only require adherence to the triangle inequality $\mathcal{D}(X, Z) \leq \mathcal{D}(X, Y) + \mathcal{D}(Y, Z)$. The triangle inequality is nevertheless central, as it ensures alignment with user intuitions, enables compositional reasoning (inferring distances between X and Z from distances between X, Y and Y, Z), and is computationally beneficial in optimization and learning tasks by enabling pruning, efficient indexing, and incremental updates.

It turns out that none of the similarity notions used in the works mentioned above constitute a metric. In this work, we ask whether it is possible to obtain a metric or a pseudo-metric between languages when the underlying similarity between words is based on the rationale of [4] and captures the *percentage* of edit operations required to transform words in X to words in Y in the limit. Since we are interested in percentages, we restrict attention to functions returning values in $[0, 1]$. That is, we seek a function $\mathcal{D} : \mathcal{P}(\Sigma^*) \times \mathcal{P}(\Sigma^*) \to [0, 1]$ that is a metric (or pseudo-metric) and captures the intuitions underlying [4].

We next present several examples that formalize the desired intuitions and requirements emerging from [4].

▶ **Requirement 1** (Percentage). *As discussed above, one may expect $\mathcal{D}(a^*, (ab)^*)$ to be $\frac{1}{2}$. Rather than insisting on a specific numerical value, we require a monotonicity property:*

$$\mathcal{D}(a^*, (a^j b)^*) < \mathcal{D}(a^*, (a^i b)^*) \quad \text{for all } j > i.$$

Intuitively, this reflects that as the percentage of symbols requiring editing increases, so does the induced distance.

▶ **Requirement 2** (Outlier-insensitive, finite-subset indifferent). *Consider repairing $X = b^*$ to $Y = a^*$. Since any word $x \in X$ is of the form b^n and such a word requires at least n edits (i.e. one edit per character), we expect $\mathcal{D}(b^*, a^*) = 1$. Consider now repairing $X' = a^* \cup b$ to Y. The percentage of edit operations for words of the form a^n is zero, while the percentage for the word b is 1. Consequently, the supremum of the edit-percentage from words in X' to Y is 1. This is undesired as it gives the impression that $a^* \cup b$ is as farthest as possible from a^*, while we expect a value reflecting they are quite similar. Similarly, we expect repairing $a^* \cup b \cup bb$ to $(acc)^*$ to return $\frac{2}{3}$ and not 1, since for all but finitely many words, the percentage is $\frac{2}{3}$. That is, we expect $\mathcal{D}$ to be outlier-insensitive, which we formulate as follows: $\mathcal{D}(X, Y) = \mathcal{D}(X \cup F, Y)$ for all infinite languages X, Y and finite languages F. Note that this requirement necessarily violates identity of indiscernibles on the full powerset, as it entails $\mathcal{D}(X \cup F, X) = \mathcal{D}(X, X) = 0$. Thus, we seek for a pseudo-metric rather than a metric.*[1]

[1] The *identity of indiscernibles* prescribes that $d(x, y) = 0$ iff $x = y$ and a *pseudo-metric* relaxes this condition to require only $d(x, x) = 0$.

▶ **Requirement 3** (Bounded-edits insensitivity)**.** *Consider now repairing $X'' = a^*ba^*$ to $Y = a^*$. In contrast to the previous example,* every *word $x \in X''$ requires at least one edit operation to reach a closest word in Y (in fact, exactly one). However, to capture the percentage nature, we note that the percentage of the number of edits required to transform a^nba^ℓ to $a^{n+\ell}$ diminishes as n or ℓ grows. Note that this is true also if we consider repairing $a^*b^ka^*$ to a^*. While now every word requires k edits, still as n or ℓ grows to infinity, the k edits required are negligible compared to the length of the word. Thus, the following formal requirement emerges from [4]: If there exists a bound k such that every $x \in X$ can be transformed into a word in Y with at most k edits and vice versa, then $\mathcal{D}(X, Y) = 0$.*

▶ Remark 1.1. One may argue that it is desirable to distinguish the language a^* from $a^* \cup b$ or from a^*ba^*. Indeed, there exist language similarity measures that make such distinctions, in particular [1]. In applications where such sensitivity is required, one may combine such a measure with the distance developed here (for example, via a product construction). In this work, however, we deliberately impose invariance under finite subsets of words and a finite number of edits, as these capture the asymptotic notion of similarity that we and [4] aim to capture.

To summarize, we seek a language similarity function with a bounded codomain, specifically $\mathcal{D} : \mathcal{P}(\Sigma^*) \times \mathcal{P}(\Sigma^*) \to [0, 1]$, that is a **pseudo-metric** and satisfies the following **requirements**:

1. *Percentage nature*: $\mathcal{D}(a^*, (a^jb)^*) < \mathcal{D}(a^*, (a^ib)^*)$ for all $j > i$.
2. *Outlier insensitivity*: $\mathcal{D}(X, Y) = \mathcal{D}(X \cup F, Y)$ for all infinite X, Y and finite F.
3. *Bounded-edits insensitivity*: If there exists a bound k such that every $x \in X$ can be transformed into a word in Y with at most k edits and vice versa, then $\mathcal{D}(X, Y) = 0$.

Since languages are sets of words, and we seek a similarity notion induced by edit distance between words, a natural question is whether there exists a general method to lift a metric d on a universe M to a metric $\mathcal{D}$ on subsets of M that meets these requirements. A classical lifting scheme is given by the Hausdorff distance. Let us first recall the common way to define a distance between an element x and a set Y. This measure, $\mathbf{d} : M \times \mathcal{P}(M) \to \mathbb{R}_{\geq 0}$ is defined as $\mathbf{d}(x, Y) = \inf_{y \in Y} \mathsf{d}(x, y)$, i.e. it measures the distance of x to the closest element y in Y.[2] Next, the directional (asymmetric) distance from set X to set Y is defined as $\mathbb{H}^{\triangleright}_{\mathsf{d}}(X, Y) = \sup_{x \in X} \mathbf{d}(x, Y)$, namely it is the distance of the farthest element in X to Y. Finally, given $\mathsf{d} : M \times M \to \mathbb{R}_{\geq 0}$, the Hausdorff distance with respect to d is the function $\mathbb{H}_{\mathsf{d}} : \mathcal{P}(M) \times \mathcal{P}(M) \to \mathbb{R}_{\geq 0}$ defined by $\mathbb{H}_{\mathsf{d}}(X, Y) = \max\{\mathbb{H}^{\triangleright}_{\mathsf{d}}(X, Y), \mathbb{H}^{\triangleright}_{\mathsf{d}}(Y, X)\}$

i.e. it takes the maximum of the directional distance from X to Y and in the other direction.

While the Hausdorff lifting yields a metric and can be used to lift edit distances between words to distances between languages, applying it to the standard Levenshtein edit distance ed [17] fails to produce a similarity measure with a percentage nature, and the resulting values are unbounded. For example, $\mathbb{H}_{\mathsf{ed}}(a^*, a^* \cup bb) = 2$ and $\mathbb{H}_{\mathsf{ed}}(a^*, (ab)^*) = \infty$.

Replacing ed by a normalized edit distance such as ned [18], ged [26], or ced [8] restores the percentage nature, but does not address a more fundamental limitation: the Hausdorff distance is inherently sensitive to outliers.[3] Indeed, while $\mathbb{H}_{\mathsf{ned}}(a^*, (ab)^*) = \frac{1}{2}$, we have $\mathbb{H}_{\mathsf{ned}}(a^*, a^* \cup bb) = 1$. This shortcoming is not specific to edit distance, but arises from the supremum-based definition of the Hausdorff construction itself.

[2] Its use in the context of formal languages goes back to [25] where d is the Levenshtein Edit Distance [17].

[3] Formal definitions of ed, ned, ged, and ced appear in Subsection 4.1. At a high level, ed denotes the minimum number of edit operations, whereas ned, ged, and ced correspond to different normalized variants.

The Hausdorff distance is well suited to lifting distances on finite universes or universes where elements are conceived as having the same size. In contrast, it is size-oblivious and ill suited to infinite universes whose elements admit an unbounded notion of size, as is the case in formal languages, where languages of interest are necessarily infinite and contain words of unbounded length. Similar limitations of Hausdorff-type constructions have been observed in prior work on language similarity measures, leading to the proposal of various alternative notions of language similarity; we review these in Section 2.

Our main contribution is a new lifting scheme $\mathbb{AH}_{\mathsf{d}} : \mathscr{P}(M) \times \mathscr{P}(M) \to \mathbb{R}_{\geq 0}$, which we term *Asymptotic Hausdorff*. This construction lifts an element-level (pseudo) metric d to a set-level pseudo-metric while remaining insensitive to finite outliers and respecting unbounded growth in element size. Unlike the classical Hausdorff distance, it is specifically suited to infinite universes equipped with a natural size notion and yields metric-valued distances whose behavior is asymptotically aligned with the underlying element-level distance.

The lifting is defined for element-level metrics satisfying a property we call the *asymptotic separation property*. We show that normalized edit distances such as ned, ged, and ced satisfy this property, and that the resulting language distance meets all of our stated requirements. We further show that this property is satisfied by other common distance functions such as the Euclidean distance, the L_p metric, and in fact every norm-induced metric. This generality suggests applications beyond the formal-languages setting, particularly in domains that employ finite representations or generators of infinitely many objects, equipped with a natural notion of size.

One such application arises in the study of graph spanner constructions. A spanner is a subgraph of a given weighted graph and thus shares its vertex set with the input. Spanner constructions are thus naturally viewed as set of objects indexed by graph size. Any metric dist comparing spanners over the same graph – for example, based on stretch, distortion, or sparsity – can be lifted via $\mathbb{AH}_{\mathsf{dist}}$ to obtain an asymptotic comparison between spanner constructions, focusing on large-scale behavior while abstracting away finite-size effects.

Another example comes from procedural texture generators in computer graphics. Such generators are commonly modeled as functions $f : \mathbb{R}^k \to [0, 1]$, for $k \in \{2, 3\}$, producing a continuous scalar field that is mapped to concrete values such as colors, materials, or block types. Well-known instances include Perlin noise, simplex noise, Worley noise, and their fractal variants. Similarity between generators is typically assessed by comparing the finite structures they induce over bounded spatial regions (often called *patches*), independently of spatial location. By equipping such patches with an appropriate distance measure that captures their similarity, and using their spatial extent as the size parameter, the lifting $\mathbb{AH}$ naturally captures asymptotic similarity between texture generators.

After presenting the abstract framework of the Asymptotic Hausdorff lifting, we return to edit-operation-based language similarity measures, study their properties, and establish complexity results for computing our primary instance, $\mathbb{AH}_{\mathsf{ned}}$. For regular languages, we prove PSPACE-hardness and give an approximation algorithm in coNEXP. For bounded context-free languages (BCFLs), we present an exact algorithm running in EXP.

Missing proofs are available in the full version.

2 Language Similarity Notions in the Literature

A variety of notions for measuring similarity between formal languages have been explored in the literature, arising from different motivations. In the following, we survey these measures through the lens of the requirements identified in the introduction, and highlight limitations that motivate our Asymptotic Hausdorff metric.

