Approximate Structural Consistency

Michel de Rougemont and Adrien Vieilleribière

Université Paris II, & LRI CNRS
LRI, Bâtiment 490, 91400 Orsay, France
{mdr,vieille}@lri.fr

Abstract. We consider documents as words and trees on some alphabet Σ and study how to compare them with some regular schemas on an alphabet Σ'. Given an input document I, we decide if it may be transformed into a document J which is ε-close to some target schema T: we show that this approximate decision problem can be efficiently solved. In the simple case where the transformation is the identity, we describe an approximate algorithm which decides if I is close to a target regular schema (DTD). This property is testable, i.e. can be solved in time independent of the size of the input document, by just sampling I. In the general case, the *Structural Consistency* decides if there is a transducer $\mathcal{T}$ with at most m states such that I is ε-close to I' and his image $\mathcal{T}(I')$ is both close to T and of size comparable to the size of I. We show that Structural Consistency is also testable, *i*.e. can be solved by sampling I.

1 Introduction

Property Testing [8, 5] considers approximations of decision problems, and is the basis for our approach. Testers for regular trees have been proposed in [6] and in [4] and extended to Data Exchange [3] where predefined constraints are given by a fixed transducer $\mathcal{T}$ [2]. In this paper we extend the approach when $\mathcal{T}$ is unknown, and we approximately decide if there exists a $\mathcal{T}$ which satisfies the Data Exchange condition.

Documents are considered as large labeled, unranked, ordered trees on an alphabet Σ with attributes [7], in some source-schema S (regular language given by regular expression on word and DTD on trees), and need to be classified. We define a canonical problem, *Structural Consistency*, which decides if a large tree τ_n close to some schema S can be transformed to a tree close to a regular target schema T. For a regular schema S, a tree τ_n of size n is ε-close to S for $0 \leq \varepsilon \leq 1$ if we can find $\tau' \in S$ such that $\mathsf{dist}(\tau, \tau') \leq \varepsilon \cdot n$. We use classical *transducers* on words and trees to transform the documents.

Some specific distance on documents, completely modifies the complexity of the basic questions, when we consider their approximate versions. Our goal is to show that some approximate classification problems can be simplified by the analysis of the statistics of schema and transducers to obtain algorithms with complexity independent of the size of the inputed structure.

Fix a source schema S, a target schema T, parameters $k = 1/\varepsilon$ (the precision),

α (the ratio) and m (the number of states of a transducer $\mathcal{T}$), where $0 < \varepsilon \leq 1$ and $0 < \alpha \leq 1$. An input I is a word w_n or an unranked ordered tree τ_n of size n following S.

Structural Consistency: Given a large input document I_n (word w_n or tree τ_n), decide if there is a transducer $\mathcal{T}$ with at most m states and an input I' ε-close to I, such that: $\alpha \cdot n \leq |\mathcal{T}(I')| \leq n/\alpha$ and $\mathcal{T}(I')$ is ε-close to T.

The transformed I' (word or tree) must satisfy two conditions: it must be of size proportional to n within a factor α, and ε-close to the schema T. A transducer which satisfies both conditions is called ε, α compatible. This problem captures the difficulties of Information Integration and Classification, as given target schemas $T_1, ..., T_k$ and an input document I, we can decide how to classify I, i.e. decide which schemas are ε, α compatible for I. For simplicity, we first consider words w_n where the techniques are simpler and generalize them to trees. Our main results are:

Theorem 4.1. Structural Consistency is testable on words.

Theorem 4.2. Structural Consistency is testable on unranked ordered trees.

We associate to a word w_n the statistics vector $\mathsf{ustat}_k(w_n)$, from which we can approximate any regular property [4]. In this paper we introduce a statistics matrix $\mathsf{ustat}_k(\tau_n))$, for an unranked ordered tree τ_n, from which we can similarly approximate any regular tree property. Regular schemas such as S and T are represented by unions of polytopes in the statistical space. A schema mapping μ is a mapping between some summits of a polytope H_S for S and some summits of H_T for T. A transducer π provides a linear transformation between the source and the target statistics and may be ε, α compatible for μ. As we can efficiently enumerate all possible ε, α compatible transformations, we obtain the results.

In section 2 we recall the basic notions on testers and the statistical embedding of [4] on words and trees. In section 3, we recall the basic results when the transformation is the identity, and in section 4 we study the Structural Consistency on words and trees.

2 Preliminaries

We consider classes of finite structures such as words and trees with possible attributes, and schemas are regular languages given by Tree-automata or DTDs. We approximate decision problems on such classes, given a distance between structures. We transform these structures with specific transducers.

Approximation The *Edit distance with moves* between two structures I and I', written $\mathsf{dist}(I, I')$, is the minimal number of elementary operations on I to obtain I', divided by $max\{|I|, |I'|\}$. An *elementary operation* on a structure I is either an *insertion, a deletion* of a node or of an edge, a *modification* of a letter (tag) or of an attribute value, or a *move*. For trees, a move consists in moving an entire subtree of τ into another position; for words, it means moving a consecutive sequence of letter into another position. For simplicity, in this paper we transform structures ignoring attribute values. We say that two structures

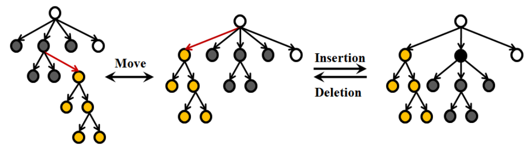

Fig. 1. Edit Distance with Moves: Elementary Operations

U_n, V_m (words or tress), whose domains are respectively of size n and m, are ε-*close* if their distance $\mathsf{dist}(U_n, V_m)$ is less than $\varepsilon \times \max\{n, m\}$. They are ε-*far* if they are not ε-close. We use a classical weak approximation:

Definition 1. *Let $\varepsilon \geq 0$ be a real. An ε-tester for a property P is a randomized algorithm A such that: (1) If I satisfies P, A always accepts; (2) If I is ε-far from P, then* $\Pr[A \text{ rejects}] \geq 2/3$.

