Electronic Colloquium on Computational Complexity, Report No. 96 (2004)

Property and Equivalence Testing on Strings*

Eldar Fischer† Frédéric Magniez‡ Michel de Rougemont§

Abstract

Using a new statistical embedding of words which has similarities with the Parikh mapping, we first construct a tolerant tester for the equality of two words, whose complexity is independent of the string size, where the distance between inputs is measured by the normalized edit distance with moves. As a consequence we get an approximation algorithm for the normalized distance, the first such algorithm whose complexity does not depend on the string size.

Then we extend our embedding to languages, and get a geometrical approximate description of regular languages by finite unions of polytopes. As an application, we have a new tester for regular languages whose complexity does not depend on the automaton. The automaton is only required in a preprocessing step, whose time is polynomial in the automaton size for a fixed threshold distance. The remaining complexity is a constant depending on the threshhold distance but not on the automaton.

Last, we introduce the notion of equivalence testing. Using the above geometrical description, we exhibit an equivalence tester for regular languages. The tester is deterministic and of polynomial time, for a fixed threshold distance. In contrast, the problem of deciding the exact equivalence of finite automata requires exponential space.

1 Introduction

We consider the approximation of several classical combinatorial problems on strings in the context of Property Testing. Inspired by the notion of Self-Testing [4, 5, 17], Property Testing has been initially defined and studied for graph properties [9]. It has been successfully extended for various classes of finite structures. Let $\mathbf{K}$ be a class of finite structures together with a distance. An ε*-tester* for a class $\mathbf{K}_0 \subseteq \mathbf{K}$ is a randomized algorithm which takes a structure $U_n \in \mathbf{K}$ of size n as an input and decides with high probability if $U_n \in \mathbf{K}_0$ or if U_n is ε-far from $\mathbf{K}_0$. A class $\mathbf{K}_0$ is testable if for every $\varepsilon > 0$ there exists an ε-tester for $\mathbf{K}_0$ whose time complexity depends only on ε, *i.e.* is independent of the size n.

Property Testing of regular languages was first considered in [1] for the *Hamming distance* and then extended to branching programs [13], where the Hamming distance between two words is the minimal number of character substitutions required to transform one word into the other. Then two words of size n are ε*-far* if they are at distance greater than εn.

The *edit distance* is a more relaxed natural distance measure for strings than the Hamming distance. The edit distance between two words is the minimal number of insertions, deletions and substitutions required to transform one word into the other. Computing the edit distance between two words is an important subproblem of many applications like text processing, genomics, web search, etc. Behind the difficulty to get a significant subquadratic time algorithm (the best known algorithm is in $O(n^2/\log n)$ [11]), several approximation algorithms have been proposed. Nevertheless, to get a linear or sublinear time algorithm, one has to consider more drastic relaxations. One can distinguish two possible approaches. The first one is a weak version of approximation based on Property Testing, and the second one considers an extension of the distance: the *edit distance with moves*, where arbitrary substrings can be moved in one step.

Concerning the Property Testing approach, in [3] a sublinear tester is constructed such that it accepts with high probability pairs of words of size n that are at edit distance $O(n^\alpha)$, for some $0 < \alpha < 1$, and it rejects with high

*Work supported in part by *ACI Sécurité Informatique: VERA* of the French Ministry of research.

†Faculty of Computer Science, Technion – Israel institute of technology, Technion City, Haifa 32000, Israel, eldar@cs.technion.ac.il
Research supported in part by an Israel Science Foundation grant number 55/03, and by a grant from the Matilde Barnett Revocable Trust.

‡CNRS–LRI, UMR 8623 Universit´e Paris–Sud, France, magniez@lri.fr

§LRI & Universit´e Paris II, France, mdr@lri.fr

ISSN 1433-8092

probability pair of words at distance $\Omega(n)$. The running time is in $\tilde{O}(n^{\max(\alpha/2, 2\alpha-1)})$. These testers can be understood as a *weak approximation* of the edit distance, since it leads to efficient approximation algorithms whenever the distance is large [15]. Nonetheless no sublinear time testers exist for the case $\alpha = 1$. Moreover, there is no hope to get a tester whose running time is size independent (even for $0 < \alpha < 1$) since a lower bound $\Omega(n^{\alpha/2})$ on the query complexity has been proven [3]. Results of same kind were proven in the sketching model [2], where a sketch (or a fingerprint) is associated to each string, which is succinct yet rich enough to approximate the edit distance. Nonetheless the complexity of computing a sketch is usually not sublinear. For instance, a near linear time algorithm is constructed to distinguish between strings at distance $O(k)$ from the ones at distance $\Omega(k^2)$, for any $k \geq 1$.

The edit distance with moves has been considered and approximated efficiently in [7, 6]. This is one of many interesting variations of the edit distance that has many applications, for instance in genomics. Whereas computing the edit distance with moves is NP-hard [18], it can be approximated within a $\tilde{O}(\log n)$ factor under near linear time [7]. If one allows other operations in the distance such as copy of subwords, then there is a linear time and constant approximation algorithm [8, 19]. In the context of property testing, the edit distance with moves has been used in [10] for testing regular languages, where the tester is more efficient and simpler than the one of [1], and can be generalized to tree regular languages.

In this paper, we develop a new statistical embedding of words which has similarities with the Parikh mapping [14]. Based on this embedding, we develop a tester (**Theorem 3.8**) for the equality between two words whose complexity is $O(\frac{\log|\Sigma|}{\varepsilon^4})$, where $|\Sigma|$ is the alphabet size, which is also *tolerant*, that is it is not only an ε-tester, but it also accepts with high probability words that are ε^2-close. This notion of tolerance, initially present in Self-Testing, was firstly not considered in Property Testing. Recently, coming back to this notion, a relation between tolerant property testing and approximation was pointed out in [15]. Based on this observation and our tolerant tester, we directly get an approximation algorithm for the normalized distance ε between two words (**Corollary 3.9**), whose complexity is $O(\frac{\log(|\Sigma|/\varepsilon)}{\varepsilon^4})$. To our knowledge this is the first approximation algorithm whose complexity is size independent. It is interesting to note that the edit distance without moves, that lies between the Hamming distance (for which there is a trivial tolerant tester) and the edit distance with moves (for which we prove the existence of a tolerant tester), is in itself hard for tolerant testing (recall the lower bound above from [3] with $\alpha = 1$).

Then we extend our embedding to languages. This leads us to an approximate geometrical description of regular languages by finite unions of polytopes, which is robust (**Theorem 4.7**). Discretizing this representation gives us (**Theorem 4.10**) a new tester for regular languages whose query complexity is $O(\frac{\log|\Sigma|}{\varepsilon^4})$ and time complexity is $2^{|\Sigma|^{O(1/\varepsilon)}}$. Whereas the complexity of previous testers for regular languages depended (exponentially) on the number of states of the corresponding automaton, here the automaton is only used in a preprocessing step to build the tester. The tester construction requires time $m^{|\Sigma|^{O(1/\varepsilon)}}$, where m is a bound on the number of states of the automaton.

Last, we introduce the notion of an equivalence tester between classes of structures. Intuitively, two classes $\mathbf{K}_1, \mathbf{K}_2$ are ε-equivalent if every but finitely many structures of $\mathbf{K}_1$ is ε-close to $\mathbf{K}_2$ and conversely. An *equivalence ε-tester* accepts equivalent classes and rejects classes which are not ε-equivalent with high probability. Using the previous discretization, we construct an equivalence ε-tester for regular languages (**Theorem 4.11**) (where the exact decision version of this problem requires exponential space [12]) in deterministic time $m^{|\Sigma|^{O(1/\varepsilon)}}$, where m is an upper bound of the number of states of the corresponding automata.