Complexity-based similarity measures

Since our focus is on formal languages, where (possibly infinite) languages are finitely represented by some computational model, one natural approach is to define a distance measure between languages via a distance between their representations. To ensure that such a measure is insensitive to the particular choice of representation, one may appeal to a canonical representation, when one exists for the class of languages under consideration.

Kolmogorov and automata-size based measures. In this spirit, [15] proposes using Kolmogorov complexity and defines the Kolmogorov distance between languages X and Y as $\mathcal{K}(X,Y) = |K(X) - K(Y)|$, where $K(L)$ denotes the Kolmogorov complexity of the language L. For regular languages, the minimal DFA can serve as a canonical representation, allowing K to be replaced by the number of DFA states. Alternatively, [15] suggests using the size of a minimal NFA, defined as the sum of its states, initial and final states, and transitions.

It is straightforward to see that these notions induce a pseudo-metric. (Indeed, any function of the form $\mathcal{S}(X,Y) = |S(X) - S(Y)|$, where $S(L)$ maps languages to $\mathbb{R}_{\geq 0}$, induces a pseudo-metric on the space of languages.) However, $\mathcal{K}$ and its variants measure differences in the complexity of languages rather than differences between the languages themselves. For example, $\mathcal{K}(a^*, b^*) = 0$, since the two languages are equally simple, even though every word in a^* needs a complete rewrite to transform into a word in b^*. In contrast, our goal is to define a pseudo-metric under which a^* and b^* are as far apart as possible.

Set-theoretic similarity measures

As mentioned in the introduction, since languages are sets, any metric on sets can be used to induce a metric on languages.

Jaccard. One of the earliest notions of set similarity, dating back to the 19th century, is the *Jaccard index*, $\bar{\mathcal{J}}$, along with its dual notion, the *Jaccard distance* $\mathcal{J}$, which serves as a dissimilarity measure. These are defined as follows:

$$\bar{\mathcal{J}}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \qquad \mathcal{J}(X,Y) = \frac{|X \triangle Y|}{|X \cup Y|} \tag{1}$$

where $\triangle$ denotes symmetric set difference. The Jaccard index and distance are undefined when $|X \cup Y|$ is 0 or ∞, and thus are inapplicable for infinite languages.

Cesáro-Jaccard. For infinite languages, one approach is to consider one of the limits

$$\lim_{n\to\infty} \mathcal{J}(X^{(n)}, Y^{(n)}) \qquad \text{or} \qquad \lim_{n\to\infty} \mathcal{J}(X^{(\leq n)}, Y^{(\leq n)}) \tag{2}$$

where $L^{(n)}$ (resp. $L^{(\leq n)}$) denotes the set of words in L of length n (resp. at most n). However, as shown in [22], these limits need not exist. For example, considering the left limit, if $X = a^*$ and $Y = (aa)^*$, then the fraction evaluates to 0 for even n and to 1 for odd n.

To address this issue, [22] propose smoothing the sequence using the Cesàro average, yielding the following distance measure between languages:

$$\mathcal{J}_{\mathcal{C}}(X,Y) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathcal{J}(X^{(\leq i)}, Y^{(\leq i)}) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \frac{|(X \triangle Y)^{(\leq i)}|}{|(X \cup Y)^{(\leq i)}|}$$

They show that $\mathcal{J}_{\mathcal{C}}$ is a pseudo-metric.

However, while the Cesàro-Jaccard distance successfully addresses the fact that formal-language theory is primarily concerned with infinite languages, and is finite-subset indifferent, it does not incorporate any notion of similarity between individual words. As a result, both $\mathcal{J}_{\mathcal{C}}(ca^*, ba^*) = 1$ and $\mathcal{J}_{\mathcal{C}}(c^*, b^*) = 1$, despite the fact that in the former pair the edit cost per word is uniformly bounded (one edit per word), while in the latter it grows unboundedly with word length. Our goal, by contrast, is a metric that reflects this asymptotic discrepancy, identifying the first pair as close and the second as far.

Discounted-sum Jaccard. Another approach to addressing the potential non-convergence of the limits in Equation 2 is to employ a discounted-sum construction. This idea was recently proposed in [5]. In particular, they show that

$$\mathcal{J}^{\lambda}_{\mathcal{DS}}(X, Y) = (1-\lambda)\sum_{n=0}^{\infty} \lambda^n \mathcal{J}(X^{(n)}, Y^{(n)}) = (1-\lambda)\sum_{n=0}^{\infty} \lambda^n \frac{|(X \triangle Y)^{(n)}|}{|(X \cup Y)^{(n)}|}$$

is a pseudo-metric for every $\lambda \in (0, 1)$.

However, this construction places greater weight on discrepancies at smaller word lengths, which limits its ability to capture asymptotic percentage behavior. In particular, differences among short words dominate the value of the distance, even when they become negligible relative to word length. For example, for $\lambda = \frac{1}{2}$ we obtain $\mathcal{J}^{\lambda}_{\mathcal{DS}}(a^{>2}, a^*) = \frac{7}{8}$ and $\mathcal{J}^{\lambda}_{\mathcal{DS}}(a^{\leq 2} \cup b^{>2}, a^*) = \frac{1}{8}$, despite the fact that the former pair differs only on finitely many short words, while the latter exhibits a persistent asymptotic discrepancy.

Shortlex vector approach. Another approach explored in [15] begins by ordering all words over the alphabet using the shortlex order (first by length and then lexicographically). A language L is then represented by an infinite binary vector $v_L \in \{0, 1\}^{\omega}$, where $v_L(i) = 1$ if and only if the i-th word belongs to L. The distance between two languages X and Y is defined by applying a chosen distance measure between binary vectors to v_X and v_Y.

Since $v_X(i) = v_Y(i)$ precisely when the i-th word either belongs to both X and Y or to neither, this construction effectively accounts for words in the symmetric difference (and intersection) of the two languages. The precise behavior of the resulting language distance depends on the specific choice of vector distance. Nevertheless, this approach does not incorporate any notion of similarity between words themselves, making it difficult to see how it could determine ba^* and ca^* as being closer than b^* and c^*.

Word-level lifted similarity measures

Predicate-lifted Jaccard. In [7], it is observed that language similarity notions can benefit from enriching set-based similarity measures, such as the Jaccard distance, with an explicit notion of distance between individual words. Rather than using the strict symmetric difference the approach of [7] introduces a predicate φ that captures when two words are considered sufficiently close. The comparison between X and Y considers only pairs of words deemed sufficiently close according to φ. For example, the predicate $\varphi(x, y)$ may be defined as $\mathsf{hamming}(x, y) \leq c_0$, where $\mathsf{hamming}$ denotes the Hamming distance between words and $c_0 \in \mathbb{N}$ is a fixed constant. Another example from [7] is $\varphi(x, y) = \mathsf{lcs}(x, y) \leq c_0$, where lcs denotes the length of the *longest common subsequence*.

Given such a predicate φ, they define $\bar{\mathcal{J}}^{\varphi}(X, Y) = \frac{|\varphi(X,Y)|}{|X \cup Y|}$ where $\varphi(X, Y) = \{x \in X \mid \exists y \in Y,\ \varphi(x, y)\} \cup \{y \in Y \mid \exists x \in X,\ \varphi(x, y)\}$. This construction essentially replaces the strict intersection in the standard Jaccard index with the set of all words that are sufficiently close according to the predicate φ.

Predicate-lifted information-rate. The focus in [7] is on extending the notion of information rate introduced by Shannon and Weaver [24] and applied to formal languages by Chomsky and Miller [6]. The information rate of a language L is defined as $\mathcal{I}(L) = \lim_{n\to\infty} \frac{\log|L^{(n)}|}{n}$. This notion pertains to a single language rather than a pair of languages and is intended to capture the *density* of a language. In [7], an extension for two languages, which additionally incorporates a predicate, is suggested: $\mathcal{I}^{\varphi}(X,Y) = \frac{\mathcal{I}(\varphi(X,Y))}{\mathcal{I}(X\cup Y)}$.

Since the information rate is not a (pseudo-)metric, it does not serve as a candidate for our purposes.

Predicate-lifted Cesàro-Jaccard. While not suggested in the literature, we note that the use of a predicate can also be applied to the Cesàro–Jaccard distance. For example, one can define $\mathcal{J}^{\varphi}_{\mathcal{C}}(X,Y)$ analogously to $\mathcal{J}_{\mathcal{C}}(X,Y)$, by replacing $(X\triangle Y)^{(\leq n)}$ with $(\overline{\varphi}(X,Y))^{(\leq n)}$ where $\overline{\varphi}(X,Y) = \{(x,y) \mid (x,y) \notin \varphi(X,Y)\}$. Taking the predicate φ to be $\mathsf{ed}(x,y) \leq 1$, we obtain $\mathcal{J}^{\varphi}_{\mathcal{C}}(ca^*, ba^*) = 0$, which is desirable and resolves the issue that $\mathcal{J}_{\mathcal{C}}(ca^*, ba^*) = 1$. Similarly, defining φ as $\mathsf{ned}(x,y) \leq \frac{1}{2}$ gives $\mathcal{J}^{\varphi}_{\mathcal{C}}(a^*, (ba)^*) = 0$ while $\mathcal{J}^{\varphi}_{\mathcal{C}}(a^*, (bba)^*) = 1$.[4]

However, the use of a predicate necessitates choosing a fixed threshold, which prevents distances from degrading gradually and makes it impossible to satisfy the monotonicity property described in Requirement 1.

We now turn to language similarity notions that lift a similarity measure between words.

Infinitum and Supremum based. As mentioned in the introduction, the measure $\mathbb{I}_{\mathsf{d}}(X,Y) = \inf_{x\in X}\inf_{y\in Y}\mathsf{d}(x,y)$, where d assigns a cost to string transformations, has been used in the literature [19, 23, 9]. However, this measure is not a metric. For example, taking d to be the normalized edit distance ned, we have $\mathbb{I}_{\mathsf{ned}}(a^+, b^*) = 1 > 0 + 0 = \mathbb{I}_{\mathsf{ned}}(a^+, a^*) + \mathbb{I}_{\mathsf{ned}}(a^*, b^*)$. Moreover, $\mathbb{I}_{\mathsf{ned}}(a^*, b^*) = 0$, illustrating the outlier sensitivity of $\mathbb{I}_{\mathsf{d}}$ (the outlier being ε). By similar reasoning, the dual notion $\mathbb{S}_{\mathsf{d}}(X,Y) = \sup_{x\in X}\sup_{y\in Y}\mathsf{d}(x,y)$ is also not a metric and suffers from outlier sensitivity; for instance, $\mathbb{S}_{\mathsf{ned}}(a^* \cup bb, a^*) = 1$.

Prefix-distance based. Considering variations in word-level similarity, several works examine the prefix distance between words. The prefix distance between two strings x and y, denoted $\mathsf{prf}(x,y)$, is defined as the number of characters in x and y that do not belong to their longest common prefix. The Hausdorff lifting of prf, denoted $\mathbb{H}_{\mathsf{prf}}$, has been studied in various works [20]. We note that $\mathbb{H}_{\mathsf{prf}}$, being a Hausdorff lifting, is sensitive to outliers. In addition, it is unbounded: for example, $\mathbb{H}_{\mathsf{prf}}(b^*, a^*) = \infty$, and it is also sensitive to bounded edits, as seen from $\mathbb{H}_{\mathsf{prf}}(ba^*, a^*) = \infty$.

