Lipschitz Robustness of Finite-state Transducers[*]

Thomas A. Henzinger, Jan Otop and Roopsha Samanta

IST Austria
`{tah,jotop,rsamanta}@ist.ac.at`

Abstract

We investigate the problem of checking if a finite-state transducer is *robust* to uncertainty in its input. Our notion of robustness is based on the analytic notion of Lipschitz continuity – a transducer is K-(Lipschitz) robust if the perturbation in its output is at most K times the perturbation in its input. We quantify input and output perturbation using *similarity functions*. We show that K-robustness is undecidable even for deterministic transducers. We identify a class of functional transducers, which admits a polynomial time automata-theoretic decision procedure for K-robustness. This class includes Mealy machines and functional letter-to-letter transducers. We also study K-robustness of nondeterministic transducers. Since a nondeterministic transducer generates a set of output words for each input word, we quantify output perturbation using *set-similarity functions*. We show that K-robustness of nondeterministic transducers is undecidable, even for letter-to-letter transducers. We identify a class of set-similarity functions which admit decidable K-robustness of letter-to-letter transducers.

1998 ACM Subject Classification D.2.4 Software/Program Verification, F.1.1 Models of Computation

Keywords and phrases Robustness; Transducers; Weighted Automata

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2014.431

1 Introduction

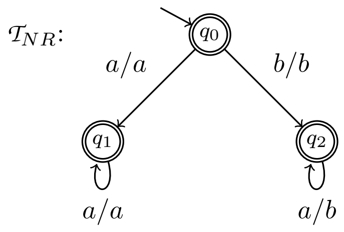
Most computational systems today are embedded in a physical environment. The data processed by such real-world computational systems is often noisy or uncertain. For instance, the data generated by sensors in reactive systems such as avionics software may be corrupted, keywords processed by text processors may be wrongly spelt, the DNA strings processed in computational biology may be incorrectly sequenced, and so on. In the presence of such input uncertainty, it is not enough for a computational system to be functionally correct. An additional desirable property is that of *continuity* or *robustness* — the system behaviour degrades smoothly in the presence of input disturbances [14].

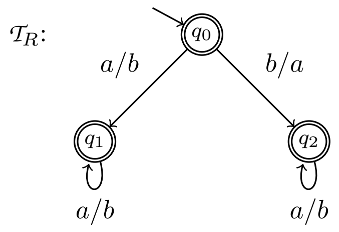
Well-established areas within control theory, such as *robust control* [16], extensively study robustness of systems. However, their results typically involve reasoning about continuous state-spaces and are not directly applicable to inherently discontinuous discrete computational systems. Moreover, uncertainty in robust control refers to differences between a system's model and the actual system; thus robust control focuses on designing controllers that function properly in the presence of perturbation in various internal parameters of a system's model. Given the above, formal reasoning about robustness of computational systems under input uncertainty is a problem of practical as well as conceptual importance.

[*] This work was supported in part by the European Research Council (ERC) under grant agreement 267989 (QUAREM) and by the Austrian Science Fund (FWF) NFN project S11402-N23 (RiSE).

In our work, we focus on robustness of finite-state transducers, processing finite or infinite words, in the presence of uncertain inputs. Transducers are popular models of input-output computational systems operating in the real world [13, 20, 3, 24]. While many decision problems about transducers have been studied thoroughly over the decades [20, 24], their behaviour under uncertain inputs has only been considered recently [22]. In [22], a transducer was defined to be robust if its output changed proportionally to every change in the input *up to a certain threshold*. In practice, it may not always be possible to determine such a bound on the input perturbation. Moreover, the scope of the work in [22] was limited to the robustness problem for *functional* transducers w. r. t. specific distance functions, and did not consider arbitrary nondeterministic transducers or arbitrary *similarity functions*.

In this paper, we formalize robustness of finite-state transducers as Lipschitz continuity. A function is Lipschitz-continuous if its output changes proportionally to *every* change in the input. Given a constant K and similarity functions d_Σ, d_Γ for computing the input, output perturbation, respectively, a functional transducer $\mathcal{T}$ is defined to be K-Lipschitz robust (or simply, K-robust) w. r. t. d_Σ, d_Γ if for all words s, t in the domain of $\mathcal{T}$ with finite $d_\Sigma(s,t)$, $d_\Gamma(\mathcal{T}(s), \mathcal{T}(t)) \leq K d_\Sigma(s,t)$. Let us consider the transducers $\mathcal{T}_{NR}$ and $\mathcal{T}_R$ below. Recall that the Hamming distance between equal length words is the number of positions in which the words differ. Let d_Σ, d_Γ be computed as the Hamming distance for equal-length words, and be ∞ otherwise. Notice that for words a^{k+1}, ba^k in the domain of the Mealy machine $\mathcal{T}_{NR}$, $d_\Sigma(a^{k+1}, ba^k) = 1$ and the distance between the corresponding output words, $d_\Gamma(a^{k+1}, b^{k+1})$, equals $k+1$. Thus, $\mathcal{T}_{NR}$ is not K-robust for any K. On the other hand, the transducer $\mathcal{T}_R$ is 1-robust: for words a^{k+1}, ba^k, we have $d_\Sigma(a^{k+1}, ba^k) = d_\Gamma((b)^{k+1}, a(b)^k) = 1$, and for all other words s, t in the domain of $\mathcal{T}_R$, either $d_\Sigma(s,t) = \infty$ or $d_\Sigma(s,t) = d_\Gamma(\mathcal{T}_R(s), \mathcal{T}_R(t)) = 0$.