2 Preliminaries

2.1 Words, Languages, and Distance

We fix a finite alphabet Σ, a positive integer k and $\varepsilon = \frac{1}{k}$. We call Σ^k the *block alphabet* and its elements the *block letters*. Any word of w of size n over Σ is also a word over Σ^k where the last $(n - k\lfloor\frac{n}{k}\rfloor)$ letters are deleted. To simplify notation, we always assume that k divides the size of the considered words. Thus, the words and the languages can be considered over both Σ and Σ^k. For a word w over Σ we denote by $|w|$ its size on Σ and by $|w|_b$ its size on Σ^k. Note that these quantities satisfy $|w| = \frac{1}{\varepsilon} \times |w|_b = k \times |w|_b$. We also denote by $w[i]$ the i-th letter of w, for $i \geq 1$, and by $w[j]_b$ the j-th block letter of w, *i.e.* the subword $w[(j-1)k+1]w[(j-1)k+2]\ldots w[jk]$ of w, for $j \geq 1$.

A *subword* of a word w is a sequence of consecutive letters of w. An *elementary operation* on a word w over Σ is either an insertion of a letter, a deletion of a letter, a substitution of a letter by another one, or the move of a subword of w into another position of w. The *edit distance with moves* $\mathsf{dist}(w, w')$ between two words w, w' over Σ is the minimal number of such elementary operations on w to obtain w'.

Define the *block Parikh equivalence* from the Parikh mapping [14] of a word on the block alphabet. Namely, let w and w' be two words of same size, then $w \equiv_k w'$ iff w' is obtained by a permutation of the block letters of w. The block Parikh equivalence preserves the distance in the following way.

Proposition 2.1. *Let w, w' be two words of size n. If $w \equiv_k w'$ then $\mathsf{dist}(w, w') \leq \varepsilon n$.*

Proof. Using the definition of the block Parikh equivalence, one can permute the block letters of w to reach w'. Such a permutation can be decomposed into at most $|w|_b = \varepsilon n$ move operations. ☐

Note that if k does not divide n, then we just need to add $k = \frac{1}{\varepsilon}$ to the previous upper bound, due to the last $(n - k\lfloor\frac{n}{k}\rfloor)$ remaining letters of w and w' that we may have to modify.

For two real vectors V, V' of dimension d, we denote by $|V - V'|$ the ℓ_1-distance between V and V', that is $|V - V'| = \sum_{i=1}^{d} |V[i] - V'[i]|$, where $V[i]$ (resp.$V'[i]$) denotes the i-th coordinate of V (resp. of V'). If the vectors V, V' denote probability distributions, then the ℓ_1-distance coincides with twice the *total variation distance*.

2.2 Property Testing

Recall the notion of Property Testing [9] on a class $\mathbf{K}$ of finite structures for which a distance function between structures has been defined. We say that two structures $U_n, V_m \in \mathbf{K}$, whose domains are respectively of size n and m, are *ε-close* if their distance is less than $\varepsilon \times \max(n, m)$. They are *$\varepsilon$-far* if they are not ε-close. The domain size is appropriate to structures such as words or trees. But for classes such as graphs, one may define the closeness relatively to the representation size (e.g., εn^2 for graphs) instead of the domain size.

Every word w over a finite alphabet Σ is a finite structure $(n, [n], l : [n] \to \Sigma)$, where $[n]$ denote the set $\{1, \ldots, n\}$. The class $\mathbf{K}$ is the set of all such structures. We will denote a subclass $\mathbf{K}_0$ of $\mathbf{K}$ as a subset $L \subseteq \Sigma^*$. In this context, a query i to some word w asks the letter $w[i] = l(i)$.

Definition 2.2 (Tester). *Let $\varepsilon > 0$ be a real number. An ε-*tester *for a class $\mathbf{K}_0 \subseteq \mathbf{K}$ is a randomized algorithm A such that, for any $U \in \mathbf{K}$ as input:*

(1) If $U \in \mathbf{K}_0$, then A accepts with probability at least $2/3$;

(2) If U is ε-far from $\mathbf{K}_0$, then A rejects with probability at least $2/3$.

If in addition the algorithm is guaranteed to always accept if $U \in \mathbf{K}_0$, then we call it a one-sided error *ε-tester.*

Definition 2.3 (Tolerant tester [15]). *Let $0 < \varepsilon_1 < \varepsilon_2$ be reals. A* tolerant $(\varepsilon_1, \varepsilon_2)$-tester *for a class $\mathbf{K}_0 \subseteq \mathbf{K}$ is a randomized algorithm A such that, for any $U \in \mathbf{K}$ as input:*

(1) If U is ε_1-close to $\mathbf{K}_0$, then A accepts with probability at least $2/3$;

(2) If U is ε_2-far from $\mathbf{K}_0$, then A rejects with probability at least $2/3$.

In [15], it was shown how to derive an approximation algorithm when a family of tolerant testers is given. We will adapt their construction for our particular family of tolerant testers.

Definition 2.4 (Approximation). *Let $\alpha, \beta : \mathbb{R} \to \mathbb{R}$. An* (α, β)-approximation *of a real function f is a randomized algorithm that on input $U \in \mathbf{K}$, outputs a value z such that $\Pr[\alpha(f(U)) \leq z \leq \beta(f(U))] \geq 2/3$.*

The *query complexity* is the number of queries made to the structure U of $\mathbf{K}$. The *time complexity* is per the usual definition, where we assume that the following operations are performed in constant time: arithmetic operations, a uniform random choice of an integer from a range given by the algorithm, and making a query to the input. A class $\mathbf{K}_0 \subseteq \mathbf{K}$ is *testable* if for every $\varepsilon > 0$, there exists an ε-tester whose time complexity depends only on ε.

In this paper, we introduce the new notion of *equivalence testing* for whole languages. For this, we first define the notion of ε-equivalence.

Definition 2.5. *Let $\varepsilon \geq 0$. Let $\mathbf{K_1}, \mathbf{K_2} \subseteq \mathbf{K}$ be two classes.*

*$\mathbf{K_1}$ is ε-*contained *in $\mathbf{K_2}$, if every but finitely many structures of $\mathbf{K_1}$ are ε-close to $\mathbf{K_2}$.*

*$\mathbf{K_1}$ is ε-*equivalent *to $\mathbf{K_2}$, if both $\mathbf{K_1}$ is ε-contained in $\mathbf{K_2}$ and $\mathbf{K_2}$ is ε-contained in $\mathbf{K_1}$.*

Definition 2.6 (Equivalence tester). *Let $\varepsilon > 0$, and let $\mathcal{R}$ be a family of finite representations of classes. A deterministic (resp. probabilistic)* ε-equivalence tester *for $\mathcal{R}$ is a deterministic (resp. probabilistic) algorithm A such that, given as input representations R_1, R_2 from $\mathcal{R}$ of classes $\mathbf{K_1}, \mathbf{K_2}$:*

(1) If $\mathbf{K}_1 = \mathbf{K}_2$, then A accepts (resp. with probability at least $2/3$);

(2) If $\mathbf{K}_1$ and $\mathbf{K}_2$ are not ε-equivalent, then A rejects (resp. with probability at least $2/3$).