$\mathcal{AC}^{\triangleright}$, Benedikt et al.'s measure. As mentioned in the introduction, Benedikt et al. [3] studied the number of edits required to repair a word in X into a word in Y. Since this quantity is often unbounded, their subsequent work [4] focused on capturing the *percentage* of edits required in the limit. Their work forms the basis for our approach. The exact formula they use is $\mathcal{AC}^{\triangleright}(X,Y) = \lim_{n\to\infty} \sup_{\substack{x\in X\\ |x|\geq n}} \inf_{y\in Y} \frac{\mathsf{ed}(x,y)}{|x|}$.

It is straightforward to see that $\mathcal{AC}^{\triangleright}(X,Y) \in [0,1]$ for any pair of languages X, Y. We note

[4] The formal definition of ed and ned are deferred to Subsection 4.1. Intuitively, ed counts minimal number of edit operations, and ned minimal percentage of edits. In particular, for every $k \in \mathbb{N}$ we have $\mathsf{ed}(ca^k, ba^k) = 1$, $\mathsf{ned}(a^{2k}, (ab)^k) = \frac{1}{2}$ and $\mathsf{ned}(a^{3k}, (bba)^k) = \frac{2}{3}$.

that $\mathcal{AC}^{\triangleright}$ is asymmetric by nature. For applications requiring symmetry, one can define $\mathcal{AC}(X,Y) = \max(\mathcal{AC}^{\triangleright}(X,Y), \mathcal{AC}^{\triangleright}(Y,X))$. However, the critical issue is that $\mathcal{AC}^{\triangleright}$ (and thus $\mathcal{AC}$) violates the triangle inequality.

▷ Claim 2.1. $\mathcal{AC}^{\triangleright}$ does not satisfy the triangle inequality.

In summary, while a variety of language similarity measures have been proposed, existing approaches face key limitations. Complexity-based measures capture overall representation size but ignore word-level structure; set- and vector-based measures fail to account for word similarity or are sensitive to infinite languages; and word-level or edit-distance–based liftings can be outlier-sensitive or fail to satisfy fundamental metric properties such as the triangle inequality. These observations motivate the need for a pseudo-metric that simultaneously accounts for word-level similarity, is robust to outliers, and preserves essential metric properties. In the next section, we introduce such a construction: the *Asymptotic Hausdorff* metric, $\mathbb{AH}_{\mathsf{d}}$, which generalizes the Hausdorff lifting to capture asymptotic behavior at the language level. We first present it at an abstract level, and later specialize to $\mathbb{AH}_{\mathsf{ned}}$, the lifting of the normalized edit distance ned [18].

3 Asymptotic Hausdorff

We turn to define the central notion of the paper, the *Asymptotic Hausdorff lifting*. In what follows we assume M is a domain (set) and $d : M \times M \to \mathbb{R}_{\geq 0}$ is a metric or a pseudo-metric. That is, we assume d satisfies the three pseudo-metric requirements:

1. *Reflexivity:* $d(x,x) = 0$ for all $x \in M$.
2. *Symmetry:* $d(x,y) = d(y,x)$ for all $x, y \in M$.
3. *Triangle inequality:* $d(x,z) \leq d(x,y) + d(y,z)$ for all $x, y, z \in M$.

3.1 Defining the Asymptotic Hausdorff Lifting

As discussed in the introduction, we are interested in domains whose elements are equipped with a natural notion of size. This allows us to distinguish between bounded and unbounded behavior and to focus on asymptotic phenomena.

▶ **Definition 3.1** (Size notion). *A* size notion *for M is a map $s : M \to \mathbb{R}_{\geq 0}$.*

Henceforth, we assume M is equipped with such a size notion s. We are interested in sets with increasing size of elements. To capture this we introduce the following definition, which considers infinite sequences of elements (rather than sets).

▶ **Definition 3.2** (s-bounded sequence). *A sequence $(x_i)_{i=1}^{\infty}$ is called* s-bounded *if there exists some $N \in \mathbb{N}$ such that $\limsup_{i\to\infty} s(x_i) \leq N$. Otherwise we say it is* s-unbounded.

We are interested in element-level distances d that separate s-bounded sequences from s-unbounded sequences, in the sense that the asymptotic distance is as large as possible.

▶ **Definition 3.3** (Asymptotic separation property). *We say d has the* asymptotic separation property *if for every s-unbounded sequence $(x_k)_{k=1}^{\infty}$, and s-bounded sequence $(y_k)_{k=1}^{\infty}$, d satisfies $\lim_{k\to\infty} d(x_k, y_k) = \sup d$ where $\sup d$ is the supremum of the image of d.*

We now have all the ingredients needed to define the Asymptotic Hausdorff lifting. As in the Hausdorff lifting, we first define a directional distance, and then take the maximum of going from X to Y and in the other direction.

▶ **Definition 3.4** (Asymptotic Hausdorff lifting). *Given a pseudo-metric d on M, the two functions $\mathbb{AH}_d^{\triangleright}$ and $\mathbb{AH}_d$ of type $\mathcal{P}(M) \times \mathcal{P}(M) \to \mathbb{R}_{\geq 0}$ are defined as follows*

$$\mathbb{AH}_d^{\triangleright}(X,Y) \stackrel{def}{=} \lim_{k\to\infty} \sup_{\substack{x\in X \\ s(x)\geq k}} \inf_{y\in Y} d(x,y) \qquad \mathbb{AH}_d(X,Y) = \max\left\{\mathbb{AH}_d^{\triangleright}(X,Y), \mathbb{AH}_d^{\triangleright}(Y,X)\right\}$$

We refer to $\mathbb{AH}_d^{\triangleright}(X,Y)$ as the asymptotic directional distance *from X to Y and to $\mathbb{AH}_d$ as the* asymptotic Hausdorff lifting *of d.*

3.2 Properties of the Asymptotic Hausdorff Lifting

We first establish that the asymptotic directional distance is well defined.

▷ Claim 3.5. $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y)$ is well defined for all $X, Y \subseteq M$.

Equivalently, the asymptotic directional distance $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y)$ can be defined as the supremum over all sequences $(x_n)_{n\geq 1} \subseteq X$ with $s(x_n) \to \infty$, of the limit $\limsup_{n\to\infty} \inf_{y\in Y} d(x_n, y)$. That is, as the worst-case asymptotic distance to Y attained along sequences of elements of X whose size grows unboundedly.

▷ Claim 3.6 (Equivalent definition to asymptotic directional distance). Let X and Y be sets. Then $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) = \sup_{\substack{(x_n)_{n=1}^{\infty}\subseteq X \\ s(x_n)\geq n,\ \forall n\in\mathbb{N}}} \limsup_{n\to\infty} \inf_{y\in Y} d(x_n, y)$

We next show that the asymptotic separation property is sufficient to lift the triangle inequality to the asymptotic setting.

▶ **Theorem 3.7** ($\mathbb{AH}_{\mathsf{d}}^{\triangleright}$ satisfies the triangle inequality). *Let d be a pseudo-metric that has the asymptotic separation property. Then $\mathbb{AH}_d^{\triangleright}$ satisfies the triangle inequality.*

The proof makes use of the following claim.

▷ Claim 3.8 (Point-wise triangle inequality). Let M be a set, d a pseudo-metric on M and Z a subset of M. Then for every $x, y \in M$ $\inf_{z\in Z} d(x,z) \leq d(x,y) + \inf_{z\in Z} d(y,z)$

Proof. Let $Z \subseteq M$ and $x, y \in M$. As d is a pseudo-metric $d(x,z) \leq d(x,y) + d(y,z)$ for every $z \in Z$. Thus $\inf_{z\in Z} d(x,z) \leq \inf_{z\in Z} \{d(x,y) + d(y,z)\} = d(x,y) + \inf_{z\in Z} d(y,z)$. ◁

Proof of Theorem 3.7. Let X, Y, Z be subsets of M and let $\varepsilon > 0$. Let $N \in \mathbb{N}$ such that

$$\sup_{y\in Y\ s(y)\geq N} \inf_{z\in Z} d(y,z) \leq \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + \tfrac{\varepsilon}{2}. \tag{3}$$

If all infinite sequences $(x_k)_{k=1}^{\infty} \subseteq X$ are s-bounded, then X is an s-bounded set and thus

$$\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Z) = \lim_{k\to\infty} \sup_{\substack{x\in X \\ s(x)\geq k}} \inf_{z\in Z} d(x,z) = \lim_{k\to\infty} \sup_{x\in\emptyset} \inf_{z\in Z} d(x,z) = 0 \leq \mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z).$$

Otherwise, let $(x_k)_{k=1}^{\infty} \subseteq X$ be some arbitrary sequence that has an infinite subsequence $(x_{k_i})_{i=1}^{\infty}$ such that $(s(x_{k_i}))_{i=1}^{\infty}$ is a non decreasing sequence that tends to ∞. We consider two cases:

Case 1. There exists a subsequence $(k_j)_{j=1}^{\infty}$ such that for every $y \in Y$ that satisfies $s(y) \geq N$, we have that $d(x_{k_j}, y) > \inf_{y \in Y} d(x_{k_j}, y) + \frac{\varepsilon}{2}$.

Let $Y_{x_{k_j}}$ be the set of elements that satisfy $d(x_{k_j}, y) \leq \inf_{y \in Y} d(x_{k_j}, y) + \frac{\varepsilon}{2}$. We get that $Y_{x_{k_j}} \subseteq Y^{\leq N}$ where $Y^{\leq N} = \{y \in Y : s(y) \leq N\}$. Therefore as d has the asymptotic separation property

$$\begin{aligned}\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) &= \lim_{k\to\infty} \sup_{\substack{x\in X\\ s(x)\geq k}} \inf_{y\in Y} d(x,y) \geq \lim_{j\to\infty} \inf_{y\in Y} d(x_{k_j}, y)\\ &= \lim_{j\to\infty} \inf_{y\in Y^{\leq N}} d(x_{k_j}, y) =^{(\text{Definition 3.3})} \sup d.\end{aligned}$$

Since $\sup d$ bounds $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(A,B)$ for any A, B, together we get that $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) = \sup d$ and

$$\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Z) \leq \sup d = \mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) \leq \mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z).$$

Case 2. There exists k' such that for every $k \geq k'$ there exists $y_k \in Y$ that satisfies $s(y_k) \geq N$ and also satisfies

$$d(x_k, y_k) \leq \inf_{y\in Y} d(x_k, y) + \tfrac{\varepsilon}{2} \tag{4}$$

As d is a pseudo-metric by Claim 3.8 we know that

$$\inf_{z\in Z} d(x_k, z) \leq d(x_k, y_k) + \inf_{z\in Z} d(y_k, z) \tag{5}$$

And because $s(y_k) \geq N$ and $a \leq \sup A$ for any $A \supseteq \{a\}$ we get

$$\inf_{z\in Z} d(y_k, z) \leq \sup_{\substack{y\in Y\\ s(y)\geq N}} \inf_{z\in Z} d(y,z) \leq^{(3)} \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + \tfrac{\varepsilon}{2} \tag{6}$$