While the K-robustness problem is undecidable even for deterministic transducers, we identify interesting classes of finite-state transducers with decidable K-robustness. We first define a class of functional transducers, called synchronized transducers, which admits a polynomial time decision procedure for K-robustness. This class includes Mealy machines and functional letter-to-letter transducers; membership of a functional transducer in this class is decidable in polynomial time. Given similarity functions computable by weighted automata, we reduce the K-robustness problem for synchronized transducers to the emptiness problem for weighted automata.

We extend our decidability results by employing an *isometry approach.* An *isometry* is a transducer, which for all words s, t satisfies $d_\Gamma(\mathcal{T}(s), \mathcal{T}(t)) = d_\Sigma(s,t)$. We observe that if a transducer $\mathcal{T}_2$ can be obtained from a transducer $\mathcal{T}_1$ by applying isometries to the input and output of $\mathcal{T}_1$, then K-robustness of $\mathcal{T}_1$ and $\mathcal{T}_2$ coincide. This observation enables us to reduce K-robustness of various transducers to that of synchronized transducers.

Finally, we study K-robustness of nondeterministic transducers. Since a nondeterministic transducer generates a set of output words for each input word, we quantify output perturbation using *set-similarity functions* and define K-robustness of nondeterministic transducers w. r. t. such set-similarity functions. We show that K-robustness of nondeterministic transducers is undecidable, even for letter-to-letter transducers. We define three classes of set-similarity functions and show decidability of K-robustness of nondeterministic letter-to-letter transducers w. r. t. one class of set-similarity functions.

In what follows, we first present necessary definitions in Sec. 2. We formalize Lipschitz robustness in Sec. 3. In Sec. 4 and Sec. 5, we study the K-robustness problem for functional transducers, showing undecidability of the general problem and presenting two classes with decidable K-robustness. We study K-robustness of arbitrary nondeterministic transducers in Sec. 6, present a discussion of related work in Sec. 7 and conclude in Sec. 8.

2 Preliminaries

In this section, we review definitions of finite-state transducers and weighted automata, and present similarity functions. We use the following notation. We denote input letters by a, b etc., input words by s, t etc., output letters by a', b' etc. and output words by s', t' etc. We denote the concatenation of words s and t by $s \cdot t$, the i^{th} letter of word s by $s[i]$, the subword $s[i] \cdot s[i+1] \cdot \ldots \cdot s[j]$ by $s[i,j]$, the length of the word s by $|s|$, and the empty word and empty letter by ϵ. Note that for an ω-word s, $|s| = \infty$.

Finite-state Transducers. A finite-state transducer (FST) $\mathcal{T}$ is a tuple $(\Sigma, \Gamma, Q, Q_0, E, F)$ where Σ is the input alphabet, Γ is the output alphabet, Q is a finite nonempty set of states, $Q_0 \subseteq Q$ is a set of initial states, $E \subseteq Q \times \Sigma \times \Gamma^* \times Q$ is a set of transitions[1], and F is a set of accepting states.

A run γ of $\mathcal{T}$ on an input word $s = s[1]s[2]\ldots$ is defined in terms of the sequence: (q_0, w'_1), (q_1, w'_2), $\ldots$ where $q_0 \in Q_0$ and for each $i \in \{1, 2, \ldots\}$, $(q_{i-1}, s[i], w'_i, q_i) \in E$. Let $\mathrm{Inf}(\gamma)$ denote the set of states that appear infinitely often along γ. For an FST $\mathcal{T}$ processing ω-words, a run is accepting if $\mathrm{Inf}(\gamma) \cap F \neq \emptyset$ (Büchi acceptance condition). For an FST $\mathcal{T}$ processing finite words, a run γ: (q_0, w'_1), $\ldots$ (q_{n-1}, w'_n), (q_n, ϵ) on input word $s[1]s[2]\ldots s[n]$ is accepting if $q_n \in F$ (final state acceptance condition). The output of $\mathcal{T}$ along a run is the word $w'_1 \cdot w'_2 \cdot \ldots$ if the run is accepting, and is undefined otherwise. The transduction computed by an FST $\mathcal{T}$ processing infinite words (resp., finite words) is the relation $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$ (resp., $[\![\mathcal{T}]\!] \subseteq \Sigma^* \times \Gamma^*$), where $(s, s') \in [\![\mathcal{T}]\!]$ iff there is an accepting run of $\mathcal{T}$ on s with s' as the output along that run. With some abuse of notation, we denote by $[\![\mathcal{T}]\!](s)$ the set $\{t : (s,t) \in [\![\mathcal{T}]\!]\}$. The input language, $\mathrm{dom}(\mathcal{T})$, of $\mathcal{T}$ is the set $\{s : [\![\mathcal{T}]\!](s)$ is non-empty$\}$.