A property is *testable* if for every sufficiently small $\varepsilon > 0$, there exists an ε-tester whose time complexity depends only on ε.

Statistical embedding on strings. For a finite alphabet Σ and a given ε, let $k = \frac{1}{\varepsilon}$. A word w of length n is embedded into a vector ustat_k of dimension $|\Sigma|^k$; $\mathsf{ustat}_k(w)[u] \stackrel{\text{def}}{=} \frac{\#u}{n-k+1}$ where $\#u$ is the number of occurrences of u (of length k) in w. This embedding is called a k-*gram* in statistics, and is related to [1] where the subwords of length k are called *shingles*.

Example 1. For $\Sigma = \{0, 1\}$, and $k = 2$, let w be the word 00011111. The statistic of w written in the lexicographical orderis $\mathsf{ustat}_2(w) = (2/7\ ,\ 1/7\ ,\ 0\ ,\ 4/7)$.

Statistical embedding on trees. We generalize k-grams on words to trees, using a matrix as in Figure 2. First, we transform an unranked tree with attributes (Fig. 2.(a)) into an extended binary tree[1], using the classical Rabin encoding[2] (Fig. 2.(b)). In this encoding, paths of length k can be paths on the right successor, *i*.e. horizontal paths in τ, paths on the left successor, *i*.e. vertical paths in τ, or zigzags. There are 2^{k-1} types of paths, and for each type we keep the classical ustat_k vector. For paths of length k, we associate their type as a boolean vector of length $k-1$. We use 0 for the left branch and 1 for the right branch. For a tree τ, let $\mathsf{ustat}_k(\tau)$ be the matrix with 2^{k-1} columns and $|\Sigma|^k$ lines. In column 1, we have the densities of paths of type 0..0, i.e. vertical paths in the original unranked tree. The last column describes paths of type 1..1, i.e. horizontal paths in the original unranked tree, and all the columns enumerate the 2^{k-1} types. As the matrix is sparse we only enumerate some of the entries with their non zero probabilities.

---

[1] An *extended* 2-ranked tree is a binary tree with a left successor, or a right successor or both.

[2] First child relations in the unranked tree are represented by left successors in the Rabin encoding, and next sibling relations are represented by right successors.

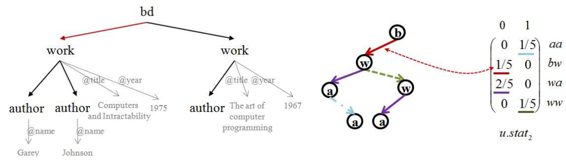

Fig. 2. A unranked tree with attributes, its Rabin encoding and its ustat_2 matrix. null entries are not represented; the first line indicates the type of the column; "author" is abbreviated by a, "bd" by b, and "work" by w.

Transformations The transformations considered are simple top-down transductions which can be implemented by linear XSLT programs. A transducer in state q transforms a letter of Σ_S (resp. a labeled node with attributes node, for trees) into a word (resp. a hedge, for trees) and continues the transformation top down, i.e. on the next letter (children, for trees) in another state q'. For instance, in state q, a transition on trees, denoted $(q, w) \rightarrow l(t, \underline{q'})$ transforms a node w into a node l with a first child τ and outputs the transformation of the children of w below l, on the right of τ, in state q'. This corresponds precisely to the linear restriction of a classical model of transduction of [7] and we restrict the study to deterministic transducers without λ transition.

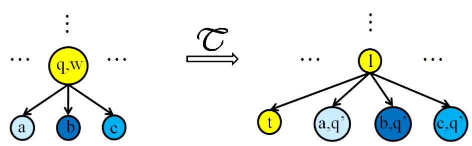

Fig. 3. Local transformation $(q, w) \rightarrow l(t, \underline{q'})$

3 Approximate Membership

This section recalls the basic membership testers for words and trees and gives a solution for Approximate Structural Consistency when the Transformation is the identity ($\mathcal{T} = id$). In all the following, let ε be fixed and $k = 1/\varepsilon$.

The Tester decides Approximate Membership (for words and trees) is based on the following property: if I is close to some schema T, I can be decomposed on simple loops, and then $\mathsf{ustat}_k(I)$ is ε-close to some polytope H_i^T of $H_T = \bigcup_i H_i^T$, *i*.e. ε-close to $H_i^T = \sum_{t_i \in C} \lambda_i \cdot t_i$, where $\sum_i \lambda_i = 1$, for some $C \subseteq \{t_1, ...t_p\}$ of size at most $d_T + 1$, where d_T is the dimension of the target vectors (Caratheodory theorem). Observe that for I large enough, I is ε-close to $I' = \Pi_{i \in C}(u_i)^{\lambda'_i . n}$ for $\lambda'_i = \frac{\lambda_i}{|u_i|}$, as λ_i reflects the density of loops u_i. In order to obtain I' from I, *moves* have been applied to regroup all identical loops and non loops have been deleted.