3 Approximating the Edit Distance with Moves

We will first study useful properties, like robustness, of our first statistics, the block statistics. Then we will extend the robustness to the uniform statistics, which have the advantage of being also sound. These two properties will directly give us a tolerant tester, based on the uniform statistics and their ℓ_1-distances. We then obtain an approximation algorithm for the normalized edit distance with moves.

3.1 Block Statistics

In this section, w and w' are two words of size n over Σ, such that k divides n. Let $\varepsilon = \frac{1}{k}$. We denote the statistics of the block letters of w by $\mathsf{b\text{-}stat}(w)$, that is the vector of dimension $|\Sigma|^k$ such that its u-coordinate, for $u \in \Sigma^k$, satisfies

$$\mathsf{b\text{-}stat}(w)[u] \stackrel{\text{def}}{=} \Pr_{j=1,\dots,n/k}[w[j]_b = u].$$

We call the vector $\mathsf{b\text{-}stat}(w)$ the *block statistics* of w.

Another equivalent and sometimes more convenient way to define the block statistics is to use the underlying distribution on words over Σ of size k, that is on block letters of Σ^k. We call the uniform distribution on block letters $w[1]_b, \dots, w[\frac{n}{k}]_b$ of w (with some possible repetitions), the *block distribution* of w. Let X be the random vector of size $|\Sigma|^k$ where all coordinates are 0 except its u-coordinate which is 1, where u is the index corresponding to a random word of size k that was chosen according to the block distribution of w. Then the expectation of X satisfies $\mathrm{E}(X) = \mathsf{b\text{-}stat}(w)$.

Last, note that $\mathsf{b\text{-}stat}(w)$ is related to the Parikh mapping [14] of w where we compute the probabilities of a block letter to occur instead of the number of occurrences of a letter. In particular, $w \equiv_k w'$ iff $\mathsf{b\text{-}stat}(w) = \mathsf{b\text{-}stat}(w')$, when w and w' have same size.

We relate the distance between two words to the ℓ_1-distance of their respective block statistics. This establishes the robustness of the block statistics construction (the notion of robustness was initially introduced for functional equations by Rubinfeld and Sudan [16, 17]). Then we will show how to efficiently estimate block statistics.

Lemma 3.1. $\mathsf{dist}(w, w') \leq (\frac{1}{2}|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')| + \varepsilon) \times n.$

Proof. If $\mathsf{b\text{-}stat}(w) = \mathsf{b\text{-}stat}(w')$, then the distance $\mathsf{dist}(w, w')$ is at most εn as we only need to move εn block letters. Otherwise, we will construct a word w'' from w such that $\mathsf{b\text{-}stat}(w'') = \mathsf{b\text{-}stat}(w')$, using at most $\frac{n}{2}|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')|$ substitutions. Applying the triangle inequality and the previous case, we obtain the desired result.

Collect in X_+ the positions i of block letters $w[i]_b$ such that $\mathsf{b\text{-}stat}(w)[w[i]_b] > \mathsf{b\text{-}stat}(w')[w[i]_b]$, and in X_- the positions j such that $\mathsf{b\text{-}stat}(w)[w'[j]_b] < \mathsf{b\text{-}stat}(w')[w'[j]_b]$. Note that X_+ and X_- have the same cardinality, which is $\frac{n}{2k}|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')|$. Initially we let $w'' = w$. Until $X_+ \neq \emptyset$ repeat the following: take any $i \in X_+$ and $j \in X_-$; replace in w'' the letters of $w''[i]_b = w[i]_b$ with those of $w'[j]_b$ (using at most k substitutions); remove i from X_+ and j from X_-. The resulting word w'' satisfies the required conditions. $\square$

In order to approximate $\mathsf{b\text{-}stat}(w)$, we will sample N subwords of w according to its block distribution, that is N random blocks letters, and we will compute a random vector $\widehat{\mathsf{b\text{-}stat}}_N(w)$, defined below, which will be close to $\mathsf{b\text{-}stat}(w)$ with high probability. Let X_i be the random vector of size $|\Sigma|^k$ whose u-coordinate is one and others are zero, where u is corresponds to a block letter chosen randomly and independently according to the block distribution of w. Recall that $\mathrm{E}(X_i) = \mathsf{b\text{-}stat}(w)$. We define

$$\widehat{\mathsf{b\text{-}stat}}_N(w) \stackrel{\text{def}}{=} \tfrac{1}{N} \sum_{i=1,\dots,N} X_i.$$

We wish to bound $\Pr[|\text{b-stat}(w) - \widehat{\text{b-stat}}_N(w)| \geq \varepsilon]$. There are several methods which can be used to obtain a Chernoff bound type on vectors. We could use the method of bounded differences. In our simple case, the use of Chernoff bound together with a direct union bound is enough.

Lemma 3.2. *There exists $N \in O(\frac{\log|\Sigma|}{\varepsilon^3})$ for which $\Pr[|\text{b-stat}(w) - \widehat{\text{b-stat}}_N(w)| \geq \varepsilon] \leq \frac{1}{3}$.*

Proof. For any block letter $u \in \Sigma^k$, $\Pr[|\text{b-stat}(w)[u] - \widehat{\text{b-stat}}_N(w)[u]| \geq t \cdot \text{b-stat}(w)[u]] \leq 2^{-8Nt^2}$, for any $t > 0$, by the Chernoff bound. Using a union bound, we conclude that: $\Pr[|\text{b-stat}(w) - \widehat{\text{b-stat}}_N(w)| \geq t] \leq |\Sigma|^k \times 2^{-8Nt^2}$. If we set $t = \varepsilon$, we get the result using an appropriate coefficient for $N \in O(\frac{\log|\Sigma|}{\varepsilon^3})$. ☐

3.2 Uniform Statistics

In this section, w and w' are again two words of size n over Σ. We want to construct a tolerant tester (as per the definition of [15]), which is not only an ε-tester, but also accepts words that are ε'-close, for some $\varepsilon' < \varepsilon$. We fix some integer k and let $\varepsilon = \frac{1}{k}$. We consider three different statistics: the original statistics of blocks $\text{b-stat}(w)$, the block uniform statistics $\text{bu-stat}(w)$, and the uniform statistics $\text{u-stat}(w)$. Any of those probabilities can be efficiently approximated as in Lemma 3.2.

We define these new statistics like the block statistics, using variants of the block distribution. The *uniform distribution* of w corresponds to a uniform and random choice of a subword of size k of w. Equivalently, the *uniform statistics* $\text{u-stat}(w)$ is defined for its u-coordinate, where $u \in \Sigma^k$, by

$$\text{u-stat}(w)[u] \stackrel{\text{def}}{=} \Pr_{j=1,\ldots,n-k+1}[w[j]w[j+1]\ldots w[j+k-1] = u].$$

To define the block uniform distribution of w we need to partition w in bigger consecutive blocks of size K, where $K = \lfloor \frac{\varepsilon^3 n}{\log|\Sigma|} \rfloor$. To simplify the discussion, we assume that k divides $(K-k-1)$, that n is divisible by K, and that $n = \Omega(\frac{\log|\Sigma|}{\varepsilon^5})$. We call the new blocks the *big blocks*. Now the *block uniform distribution* is defined by the following two-step procedure: First, in each big block choose uniformly a random $0 \leq t \leq k-1$, and delete the first t letters and the last $k-1-t$ letters; then take uniformly a random block letter in the remaining subword of the original word. The *block uniform statistics* $\text{bu-stat}(w)$ is therefore just the expectation of block statistics of the big blocks where the t first letters and the $k-1-t$ last letters are first removed, for a randomly chosen $0 \leq t \leq k-1$.