An FST $\mathcal{T}$ is called *functional* if the relation $[\![\mathcal{T}]\!]$ is a function. In this case, we use $[\![\mathcal{T}]\!](s)$ to denote the unique output word generated along any accepting run of $\mathcal{T}$ on input word s. Checking if an arbitrary FST is functional can be done in polynomial time [12]. An FST $\mathcal{T}$ is *deterministic* if $\forall q \in Q, a \in \Sigma$: $|\{q' : (q, a, w', q') \in E\}| \leq 1$. An FST $\mathcal{T}$ is a *letter-to-letter* transducer if for every transition of the form $(q, a, w', q') \in E$, $|w'| = 1$. A *Mealy machine* is a deterministic, letter-to-letter transducer, with every state being an accepting state. In what follows, we use transducers and finite-state transducers interchangeably.
Composition of transducers. Consider transducers $\mathcal{T}_1 = (\Sigma, \Delta, Q_1, Q_{1,0}, E_1, F_1)$ and $\mathcal{T}_2 = (\Delta, \Gamma, Q_2, Q_{2,0}, E_2, F_2)$ such that for every $s \in \mathrm{dom}(\mathcal{T}_1)$, $[\![\mathcal{T}]\!](s) \in \mathrm{dom}(\mathcal{T}_2)$. We define $\mathcal{T}_2 \circ \mathcal{T}_1$, the *composition* of $\mathcal{T}_1$ and $\mathcal{T}_2$, as the transducer $(\Sigma, \Gamma, Q_1 \times Q_2, Q_{1,0} \times Q_{2,0}, E, F_1 \times F_2)$, where E is defined as follows: $(\langle q_1, q_2 \rangle, a, w', \langle q'_1, q'_2 \rangle) \in E$ iff $(q_1, a, t', q'_1) \in E_1$ and upon reading t', $\mathcal{T}_2$ generates w' and changes state from q_2 to q'_2, i.e., iff $(q_1, a, t', q'_1) \in E_1$ and

[1] Note that we disallow ϵ-transitions where the transducer can change state without moving the reading head.

there exist $(q_2, t'[1], w'_1, q_2^1)$, $(q_2^1, t'[2], w'_2, q_2^2)$, …, $(q_2^{k-1}, t'[k], w'_k, q'_2) \in E_2$ such that $k = |t'|$ and $w' = w'_1 \cdot w'_2 \cdot \ldots \cdot w'_k$. Observe that if $\mathcal{T}_1, \mathcal{T}_2$ are functional, $\mathcal{T}_2 \circ \mathcal{T}_1$ is functional and $[\![\mathcal{T}_2 \circ \mathcal{T}_1]\!] = [\![\mathcal{T}_2]\!] \circledcirc [\![\mathcal{T}_1]\!]$, where $\circledcirc$ denotes function composition.

Weighted automata. Recall that a finite automaton (with Büchi or final state acceptance) can be expressed as a tuple (Σ, Q, Q_0, E, F), where Σ is the alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is a set of initial states, $E \subseteq Q \times \Sigma \times Q$ is a transition relation, and $F \subseteq Q$ is a set of accepting states. A weighted automaton (WA) is a finite automaton whose transitions are labeled by rational numbers. Formally, a WA $\mathcal{A}$ is a tuple $(\Sigma, Q, Q_0, E, F, c)$ such that (Σ, Q, Q_0, E, F) is a finite automaton and $c : E \mapsto \mathbb{Q}$ is a function labeling the transitions of $\mathcal{A}$. The transition labels are called *weights*.