Putting it all together we get

$$\begin{aligned}\inf_{z\in Z} d(x_k, z) &\leq^{(5)} d(x_k, y_k) + \inf_{z\in Z} d(y_k, z)\\ &\leq^{(4)} \inf_{y\in Y} d(x_k, y) + \tfrac{\varepsilon}{2} + \inf_{z\in Z} d(y_k, z)\\ &\leq^{(6)} \inf_{y\in Y} d(x_k, y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + \varepsilon.\end{aligned} \tag{7}$$

Note that this inequality is satisfied for an arbitrary sequence in X with size that tends to ∞. Using Claim 3.6 there exists $(p_k)_{k=1}^{\infty}$ a sequence that satisfies

$$\varepsilon + \limsup_{k\to\infty} \inf_{z\in Z} d(p_k, z) \geq \mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Z). \tag{8}$$

Thus, we finally get

$$\begin{aligned}\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Z) &\leq^{(8)} \limsup_{k\to\infty} \inf_{z\in Z} d(p_k, z) + \varepsilon\\ &\leq^{(7)} \limsup_{k\to\infty} \left\{ \inf_{y\in Y} d(p_k, y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + 2\varepsilon \right\}\\ &= \limsup_{k\to\infty} \left\{ \inf_{y\in Y} d(p_k, y) \right\} + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + 2\varepsilon\\ &\leq \sup_{\substack{(x_k)_{k=1}^{\infty}\subseteq X\\ s(x_k)\geq k,\ \forall k\in\mathbb{N}}} \limsup_{k\to\infty} \left\{ \inf_{y\in Y} d(x_k, y) \right\} + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + 2\varepsilon\\ &=^{Claim\ 3.6} \mathbb{AH}_{\mathsf{ned}}^{\triangleright}(X,Y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z) + 2\varepsilon.\end{aligned}$$

As ε is arbitrary small we get that $\mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Z) \leq \mathbb{AH}_{\mathsf{d}}^{\triangleright}(X,Y) + \mathbb{AH}_{\mathsf{d}}^{\triangleright}(Y,Z)$. ◀

Since $\mathbb{AH}_{\mathsf{d}}$ obviously satisfies reflexivity and symmetry, an immediate corollary of Theorem 3.7 is that $\mathbb{AH}_{\mathsf{d}}$ is a pseudo-metric.

▶ **Theorem 3.9** (Asymptotic Hausdorff is a pseudo-metric). *$\mathbb{AH}_d$ is a pseudo-metric when d has the asymptotic separation property.*

Returning to our primary motivating setting of words and language, in the next section, we show that many word similarity measures satisfy the asymptotic separation property, and therefore can be lifted to pseudo-metrics between languages. We expect that similar results hold for metrics on trees and graphs, but we do not pursue this direction here.

More broadly, this naturally raises a stronger question: how restrictive is the asymptotic separation property in general? We therefore turn to identifying broad classes of distance functions for which this property holds. The following theorem shows that the asymptotic separation property is in fact quite common: every metric induced by a norm satisfies it.

▶ **Theorem 3.10** (Norm-induced metrics satisfy asymptotic separation). *Let $(M, \|\cdot\|)$ be a normed space with $d(x, y) = \|x - y\|$ and $s(x) = \|x\|$. Then d has the asymptotic separation property.*

▶ **Corollary 3.11.** *Let $(M, \|\cdot\|)$ be a normed space and $d(x, y) = \|x - y\|$. Then $\mathbb{AH}_d$ is a pseudo-metric on $\mathcal{P}(M)$.*

As an immediate consequence of the corollary, the Asymptotic Hausdorff lifting applies to all metrics induced by norms. This includes, in particular, the Euclidean distance and, more generally, L_p distances for any $p \geq 1$. Hence, $\mathbb{AH}_d$ can be meaningfully applied in continuous settings alongside the discrete ones considered earlier.

While we arrived at this notion from the perspective of formal languages, as discussed in the introduction, we believe that it is applicable in a much broader range of settings. In particular, it is well suited to contexts in which infinite sets arise from finite representations or generators, are equipped with a natural notion of size, and contain elements whose size is unbounded. This situation commonly occurs, for example, when considering functions defined over infinite domains, where each input x induces an element $f(x)$ in the set generated by the function.

4 Formal Languages and Distances

We now focus on language similarity notions obtained via the Asymptotic Hausdorff lifting. This construction applies to a variety of word-level metrics, including both classical and normalized edit distances. While all are suitable for lifting, only normalized metrics are compatible with the percentage-based requirements discussed in the introduction. Among these, the normalized edit distance ned will play a central role and serve as our main point of reference. We begin by briefly reviewing standard distances between words.

4.1 Metrics on words (Preliminaries)

Let Σ be a finite alphabet. For a word $x \in \Sigma^*$, we write $|x|$ for its length and $x[i]$ for its i-th letter. The empty word is denoted by ε.

One of the oldest metrics on words is the *Hamming distance*. It measures the distance between x and y as the number of letters on which they differ plus the difference between their lengths. Formally, if $\ell_x = |x|$ and $\ell_y = |y|$ then $\mathsf{hamming}(x, y) = \left|\{i \colon x[i] \neq y[i], i \leq \min(\ell_x, \ell_y)\}\right| + \left|\ell_x - \ell_y\right|$. Another common simple metric between words is the *prefix distance* [20]. It measures the number of letters in x and y that are not in the longest common

prefix of x and y. Formally, $\mathsf{prf}(x,y) = |x| + |y| - 2 \cdot \max\{|z| \colon x, y \in z \cdot \Sigma^*\}$. Beyond these position-based notions, many widely used distances between words are defined in terms of edit operations, which we review next.

Edit operations and edit paths. We work with the standard edit operations: insertion, deletion, substitution, and no-op. Let $\hat{\Gamma} = (\Sigma \cup \{\varepsilon\})^2$, and write $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ for the pair (a,b). The set of *edit operations* is $\Gamma = \hat{\Gamma} \setminus \{\left[\begin{smallmatrix} \varepsilon \\ \varepsilon \end{smallmatrix}\right]\}$, where $\left[\begin{smallmatrix} a \\ b \end{smallmatrix}\right]$ denotes substitution of a by b, $\left[\begin{smallmatrix} a \\ a \end{smallmatrix}\right]$ a no-op, $\left[\begin{smallmatrix} a \\ \varepsilon \end{smallmatrix}\right]$ deletion of a, and $\left[\begin{smallmatrix} \varepsilon \\ a \end{smallmatrix}\right]$ insertion of a.

An *edit path* from x to y is a finite sequence p over Γ such that, writing $p = \left[\begin{smallmatrix} a_1 \\ b_1 \end{smallmatrix}\right]\left[\begin{smallmatrix} a_2 \\ b_2 \end{smallmatrix}\right]\cdots\left[\begin{smallmatrix} a_n \\ b_n \end{smallmatrix}\right]$, we have $a_1 \cdots a_n = x$ and $b_1 \cdots b_n = y$. We denote by $|p| = n$ the length of p. Let $d : \Gamma \to [0,1]$ assign weights to edit operations. The *weight* of an edit path p is $\mathsf{wgt}(p) = \sum_{i=1}^{n} d(a_i, b_i)$. Its *cost* is defined as $\mathrm{cost}(p) = \mathsf{wgt}(p)/|p|$. We write $p : x \rightsquigarrow y$ to denote that p is an edit path from x to y.

Edit-distance notions. Several notions of edit distance have been proposed in the literature, differing mainly in how edit operations are aggregated and normalized. The most basic notion is the *Levenshtein (edit) distance* [17], denoted ed, which returns the minimum total weight of an edit path transforming x into y.

For applications involving words of substantially different lengths, normalization becomes essential. When words have equal length, normalizing by the word length is straightforward; however, for unequal lengths, naïve normalizations – such as dividing by the maximum, minimum, or sum of the lengths – generally fail to preserve the metric properties (cf. [26]). To address this issue, several normalized variants of edit distance have been proposed, including the *normalized edit distance* ned [18], the *generalized edit distance* ged [26], and the *contextual edit distance* ced [8]. Despite differing in their normalization schemes, all these notions yield bounded distances that satisfy the metric axioms, and are therefore well suited for comparing words of varying lengths.

A common choice of weights is the *uniform weight*, in which no-op operations have cost 0 and all other edit operations have weight 1. All notions but ced allow non-uniform weights. When non-uniform weights are considered, they must satisfy additional conditions to ensure that the induced distances ed_d, ned_d, and ged_d are metrics [26, 10, 11]. Unless stated otherwise, we henceforth assume the uniform weight.

▶ **Definition 4.1** (Edit-distance notions). *Let $x, y \in \Sigma^*$.*

- Levenshtein (edit) distance [17]: *ed_d minimizes the weight of an edit path:*

$$\mathsf{ed}_d(x,y) = \min\{\mathsf{wgt}(p) \mid p \text{ is an edit path from } x \text{ to } y\}.$$

- Normalized edit distance [18, 10, 11]: *ned_d minimizes the average cost per operation, by dividing by the edit path length:*

$$\mathsf{ned}_d(x,y) = \min\{\mathrm{cost}(p) \mid p \text{ is an edit path from } x \text{ to } y\}.$$

- Generalized edit distance [26]: *ged is another way to obtain an averaged cost:*

$$\mathsf{ged}_d(x,y) = \tfrac{2 \cdot \mathsf{ed}_d(x,y)}{|x|+|y|+\mathsf{ed}_d(x,y)}.$$

- Contextual edit distance [8]: *Last, ced provides an averaged cost by considering the context of the edits. Formally, for strings s, s' for which $\mathsf{ed}(s,s') = 1$, one defines $\mathsf{ced}(s,s') = 1/\max(|s|,|s'|)$. For a sequence $\rho = (s_0, \ldots, s_k)$ satisfying $\mathsf{ed}(s_i, s_{i+1}) = 1$ for every $i < k$, let $\mathsf{ced}(\rho) = \sum_{i=1}^{k} \mathsf{ced}(s_{i-1}, s_i)$. Then*

$$\mathsf{ced}(x,y) = \min\{\mathsf{ced}(\rho) \mid \rho{=}(s_0, \ldots, s_k),\ s_0{=}x,\ s_k{=}y\}.$$

▶ **Example 4.2.** Consider $x = aab$ and $y = abac$. One edit path from x to y is $p_1 = \left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\c\end{smallmatrix}\right]$. Another edit path is $p_2 = \left[\begin{smallmatrix}a\\\varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\c\end{smallmatrix}\right]$. We have $\mathsf{wgt}(p_1) = \mathsf{wgt}(p_2) = 3$. Since no edit path has smaller weight, it follows that $\mathsf{ed}(x, y) = 3$. Applying this value in the definition of ged, we obtain $\mathsf{ged}(x, y) = \frac{2\cdot 3}{3+4+3} = \frac{3}{5}$. Since $|p_1| = 4$ and $|p_2| = 5$, we have $\mathrm{cost}(p_1) = \frac{3}{4}$ and $\mathrm{cost}(p_2) = \frac{3}{5}$. As no edit path has smaller cost, we conclude that $\mathsf{ned}(x, y) = \frac{3}{5}$. For ced, consider the sequence of strings $s_0 = aab$, $s_1 = abb$, $s_2 = aba$, and $s_3 = abac$. Note that $\mathsf{ed}(s_{i-1}, s_i) = 1$ for all $1 \le i \le 3$. Therefore, $\mathsf{ced}(s_0, s_1, s_2, s_3) = \frac{1}{3} + \frac{1}{3} + \frac{1}{4} = \frac{11}{12}$. However, a different sequence yields a smaller value. In particular, $\mathsf{ced}(aab, aabc, abbc, abac) = \frac{1}{4} + \frac{1}{4} + \frac{1}{4} = \frac{3}{4}$. Thus, $\mathsf{ced}(x, y) \le \frac{3}{4}$.