We will prove that u-stat is both robust and sound, which leads to an estimator of the distance for far away instances, whereas b-stat is only robust. For instance, the words $(01)^n$ and $(10)^n$ are $\frac{1}{2n}$-close, whereas their block statistics are $\Omega(1)$-far. The proof of the robustness of u-stat will use in an intermediate step the robustness of the block uniform statistics bu-stat. For the soundness of u-stat, the proof is much simpler.

Lemma 3.3 (Soundness). *If $\text{dist}(w, w') \leq \varepsilon^2 n$ then $|\text{u-stat}(w) - \text{u-stat}(w')| \leq 6\varepsilon$.*

Proof. Assume that $\text{dist}(w, w') = 1$. In case of a simple edit operation (insertion, deletion, substitution) on a letter, $|\text{u-stat}(w) - \text{u-stat}(w')| \leq 2 \times \frac{k}{n}$. For a move operation, if $w = A \cdot B \cdot C \cdot D$ and $w' = A \cdot C \cdot B \cdot D$ where a subword B has been moved, there are three border areas where we may choose a word of length k in w which does not exist in w'. Conversely, there are similar borders in w'. For each border, there are k possible subwords that intersect it, hence $|\text{u-stat}(w) - \text{u-stat}(w')| \leq 2 \times \frac{3k}{n}$.

If $\text{dist}(w, w') \leq \varepsilon^2 n$ then by the triangle inequality $|\text{u-stat}(w') - \text{u-stat}(w')| \leq \varepsilon^2 n \times \frac{6k}{n} = 6\varepsilon$, since $k = \frac{1}{\varepsilon}$. ☐

We show that the robustness for $\text{b-stat}(w)$ implies the robustness for $\text{bu-stat}(w)$, which then will imply the robustness for $\text{u-stat}(w)$. For a big block B_i, where $i = 1, \ldots, \frac{n}{K}$, we denote by v_{i,t_i} the subword of B_i after deleting the first t_i letters and the last $k-1-t_i$ letters of B_i. Let v be the concatenations of the words v_{i,t_i}. Then by the definition of $\text{bu-stat}(w)$ we have

$$\text{bu-stat}(w) = \frac{K}{n} \sum_{i=1}^{n/K} \mathop{\mathrm{E}}_{t_i=0,\ldots,k-1}(\text{b-stat}(v_{i,t_i})) = \mathop{\mathrm{E}}_{v}(\text{b-stat}(v)).$$

Intuitively one would like to use this equation directly for extending the robustness of $\mathsf{b\text{-}stat}$ to $\mathsf{bu\text{-}stat}$. Nonetheless, this will not work since one would need to use a triangle inequality in the wrong direction. Instead we use a more elaborate proof based on a Chernoff bound argument.

Lemma 3.4. *Let w be a word over Σ of size n. There exists a word v obtained from w after deleting $O(\frac{\log|\Sigma|}{\varepsilon^4})$ letters, so that $|\mathsf{bu\text{-}stat}(w) - \mathsf{b\text{-}stat}(v)| \leq \frac{\varepsilon}{2}$.*

Proof. Fix a coordinate $u \in \Sigma^k$. For every $i = 1, \dots, \frac{n}{K}$, let X_i be the random variable $X_i \stackrel{\text{def}}{=} \mathsf{b\text{-}stat}(v_{i,t_i})[u]$, where t_i is chosen uniformly in $\{0, \dots, k-1\}$. We denote by v the random word obtained by the concatenation of the words v_{i,t_i}. Note that v is obtained from w after deleting $(k-1) \times \frac{n}{K} = O(\frac{\log|\Sigma|}{\varepsilon^4})$ letters.

The variables $(X_i)_i$ are independent random variables such that $0 \leq X_i \leq 1$ and $\mathrm{E}_v(\mathsf{b\text{-}stat}(v)[u]) = \frac{K}{n}\sum_i \mathrm{E}(X_i) = \mathsf{bu\text{-}stat}(w)[u]$. By the Chernoff bound we then get that, for any $t > 0$,

$$\Pr[|\mathsf{bu\text{-}stat}(w)[u] - \mathsf{b\text{-}stat}(v)[u]| \geq t \cdot \mathsf{bu\text{-}stat}(w)[u]] \leq 2^{-8(\frac{n}{K})t^2}.$$

We repeat the same argument for every u-coordinate, and using a union bound, we conclude that:

$$\Pr[|\mathsf{bu\text{-}stat}(w) - \mathsf{b\text{-}stat}(v)| \geq t] \leq |\Sigma|^k \times 2^{-8(\frac{n}{K})t^2}.$$

If we set $t = \frac{\varepsilon}{2} = \frac{1}{2k}$, and use the fact that $K = \lfloor \frac{\varepsilon^3 n}{\log|\Sigma|} \rfloor$, we conclude that there exists with non zero probability a word v that satisfies the required property about the statistics, completing the proof. $\square$

The following simple result is well known and easy to check.

Proposition 3.5. *Let $A \subseteq B$ be two finite subsets and let μ_A, μ_B be their respective uniform distributions. Then $|\mu_A - \mu_B| = 2\frac{|B|-|A|}{|B|}$.*

Lemma 3.6. *Let w be a word over Σ of size n. Then $|\mathsf{bu\text{-}stat}(w) - \mathsf{u\text{-}stat}(w)| = O(\frac{\log|\Sigma|}{\varepsilon^4 n})$.*

Proof. The proof consists in proving that both underlying block distributions are at ℓ_1-distance at most $O(\frac{\log|\Sigma|}{\varepsilon^4 n})$. Then the definitions of the vectors $\mathsf{u\text{-}stat}$ and $\mathsf{bu\text{-}stat}$ directly imply the result.

The uniform distribution consists in choosing uniformly at random a subword u of w of length k, that is an integer $z \in \{1, 2, \dots, n-k-1\}$. The block uniform distribution consists in choosing uniformly at random a big block, an integer $0 \leq t \leq k-1$, and then a subword u of length k at position i in the big block that satisfies $(i-1) = t \mod k$. In an equivalent way, the block uniform distribution is the uniform distribution over all subwords of size k that are inside some big block.

The number of subwords of size k that cross boundaries of big blocks is $(k-1) \times (\frac{n}{K} - 1)$. Therefore using Proposition 3.5, the ℓ_1-distance between the distributions is upper bounded by $2 \times (k-1)(\frac{n}{K} - 1) \times \frac{1}{n-k} = O(\frac{\log|\Sigma|}{\varepsilon^4 n})$. $\square$

Combining the previous lemmas and the robustness of block statistics, we get our robustness lemma.

Lemma 3.7 (Robustness). *Let $n = \Omega(\frac{\log|\Sigma|}{\varepsilon^5})$. If $\mathsf{dist}(w, w') \geq 5\varepsilon n$ then $|\mathsf{u\text{-}stat}_k(w) - \mathsf{u\text{-}stat}_k(w')| \geq 6.5\varepsilon$.*

Proof. We assume that $n/(\frac{\log|\Sigma|}{\varepsilon^5})$ is large enough so that the $O(\frac{\log|\Sigma|}{\varepsilon^4})$ of Lemma 3.4 is upper bounded by $\frac{\varepsilon n}{16}$, and the $O(\frac{\log|\Sigma|}{\varepsilon^4 n})$ of Lemma 3.6 is upper bounded by $\frac{\varepsilon}{8}$.