Recall that a run π of a finite automaton on a word $s = s[1]s[2]\ldots$ is defined as a sequence of states: $q_0, q_1, \ldots$ where $q_0 \in Q_0$ and for each $i \in \{1, 2, \ldots\}$, $(q_{i-1}, s[i], q_i) \in E$. A run π in a finite automaton processing ω-words (resp., finite words) is accepting if it satisfies the Büchi (resp., final state) acceptance condition. The set of accepting runs of an automaton on a word s is denoted $\mathsf{Acc}(s)$. Given a word s, every run π of a WA $\mathcal{A}$ on s defines a sequence $c(\pi) = (c(q_{i-1}, s[i], q_i))_{1 \le i \le |s|}$ of weights of successive transitions of $\mathcal{A}$; such a sequence is also referred to as a weighted run. To define the semantics of weighted automata we need to define the value of a run (that combines the sequence of weights of the run into a single value) and the value across runs (that combines values of different runs into a single value). To define values of runs, we consider *value functions* f that assign real numbers to sequences of rational numbers, and refer to a WA with a particular value function f as an f-WA. Thus, the value $f(\pi)$ of a run π of an f-WA $\mathcal{A}$ on a word s equals $f(c(\pi))$. The value of a word s assigned by an f-WA $\mathcal{A}$, denoted $\mathcal{L}_{\mathcal{A}}(s)$, is the *infimum* of the set of values of all accepting runs, i.e., $\mathcal{L}_{\mathcal{A}}(s) = \inf_{\pi \in \mathsf{Acc}(s)} f(\pi)$ (the infimum of an empty set is infinite).

In this paper, we consider the following value functions: (1) the sum function $\textsc{Sum}(\pi) = \sum_{i=1}^{|\pi|}(c(\pi))[i]$, (2) the discounted sum function $\textsc{Disc}_\delta(\pi) = \sum_{i=1}^{|\pi|} \delta^i (c(\pi))[i]$ with $\delta \in (0, 1)$ and (3) the limit-average function $\textsc{LimAvg}(\pi) = \limsup_{k \to \infty} \frac{1}{k} \sum_{i=1}^{k}(c(\pi))[i]$. Note that the limit-average value function cannot be used with finite sequences. We define $\textsc{ValFunc} = \{\textsc{Sum}, \textsc{Disc}_\delta, \textsc{LimAvg}\}$.

A WA $\mathcal{A}$ is *functional* iff for every word s, all accepting runs of $\mathcal{A}$ on s have the same value.

Decision questions. Given an f-WA $\mathcal{A}$ and a threshold λ, the *emptiness* question asks whether *there exists* a word s such that $\mathcal{L}_{\mathcal{A}}(s) < \lambda$ and the *universality* question asks whether *for all* words s we have $\mathcal{L}_{\mathcal{A}}(s) < \lambda$. The following results are known.

▶ **Lemma 1.** (1) *For every $f \in \textsc{ValFunc}$, the emptiness problem is decidable in polynomial time for nondeterministic f-automata [11, 10].* (2) *The universality problem is undecidable for* $\textsc{Sum}$*-automata with weights drawn from* $\{-1, 0, 1\}$ *[17, 1].*

▶ Remark. Weighted automata have been defined over semirings [10] as well as using value functions (along with infimum or supremum) as above [5, 6]. These variants of weighted automata have incomparable expression power. We use the latter definition as it enables us to express long-run average and discounted sum, which are inexpressible using weighted automata over semirings. Long-run average and discounted sum are widely used in quantitative verification and define natural distances (Example 9). Moreover, unlike the semiring-based definition, the value-function-based definition extends easily from finite to infinite words.

Similarity Functions. In our work, we use similarity functions to measure the similarity between words. Let $\mathbb{Q}^\infty$ denote the set $\mathbb{Q} \cup \{\infty\}$. A similarity function $d : S \times S \to \mathbb{Q}^\infty$ is a function with the properties: $\forall x, y \in S$: (1) $d(x, y) \geq 0$ and (2) $d(x, y) = d(y, x)$. A similarity function d is also a distance (function or metric) if it satisfies the additional properties: $\forall x, y, z \in S$: (3) $d(x, y) = 0$ iff $x = y$ and (4) $d(x, z) \leq d(x, y) + d(y, z)$. We emphasize that in our work we do not need to restrict similarity functions to be distances.

An example of a similarity function is the *generalized Manhattan distance* defined as: $d_M(s, t) = \sum_{i=1}^{\infty} \texttt{diff}(s[i], t[i])$ for infinite words s, t, where `diff` is the mismatch penalty for substituting letters. For finite words s, t, $d_M(s, t) = \sum_{i=1}^{max(|s|,|t|)} \texttt{diff}(s[i], t[i])$. The mismatch penalty is required to be a distance function on the alphabet (extended with a special end-of-string letter # for finite words). When $\texttt{diff}(a, b)$ is defined to be 1 for all a, b with $a \neq b$, and 0 otherwise, d_M is called the *Manhattan distance*.