3.1 Word Case

The word embedding associates $\{\mathsf{ustat}_k(w) : w \in r\}$ to a regular expression r, a union of polytopes H in the same space, such that the distance (for the L_1 norm) between a vector and a union of polytopes is approximately $\mathsf{dist}(w, L(r))$, as shown in [4]. For a simple regular expression such as $(001)^*$, the polytope is a unique summit, the *base vector*, which by definition is $lim_{n\to\infty}\mathsf{ustat}((001)^n)$. For a more general regular expression, the polytope is the convex hull of the base vectors, associated with compatible *simple loops*, i.e. simple loops for which there is a run which follows them. Consider a word w as the input I, and its $\mathsf{ustat}_k(w)$ vector. The embedding associates with T a finite set H^T of polytopes H_i^T, with summits $t_1, ...t_p$. Each geometrical summit t_i is associated with a set $U_i = \{u_i^j\}$ of loops u_i^j (j is an index), and all the u_i^j have the same ustat vectors, i.e. correspond to the same geometrical summit t_i. Some of these loops u_i^j may be decomposed as smaller loops: if ab is a loop for an automaton associated with a schema T, so is $abab$, ba, ... For $k = 2$, they all have the same statistics. A finite set of loops $\{u_i^j\}$ for $i = 1, \ldots, p$ is *compatible* if there is an input w which follows all these loops.

Example 2. Let $k = 2 = 1/\varepsilon$, $w = 000111$, $T = (001)^*.1^*$. For a lexicographic enumeration of the length 2 binary words, $\mathsf{ustat}_2(w) = (2/5, 1/5, 0, 2/5)$. Let $t = (1/3, 1/3, 1/3, 0)$ the base vector of the regular expression $(001)^*$, and similarly $t' = (0, 0, 0, 1)$ for 1^*. The polytope H associated with T is $\mathrm{Convex-Hull}(t, t') = \{\lambda.t + (1 - \lambda).t', \lambda \in [0, 1]\}$ and it approximates the set of $\mathsf{ustat}_2(w)$ when $w \in T$.The word w is at distance $1/6$ to T as it requires the removal of the first 0 to yield the corrected word $001.11 \in T$.

The $\mathsf{ustat}_k(w)$ vector can be approximated for the L_1 norm by taking N random samples to define the random variable $\widehat{\mathsf{ustat}_k(w)}$ which approximates $\mathsf{ustat}_k(w)$. These techniques yield the simple testers of [4] for Membership (w, r) between a word w and a regular expression r. Take $N \in O(\frac{|\Sigma|^{2/\varepsilon}.\ln|\Sigma|}{\varepsilon^3})$ samples, and let $\widehat{\mathsf{ustat}_k(w)}$ be the ustat_k of the samples. We compute the set of polytopes H associated with r in the same space and reject if the geometrical distance from the point $\widehat{\mathsf{ustat}_k(w)}$ to H is greater than ε. If w is in r then $\mathsf{ustat}_k(w)$ is close to H and the membership test accepts. On the other hand, if w is ε-far from the regular expression r, then the tester rejects with high probabilities. This shows that the approximate Membership is testable on words.

3.2 Tree Case

Sampling. The $\mathsf{ustat}_k(\tau)$ can be approximated, for the L_1 norm, by taking random samples as follows. Select with the uniform distribution a random node i of τ and let $\widehat{\mathsf{ustat}_k}(\tau)$ be the random matrix where we add each path of length $k-1$ from i as a unit in the corresponding position (type, labels). After N samples, we divide by the numbers of units. Observe that $\mathrm{E}(\widehat{\mathsf{ustat}_k}(\tau)) = \mathsf{ustat}_k(\tau)$, and that a Chernoff bound will determine $N = O(k^5 \cdot |2\Sigma|^{2k} \cdot \ln(\Sigma))$ with $k = 1/\varepsilon$, to insure that $|\mathsf{ustat}_k(\tau) - \widehat{\mathsf{ustat}_k}(\tau)| \leq \varepsilon$, with high probabilities.

DTD Embedding We now generalize the notion of base loop from words to trees. We associate a set of *base loops* τ_i to a DTD T, *i*.e. a set of minimal 2-extended tree τ_i in a Rabin Encoding with a distinguished leaf *compatible* with the root of τ_i, *i*.e. with the same label and free successors to accept iterations. If the root of τ_i has one left successor, the distinguished element of τ_i has a free left successor, and similarly for the right successor or both successors. The base loop τ_i has at least two nodes and the distinguished element is underlined, for example $\tau_i = a(b, \underline{a})$ or $a(., \underline{a})$. We define $(\tau_i)^m$ as the m-th iteration of the tree τ_i on the distinguished element. Let τ_a be a *terminal tree* with a root labeled a associated with a DTD, in a Rabin Encoding, *i*.e. a valid subtree for the label a and no label occurs twice in a path. For each base loop τ_i, a *derived loop* from τ_i is a base loop τ_i where some terminal trees τ_a are connected to possible nodes a of τ_i. There are finitely many distinct terminal trees τ_a for each letter a. With each base loop and derived loop, we associate a *base matrix* $t_i = lim_{n \to \infty} \mathsf{ustat}_k((\tau_i)^n)$. The set of $\mathsf{ustat}_k(\tau)$ for $\tau \in T$ is a union of polytope H_i^T which is the Convex-Hull $(\tau_1^*, ... \tau_l^*)$ of the base vectors of compatible base loops, restricted to some additional linear constraints. If $\mathsf{ustat}_k(\tau)$ is ε-close to H_i, then it is also close to $\sum_{s_i \in C} \lambda_i \cdot \tau_i^*$ where $C \subseteq \{\tau_1^*, \ldots, \tau_p^*\}$ of size at most $d_T + 1$, where d_T is the dimension of the vectors, *i*.e. $2^{k-1}. \mid \Sigma_T \mid$.