Using Lemmas 3.4 and 3.6, we get subwords v and v' that respectively come from w and w' after deleting at most $\frac{\varepsilon n}{16}$ letters from each, so that $|\mathsf{u\text{-}stat}(w) - \mathsf{b\text{-}stat}(v)| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{8}$ and $|\mathsf{u\text{-}stat}(w') - \mathsf{b\text{-}stat}(v')| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{8}$.

From the hypothesis on w and w', and using the triangle inequality on dist, we obtain that $\mathsf{dist}(v, v') \geq 5\varepsilon n - \frac{\varepsilon n}{8}$. Therefore, using Lemma 3.1, we get that $|\mathsf{b\text{-}stat}(v) - \mathsf{b\text{-}stat}(v')| \geq 8\varepsilon - \frac{\varepsilon}{4}$, which implies from the construction of v, v' that $|\mathsf{u\text{-}stat}(w) - \mathsf{u\text{-}stat}(w')| \geq 8\varepsilon - \frac{\varepsilon}{4} - 2(\frac{\varepsilon}{2} + \frac{\varepsilon}{8}) = 6.5\varepsilon$. $\square$

Using the Soundness and Robustness Lemmas, we can construct a one-sided error tester for the equality of two words which is also $(\varepsilon^2, 5\varepsilon)$-tolerant:

Uniform Tester(w, w', ε)**:**
Let $N = \Theta(\frac{\log|\Sigma|}{\varepsilon^3})$, and $k = \frac{1}{\varepsilon}$
Compute $\widehat{\mathsf{u\text{-}stat}}_N(w)$ and $\widehat{\mathsf{u\text{-}stat}}_N(w')$ using the same N random and uniform indices in $\{1, \ldots, n-k+1\}$
Accept if $|\widehat{\mathsf{u\text{-}stat}}_N(w) - \widehat{\mathsf{u\text{-}stat}}_N(w')| \leq 6.25\varepsilon$
Reject otherwise

From the above lemmas it is clear that this algorithm satisfies the requirements of the following theorem.

Theorem 3.8. *For any $\varepsilon > 0$, and two words w, w' of the same size of order $\Omega(\frac{\log|\Sigma|}{\varepsilon^5})$, the previous test,*
(1) accepts if $w = w'$ with probability 1,
(2) accepts if w and w' are ε^2-close with probability at least $2/3$,
(3) rejects if w and w' are 5ε-far with probability at least $2/3$.
Moreover its query and time complexities are in $O(\frac{\log|\Sigma|}{\varepsilon^4})$.

From this $(\varepsilon^2, 5\varepsilon)$-tolerant tester, one can derive an $(\frac{\varepsilon^2}{25}, 5\sqrt{\varepsilon})$-approximation algorithm of the distance following the approach of [15].

Corollary 3.9. *There exists a randomized algorithm A such that, given two words w, w' of the same size of order $\Omega(\frac{\log|\Sigma|}{\varepsilon^5})$, A outputs ε' satisfying $\varepsilon' \in [\frac{\varepsilon^2}{25}, 5\sqrt{\varepsilon}]$ with probability at least $2/3$, where $\varepsilon = \frac{\mathsf{dist}(w,w')}{|w|}$. Moreover the query and time complexities of A are in $O(\frac{\log(|\Sigma|/\varepsilon)}{\varepsilon^4})$.*

4 Geometric Embedding of a Language

4.1 General Observations

In this section, we want to use the notion of block statistics in order to efficiently characterize a language. We choose this statistics vector not only since it is the simplest to manipulate, but it is also the most appropriate one for our purpose as we will see below.

Using the previous section, we can embed a word w into its block statistics $\mathsf{b\text{-}stat}(w) \in \mathbb{R}^{|\Sigma|^{1/\varepsilon}}$. This characterization is approximately one-to-one from Lemma 3.1 if the size of the words is fixed. It is not the case for words of different lengths as $\mathsf{b\text{-}stat}(w_0) = \mathsf{b\text{-}stat}(w_0^t)$ for every positive integer t, if w_0 is any word of size $k = \frac{1}{\varepsilon}$.

This means that the set of block statistics $\mathsf{b\text{-}stat}(w)$ of all the elements $w \in L$ is not a good characterization of a general language L. For instance, the word $w_0^{3\times 2^{s-1}}$ is $(1 - 1/k^{2^{s-1}})$-far from the language $\{w_0^{2^t} : t \geq 1\}$, for every positive integer s.

This example shows that one might consider only block statistics of loops of a language. This characterization makes sense when any word of a language can be decomposed into loops up to few remaining letters. Such languages are essentially the regular languages. Observe also that the fact that any iteration of the same loop is mapped into one point of $\mathbb{R}^{|\Sigma|^{1/\varepsilon}}$ is a property of the block statistics which does not hold for uniform statistics.

4.2 Regular Languages

We fix a finite alphabet Σ, and an automaton A (possibly non deterministic) on Σ with a set of states Q of size m, that recognizes a regular language L. Let k be a positive integer and $\varepsilon = \frac{1}{k}$. We consider only words whose size is divisible by k, as any word of length n of L, for n large enough, is close to such a word. Define A^k, the *k-th power of A*, the automaton on Σ^k with set of states Q such that transitions of A^k are exactly k consecutive transitions of A. Then A and A^k recognize the same language. In the general case, one can modify A^k such that A^k recognizes the language of words of L where the last $(|w| - k\lfloor\frac{|w|}{k}\rfloor)$ letters are deleted.

We will characterize L by the block statistics of its loops on the block alphabet. We remark that the statistics of A^k-loops basically only depend on L and k, from Proposition A.2 in Appendix A.

Definition 4.1. *A word v over Σ^k is an A^k-*loop *if there exist two words u, w over Σ^k and an accepting path of A^k for uvw, such that the state of the automaton after reading u (following the above accepting path) is identical to the state after reading uv.*
*A finite set of A^k-loops is A^k-*compatible *if all the loops can occur one after the other (in any order) in one accepting path of A^k.*

We define the geometric embedding of L by the union of convex hulls of every compatible set of loops.

Definition 4.2.

$$\mathcal{H} \stackrel{\text{def}}{=} \bigcup_{\substack{v_1,\ldots,v_t:\\ A^k\text{-compatible loops}\\ t\geq 0}} \mathsf{Convex\text{-}Hull}(\mathsf{b\text{-}stat}(v_1),\ldots,\mathsf{b\text{-}stat}(v_t)).$$

This definition is motivated by a standard result on finite automata: one can rearrange any word of a regular language into a sequence of small compatible loops. We formulate this fact in our context.

Proposition 4.3. *Let $w \in L$. Then $w \equiv_k vu_1u_2\ldots u_l$, where $|v|_b, |u_1|_b, \ldots, |u_l|_b \leq m$ and $\{u_1, u_2, \ldots, u_l\}$ is an A^k-compatible set of A^k-loops (non necessarily pairwise distinct).*

A consequence of this proposition is that if a word $w \in L$, then it has to satisfy approximately $\mathsf{b\text{-}stat}(w) \in \mathcal{H}$ (Lemma 4.5 below). The converse is also approximately true (Theorem 4.7 below).