Notation: We use $s_1 \otimes \ldots \otimes s_k$ to denote *convolution* of words $s_1, \ldots, s_k$, for $k > 1$. The convolution of k words merges the arguments into a single word over a k-tuple alphabet (accommodating arguments of different lengths using # letters at the ends of shorter words). Let $s_1, \ldots, s_k$ be words over alphabets $\Sigma_1, \ldots, \Sigma_k$. Let $\Sigma_1 \otimes \ldots \otimes \Sigma_k$ denote the k-tuple alphabet $(\Sigma_1 \cup \{\#\}) \times \ldots \times (\Sigma_k \cup \{\#\})$. The convolution $s_1 \otimes \ldots \otimes s_k$ is an infinite word (resp., a finite word of length $max(|s_1|, \ldots, |s_k|)$), over $\Sigma_1 \otimes \ldots \otimes \Sigma_k$, such that: for each $i \in \{1, \ldots, |s_1 \otimes \ldots \otimes s_k|\}$, $(s_1 \otimes \ldots \otimes s_k)[i] = \langle s_1[i], \ldots, s_k[i]\rangle$ (with $s_j[i] = \#$ if $i > |s_j|$). For example, the convolution $aa \otimes b \otimes add$ is the 3 letter word $\langle a, b, a\rangle\langle a, \#, d\rangle\langle \#, \#, d\rangle$.

▶ **Definition 2** (Automatic Similarity Function). A similarity function $d : \Sigma_1^\omega \times \Sigma_2^\omega \mapsto \mathbb{Q}$ is called automatic if there exists a WA $\mathcal{A}_d$ over $\Sigma_1 \otimes \Sigma_2$ such that $\forall s_1 \in \Sigma_1^\omega, s_2 \in \Sigma_2^\omega$: $d(s_1, s_2) = \mathcal{L}_{\mathcal{A}_d}(s_1 \otimes s_2)$. We say that d is computed by $\mathcal{A}_d$.

One can similarly define automatic similarity functions over finite words.

3 Problem Definition

Our notion of robustness for transducers is based on the analytic notion of Lipschitz continuity. We first define K-Lipschitz robustness of functional transducers.

▶ **Definition 3** (K-Lipschitz Robustness of Functional Transducers). Given a constant $K \in \mathbb{Q}$ with $K > 0$ and similarity functions $d_\Sigma : \Sigma^\omega \times \Sigma^\omega \to \mathbb{Q}^\infty$ (resp., $d_\Sigma : \Sigma^* \times \Sigma^* \to \mathbb{Q}^\infty$) and $d_\Gamma : \Gamma^\omega \times \Gamma^\omega \to \mathbb{Q}^\infty$ (resp., $d_\Gamma : \Gamma^* \times \Gamma^* \to \mathbb{Q}^\infty$), a functional transducer $\mathcal{T}$, with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$ (resp., $[\![\mathcal{T}]\!] \subseteq \Sigma^* \times \Gamma^*$,), is called K-Lipschitz robust w. r. t. d_Σ, d_Γ if:

$$\forall s, t \in \mathrm{dom}(\mathcal{T}) : \; d_\Sigma(s, t) < \infty \Rightarrow d_\Gamma([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) \leq K d_\Sigma(s, t).$$

Recall that when $\mathcal{T}$ is an arbitrary nondeterministic transducer, for each $s \in \mathrm{dom}(\mathcal{T})$, $[\![\mathcal{T}]\!](s)$ is a set of words in Γ^ω (resp., Γ^*). Hence, we cannot use a similarity function over Γ^ω (resp., Γ^*) to define the similarity between $[\![\mathcal{T}]\!](s)$ and $[\![\mathcal{T}]\!](t)$, for $s, t \in \mathrm{dom}(\mathcal{T})$. Instead, we must use a *set-similarity function* that can compute the similarity between sets of words in Γ^ω (resp., Γ^*). We define K-Lipschitz robustness of nondeterministic transducers using such set-similarity functions (we use the notation d and D for similarity functions and set-similarity functions, respectively).

▶ **Definition 4** (K-Lipschitz Robustness of Nondeterministic Transducers). Given a constant $K \in \mathbb{Q}$ with $K > 0$, a similarity function $d_\Sigma : \Sigma^\omega \times \Sigma^\omega \to \mathbb{Q}^\infty$ (resp., $d_\Sigma : \Sigma^* \times \Sigma^* \to \mathbb{Q}^\infty$) and a set-similarity function $D_\Gamma : 2^{\Gamma^\omega} \times 2^{\Gamma^\omega} \to \mathbb{Q}^\infty$ (resp., $D_\Gamma : 2^{\Gamma^*} \times 2^{\Gamma^*} \to \mathbb{Q}^\infty$), a nondeterministic transducer $\mathcal{T}$, with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$ (resp. $[\![\mathcal{T}]\!] \subseteq \Sigma^* \times \Gamma^*$), is called K-Lipschitz robust w.r.t. d_Σ, D_Γ if:

$$\forall s,t \in \mathrm{dom}(\mathcal{T}) : \; d_\Sigma(s,t) < \infty \Rightarrow D_\Gamma([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) \le K d_\Sigma(s,t).$$

In what follows, we use K-robustness to denote K-Lipschitz robustness. The results in the remainder of this paper hold both for machines processing ω-words as well as for those processing finite words. To keep the presentation clean, we present all results in the context of machines over ω-words, making a distinction as needed. Moreover, we only present some (partial) proofs in this paper. We direct the interested reader to [15] for the complete proofs.