Example 3. Consider the DTD given by the four rules: $\{root: \ a^*b\,; \ \ a: \ c.d\,; \ \ c: \ a.f + g\,; \ \ b: \ e^*\}$. The base loops are: $\tau_1 = a(., \underline{a})$, $\tau_2 = e(., \underline{e})$, $\tau_3 = a(c(\underline{a}(., f), d), .)$, as the $''.''$ indicates the absence of successor. A terminal tree for a is $\tau_a = a(c(g, d), .)$ and a terminal tree for c is $\tau_c = c(g, .)$. A derived loop from τ_1 is $\tau_4 = a(c(g, d), \underline{a})$. On the unranked trees, the base loops are equivalent to: $a*$, $e*$ and $a(c(\underline{a}(., f), d), .)*$. The base matrices for $k = 2$, with the notation of sparse matrices, are:

t_1	0 1
aa	0 1

,

t_2	0 1
ee	0 1

,

t_3	0	1
ac	1/4	0
af	0	1/4
ca	1/4	0
cd	0	1/4

and

t_4	0	1
aa	1/4	0
ac	0	1/4
cd	1/4	0
cf	0	1/4

for the derived loop t_4.

If τ is ε-close to the DTD, then $\mathsf{ustat}_k(\tau)$ is ε-close to

$$H = \left\{ \lambda_1 \cdot t_1 + \lambda_2 \cdot t_2 + \lambda_3 \cdot t_3 + \lambda_4 \cdot t_4 \ \ \mid \ \ \sum_i \lambda_i = 1 \right\}.$$

ε-Membership Tester.

1. Sample τ (in a Rabin encoding) with $N \in O\left(\frac{|2\Sigma_S|^{2/\varepsilon} \ln(\Sigma_S)}{\varepsilon^5}\right)$ samples, and let $\widehat{\mathsf{ustat}_k(\tau)}$ be the estimation of $\mathsf{ustat}_k(\tau)$.
2. Enumerate all possible polytope H_T and Accept if one is ε-close to $\widehat{\mathsf{ustat}_k(\tau)}$, else Reject.

This shows that Membership is testable on unranked trees. The argument is similar to the case of words and the complexity is $O(1)$ for the size n of the tree τ but exponential in the size of the DTDs.

4 Approximate Structural Consistency

Fix a source schema S, a target schema T and parameters $k = 1/\varepsilon$ (the precision), α (the ratio) and m (the number of states of a transducer $\mathcal{T}$). Given an input (word or tree of size n) in S, we decide if there is a transducer $\mathcal{T}$ with m states, I' ε-close to I such that $\mathcal{T}(I')$ is ε-close to T and $\alpha \cdot n \leq |\mathcal{T}(I')| \leq n/\alpha$. We approximate the k statistics of I by sampling, consider some possible *base mappings* or *schema mappings* μ between the summits of H_S and the summits of H_T, and some compatible 1-state transducer π associated with μ. The important observation is that we can enumerate all possible μ, π in time independent of n. We then decide if there are m input mappings $\pi_1, ..., \pi_m$ defining $\mathcal{T}$ such that I is ε-close to $I' = I'_1.I'_2, ..., I'_m$ such that $\mathcal{T}(I') = \pi_1(I'_1), ..., \pi_m(I'_m)$ and $\alpha \cdot n \leq |\mathcal{T}(I')| \leq n/\alpha$. The total number of operations is independent of n.

Sampling and decomposition The embedding associates with S a finite set H^S of polytopes H^S_i, with summits $s_1, ..., s_p$. Each geometrical summit s_i is associated with a set U_i of base loops. $U_i = \{u^j_i\}$, and all u^j_i have the same ustat vectors, i.e. correspond to a summit s^j_i which coincide with the geometrical summit s_i. These loops cannot be decomposed as smaller loops, and are compatible, i.e. there is an input I which follows these loops. Similarly for T, we have a finite set of polytopes H^T_j with summits $t_1, ...t_q$. Given an input I (word w_n or tree t_n) close to some schema S, we first take N samples as before, and $x = \widehat{\mathsf{ustat}_k}(I)$. We decompose x on a simplex for some polytope H^S_i of H_S, i.e. $\mathsf{ustat}_k(I)$ is ε-close to $\sum_{s_i \in C} \lambda_i \cdot s_i$, where $\sum_i \lambda_i = 1$, for some $C \subseteq \{s_1, ...s_p\}$ of size at most $d_S + 1$, where d_S is the dimension of the source vectors. Observe that for large enough I, it is ε-close to $I' = \Pi_{i \in C} (u_i)^{\lambda'_i . n}$ for $\lambda'_i = \frac{\lambda_i}{|u_i|}$, as λ_i reflects the density of loops u_i. In order to obtain w' from w, some *moves* are applied to regroup all identical loops and non loops are deleted. In the decomposition, we can assume that each $\lambda_i > \frac{\varepsilon}{d_S} = c$. Otherwise we can find another input ε-close to I by deleting a few symbols and rounding the small coefficient to 0. The symbols are characters for the words and nodes for the trees.