Another consequence together with Caratheodory's theorem is that one can equivalently define $\mathcal{H}$ when the loop sizes and the number of compatible loops are bounded (see Appendix B for the proof). Recall that even if this new characterization explicitly depends on A^k (that is on A and ε), the set $\mathcal{H}$ only depends on L and ε (see Appendix A).

Proposition 4.4.

$$\mathcal{H} = \bigcup_{\substack{v_1,\ldots,v_t:\\ A^k\text{-compatible loops}\\ t=|\Sigma|^{1/\varepsilon}+1, |v_i|_b\leq m}} \mathsf{Convex\text{-}Hull}(\mathsf{b\text{-}stat}(v_1),\ldots,\mathsf{b\text{-}stat}(v_t)).$$

The following lemma gives one direction in the correspondence between L and $\mathcal{H}$ that we are looking for. It can be understood as an approximate Parikh classification of regular languages, whereas the original Parikh characterization was for context-free languages [14].

Lemma 4.5. *For every $w \in L$ there exists w', so that*

$$0 \leq |w| - |w'| \leq \tfrac{m}{\varepsilon},\ \mathsf{dist}(w, w') \leq \tfrac{m}{\varepsilon},$$

$$|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')| \leq \tfrac{2m}{\varepsilon|w|},\ \textit{and}\ \mathsf{b\text{-}stat}(w') \in \mathcal{H}.$$

Proof. First, recall that the block statistics $\mathsf{b\text{-}stat}(w)$ is invariant under block letter permutations. Moreover, if w' is w on which one block letter has been either deleted or inserted then $|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')| \leq \frac{2}{\varepsilon|w|}$.

Let $w \in L$. Applying Proposition 4.3, we can delete less than m block letters from w so that the resulting word is a concatenation $w' \equiv_k u_1u_2\ldots u_l$, where the u_i are A^k-compatible A^k-loops. Since w' is obtained from w using at most m deletions of block letters, we have $|\mathsf{b\text{-}stat}(w) - \mathsf{b\text{-}stat}(w')| \leq \frac{2m}{\varepsilon|w|}$. This concludes the proof since $\mathsf{b\text{-}stat}(w') \in \mathcal{H}$. ∎

Lemma 4.6. *For every $X \in \mathcal{H}$ and every n there exists $w \in L$, such that*

$$0 \leq |w| - n \leq (|\Sigma|^{1/\varepsilon} + 3)\tfrac{2m}{\varepsilon},\ \textit{and}\ |X - \mathsf{b\text{-}stat}(w)| \leq (|\Sigma|^{1/\varepsilon} + 2)\tfrac{3m}{\varepsilon n}.$$

Proof. Let $X \in \mathcal{H}$, that is $X = \sum_{i=1}^{l} \lambda_i \cdot \mathsf{b\text{-}stat}(u_i)$, where $l = |\Sigma|^k + 1$, $|u_i|_b \leq m$, $0 \leq \lambda_i \leq 1$ and $\sum_i \lambda_i = 1$. Fix any integer n. We choose non negative integers $(r_i)_{i=1,2,\ldots,l}$ that respectively approximate $\lambda_i \frac{\varepsilon n}{|u_i|_b}$, that is satisfy $0 \leq |r_i - \lambda_i \frac{\varepsilon n}{|u_i|_b}| \leq 1$, and such that $0 \leq \sum_i r_i |u_i|_b - \varepsilon n \leq m$. It is always possible to satisfy this last condition due to the degree of freedom on the choices of r_i and the upper bound $|u_i|_b \leq m$: We let $j \geq 0$ be the minimum integer so that $\sum_{i=1}^{j} \lceil \lambda_i \frac{\varepsilon n}{|u_i|_b} \rceil |u_i|_b + \sum_{i=j+1}^{l} \lfloor \lambda_i \frac{\varepsilon n}{|u_i|_b} \rfloor |u_i|_b \geq 0$, and set $r_i = \lceil \lambda_i \frac{\varepsilon n}{|u_i|_b} \rceil$ for $i \leq j$ and $r_i = \lfloor \lambda_i \frac{\varepsilon n}{|u_i|_b} \rfloor$ for $i > j$.

Define the word $w' = u_1^{r_1} u_2^{r_2} \ldots u_l^{r_l}$. Then its block length is close to εn: $0 \leq |w'|_b - \varepsilon n \leq m$. Moreover its block statistics satisfies

$$
\begin{aligned}
|\mathsf{b\text{-}stat}(w') - X| &= \left| \sum_i \left(r_i \frac{|u_i|_b}{|w'|_b} - \lambda_i \right) \mathsf{b\text{-}stat}(u_i) \right| \\
&\leq \sum_i |r_i \frac{|u_i|_b}{|w'|_b} - \lambda_i| \\
&\leq \sum_i |r_i \frac{|u_i|_b}{|w'|_b} - r_i \frac{|u_i|_b}{\varepsilon n}| + \sum_i |r_i \frac{|u_i|_b}{\varepsilon n} - \lambda_i| \\
&\leq \sum_i r_i |u_i|_b \times |\frac{1}{|w'|_b} - \frac{1}{\varepsilon n}| + \sum_i \frac{m}{\varepsilon n} \\
&\leq (m + \varepsilon n) \times (\frac{1}{\varepsilon n} - \frac{1}{m+\varepsilon n}) + l \frac{m}{\varepsilon n} = \frac{m}{\varepsilon n} + l \frac{m}{\varepsilon n}.
\end{aligned}
$$

Using A^k-compatibility, we can get a word of L from w' by inserting few block letters. Let $v_0 u_{i_1} v_1 u_{i_2} v_2 \ldots u_{i_l} v_l \in L$ be the witness of the A^k-compatibility of the loops $u_1, \ldots, u_l$, such that $|v_j|_b \leq m$ for every j, and where $(i_1, \ldots, i_l)$ is a permutation of $(1, \ldots, l)$. Then the word $w = v_0 u_{i_1}^{r_{i_1}} v_1 u_{i_2}^{r_{i_2}} v_2 \ldots u_{i_l}^{r_{i_l}} v_l \in L$ by construction. Moreover $0 \leq |w|_b - |w'|_b \leq (l+1)m$, and $|\mathsf{b\text{-}stat}(w') - \mathsf{b\text{-}stat}(w)| \leq \frac{2(l+1)m}{\varepsilon n}$, so we conclude. □

Theorem 4.7. *Let $w \in \Sigma^n$ and $X \in \mathcal{H}$ be such that $|\mathsf{b\text{-}stat}(w) - X| \leq \delta$. Then*

$$\mathsf{dist}(w, L) \leq \left(\frac{\delta}{2} + \left(1 + O(\frac{m|\Sigma|^{1/\varepsilon}}{\varepsilon^2 n})\right)\varepsilon \right) n.$$

Proof. Let $n = |w|$. For simplicity, we assume that k divides n, otherwise we just delete at most $k-1$ letters from w so that the new length is dividable by k. From Lemma 4.6, there exists a word $w' \in L$, such that $0 \leq |w'| - n \leq (l+2)\frac{2m}{\varepsilon}$ and $|\mathsf{b\text{-}stat}(w') - X| \leq (l+1)\frac{3m}{\varepsilon n}$, where $l = 1 + |\Sigma|^k$. We again assume that k divides $|w'|$.

Assume that $|w| = |w'|$. Then, using Lemma 3.1, we get that $\mathsf{dist}(w, w') \leq (\frac{1}{2}(\delta + (l+1)\frac{3m}{\varepsilon n}) + \varepsilon)n$.