4 Synchronized (Functional) Transducers

In this section, we define a class of functional transducers which admits a decision procedure for K-robustness.

▶ **Definition 5** (Synchronized Transducers). A functional transducer $\mathcal{T}$ with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$ is synchronized iff there exists an automaton $\mathcal{A}_\mathcal{T}$ over $\Sigma \otimes \Gamma$ recognizing the language $\{s \otimes [\![\mathcal{T}]\!](s) : s \in \mathrm{dom}(\mathcal{T})\}$.

Let $\mathcal{T}$ be an arbitrary functional transducer. In each transition, $\mathcal{T}$ reads a single input letter and may generate an empty output word or an output word longer than a single letter. To process such non-aligned input and output words, the automaton $\mathcal{A}_\mathcal{T}$ needs to internally implement a buffer. Thus, $\mathcal{T}$ is synchronized iff there is a bound B on the required size of such a buffer. We can use this observation to check if $\mathcal{T}$ is synchronized. Note that letter-to-letter transducers are synchronized, with B being 0.

▶ **Proposition 6.** *Synchronicity of a functional transducer is decidable in polynomial time.*

Synchronized transducers admit an automata-theoretic decision procedure for checking K-robustness w.r.t. similarity functions satisfying certain properties.

▶ **Theorem 7.** *For every $f \in$ ValFunc, if d_Σ, d_Γ are similarity functions computed by functional f-WA $\mathcal{A}_{d_\Sigma}$, $\mathcal{A}_{d_\Gamma}$, respectively, and $\mathcal{T}$ is a synchronized transducer, K-robustness of $\mathcal{T}$ w.r.t. d_Σ, d_Γ is decidable in polynomial time in the sizes of $\mathcal{T}$, $\mathcal{A}_{d_\Sigma}$ and $\mathcal{A}_{d_\Gamma}$.*

We show that for every $f \in$ ValFunc, if the conditions of Theorem 7 are met, K-robustness of $\mathcal{T}$ can be reduced to the emptiness problem for f-weighted automata, which is decidable in polynomial time.

Similarity functions computed by nondeterministic automata. If we permit the weighted automata computing the similarity functions d_Σ, d_Γ to be nondeterministic, K-robustness becomes undecidable. We can show that the universality problem for nondeterministic weighted automata reduces to checking 1-robustness. Indeed, given a nondeterministic weighted automaton $\mathcal{A}$, consider (1) d_Σ such that $\forall s,t \in \Sigma^\omega$: $d_\Sigma(s,t) = \lambda$ if $s = t$, and undefined otherwise, (2) $\mathcal{T}$ encoding the identity function, and (3) d_Γ such that $\forall s',t' \in \Sigma^\omega$: $d_\Gamma(s',t') = \mathcal{L}_\mathcal{A}(s')$ if $s' = t'$, and undefined otherwise. Note that d_Γ is computed by a nondeterministic weighted automaton obtained from $\mathcal{A}$ by changing each transition (q,a,q') in $\mathcal{A}$ to $(q,(a,a),q')$ while preserving the weight. Then, $\mathcal{T}$ is 1-robust w.r.t. d_Σ, d_Γ iff for all words s, $\mathcal{L}_\mathcal{A}(s) \le \lambda$. Since the universality problem for f-weighted automata is undecidable (e.g., for $f =$ Sum), it follows that checking 1-robustness of transducers with similarity functions computed by nondeterministic weighted automata is undecidable.

We now present examples of synchronized transducers and automatic similarity functions satisfying the conditions of Theorem 7.

▶ **Example 8** (Mealy machines and generalized Manhattan distances.). Mealy machines are perhaps the most widely used transducer model. Prior work [22] has shown decidability of robustness of Mealy machines with respect to generalized Manhattan distances given a fixed bound on the amount of input perturbation. In what follows, we argue the decidability of robustness of Mealy machines (processing infinite words) with respect to generalized Manhattan distances in the presence of unbounded input perturbation.