Base mappings Let $\{s_1....s_p\}$ the set of summits of H^S_i, and $C \subseteq \{s_1, ...s_p\}$ of size at most $d_S + 1$, where d_S is the dimension of the source vectors. If $\mathsf{ustat}_k(w)$ is ε-close to H_i, then it is also close to $\sum_{s_i \in C} \lambda_i \cdot s_i$ by Caratheodory's theorem, and the λ_i are larger than a fixed value c. Similarly let H^T_j be one of the polytopes associated with the schema T and let $D \subseteq \{t_1, ...t_q\}$ of size at most $d_T + 1$, where d_T is the dimension of the target vectors.

Definition 2. *A (partial)* base mapping *μ between S and T is a partial function $H_i^S \rightarrow H_j^T$, only defined on some summits of the polytope, i.e. $\mu(s_i) = t_j$ for $s_i \in C$ and $t_j \in D$. A 1-state transducer π between S and T, is* compatible with *μ, if $\pi(u_i) = v_j$ if $\mu(s_i) = t_j$, $s_i = \mathsf{ustat}_k(u_i)$ and $t_j = \mathsf{ustat}_k(v_j)$.*

In the case of words, $\pi : \Sigma_S \rightarrow \Sigma_T^*$ and we talk about a π mapping. The domain of μ is C and the range is D. Each summit s_i corresponds to a base loop of the regular schema S, i.e. a minimal word u_i which can be iterated, and similarly each t_j corresponds to a base loop v_j for the regular schema T. Let $\alpha_i = \frac{|v_j|}{|u_i|}$ be the ratio between the length of the target loop and the source loop and $\lambda_i' = \frac{\lambda_i}{|u_i|}$. A μ-compatible mapping π is *(ε, α)-feasible for the decomposition $\sum_{i \in C} \lambda_i \cdot s_i$ on* C if there exists $w' = \Pi_{i \in C}(u_i)^{\lambda_i'.n}$, ε-close to w, such that $\alpha \leq \frac{\sum_{s_i \in C} \alpha_i.\lambda_i'}{\sum_{s_i \in C} \lambda_i'} \leq \frac{1}{\alpha}$.

4.1 Words

Notice that if $W = \pi(w')$ is the *source transformed by* π, then $\alpha.|w| \leq |W| \leq |w|/\alpha$. An (ε, α)-feasible μ-compatible mapping π yields directly a 1-state transducer.

Lemma 1. *If w is ε-close to S and there exists a μ-compatible mapping π which is (ε, α)-feasible, there exists an (ε, α)-feasible 1-state transducer $\mathcal{T}$ for w, S, T.*

Proof. If there is a μ-compatible mapping π , then $\mathsf{ustat}_k(w)$ is ε-close to $\sum_{s_i \in C} \lambda_i \cdot s_i$, where $\sum_i \lambda_i = 1$, for some $C \subseteq \{s_1, ...s_p\}$ of size at most $d_S + 1$. Observe that for w large enough, w is ε-close to $w' = \Pi_{i \in C}(u_i)^{\lambda_i'.n}$, as λ_i is the density of loops u_i in w and the number of iteration of each loop is $\lambda_i'.n = \frac{\lambda_i}{|u_i|}.n$ after some rounding. If we erase all the letters of w which are not loops, and apply moves to regroup all identical loops, we obtain w' ε-close to w. Let $\mathcal{T}$ the 1-state transducer associated with π, with transitions $a/\pi(a)$ for $a \in \Sigma_S$. By definition, α_i is the expansion on loop u_i and the total expansion on w' is less than α. Because $\mathsf{ustat}_k(w') = \sum_{i \in C} \lambda_i \cdot s_i$, $W = \pi(w')$ is such that $\mathsf{ustat}_k(\pi(w') = \sum_{i \in C} \lambda_i \cdot \mu(s_i) = \sum_{j \in D} \lambda_j^T \cdot t_j$ where $\lambda_j^T = \sum_{i \in C, \mu(s_i) = t_j} \lambda_i$, hence $\pi(w')$ is ε close to T.

Example 4. Let $k = 2 = 1/\varepsilon$ and $S = (001)^*.(01)^*.1^*.(011)^*$ with $\Sigma_S = \{0, 1\}$ as in the previous example, with summits $\{s_0, s_1, s_2, s_3\}$ associated with the simple loops $u_0 = (001), u_1 = (01), u_2 = 1, u_3 = (011)$ and $d_S = 2^2$. Let $T = (ab)^*.a^*.(aabc)^*$ with $\Sigma_T = \{a, b, c\}$, $d_T = 3^2$, and H_T be the polytope with summits $\{t_0, t_1, t_2\}$ associated with the simple loops $v_0 = ab$, $v_1 = a$ and $v_2 = aabc$. Let $w_1 = 0101111 = (01)^2.1^3 \in S$. Let $\mu(s_1) = t_0, \mu(s_2) = t_1$ as in figure **??** and let $\pi(0) = b, \pi(1) = a$, which is μ-compatible as $\pi(u_1) = \pi(01) = ba$, $\pi(u_2) = \pi(1) = a$, and $\mathsf{ustat}_2(\pi(u_1)^*) = \mathsf{ustat}_2((ba)^*) = \mathsf{ustat}_2((ab)^*) = \mathsf{ustat}_2((v_0)^*)$ and $\mathsf{ustat}_2(\pi(u_2)^*) = \mathsf{ustat}_2((v_1)^*)$. In this case, the 1-state transducer $\mathcal{T}$ is such that $\mathcal{T}(0) = b$ and $\mathcal{T}(1) = b$, and $(0,1)$-feasible for w_1, i.e. $\alpha = 1$, and $\varepsilon = 0$. If we consider $w = 000111$, $1/6$-close to $w' = 00111 \in S$, $\mathcal{T}(w') = bbaaa$, at distance $2/5$ from T and $\alpha = 5/6$. Therefore $\mathcal{T}$ is $(2/5, 5/6)$-feasible for w.