If w and w' have different sizes, we artificially increment the size of w by adding at most $(l+2)2m$ block letters at the end of w (recall that adding a block letter adds k to the word size). The deviation of its block statistics is then at most $(l+2)2m \times \frac{2}{\varepsilon n}$, so we asymptotically get the same bound. □

4.3 Construction of $\mathcal{H}$

One of the remaining tasks is to efficiently construct $\mathcal{H}$ for a given automaton A with m states. One could try to enumerate all A^k-loops of size at most m over Σ^k. This is not efficient enough due to the possible large number of loops, $O(|\Sigma|^{km})$. Nevertheless, the number of possible corresponding block statistics is exponentially smaller, $O(m^{|\Sigma|^k})$, since a block statistics of a word v of size at most m over Σ^k has at most $|v|_b \leq m$ nonzero coordinates of the form $\frac{a}{|v|_b}$, where $a = 1, \ldots, |v|_b$. We now explain how to enumerate such block statistics.

We proceed recursively on the length t of paths between two possible states of A^k, for $t = 1, \ldots, m$. Let P_t be an $m \times m$ matrix where the entry (i, j) is the set of block statistics corresponding to a path of length t between the states i and j. Let us consider the algebra of sets of distributions over Σ^k with the operations $\cup, \odot_t$, where $\odot_t$ is distributive over $\cup$ and defined for singletons by $\{\overrightarrow{x}\} \odot_t \{\overrightarrow{y}\} = \{\frac{1}{t+1}\overrightarrow{x} + \frac{t}{t+1}\overrightarrow{y}\}$. If we denote by $\circ_t$ the matrix multiplication over this algebra, then the matrices P_t satisfy the following simple inductive equation, where P_1 is directly given by A^k (by setting each non-empty entry of P_1 to be the set of unit vectors corresponding to the block letters labeling the corresponding arcs in A^k):

$$P_{t+1} = P_1 \circ_t P_t.$$

Lemma 4.8. *Given A and ε, a set H of $(|\Sigma|^{1/\varepsilon} + 1)$-tuples of vectors can be computed in time $m^{|\Sigma|^{O(1/\varepsilon)}}$ such that*

$$|H| \leq m^{|\Sigma|^{O(1/\varepsilon)}} \quad \textit{and} \quad \mathcal{H} = \bigcup_{S \in H} \mathsf{Convex\text{-}Hull}(S).$$

Proof. We first compute as we explained above the matrices $(R_t)_{t=1,\dots,m}$. At the end of the process, the diagonals of those matrices contain the block statistics of all A^k-loops of length at most m. Then, a tuple of $(|\Sigma|^{1/\varepsilon}+1)$ loops is compatible if and only if there exists an accepting path of the automaton which passes through all states of the respective origins of the loops, a condition that can also be checked in polynomial time by using matrix multiplication over an appropriate algebra. Using Proposition 4.4, we know that including in H the statistics of the corresponding compatible sets is sufficient. The upper bounds on the size and the time complexity of the decomposition come from the previous observation that at most $m^{|\Sigma|^k}$ block statistics are considered. □

For a regular language, the set $\mathcal{H}$ is a subset of the unit ball of $\mathbb{R}^{|\Sigma|^k}$ for the ℓ_1-norm. Let us consider the grid $\mathcal{G}_\varepsilon = \{0, \frac{\varepsilon}{|\Sigma|^k}, \frac{2\varepsilon}{|\Sigma|^k}, \dots, 1\}^{|\Sigma|^k}$ of the cube $[0,1]^{|\Sigma|^k}$ with step $\frac{\varepsilon}{|\Sigma|^k}$. Let $\mathcal{H}_\varepsilon$ be the set of points of $\mathcal{G}_\varepsilon$ that are at distance at most $\frac{\varepsilon}{2}$ from $\mathcal{H}$ (for the ℓ_1-distance). Since $|\mathcal{G}_\varepsilon| = (k|\Sigma|^k+1)^{|\Sigma|^k} = 2^{|\Sigma|^{O(1/\varepsilon)}}$, then $|\mathcal{H}_\varepsilon| = 2^{|\Sigma|^{O(1/\varepsilon)}}$. Moreover, one can easily construct it from H.

Proposition 4.9. *Given A and ε, the set $\mathcal{H}_\varepsilon$ can be computed in time $m^{|\Sigma|^{O(1/\varepsilon)}}$.*

4.4 Applications

Theorem 4.10. *For every real $\varepsilon > 0$ and regular language L over a finite alphabet Σ, there exists an ε-tester for L whose query complexity is in $O(\frac{\log|\Sigma|}{\varepsilon^4})$ and time complexity is in $2^{|\Sigma|^{O(1/\varepsilon)}}$.*
Moreover, given an automaton with m states that recognizes L, the tester can be constructed in time $m^{|\Sigma|^{O(1/\varepsilon)}}$.

Proof. We fix $\varepsilon > 0$, and automaton A with m states that recognizes L. We will construct a 3ε-tester for L. Let w be a word given as input. We assume that $|w|/(\frac{m|\Sigma|^{1/\varepsilon}}{\varepsilon^2})$ is large enough, otherwise we just run the automaton on w.

The tester is in two steps: a preprocessing step and the testing step itself. Given A and ε, one can compute $\mathcal{H}_\varepsilon$ in time $m^{|\Sigma|^{O(1/\varepsilon)}}$ from Proposition 4.9. Now the testing part consists in computing the estimation $\widehat{\mathsf{b\text{-}stat}}_N(w)$ of $\mathsf{b\text{-}stat}(w)$ as in Lemma 3.2, where $N = \Theta(\frac{\log|\Sigma|}{\varepsilon^3})$, using $O(\frac{\log|\Sigma|}{\varepsilon^4})$ queries to w. Then if $\widehat{\mathsf{b\text{-}stat}}_N(w)$ is at distance at most 2ε from $\mathcal{H}_\varepsilon$, the tester accepts, and otherwise it rejects.

From Lemma 3.2, $\widehat{\mathsf{b\text{-}stat}}_N(w)$ is at ℓ_1-distance at most ε from $\mathsf{b\text{-}stat}(w)$, with high probability. Now, if $w \in L$, using Lemma 4.5, $\mathsf{b\text{-}stat}(w)$ is at ℓ_1-distance at most 0.25ε from $\mathcal{H}$ and at most 0.75ε from $\mathcal{H}_\varepsilon$, and therefore the tester accepts w with high probability. If w is 3ε-far from L, then by the contraposition of Theorem 4.7, $\mathsf{b\text{-}stat}(w)$ is at ℓ_1-distance at least $(4-0.25)\varepsilon$ from $\mathcal{H}$ and at least 3.25ε from $\mathcal{H}_\varepsilon$, so the tester rejects w with high probability. □

We end with the existence of a deterministic equivalence tester for regular languages.

Theorem 4.11. *There exists a deterministic algorithm T such that, given two automata A and B over a finite alphabet Σ with at most m states and a real $\varepsilon > 0$, $T(A,B,\varepsilon)$*

(1) accepts if A and B recognize the same language,

(2) rejects if A and B recognize languages that are not ε-equivalent.

Moreover the time complexity of T is in $m^{|\Sigma|^{O(1/\varepsilon)}}$.

Proof. Fix an $\varepsilon > 0$. The algorithm simply computes the respective discrete approximations $\mathcal{H}_{A,\varepsilon}$ and $\mathcal{H}_{B,\varepsilon}$ of $\mathcal{H}_A$ and $\mathcal{H}_B$ corresponding to the automata A and B. If they are equal, the tester accepts, and otherwise it rejects.