The values of ed are clearly unbounded. In contrast, the values of ned and ged are bounded by 1 and may attain this bound. The values of ced are unbounded; however, they can be made bounded by considering the variant $\mathsf{ced}'(x, y) = \max\{1, \mathsf{ced}(x, y)\}$ [10, 8]. Finally, we note that $\mathsf{ged}(x, y) \le \mathsf{ned}(x, y)$ for all x, y; see the full version.

We establish that each of these notions satisfies the asymptotic separation property, and thus is amenable to lifting via the Asymptotic Hausdorff construction.

▷ Claim 4.3. The asymptotic separation property holds for ed, ned, ged, ced, and prf.

▶ **Corollary 4.4.** *$\mathbb{AH}_{ed}$, $\mathbb{AH}_{ned}$, $\mathbb{AH}_{ged}$, $\mathbb{AH}_{ced}$, and $\mathbb{AH}_{prf}$, are all pseudo-metrics on the set of languages.*

4.2 $\mathbb{AH}$ for Normalized Edit Distances

Recall the three requirements given in the introduction. We claim that the three notions of normalized edit distance ned, ged and ced, all satisfy these properties.

▷ Claim 4.5 ($\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}$ are finite-subset indifferent). Let X, Y be infinite language and F a finite languages. Then $\mathcal{D}(X, Y) = \mathcal{D}(X \cup F, Y)$ for every $\mathcal{D} \in \{\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}\}$.

▷ Claim 4.6 ($\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}$ are bounded-edits insensitive). Let $X, Y \subseteq \Sigma^*$. If $\exists k \in \mathbb{N}$ such that for every $x \in X$ there exists $y \in Y$ such that $\mathsf{ed}(x, y) < k$ and vice versa then $\mathcal{D}(X, Y) = 0$ for every $\mathcal{D} \in \{\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}\}$.

▷ Claim 4.7 ($\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}$ have a percentage nature). For every $\mathcal{D} \in \{\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}\}$ we have that $\mathcal{D}(a^*, (a^j b)^*) < \mathcal{D}(a^*, (a^i b)^*)$ for all $j > i$.

▶ **Corollary 4.8.** *$\mathbb{AH}_{ned}, \mathbb{AH}_{ged}, \mathbb{AH}_{ced}$ satisfy all of our requirements.*

Recall (cf. Requirement 1) that one may expect $\mathcal{D}(a^*, (ab)^*) = \frac{1}{2}$ and, more generally, $\mathcal{D}(a^*, (a^k b)^*) = \frac{1}{k+1}$. We deliberately adopted a more relaxed requirement, as enforcing these equalities exactly would be overly restrictive. Nevertheless, we show that $\mathbb{AH}_{\mathsf{ned}}$ does satisfy this stricter behavior, whereas $\mathbb{AH}_{\mathsf{ged}}$ and $\mathbb{AH}_{\mathsf{ced}}$ do not.

▶ **Requirement 4** (Strict percentage property). *We say that a language metric $\mathcal{D}$ satisfies the* strict percentage property *if the following hold:*

- $\mathcal{D}\left(a^*, (a^k b^l)^*\right) = \frac{l}{k+l}$ *for all $k, l \ge 0$ with $k + l \ge 1$;*
- $\mathcal{D}\left((a^i b)^*, (a^k b)^*\right) = \frac{k-i}{k+1}$ *for all $i < k$.*

▷ Claim 4.9 (Satisfaction of the strict percentage property). The strict percentage property is satisfied by $\mathbb{AH}_{\mathsf{ned}}$, but not by $\mathbb{AH}_{\mathsf{ged}}$ or $\mathbb{AH}_{\mathsf{ced}}$.

For this reason, we prefer $\mathbb{AH}_{\mathsf{ned}}$ over $\mathbb{AH}_{\mathsf{ged}}$ and $\mathbb{AH}_{\mathsf{ced}}$.

▶ Remark 4.10 (Relations). The following relations between the metrics hold for every pair of languages X, Y: $\mathbb{AH}_{\mathsf{ged}}(X,Y) \leq \mathbb{AH}_{\mathsf{ned}}(X,Y) \leq \mathbb{H}_{\mathsf{ned}}(X,Y) \leq \mathbb{H}_{\mathsf{ed}}(X,Y)$

While $\mathbb{AH}_{\mathsf{ned}}$ satisfies all of our stated requirements, its insensitivity to finite outliers and bounded local edits implies that it is a pseudo-metric rather than a metric. In scenarios where one wishes to additionally distinguish, for example, $a^* \cup b$ from a^* (i.e., to forgo outlier insensitivity), or a^*b from a^* (i.e., to forgo bounded-edit insensitivity), this can be achieved by combining $\mathbb{AH}_{\mathsf{ned}}$ with an additional language metric $\mathcal{D}$ (e.g. $\mathbb{H}_{\mathsf{ed}}$, $\mathbb{H}_{\mathsf{prf}}$, etc.). Specifically, one may consider the refined generalized metric $\mathbb{D}_{\times}(X,Y) = (\mathbb{AH}_{\mathsf{ned}}(X,Y), \mathcal{D}(X,Y))$, ordered lexicographically so that the $\mathbb{AH}_{\mathsf{ned}}$ component is dominant.[5] If $\mathcal{D}$ takes values in a domain that is bounded above by a constant C (as is the case, for example, for $\mathbb{H}_{\mathsf{ned}}$), then this generalized metric can be collapsed into a genuine metric by defining $\mathbb{D}_{\otimes}(X,Y) = (C{+}1) \cdot \mathbb{AH}_{\mathsf{ned}}(X,Y) + \mathcal{D}(X,Y)$, which preserves the dominance of the asymptotic distance.

4.3 Asymptotic Essence of Regular Languages

A pseudo-metric $\mathcal{D}$ naturally induces an equivalence relation $\equiv_{\mathcal{D}}$, where $X \equiv_{\mathcal{D}} Y$ if and only if $\mathcal{D}(X,Y) = 0$. In our context, two languages are considered equivalent with respect to language-metric $\mathcal{D}$ if their $\mathcal{D}$-distance is zero.

Recalling the discussion in the introduction, which was illustrated using regular expressions, it appears that removing all parts of a regular expression that are not under a Kleene star yields a language that is equivalent to the original one under the desired pseudo-metric $\mathcal{D}$. Intuitively, the *asymptotic essence* of a regular expression is captured by the subexpressions occurring under Kleene closure, whereas all other subexpressions are asymptotically negligible. Accordingly, we denote by $\mathcal{E}_{\mathrm{REG}}(r)$ the regular expression obtained by retaining only these Kleene-starred subexpressions.

Note that applying this procedure to different regular expressions defining the same language may result in different languages. For example, consider $r_1 = a(aa)^* \cup (aa)^*$ and $r_2 = a^*$. Although $[\![r_1]\!] = [\![r_2]\!]$, we have $[\![\mathcal{E}_{\mathrm{REG}}(r_1)]\!] \neq [\![\mathcal{E}_{\mathrm{REG}}(r_2)]\!]$, since $\mathcal{E}_{\mathrm{REG}}(r_1) = (aa)^* \cup (aa)^*$ whereas $\mathcal{E}_{\mathrm{REG}}(r_2) = a^*$.

Since regular languages do not admit a canonical regular expression, we seek to define asymptotic essence at the level of automata, so that it can later be applied to the minimal DFA, which is canonical up to isomorphism. Given a DFA or an NFA A recognizing a language L, we seek an intuitively simpler language L' that is equivalent to L under $\equiv_{\mathcal{D}}$. To this end, we decompose A into its strongly connected components (SCCs) and replace all transitions that are not contained in some SCC by ε-transitions. The resulting automaton is a simpler NFA A' that recognizes a language L' which, intuitively, preserves exactly the asymptotically significant behavior of A.[6]

We denote by $\mathcal{E}_{\mathrm{FA}}(A)$ the NFA obtained by this construction. The *asymptotic essence* of a regular language L, denoted $\mathcal{E}(L)$, is then defined as $\mathcal{E}_{\mathrm{FA}}(A_L)$, where A_L is the minimal DFA recognizing L.

The following claim shows that these constructions – whether applied to regular expressions or to automata – yield languages that are equivalent under $\equiv_{\mathcal{D}}$ when $\mathcal{D}$ is instantiated as one of the Asymptotic Hausdorff distances using a normalized word metric.

---

[5] Here we use the term *generalized metric* to refer to a distance function whose codomain is a totally ordered set (rather than $\mathbb{R}_{\geq 0}$), and which satisfies the triangle inequality with respect to that order; see, e.g., [16, 12].

[6] Here, we use the strict notion of SCC: a singleton state forms an SCC only if it has a self-loop.

▷ Claim 4.11 (Asymptotic Essence Equivalences). Let L be a regular language, and r and A a regular expression and a NFA recognizing L, resp. Then, for every $\mathcal{D} \in \{\mathbb{AH}_{\mathsf{ned}}, \mathbb{AH}_{\mathsf{ged}}, \mathbb{AH}_{\mathsf{ced}}\}$

$$L \equiv_{\mathcal{D}} [\![\mathcal{E}(L)]\!] \equiv_{\mathcal{D}} [\![\mathcal{E}_{\text{REG}}(r)]\!] \equiv_{\mathcal{D}} [\![\mathcal{E}_{\text{FA}}(A)]\!]$$

Sketch proof. The proof follows from Claim 4.5 since $\mathcal{E}_{\text{REG}}$ and $\mathcal{E}_{\text{FA}}$ (and thus $\mathcal{E}$) yield languages that differ from the original language by finitely many words and require finitely many edits. ◁

We note that this claim does not hold in general for $\mathbb{AH}_{\mathsf{d}}$. For example, it fails for $\mathbb{AH}_{\mathsf{ed}}$, since $\mathcal{E}_{\text{REG}}(a^*bb) = a^*$ while $\mathbb{AH}_{\mathsf{ed}}(a^*bb, a^*) = 2$ implying $a^*bb \not\equiv_{\mathbb{AH}_{\mathsf{ed}}} a^*$.

5 On the Computation of $\mathbb{AH}_{\mathsf{ned}}$

We now turn to the computation of $\mathbb{AH}_{\mathsf{ned}}$, our primary notion of interest (cf. Corollary 4.8 and Claim 4.9). We begin, in Subsection 5.1, by considering regular languages, where we establish a PSPACE-hardness result and present an approximation algorithm in coNEXP. We then move to Subsection 5.2, where we develop a detailed algorithm for bounded context-free languages.

5.1 $\mathbb{AH}_{\mathsf{ned}}$ for Regular Languages

For regular languages, we establish both a hardness result and an approximation bound for computing $\mathbb{AH}_{\mathsf{ned}}$.