We now generalize to transducers with m states and consider m distinct base mappings $\mu_1, ...\mu_m$, and μ-compatible mappings $\pi_1, ...\pi_m$ for the same H_i^S and H_j^T. We first describe a Verification Algorithm, which given $\pi_1, ...\pi_m$, C a subset of summits of H_i^S, D a subset of summits of H_j^T, such that $\mathsf{ustat}_k(w)$ is ε-close to $\sum_{s_i \in C} \lambda_i \cdot s_i$, where $\sum_i \lambda_i = 1$, decides if there is an (ε, α)-feasible transducer $\mathcal{T}$ with m states for w, C, D. We can find w' ε-close to w, such that $w' = \Pi_{i \in C}(u_i)^{\lambda'_i . n}$ for $\lambda'_i = \frac{\lambda_i}{|u_i|}$ and can also decompose w' into m components $w'_1, ...w'_m$, i.e. $w'' = w'_1, ...w'_m = (\Pi_{i \in C}(u_i)^{\lambda_i^{'1.n}})_1 ....(\Pi_{i \in C}(u_i)^{\lambda_i^{'m.n}})_m$ such that $\sum_{j=1...m} \lambda_i^{'j} = \lambda'_i$. We divide each λ'_i into positive $\lambda_i^{'j}$ associated with the mapping π_j for $j = 1, ...m$ and some of the $\lambda_i^{'j}$ are 0. Each π_j gives an expansion α_i^j for each loop u_i such that $s_i \in C$. The general expansion on u_i is $\alpha_i = \frac{\sum_{j=1,...m} \lambda_i^{'j} . \alpha_i^j}{\lambda'_i}$ and the global expansion is $\alpha_g = \frac{\sum_{s_i \in C} \lambda'_i . \alpha_i}{\sum_{s_i \in C} \lambda'_i}$ which is either larger or smaller than 1. We can then write two Linear programs with positive variables $\{\lambda_i^{'j}, \alpha_i, \alpha_g\}$, whereas λ'_i, λ_i and α_i^j are constants, and $s_i \in C$, one for the case $\alpha_g \leq 1$ and the other for the case $\alpha_g \geq 1$:

Linear Programs $P(C, \lambda_i)$

$Min\ (1 - \alpha_g) \geq 0$ [case of $\alpha_g \leq 1$]

$Min\ (\alpha_g - 1) \geq 0$ [case of $\alpha_g \geq 1$]

$$\alpha_i = \frac{\sum_{j=1,...m} \lambda_i^{'j} . \alpha_i^j}{\lambda'_i}, \quad \alpha_g = \frac{\sum_{s_i \in C} \lambda'_i . \alpha_i}{\sum_{s_i \in C} \lambda'_i}, \quad \lambda'_i = \frac{\lambda_i . n}{| u_i |}, \quad \sum_{j=1...m} \lambda_i^{'j} = \lambda'_i.$$

Let $\mathcal{T}_m$ be the transducer with m states operating on w'', ε-close to w, i.e. applying π_j on w'_j in state j, for $j = 1, ...m$. We solve both linear systems and compare the parameter α_g to α to decide if the transducer $\mathcal{T}_m$ is (ε, α)-feasible.

Lemma 2. *If the solution of the linear programs P is such that $\alpha \leq \alpha_g \leq 1$ or $1 \leq \alpha_g \leq \frac{1}{\alpha}$, then $\mathcal{T}_m$ is (ε, α)-feasible for large enough inputs.*

Proof. Notice that w is ε-close to $w' = \Pi_{i \in C}(u_i)^{\lambda'_i}$ for $\lambda'_i = \frac{\lambda_i}{|u_i|}$ and to

$$w'' = w'_1, ...w'_m = (\Pi_{i \in C}(u_i)^{\lambda_i^{'1}.n})_1 ....(\Pi_{i \in C}(u_i)^{\lambda_i^{'m}}.n)_m \text{ such that } \sum_{j=1...m} \lambda_i^{'j} = \lambda'_i.$$

The solution of the linear program, if it exists, insures that the expansion factor is within α but may yield non integer values to the $\lambda_i^{'j}$. We round to the closest integer modifying slightly w'' into w''', which is still ε-close to w. By construction the image $\mathcal{T}_m(w''')$ is in T and the transducer $\mathcal{T}_m$ is (ε, α)-feasible.

Verification Algorithm $A(w, C, D, \pi_1, ...\pi_m)$.

1. Decompose $\widehat{\mathsf{ustat}_k(w)}$ (which approximates $\mathsf{ustat}_k(w)$) ε-close to $\sum_{i \in C} \lambda_i \cdot s_i$, otherwise reject.