A Mealy machine $\mathcal{T} : (\Sigma, \Gamma, Q, \{q_0\}, E_{\mathcal{T}}, Q)$ is a synchronized transducer with $\mathcal{A}_{\mathcal{T}}$ given by $(\Sigma \otimes \Gamma, Q, \{q_0\}, E_{\mathcal{A}_{\mathcal{T}}}, Q)$, where $E_{\mathcal{A}_{\mathcal{T}}} = \{(q, a \otimes a', q') : (q, a, a', q') \in E_{\mathcal{T}}\}$. The generalized Manhattan distance $d_M : \Sigma^\omega \times \Sigma^\omega \to \mathbb{Q}^\infty$ can be computed by a functional Sum-weighted automaton $\mathcal{A}_M$ given by the tuple $(\Sigma \otimes \Sigma, \{q_0\}, \{q_0\}, E_M, \{q_0\}, c)$. Here, q_0 is the initial as well as the accepting state, $E_M = \{(q_0, a \otimes b, q_0) : a \otimes b \in \Sigma \otimes \Sigma\}$, and the weight of each transition $(q_0, a \otimes b, q_0)$ equals $\texttt{diff}(a, b)$.

Thus, all the conditions of Theorem 7 are satisfied. K-robustness of Mealy machines, with d_Σ, d_Γ being the generalized Manhattan distance, is decidable in polynomial time.

▶ **Example 9** (Piecewise-linear functions.). Let us use $\underline{q}$ to denote an infinite word over $\{0, \ldots, 9, +, -\}$ representing the fractional part of a real number in base 10. E. g., $\underline{-0.21} = -21$ and $\underline{\pi - 3} = 1415\ldots$ Then, $\underline{q_1} \otimes \ldots \otimes \underline{q_k}$ is a word over $\{0, \ldots, 9, +, -\} \otimes \ldots \otimes \{0, \ldots, 9, +, -\}$ that represents a k-tuple of real numbers $q_1, \ldots, q_k$ from the interval $(-1, 1)$. Now, observe that one can define letter-to-letter transducers that compute the following functions: (1) swapping of arguments, $[\![\mathcal{T}]\!](\underline{q_1}, \ldots, \underline{q_l}, \ldots, \underline{q_m}, \ldots, \underline{q_k}) = (\underline{q_1}, \ldots, \underline{q_m}, \ldots, \underline{q_l}, \ldots, \underline{q_k})$, (2) addition, $[\![\mathcal{T}]\!](\underline{q_1}, \ldots, \underline{q_k}) = (\underline{q_1 + q_2}, \underline{q_2}, \ldots, \underline{q_k})$, (3) multiplication by a constant c, $[\![\mathcal{T}]\!](\underline{q_1}, \ldots, \underline{q_k}) = (\underline{cq_1}, \ldots, \underline{cq_k})$, (4) projection, $[\![\mathcal{T}]\!](\underline{q_1}, \ldots, \underline{q_k}) = (\underline{q_1}, \ldots, \underline{q_{k-1}})$, and (5) conditional expression, $[\![\mathcal{T}]\!](\underline{q_1}, \ldots, \underline{q_k})$ equals $[\![\mathcal{T}_1]\!](\underline{q_1}, \ldots, \underline{q_k})$, if $\underline{q_1} > 0$, and $[\![\mathcal{T}_2]\!](\underline{q_1}, \ldots, \underline{q_k})$ otherwise. We assume that the transducers reject if the results of the corresponding functions lie outside the interval $(-1, 1)$. We can model a large class of piecewise-linear functions using transducers obtained by composition of transducers (1)-(5). The resulting transducers are functional letter-to-letter transducers.

Now, consider d_Σ, d_Γ defined as the *L1*-norm over $\mathbb{R}^k$, i.e., $d_\Sigma(\underline{q_1} \otimes \ldots \otimes \underline{q_k}, \underline{q'_1} \otimes \ldots \otimes \underline{q'_k})$ $= d_\Gamma(\underline{q_1} \otimes \ldots \otimes \underline{q_k}, \underline{q'_1} \otimes \ldots \otimes \underline{q'_k}) = \sum_{i=1}^{k} \mathrm{abs}(q_i - q'_i)$. Observe that d_Σ, d_Γ can be computed by deterministic Disc$_\delta$-weighted automata, with $\delta = \frac{1}{10}$. Therefore, 1-robustness of $\mathcal{T}$ can be decided in polynomial time (Theorem 7). Finally, note that K-robustness of a transducer computing a piecewise-linear function h w. r. t. the above similarity functions is equivalent to Lipschitz continuity of h with coefficient K.

5 Functional Transducers

It was shown in [22] that checking K-robustness of a functional transducer w. r. t. to a fixed bound on the amount of input perturbation is decidable. In what follows, we show that when the amount of input perturbation is unbounded, the robustness problem becomes undecidable even for deterministic transducers.