Hardness. We begin by showing that, as in the case of $\mathcal{AC}^{\triangleright}$, which serves as our starting point, the problem is already PSPACE-hard when languages are regular and $X = \Sigma^*$.

▶ **Theorem 5.1** (PSPACE-hardness). *Let Σ be an alphabet, T an NFA over Σ and ν a rational number. The problem of deciding whether $\mathbb{AH}^{\triangleright}_{ned}(\Sigma^*, [\![T]\!]) \leq \nu$ is PSPACE-hard.*

Sketch proof. The proof proceeds by a reduction from the universality problem for NFAs, adapting ideas from the PSPACE-hardness construction for $\mathcal{AC}^{\triangleright}$ [4] to account for normalization by edit-path length rather than word length. Given an NFA A over Σ, we construct an NFA T over the extended alphabet $\Gamma = \Sigma \cup \{\#\}$ that recognizes the language $(\# \cdot [\![A]\!])^*$. We then show that if A is universal, i.e. $[\![A]\!] = \Sigma^*$, then $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(\Gamma^*, [\![T]\!]) = 0$, whereas if A is not universal, then $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(\Gamma^*, [\![T]\!]) > 0$.

To establish the latter case, we consider the sequence $(x_k)_{k\geq 1}$ defined by $x_k = (\#x)^k$ for some $x \in \Sigma^* \setminus [\![A]\!]$. We first analyze the cost of an optimal edit path for x_k and then study the limit behavior arising in the computation of $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}$. ◀

The complete proof makes use of the following claim which we also use for the approximation result.

▷ Claim 5.2. Let x be a word and let Y be a language recognized by an NFA with n states. Then there exists a word $y_x \in Y$ attaining the value $\inf_{y\in Y} \mathsf{ned}(x, y)$. Moreover, there exists an optimal edit path from x to y_x of length at most $n(|x|+1)$.

Approximation. Next, we establish bounds relating $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}$ and $\mathcal{AC}^{\triangleright}$. In particular, $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y)$ is sandwiched between $\mathcal{AC}^{\triangleright}(X,Y)$ and $\frac{1}{d}\cdot\mathcal{AC}^{\triangleright}(X,Y)$, where d is the number of states in the minimal DFA for Y.

▷ Claim 5.3. Let X and Y be regular languages. Then $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y) \leq \mathcal{AC}^{\triangleright}(X,Y)$.

▷ Claim 5.4. Let X and Y be regular languages. Then $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y) \geq \frac{1}{d}\cdot\mathcal{AC}^{\triangleright}(X,Y)$ where d is the number of states in the minimal DFA of Y.

Using these bounds and the coNExp-time algorithm for $\mathcal{AC}^{\triangleright}$ developed in [4], which relies on substantial technical machinery, we obtain a coNExp-time approximation algorithm for $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}$.

▶ **Lemma 5.5.** *Let X and Y be regular languages. Then there is a coNExp algorithm finding c such that $\mathbb{AH}^{\triangleright}_{ned}(X,Y) \in [\frac{1}{d},1]c$ where d is the number of states in the minimal DFA of Y.*

5.2 $\mathbb{AH}_{\mathsf{ned}}$ for Bounded Context-Free Languages

We now turn to *bounded context-free languages* (BCFLs), a well-studied subclass of context-free languages with rich structural properties and strong decidability results. We assume familiarity with standard definitions and basic properties of context-free languages, and provide the necessary definitions for *bounded* CFLs.

Bounded CFL (Preliminaries). A language L is said to be *bounded* if there exist fixed words $w_1, w_2, \ldots, w_n \in \Sigma^*$ such that $L \subseteq w_1^* w_2^* \cdots w_n^*$. A language is a *bounded context-free language* if it is both bounded and context-free.

Let $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$. The *Parikh vector* of a word $w \in \Sigma^*$ is the k-tuple $\mathit{Parikh}(w) = (|w|_{\sigma_1}, \ldots, |w|_{\sigma_k})$, where $|w|_\sigma$ is the number of occurrences of σ in w. For bounded languages, it is often convenient to work with a Parikh representation relative to the bounding words. Specifically, given $w_1, \ldots, w_n$ and a word $w = w_1^{n_1} w_2^{n_2} \cdots w_n^{n_n}$, we define the Parikh vector of w with respect to these words to be $(n_1, n_2, \ldots, n_n)$.

Parikh's Theorem states that for every context-free language L, the set of Parikh vectors $\mathit{Parikh}(L) = \{\mathit{Parikh}(w) \mid w \in L\}$ is a *semi-linear set* [21]. A set $S \subseteq \mathbb{N}^n$ is *linear* if there exist vectors $U = \{\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_m\} \subseteq \mathbb{N}^n$ such that $S = \{\mathbf{u}_0 + t_1\mathbf{u}_1 + \cdots + t_m\mathbf{u}_m \mid t_1, \ldots, t_m \in \mathbb{N}\}$. The vector $\mathbf{u}_0$ is called the *constant vector*, and the remaining vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are called *period vectors*. The matrix in $\mathbb{N}^{n\times(m+1)}$ whose columns are the vectors $\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_m$ is called the *generator matrix* of S. A set is *semi-linear* if it is a finite union of linear sets.

▶ **Example 5.6.** As an example, consider the language $L = \{(aba)^n(bcc)^{3n}d^k e^m c^{2m+1} \mid n,k,m \geq 0\}$. This language is bounded, since $L \subseteq w_1^* w_2^* w_3^* w_4^* w_5^*$ for w_1=aba, w_2=bcc, w_3=d, w_4=e, and w_5=c. For the word $w = (aba)(bcc)^3 d^2 e c^3$, the corresponding Parikh vector is $(1,3,2,1,3)$. The Parikh image of L is generated by the constant vector $\mathbf{u}_0 = (0,0,0,0,1)$ together with the period vectors $\mathbf{u}_1 = (1,3,0,0,0)$, $\mathbf{u}_2 = (0,0,1,0,0)$, $\mathbf{u}_3 = (0,0,0,1,2)$, that correspond to the powers n, k, and m, respectively.

Ginsburg and Spanier [13] characterized bounded context-free languages by showing that a bounded language is context-free if and only if its Parikh image (with respect to the bounding words) is a *stratified semi-linear set*. A linear set is *stratified* if (i) each of its period vectors has at most two non-zero coordinates, and (ii) the non-zero coordinates of distinct period vectors do not interleave. Formally, if $\mathbf{u}$ has non-zero coordinates at indices i_1<i_2

and $\mathbf{v}$ has non-zero coordinates at $j_1<j_2$, then it is not the case that $i_1<j_1<i_2<j_2$. In the example above, the period vectors satisfy these conditions: each has at most two non-zero entries, and the index sets $\{1,2\}$, $\{3\}$, and $\{4,5\}$ are pairwise non-interleaving.

Towards solving $\mathbb{AH}_{\mathsf{ned}}$ for BCFLs

For simplicity, we focus on BCFLs whose Parikh image is a stratified linear (rather than semi-linear) set. Henceforth, let $X \subseteq x_1^* \cdots x_n^*$ and $Y \subseteq y_1^* \cdots y_m^*$ be BCFLs with generators $U \in \mathbb{N}^{n\times(k+1)}$ and $V \in \mathbb{N}^{m\times(r+1)}$, respectively. Let X_0 be the BCFL with the same generating vector set U as X but where $\mathbf{u}_0 = \mathbf{0}$. Thus with $\sum_{i=1}^{n} \mathbf{u}_0[i] \cdot |w_i|$ edit operations we can move between X and X_0 and hence by Claim 4.6 $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X_0, X) = 0$. From this point onward we thus assume $\mathbf{u}_0 = \mathbf{v}_0 = \mathbf{0}$ and hence $U \in \mathbb{N}^{n\times k}$ and $V \in \mathbb{N}^{m\times r}$.

▶ **Example 5.7.** We use $X = \{(a)^n(bba)^{2n} \mid n \in \mathbb{N}\}$ and $Y = \{(abbb)^k(ab)^{2m}(c)^m \mid k, m \in \mathbb{N}\}$ as a running example. The generators of X and Y are $U \in \mathbb{N}^{2\times 1}$ and $V \in \mathbb{N}^{3\times 2}$ where $\mathbf{u}_1 = (1, 2)$ $\mathbf{v}_1 = (1, 0, 0)$ and $\mathbf{v}_2 = (0, 2, 1)$.

We compute $\mathbb{AH}_{\mathsf{ned}}$ for BCFLs using linear programs. This requires several auxiliary notions, which we introduce next.

The edit graph. Let $x = a_1a_2 \dots a_n$ and $y = b_1b_2 \dots b_m$. The edit graph $G_{x,y}$ is the directed graph whose vertices are the grid points $\{0, \dots, n\} \times \{0, \dots, m\}$, with edges from (i, j) to $(i{+}1, j)$, $(i, j{+}1)$, and $(i{+}1, j{+}1)$ whenever the target vertex exists. An edge to $(i{+}1, j)$ corresponds to deleting a_i, and has weight 1; an edge to $(i, j{+}1)$ corresponds to inserting b_j, and has weight 1; and an edge to $(i{+}1, j{+}1)$ corresponds either to a no-op (with weight 0) if $a_i = b_j$, or to a substitution of a_i by b_j (with weight 1) otherwise.

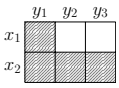
▶ **Example 5.8.** The edit graph of $x = abbabba$ and $y = abbbababc$ is given in Figure 1 (left). Edges that weigh 0 are dashed whereas edges that weigh 1 are solid. A minimum cost-path is marked on the graph in black. It corresponds to the edit path $\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\\varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\c\end{smallmatrix}\right]$. It has 10 edges, out of which 4 weigh 1 so its cost is $\frac{4}{10}$. Accordingly $\mathsf{ned}(x, y) = \frac{4}{10}$.

Blocks. Note that words in X (resp. Y) are parameterized by vectors in $\mathbb{N}^n$ (resp. $\mathbb{N}^m$). Let $\mathbf{n} = (n_1, \dots, n_n) \in \mathbb{N}^n$ and $\mathbf{m} = (m_1, \dots, m_m) \in \mathbb{N}^m$. These vectors induce the words $w^{\mathrm{X}}_{\mathbf{n}} = x_1^{n_1} \cdots x_n^{n_n}$ and $w^{\mathrm{Y}}_{\mathbf{m}} = y_1^{m_1} \cdots y_m^{m_m}$, resp. We denote their lengths by $\ell^{\mathrm{X}}_{\mathbf{n}} = |w^{\mathrm{X}}_{\mathbf{n}}|$ and $\ell^{\mathrm{Y}}_{\mathbf{m}} = |w^{\mathrm{Y}}_{\mathbf{m}}|$. Consider now $x = w^{\mathrm{X}}_{\mathbf{n}}$ and $y = w^{\mathrm{Y}}_{\mathbf{m}}$. The edit graph of x and y can be partitioned into $n \times m$ rectangular subgrids, which we henceforth call *blocks*. The (i, j)-block corresponds to the edit graph of $x_i^{n_i}$ and $y_j^{m_j}$.