2. Solve the linear programs $P(C, \lambda_i)$.
3. If $\alpha \leq \alpha_g \leq 1$ or $1 \leq \frac{1}{\alpha_g} \leq \frac{1}{\alpha}$, accept else reject.

Tester for the Existence of an (ε, α)**-feasible transducer**: $TE(w, S, T, \varepsilon, \alpha, m)$.

1. Sample w with $N \in O(\frac{|\Sigma_S|^{2/\varepsilon} \cdot \ln|\Sigma_S|}{\varepsilon^3})$ samples, and let $\widehat{\mathsf{ustat}_k(w)}$ be the estimation of $\mathsf{ustat}_k(w)$.
2. Choose a possible polytope H_S, ε-close to $\widehat{\mathsf{ustat}_k(w)}$.
3. Enumerate all possible C, D, all possible base mappings μ, and all μ-feasible π. Accept if one $A(w, C, D, \pi_1, ...\pi_m)$ accepts, else Reject.

Theorem 1. *If there exists an (ε, α)-feasible transducer with at most m states, then $TE(w, S, T, \varepsilon, \alpha, m)$ accepts. If w is ε-far from any w' such that there exists an (ε, α)-feasible transducer with at most m states, then $TE(w, S, T, \varepsilon, \alpha, m)$ rejects with high probabilities.*

Proof. If there exists an (ε, α)-feasible transducer for w, it must transform a simple loop of S into a simple loop of T. Otherwise if $w = (u_i)^j$, its image may be far from T. Therefore each state corresponds to some base mapping μ and to some μ-compatible mapping π. We generate all possible mappings for m states. Because each $\lambda_i > c$, the number of possible mappings π is independent of n, and only depends on ε and the dimensions. For the right choice, the Verification will accept and so will do the Tester TE. If w is ε-far from any w' for which there exists an (ε, α)-feasible transducer for w, then either $\mathsf{ustat}_k(w)$ is ε-far from any polytope H_S and this condition is detected with high probability from $\widehat{\mathsf{ustat}_k(w)}$, or $\mathsf{ustat}_k(w)$ is ε-close to a polytope H_S and to some $\sum_{i \in C} \lambda_i \cdot s_i$ but no mapping can map simple loops of s_i to simple loops of t_j. This last condition is detected as we analyze all possibilities.

4.2 Trees

Recall that we associate a union of polytopes to a DTD. Let $C \subseteq \{s_1, ...s_p\}$ be a polytope described by its summits. To each summit s_i corresponds a set of base loops τ_i, i.e. extended binary trees (in a Rabin encoding) which can be iterated, with the same statistics. As in definition 2, a (partial) base mapping μ between two schemas S and T is a partial function $H_i^S \to H_j^T$, only defined on some summits of the polytopes, i.e. $\mu(s_i) = t_j$ for $s_i \in C$ and $t_j \in D$. A 1-state transducer π between S and T, is *compatible with* μ, if $\pi(\tau_i) = \tau_j$ if $\mu(s_i) = t_j$, $\mathsf{ustat}_k(\tau_i) = s_i$ and $\mathsf{ustat}_k(\tau_j) = t_j$. In this case π transforms trees. We follow an approach similar to the word case. We first estimate $\widehat{\mathsf{ustat}_k(\tau)}$ which approximates $\mathsf{ustat}_k(\tau)$. If τ is close to S, then $\mathsf{ustat}_k(\tau)$ has a decomposition on a polytope H^S, i.e. $\mathsf{ustat}_k(\tau)$ is close to $\sum_{i \in C} \lambda_i \cdot s_i$ for some summits C of H, where s_i are the base matrices. The tree τ is close to $\tau' = \Pi_{i \in C}(t_i)^{\lambda'_i.n}$ for n large enough, where each base or derived loop τ_i is iterated $\lambda'_i.n$ times after some rounding, as we regroup similar loops with moves and erase the other subtrees, for $\lambda'_i = \frac{\lambda_i}{|t_i|}$. For a given π, let $\alpha_i = \frac{|t^m_j|}{|t^l_i|}$ the ratio between the length of the target loop and the source loop. A μ-compatible mapping π is ε, α*-feasible* for the decomposition $\sum_{i\in C} \lambda_i \cdot s_i$ on C if there exists τ' ε-close to τ such that $\frac{\sum_{s_i\in C} \alpha_i.\lambda'_i}{\sum_{s_i\in C} \lambda'_i} \leq \alpha$ and $\alpha.|\tau| \leq |\pi(\tau')| \leq |\tau|/\alpha$.

Example 5. Consider the following source **S**, target **T** DTDs and $\epsilon = \frac{1}{2}$ fixed, *i*.e. $k = 2$. **S**

```
<!ELEMENT bd (work*)>
<!ELEMENT work (author+)>
<!ATTLIST work title CDATA year CDATA>
<!ELEMENT author (EMPTY)>
<!ATTLIST author name  CDATA #REQUIRED>
```

T

```
<!ELEMENT bib (livre*,editeur)>
<!ELEMENT livre (titre, auteur+)>
<!ELEMENT auteur #PCDATA>
<!ELEMENT titre #PCDATA>
```

We use the standard abbreviations of the tags, where *bd* and *bib* are abbreviated by *b*. Let τ be a large tree of **S** and assume that sampling τ gives us a matrix: $\widehat{\mathsf{ustat}_2(\tau)}$. Let us explicit a $(1/2, 3/4)$-feasible one state transducer. The loops of the schema S are $\tau_1 = w(a, \underline{w})$ and $\tau_2 = a(., \underline{a})$. For $k = 2$, their statistical representation are s_1 and s_2. The schema **T** has two loops: $l(t(., a), \underline{l})$ and $a(., \underline{a})$, whose statistical representations are t'_1 and t'_2 :

s_1	0	1
wa	1/2	0
ww	0	1/2

,

s_2	0	1
aa	0	1

,

$\widehat{\mathsf{ustat}_2(\tau)}$	0	1
aa	0	0.59
bw	0.01	0
wa	0.2	0
ww	0	0.2

,

t'_1	0	1
ll	0	1/3
lt	1/3	0
ta	0	1/3

and

t'_2	0	1
aa	0	1

.