If A and B recognize the same language, then one can verify that their respective sets $\mathcal{H}_A$ and $\mathcal{H}_B$ are equal (see for instance Proposition A.4 in Appendix A), and so their discrete approximation $\mathcal{H}_{A,\varepsilon}$ and $\mathcal{H}_{B,\varepsilon}$ are also identical. Therefore the algorithm accepts.

Assume now that A and B are not 2ε-equivalent. For instance assume that A is not 2ε-contained in B. Let (w_n) be an infinite sequence of words accepted by A but 2ε-far from B. We only consider just one word w_n such that $|w_n| = \Omega(\frac{m|\Sigma|^{1/\varepsilon}}{\varepsilon^2})$. Then from Lemma 4.5, $\mathsf{b\text{-}stat}(w_n)$ is at most 0.25ε from $\mathcal{H}_A$ and at most 0.75ε from $\mathcal{H}_{A,\varepsilon}$. Now by the contraposition of Theorem 4.7, since w_n is 2ε-far from the language that B recognizes, we get that $\mathsf{b\text{-}stat}(w_n)$ is at ℓ_1-distance at least $(2-0.25)\varepsilon$ from $\mathcal{H}_{B,\varepsilon}$ and at least 1.25ε from $\mathcal{H}_{B,\varepsilon}$. Therefore the algorithm rejects. □

References

[1] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy. Regular languages are testable with a constant number of queries. *SIAM Journal on Computing*, 30(6):1842–1862, 2000.

[2] Z. Bar-Yossef, T.S. Jayram, R. Krauthgamer, and R. Kumar. Approximating edit distance effi ciently. In *Proceedings of the ACM Symposium on Theory of Computing*, 2004.

[3] T. Batu, F. Ergun, J. Kilian, A. Magen, S. Raskhodnikova, R. Rubinfeld, and R. Sami. A sublinear algorithm for weakly approximating edit distance. In *Proceedings of the ACM Symposium on Theory of Computing*, pages 316–324, 2003.

[4] M. Blum and S. Kannan. Designing programs that check their work. *Journal of the ACM*, 42(1):269–291, 1995.

[5] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[6] G. Cormode. *Sequence Distance Embeddings*. PhD thesis, University of Warwick, 2003.

[7] G. Cormode and S. Muthukrishnan. The sring edit distance matchning problem with moves. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 667–676, 2002.

[8] F. Ergün, S. Muthukrishnan, and S. Sahinalp. Comparing sequences with segment rearrangements. In *Proceedings of Foundations of Software Technology and Theoretical Computer Science*, pages 183–194, 2003.

[9] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[10] F. Magniez and M. de Rougemont. Property testing of regular tree languages. In *Proceedings of 31st International Colloquium on Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 932–944. Verlag, 2004.

[11] W. Masek and M. Paterson. A faster algorithm for computing string edit distance. *Journal of Computer and System Sciences*, 20(1):18–31, 1980.

[12] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 125–129, 1972.

[13] I. Newman. Testing membership in languages that have small width branching programs. *SIAM Journal on Computing*, 3142(5):1557–1570, 2002.

[14] R. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

[15] M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. Technical Report TR04-010, ECCC, 2004. `http://eccc.uni-trier.de/eccc-reports/2004/TR04-010/index.html`.

[16] R. Rubinfeld. On the robustness of functional equations. *SIAM Journal on Computing*, 28(6):1972–1997, 1999.

[17] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):23–32, 1996.

[18] D. Shapira and J. Storer. Edit distance with move operations. In *Proceedings of Symposium on Combinatorial Pattern Matching*, volume 2373 of *Lecture Notes in Computer Science*, pages 85–98. Verlag, 2002.

[19] D. Shapira and J. Storer. Large edit distance with multiple block operations. In *Proceedings of Symposium on String Processing and Information Retrievial*, pages 369–377, 2003.

A Automata Independence Proofs

We show that loops in different automata for the same language are related, by correlating them with a definition that only depends on the language itself.

Definition A.1. *A word v over Σ^k is an (L, k)-*loop *if there exist two words u, w over Σ^k such that $uv^t w \in L$ for every integer t.*

One can remark that A^k-loops and (L, k)-loops are nearly the same, and in particular share the same statistics, according to the following result.

Proposition A.2. *Every A^k loop is also an (L, k)-loop, and on the other hand for every (L, k)-loop v there exists $t \geq 1$ such that v^t is an A^k loop. In particular, the set of statistics of loops of an automaton deciding the language L depends only on L and k.*

Proof. The first direction is clear.

The second direction requires some indications. Let us now consider the family of L words $(uv^t w)_{t\geq 1}$, and in particular let us consider one of the accepting paths for $uv^m w$. By a counting argument, there exist $0 \leq t < t' \leq m$, such that this accepting path reaches the same state after both $|uv^t|_b$ steps and $|uv^{t'}|_b$ steps. Hence the automaton contains some loop that corresponds to $v^{t'-t}$. Moreover, there is an accepting path that contains this loop, namely the one for $uv^m w$, and so it is indeed an A^k-loop. □

However, for our purpose we need to consider not only loops, but sets of compatible loops. Here is the corresponding definition that depends on the language.

Definition A.3. *A sequence of words $v_1, \ldots, v_l$ over Σ^k is a* compatible (L, k)-loop sequence *if there exists a permutation $\sigma : \{1, \ldots, l\} \to \{1, \ldots, l\}$, and words $u_0, u_1, \ldots, u_l$ over Σ^k, such that for every $t_1, \ldots, t_l$ we have $u_0 v_{\sigma(1)}^{t_1} u_1 v_{\sigma(2)}^{t_2} u_2 \ldots u_{l-1} v_{\sigma(l)}^{t_l} u_l \in L$ (note that this in particular implies that $v_1, \ldots, v_l$ are (L, k)-loops).*

Proposition A.4. *Every compatible sequence of A^k-loops is also a compatible sequence of (L, k)-loops. On the other hand, for every compatible sequence $v_1, \ldots, v_l$ of (L, k)-loops there exist $t_1, \ldots, t_l \geq 1$ such that $v_1^{t_1}, \ldots, v_l^{t_l}$ is a compatible sequence of A^k-loops. In particular, the geometric set $\mathcal{H}$, constructed from any automaton A^k deciding the language L, depends only on L and k.*

Proof. The proof follows the very same methods of the proof of Proposition A.2. □

B Proof of Proposition 4.4

Proof of Proposition 4.4. The inclusion $\supseteq$ is straightforward.

For the $\subseteq$ inclusion, let us first state Caratheodory's theorem: In dimension d, any convex hull of N points $p_1, \ldots, p_N$ can be decomposed into the union of convex hulls of $(d + 1)$ points $p_{i_1}, \ldots, p_{i_{d+1}}$ (with some possible repetitions), where the union is over every possible choices of these points. Hence this inclusion would have been also straightforward without the length assumption on the loops as the dimension $d = |\Sigma|^k$. To overcome the length constraint we use the fact that any loop $v = v_i$ of size $|v|_b > m$ can be decomposed into $v = u_1 u_2$, where u_1 and u_2 are loops which are also compatible with the other loops v_j. Repeating this argument inductively, we first prove that $\mathcal{H}$ is not smaller if we upper bound the loop sizes by m. Then applying Caratheodory's theorem, we conclude the proof. □

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'