▶ **Example 5.9.** Let $X = \{(a)^n(bba)^{2n} \mid n \in \mathbb{N}\}$ and $Y = \{(abbb)^k(ab)^{2m}(c)^m \mid k, m \in \mathbb{N}\}$. Let $\mathbf{n} = (1, 2)$ and $\mathbf{m} = (1, 2, 1)$ Figure 1 (middle) shows the edit graph of $w^{\mathrm{X}}_{\mathbf{n}} = a \cdot bba \cdot bba$ and $w^{\mathrm{Y}}_{\mathbf{m}} = aabb \cdot ab \cdot ab \cdot c$ partitioned into its six blocks by the bold gray lines.

Interleavings. Let p be an edit path from x to y. We consider the $n \times m$ grid, and say that a cell (i, j) is *lit* if the edit path p passes through the (i, j)-block. We write $\pi(p)$ for the set of cells lit by p, and refer to $\pi(p)$ as the *interleaving* imposed by p. Observe that $\pi(p)$ forms a monotone path from $(1, 1)$ to (n, m) in the $n \times m$ grid. We use Π to denote the set of all interleavings from $(1, 1)$ to (n, m).

▶ **Example 5.10.** Continuing Ex.5.9, the 2×3 grid below shows that the marked edit path passes through blocks $(1,1)$, $(2,1)$, $(2,2)$, and $(2,3)$; the cells corresponding to these blocks are diagonally hatched.



▶ Remark 5.11. Note that $|\Pi| = \binom{n+m-2}{n-1}$, since any such interleaving consists of a total of $n-1$ rightward steps and $m-1$ downward steps, and is therefore determined by the choice of which $n-1$ of the $n+m-2$ steps are rightward.

Sub-blocks and the cyclic-edit graph. In our linear program, we want to reason simultaneously about many words $x \in X$ and $y \in Y$. To this end, we observe that each (i,j)-block is composed of $n_i \times m_j$ identical copies of the edit graph of x_i and y_j, which we call *sub-blocks*. Figure 1 (right) illustrates the partition of the $(2,2)$-block into four identical sub-blocks corresponding to the edit graph of x_2 to y_2.

To reason uniformly about powers of x_i and y_j, we introduce the *cyclic edit graph* of x_i and y_j, denoted $G^{\circlearrowleft}_{i,j}$. Let $x_i = a_1 \dots a_{|x_i|}$ and $y_j = b_1 \dots b_{|y_j|}$. The vertices of $G^{\circlearrowleft}_{i,j}$ form a quotient of the set $V = \{(l,k) \mid 0 \le l \le |x_i|,\ 0 \le k \le |y_j|\}$ where $(0,k)$ is identified with $(|x_i|,k)$, and $(l,0)$ is identified with $(l,|y_j|)$.

These identifications reflect the adjacency of sub-blocks: for instance, the vertex $(0,k)$ of a given sub-block (e.g., the orange block in Figure 1 (right)) coincides with $(|x_i|,k)$ of the sub-block to its left (the cyan one), and analogously for the vertical direction.

As a result, there is a bijection between edit paths from $x_{i_{\text{SUFF}}} {x_i}^* x_{i_{\text{PREF}}}$ to $y_{j_{\text{SUFF}}} {y_j}^* y_{j_{\text{PREF}}}$ and walks in $G^{\circlearrowleft}_{i,j}$, where $x_{i_{\text{SUFF}}}$ and $y_{j_{\text{SUFF}}}$ are suffixes and $x_{i_{\text{PREF}}}$ and $y_{j_{\text{PREF}}}$ are prefixes of x_i and y_j, respectively. This bijection preserves both weight and length. Moreover, every walk starting at $(0,0)$ corresponds to an edit path from some word in $x_i^* x_{i_{\text{PREF}}}$ to some word in $y_j^* y_{j_{\text{PREF}}}$, and similarly for walks ending at $(0,0)$ and words in $x_{i_{\text{SUFF}}} x_i^*$ and $y_{j_{\text{SUFF}}} y_j^*$.

▶ **Example 5.12.** Consider the edit path in the $(2,2)$-block of Figure 1 (right). It induces the edit path $\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\b\end{smallmatrix}\right]\left[\begin{smallmatrix}b\\\varepsilon\end{smallmatrix}\right]\left[\begin{smallmatrix}a\\a\end{smallmatrix}\right]\left[\begin{smallmatrix}\varepsilon\\b\end{smallmatrix}\right]$ which corresponds to the walk $(0,2) \to (1,3) \equiv (1,0) \to (2,1) \to (2,2) \equiv (0,2) \to (1,3) \to (2,3)$ in $G^{\circlearrowleft}_{2,2}$. This walk represents the alignment of the words $a \cdot bba$ and $ab \cdot ab$, that is, ax_2 and y_2y_2, respectively, where a is indeed a suffix of x_2.

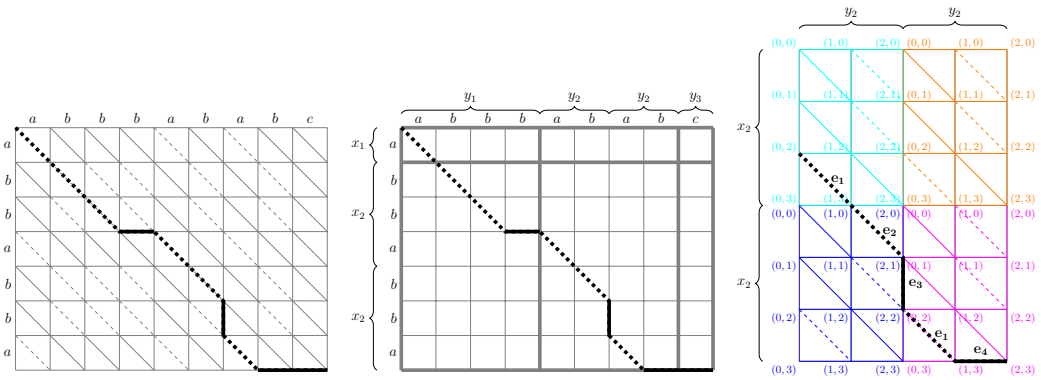


Figure 1 Left: $G_{x,y}$, **Middle:** the blocks of $G_{x,y}$, **Right:** the sub-blocks of the $(2,2)$-block.

Directions. Another means to simultaneously reason on many $x \in X$ and $y \in Y$ is the notion of *direction*. Consider again $\mathbf{n} = (n_1, \ldots, n_n) \in \mathbb{N}^n$, $w^{\mathrm{X}}_{\mathbf{n}} = x_1^{n_1} \cdots x_n^{n_n}$, and $\ell^{\mathrm{X}}_{\mathbf{n}} = |w^{\mathrm{X}}_{\mathbf{n}}|$. Note that $\ell^{\mathrm{X}}_{\mathbf{n}} = \sum_{i=1}^{n} n_i \cdot |x_i|$. The *direction* induced by $\mathbf{n}$ is the vector $\mathrm{dir}(\mathbf{n}) = (d_1, \ldots, d_n)$, where $d_i = n_i / \ell^{\mathrm{X}}_{\mathbf{n}}$.

▶ **Example 5.13.** For instance, for $X = \{(a)^n (bba)^{2n} \mid n \in \mathbb{N}\}$, $\mathbf{n} = (1, 2)$, we get that $w^{\mathrm{X}}_{\mathbf{n}} = abbabba$, $\ell^{\mathrm{X}}_{\mathbf{n}} = 7$ and $\mathrm{dir}(\mathbf{n}) = (\frac{1}{7}, \frac{2}{7})$.

For every $\mathbf{n} \in \mathbb{N}^n$, the induced direction $\mathrm{dir}(\mathbf{n})$ lies in the set $\Delta_{\mathrm{X}} \stackrel{\text{def}}{=} \{\mathbf{d} \in \mathbb{R}^n_{\geq 0} : \sum_{i=1}^{n} \mathbf{d}[i] \cdot |x_i| = 1\}$. Note that each direction $\mathbf{a} \in \Delta_{\mathrm{X}}$ is induced by infinitely many vectors in $\mathbb{N}^n$: if $\mathbf{n}$ induces $\mathbf{a}$, then so does $c\mathbf{n}$ for any $c \in \mathbb{N}$.

Not every vector $\mathbf{n} \in \mathbb{N}^n$ induces a word $w^{\mathrm{X}}_{\mathbf{n}}$ that belongs to X. We therefore restrict attention to directions that are realizable by words in X, at least asymptotically. Let $S_{\mathrm{X}} \subseteq \mathbb{N}^n$ denote the Parikh image of X with respect to $x_1, \ldots, x_n$. We define the *direction set* of X as $D_{\mathrm{X}} \stackrel{\text{def}}{=} \overline{\{\mathrm{dir}(\mathbf{n}) \mid \mathbf{n} \in S_{\mathrm{X}}\}} \subseteq \Delta_{\mathrm{X}}$, where the closure is taken in the standard topology on $\mathbb{R}^n$. Since S_{X} is a linear set, D_{X} is a polytope.

The Linear Program

Fix a direction $\mathbf{a} \in D_{\mathrm{X}}$ and an interleaving $\pi \in \Pi$. Our goal is to describe an optimal edit path from a word in X inducing the direction $\mathbf{a}$ to a word in Y, under the restriction that the path respects the interleaving π. We use a linear program to solve the infimum for any $\mathbf{b} \in D_{\mathrm{Y}}$ and $\lambda \in (0, 1]$, where $\mathbf{b}$ is the "closest" direction to $\mathbf{a}$ in S_{Y} and if p is a corresponding edit path then λ represents the fraction $\frac{|x|}{|p|}$. To this aim we define a linear program over the following set of variables:

- $f_e^{i,j} \geq 0$ for each $(i, j) \in \pi$ and $e \in E(G^{\circlearrowleft}_{i,j})$
- $\tau_t \geq 0$ for each $1 \leq t \leq \ell$
- $\lambda \in (0, 1]$

Intuitively the variable $f_e^{i,j}$ holds the fraction corresponding to the number of times edge e is used in $G^{\circlearrowleft}_{i,j}$ along π divided by the length of the edit path p. Thus, the sum overall these variables should be 1, and this is Constraint 1 below. Working with edit paths whose length is normalized to 1 simplifies the reasoning for ned: under this condition the cost of the path is the same as its weight.

Not every assignment to the variables $f_e^{i,j}$ corresponds to a valid edit path. In a path, for every vertex (except for the source and target) every visited vertex has a corresponding entry and exit edges. This gives rise to Constraint 2 below.[7]

We further require that the resulting edit path corresponds to a word in X with direction $\mathbf{a}$, and transforms it into an optimal word in Y. We use the indicators $\mathbb{1}_{x_i}(e)$ and $\mathbb{1}_{y_j}(e)$ where $\mathbb{1}_{x_i}(e)$ equals 1 if the edge e consumes a letter of x_i – that is, if it corresponds to a substitution, deletion, or no-op – and equals 0 otherwise (and analogously for $\mathbb{1}_{y_j}(e)$). Constraint 3 ensures that the total use of edit operations reading symbols from the source block x_i matches the contribution prescribed by the direction $\mathbf{a} \in D_{\mathrm{X}}$. Recall that parameter λ corresponds to the ratio $\frac{|x|}{|p|}$. It is introduced to account for the fact that the expression $\sum_{(i,j)\in\Pi,\ e\in E(G^{\circlearrowleft}_{i,j})} f_e^{i,j}\, \mathbb{1}_{x_i}(e)$ naturally yields $n_i|x_i|$ normalized by $|p|$, whereas the direction vector $\mathbf{a} \in D_{\mathrm{X}}$ is defined in terms of normalization by $|x|$. Multiplication by λ therefore corrects the denominator, aligning the normalization with that of the source direction.