Take the base mapping $\mu(s_1) = t'_1$ and $\mu(s_2) = t'_2$. A possible admissible one state transducer π compatible with μ given below by the transition rules in a compact formalism : $(q, b) \to b(\underline{q}, .)$; $(q, w) \to l(t, \underline{q})$; $(q, a) \to a, \underline{q}$. Since the nodes 'bd', 'bib' and 'editeur' have a small impact on the statistics for big trees of **S** and **T**, only loops are considered. Decomposed over H_S, $\widehat{\mathsf{ustat}_2} \approx 0.6 \cdot s_1 + 0.4 \cdot s_2$. The distortion produced by the second rule is 3/2 and the third rule preserves size *i*.e. the total distortion is $0.6 \times \frac{3}{2} + 0.4 \times 1 = 1.3$ and it's inverse (≈ 0.76923) is higher that 3/4.

The generalization to transducers with m states is similar to the case of words. We consider m distinct base mappings $\mu_1, ...\mu_m$, μ-compatible mappings $\pi_1, ...\pi_m$ and decompose the tree τ into a forest with m components $\tau'_1, ...\tau'_m$, i.e. $\tau'' = \tau'_1, ...\tau'_m = (\Pi_{i\in C}(t_i)^{\lambda'^1_i})_1....(\Pi_{i\in C}(\tau_i)^{\lambda'^m_i})_m$ such that $\sum_{j=1...m} \lambda'^j_i = \lambda'_i$. The forest τ'' is ε-close to τ, as we apply a limited number of moves. We use a linear program $P'(C, \lambda_i)$ to decide if we can find the λ'^j_i, and a verification algorithm $A'(t, S, T; C, \pi_1, ...\pi_m)$, as in the case of words. We can use the same Verification algorithm and the Tester but for $N \in O\left(\frac{|2\Sigma_S|^{2/\varepsilon} \cdot \ln(|\Sigma_S|)}{\varepsilon^5}\right)$ samples. The number of possible C is bounded by the dimension $|\Sigma_S|^k \cdot 2^{k-1}$, and the number of possible μ is also bounded. We need to bound the number of possible π, as in the case of words by the following lemma:

Lemma 3. *If there exists an* ε, α*-feasible* μ*-compatible mapping* π*, then* $|\pi(a)| \leq \frac{1}{c\cdot\alpha}$ *for each letter* $a \in \Sigma_S$*.*

Proof. Recall that as in the case of words, the λ_i coefficients of the decomposition can be supposed greater than a constant $c < 1$. If the expansion $|\pi(a)|$ was larger than $\frac{1}{c \cdot \alpha}$, the global expansion would be larger than α.

We can finally state our main result:

Theorem 2. *If there exists an (ε, α)-feasible transducer with at most m states, then $TES(\tau, S, T, \varepsilon, \alpha, m)$ accepts. If τ is ε-far from any τ' such that there exists an (ε, α)-feasible transducer with at most m states, then $TES(\tau, S, T, \varepsilon, \alpha, m)$ rejects with high probabilities.*

5 Conclusion

The approximate embedding of trees and tree languages proposed in this paper gives an efficient solution to decide Approximate Structural Consistency, as the complexity of the algorithms only depends on the accuracy parameters. The methods are also robust to some noise ratio, as the statistics matrices are close on close inputs. We did not specify the exact complexity of the algorithms as a function of the size of the DTD and leave it as an open problem. Structural Consistency can also be applied to documents which do not have a schema, such as data words or streams but an input schema guarantees a much smaller number of potential mappings. General problems in formal languages and rewriting systems are often hard in their exact versions and approximate solutions are natural.

References

1. A. Broder. On the resemblance and containment of documents. In *SEQUENCES '97: Proceedings of the Compression and Complexity of Sequences 1997*, 1997.
2. M. de Rougemont and A. Vieilleribière. Approximate data exchange. In *International Conference on Database Technology (ICDT)*, pages 44–58, 2007.
3. Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian P opa. Data exchange: Semantics and query answering. In *ICDT '03: Proceedings of the 9th International Conference on Database Theory*, pages 207–224, London, UK, 2002. Springer-Verlag.
4. E. Fischer, F. Magniez, and M. de Rougemont. Approximate satisfiability and equivalence. In *IEEE Logic in Computer Science*, pages 421–430, 2006.
5. 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.
6. F. Magniez and M. de Rougemont. Property testing of regular tree languages. In *International Conference on Automata Languages and Programming (ICALP)*, pages 932–944, 2004.
7. W. Martens and F. Neven. Typechecking top-down uniform unranked tree transducers. In *International Conference on Database Theory*, pages 64–78, 2002.
8. R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):23–32, 1996.