[7] The discrepancy for the source and target is taken care of in the proof details.

For the target word, we do not fix a direction in advance. Instead, we allow the linear program to choose an optimal target direction. This is achieved using the variables $\tau_1, \ldots, \tau_r$, which induce a normalized direction vector $\mathbf{b}$ for Y via $b_j = \sum_{s=1}^{r} \tau^s \cdot \mathbf{v}_s[j]$ for each $1 \leq j \leq m$. Intuitively, the variables $\tau_1, \ldots, \tau_r$ normalize by $|p|$ a natural combination of the generators $\mathbf{v}_1, \ldots, \mathbf{v}_r$ of Y. Accordingly, b_j represents the $|p|$-normalized contribution of the target block y_j. Constraint 4 ensures that the edit path consumes symbols from the target blocks in accordance with this induced direction.

Constraints. The constraints of the linear program $\mathrm{LP}_{\mathbf{a},\pi}$ are as follows:

1. *Normalization constraint*:

$$\sum_{(i,j)\in\pi} \sum_{e\in E(G^{\circlearrowleft}_{i,j})} f_e^{i,j} = 1.$$

2. *Flow conservation in each block*: for every $(i,j) \in \pi$ and every vertex $v \in V(G^{\circlearrowleft}_{i,j})$

$$\sum_{e\in \mathrm{Out}(v)} f_e^{i,j} = \sum_{e\in \mathrm{In}(v)} f_e^{i,j}.$$

3. *Source-consumption constraints*: for all $i \in \{1, \ldots, n\}$

$$\sum_{(i,j)\in\pi} \sum_{e\in E(G^{\circlearrowleft}_{i,j})} f_e^{i,j}\, \mathbb{1}_{x_i}(e) = |x_i| \cdot a_i \cdot \lambda.$$

4. *Target-consumption constraints*: for all $j \in \{1, \ldots, m\}$

$$\sum_{(i,j)\in\pi} \sum_{e\in E(G^{\circlearrowleft}_{i,j})} f_e^{i,j}\, \mathbb{1}_{y_j}(e) = |y_j| \cdot b_j$$

Objective. $$\min \sum_{(i,j)\in\pi} \sum_{e\in E(G^{\circlearrowleft}_{i,j})} f_e^{i,j}\, c(e).$$

where $c(e)$ is the weight of e. The objective thus looks for an edit path with minimum cost.

For a fixed direction $\mathbf{a} \in D_{\mathrm{X}}$, each interleaving $\pi \in \Pi$ induces a linear program $\mathrm{LP}_{\mathbf{a},\pi}$. We denote by $\mathrm{OPT}_{\mathbf{a},\pi}$ the optimal value of $\mathrm{LP}_{\mathbf{a},\pi}$. We aggregate these by taking the best value over all interleavings, and define $F(\mathbf{a}) \stackrel{\text{def}}{=} \min_{\pi\in\Pi} \mathrm{OPT}_{\mathbf{a},\pi}$. Continuity of F is used in the proof of the following theorem.

▷ Claim 5.14. The function $F : D_{\mathrm{X}} \to \mathbb{R}$ is continuous.

We can finally state the main theorem for this section.

▶ **Theorem 5.15.** *Let X and Y be bounded CFLs, and let Π and D_X be the corresponding interleaving and directions set. Then* $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y) = \sup_{\mathbf{a}\in D_X} \min_{\pi\in\Pi} \mathrm{OPT}_{\mathbf{a},\pi} = \sup_{\mathbf{a}\in D_X} F(\mathbf{a})$.

Putting the above ingredients together, we obtain the following complexity bound for computing $\mathbb{AH}_{\mathsf{ned}}$ on bounded context-free languages.

▶ **Lemma 5.16.** *There exists an Exp algorithm that computes $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y)$ for bounded context-free languages X and Y.*

Proof sketch. By the LP characterization, $\mathbb{AH}^{\triangleright}_{\mathsf{ned}}(X,Y) = \sup_{\mathbf{a}\in D_X} F(\mathbf{a})$, where $F(\mathbf{a}) = \min_{\pi\in\Pi} \mathrm{OPT}_{\mathbf{a},\pi}$ and $\mathrm{OPT}_{\mathbf{a},\pi}$ is the optimum of a linear program whose constraints are independent of $\mathbf{a}$ and whose right-hand side depends affinely on it. By LP duality, for each π, $\mathrm{OPT}_{\mathbf{a},\pi}$ is a convex, piecewise-linear function of $\mathbf{a}$. Hence, F is also piecewise linear.

Since D_X is a compact polytope, the supremum of F is attained at a vertex of the arrangement induced by these linear pieces. The number of defining hyperplanes is exponential, as it depends on the number of interleavings and dual vertices. Standard bounds on hyperplane arrangements imply that the number of candidate vertices is exponential.

The algorithm enumerates these candidates, evaluates F at each by solving the corresponding linear programs, and returns the maximum. This yields an exponential algorithm. ◀

6 Discussion

The Asymptotic Hausdorff lifting captures a notion of similarity that is inherently asymptotic, comparing infinite sets by their long-run behavior while deliberately abstracting away from finite deviations. This choice is intentional: in formal verification, two languages are considered similar when similarity is witnessed on increasingly long words. In particular, in applications such as repair, robustness, and automata learning, asymptotic behavior with respect to a normalized edit distance between words reflects the semantic core of the compared languages.

Our results highlight an inherent trade-off between robustness and sensitivity. Classical Hausdorff-style constructions are highly sensitive to outliers, whereas asymptotic notions necessarily collapse certain local distinctions. The Asymptotic Hausdorff lifting makes this trade-off explicit and allows additional sensitivity to be reintroduced in a controlled manner by combining it with complementary distances.

Although formal languages serve as a central motivating domain, the lifting scheme itself is more general and applies to any infinite domain equipped with a meaningful notion of size and an element-level metric.

From an algorithmic perspective, our results indicate that computing $\mathbb{AH}_{\mathsf{ned}}$ is already challenging for regular and bounded context-free languages, and extending the proposed techniques to richer classes or more efficient exact algorithms is an interesting direction for future work.

References

1 Michael Benedikt, Gabriele Puppis, and Cristian Riveros. The cost of traveling between languages. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, volume 6756 of *Lecture Notes in Computer Science*, pages 234–245. Springer, 2011. doi:10.1007/978-3-642-22012-8_18.

2 Michael Benedikt, Gabriele Puppis, and Cristian Riveros. Regular repair of specifications. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science, LICS 2011, June 21-24, 2011, Toronto, Ontario, Canada*, pages 335–344. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.43.

3 Michael Benedikt, Gabriele Puppis, and Cristian Riveros. Bounded repairability of word languages. *J. Comput. Syst. Sci.*, 79(8):1302–1321, 2013. doi:10.1016/J.JCSS.2013.06.001.

4 Michael Benedikt, Gabriele Puppis, and Cristian Riveros. The per-character cost of repairing word languages. *Theoretical Computer Science*, 539:38–67, 2014. doi:10.1016/j.tcs.2014.04.021.

5 Florian Bruse, Maurice Herwig, and Martin Lange. A similarity measure for formal languages based on convergent geometric series. In *Implementation and Application of Automata: 26th International Conference, CIAA 2022, Rouen, France, June 28 – July 1, 2022, Proceedings*, Berlin, Heidelberg, 2022. Springer-Verlag. doi:10.1007/978-3-031-07469-1_6.

6 Noam Chomsky and George A. Miller. Finite state languages. *Inf. Control.*, 1(2):91–112, 1958. doi:10.1016/S0019-9958(58)90082-2.

7 Cewei Cui, Zhe Dang, Thomas R. Fischer, and Oscar H. Ibarra. Similarity in languages and programs. *Theoretical Computer Science*, 498:58–75, 2013. doi:10.1016/j.tcs.2013.05.040.

8 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. doi:10.1109/ICDEW.2008.4498345.

9 Emmanuel Filiot, Nicolas Mazzocchi, Jean-François Raskin, Sriram Sankaranarayanan, and Ashutosh Trivedi. Weighted transducers for robustness verification. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, Vienna, Austria (Virtual Conference), September 1-4, 2020*, volume 171 of *LIPIcs*, pages 17:1–17:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.17.

10 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, June 27-29, 2022, Prague, Czech Republic*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CPM.2022.17.

11 Dana Fisman and Ilay Tzarfati. When is the normalized edit distance over non-uniform weights a metric? In Shunsuke Inenaga and Simon J. Puglisi, editors, *35th Annual Symposium on Combinatorial Pattern Matching, CPM 2024, Fukuoka, Japan, June 25-27, 2024*, volume 296 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CPM.2024.14.

12 Robert C. Flagg. Quantales and continuity spaces. *Algebra Universalis*, 37(3):257–276, 1997.

13 Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

14 Thomas A. Henzinger, Jan Otop, and Roopsha Samanta. Lipschitz robustness of finite-state transducers. In Venkatesh Raman and S. P. Suresh, editors, *34th International Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2014, New Delhi, India, December 15-17, 2014*, volume 29 of *LIPIcs*, pages 431–443. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2014. doi:10.4230/LIPIcs.FSTTCS.2014.431.

15 Manfred Kudlek and Benedek Nagy. Distances of formal languages. *PU.M.A. Pure Mathematics and Applications*, 17, January 2006.

16 F. William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43(1):135–166, 1973.

17 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.

18 A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):926–932, 1993. doi:10.1109/34.232078.

19 Mehryar Mohri. Edit-distance of weighted automata. In *Proceedings of the 7th International Conference on Implementation and Application of Automata*, CIAA'02, pages 1–23, Berlin, Heidelberg, 2002. Springer-Verlag. doi:10.1007/3-540-44977-9_1.

20 Timothy Ng, David Rappaport, and Kai Salomaa. Relative prefix distance between languages. In *International Conference on Developments in Language Theory*, pages 284–295. Springer, 2017. doi:10.1007/978-3-319-62809-7_21.

21 Rohit J. Parikh. On context-free languages. *Journal of the Association for Computing Machinery*, 13(4):570–581, 1966. doi:10.1145/321356.321364.

22 Austin J. Parker, Kelly B. Yancey, and Matthew P. Yancey. Regular language distance and entropy, 2016. arXiv:1602.07715.

23 Roopsha Samanta, Jyotirmoy V. Deshmukh, and Swarat Chaudhuri. Robustness analysis of string transducers. In Dang Van Hung and Mizuhito Ogawa, editors, *Automated Technology for Verification and Analysis*, pages 427–441, Cham, 2013. Springer International Publishing. doi:10.1007/978-3-319-02444-8_30.

24 C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001. doi:10.1145/584091.584093.

25 Robert A. Wagner. Order-n correction for regular languages. *Commun. ACM*, 17(5):265–268, 1974. doi:10.1145/360980.360995.

26 Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, 2007. doi:10.1109/TPAMI.2007.1078.