▶ **Theorem 10.** *1-robustness of deterministic transducers is undecidable.*

Proof. The Post Correspondence Problem (PCP) is defined as follows. Given a set of word pairs $\{\langle v_1, w_1\rangle, \ldots, \langle v_k, w_k\rangle\}$, does there exist a sequence of indices $i_1, \ldots, i_n$ such that $v_{i_1} \cdot \ldots \cdot v_{i_n} = w_{i_1} \cdot \ldots \cdot w_{i_n}$? PCP is known to be undecidable.

Let $\mathcal{G}_{pre} = \{\langle v_1, w_1\rangle, \ldots, \langle v_k, w_k\rangle\}$ be a PCP instance with $v_i, w_i \in \{a, b\}^*$ for each $i \in [1, k]$. We define a new instance $\mathcal{G} = \mathcal{G}_{pre} \cup \{\langle v_{k+1}, w_{k+1}\rangle\}$, where $\langle v_{k+1}, w_{k+1}\rangle = \langle \$, \$\rangle$. Observe that for $i_1, \ldots, i_n \in [1, k]$, $i_1, \ldots, i_n, k+1$ is a solution of $\mathcal{G}$ iff $i_1, \ldots, i_n$ is a solution of $\mathcal{G}_{pre}$. We define a deterministic transducer $\mathcal{T}$ processing finite words and generalized Manhattan distances d_Σ, d_Γ such that $\mathcal{T}$ is *not* 1-robust w. r. t. d_Σ, d_Γ iff $\mathcal{G}$ has a solution of the form $i_1, \ldots, i_n, k+1$, with $i_1, \ldots, i_n \in [1, k]$.

We first define $\mathcal{T}$, which translates indices into corresponding words from the PCP instance $\mathcal{G}$. The input alphabet Σ is the set of indices from $\mathcal{G}$, marked with a *polarity*, L or R, denoting whether an index i, corresponding to a pair $\langle v_i, w_i\rangle \in \mathcal{G}$, is translated to v_i or w_i. Thus, $\Sigma = \{1, \ldots, k+1\} \times \{L, R\}$. The output alphabet Γ is the alphabet of words in $\mathcal{G}$, marked with a polarity. Thus, $\Gamma = \{a, b, \$\} \times \{L, R\}$. The domain of $[\![\mathcal{T}]\!]$ is described by the following regular expression: $\mathrm{dom}(\mathcal{T}) = \Sigma_L^* \langle k+1, L\rangle + \Sigma_R^* \langle k+1, R\rangle$, where for $P \in \{L, R\}$, $\Sigma_P = \{1, \ldots, k\} \times \{P\}$. Thus, $\mathcal{T}$ only processes input words over letters with the same polarity, rejecting upon reading an input letter with a polarity different from that of the first input letter. Moreover, $\mathcal{T}$ accepts iff the first occurrence of $\langle k+1, L\rangle$ or $\langle k+1, R\rangle$ is in the last position of the input word. Note that the domain of $\mathcal{T}$ is *prefix-free*, i.e., if $s, t \in \mathrm{dom}(\mathcal{T})$ and s is a prefix of t, then $s = t$. Let u^P denote the word $u \otimes P^{|u|}$. Along accepting runs, $\mathcal{T}$ translates each input letter $\langle i, L\rangle$ to v_i^L and each letter $\langle i, R\rangle$ to w_i^R, where $\langle v_i, w_i\rangle$ is the i^{th} word pair of $\mathcal{G}$. Thus, the function computed by $\mathcal{T}$ is:

$$[\![\mathcal{T}]\!](\langle i_1, L\rangle \ldots \langle i_n, L\rangle\langle k+1, L\rangle) = v_{i_1}^L \ldots v_{i_n}^L v_{k+1}^L$$
$$[\![\mathcal{T}]\!](\langle i_1, R\rangle \ldots \langle i_n, R\rangle\langle k+1, R\rangle) = w_{i_1}^R \ldots w_{i_n}^R w_{k+1}^R$$

We define the output similarity function d_Γ as a generalized Manhattan distance with the following *symmetric* $\mathtt{diff}_\Gamma$ where $P, Q \in \{L, R\}$ and $\alpha, \beta \in \{a, b, \$\}$ with $\alpha \neq \beta$:

$$\mathtt{diff}_\Gamma(\langle \alpha, P\rangle, \langle \alpha, P\rangle) = 0 \qquad \mathtt{diff}_\Gamma(\langle \alpha, L\rangle, \langle \alpha, R\rangle) = 2$$
$$\mathtt{diff}_\Gamma(\langle \alpha, P\rangle, \langle \beta, Q\rangle) = 1 \qquad \mathtt{diff}_\Gamma(\langle \alpha, P\rangle, \texttt{#}) = 1$$

Note that for $s', t' \in \Gamma^*$ with different polarities, $d_\Gamma(s', t')$ equals the sum of $max(|s'|, |t'|)$ and $\mathcal{N}(s', t')$, where $\mathcal{N}(s', t')$ is the number of positions in which s' and t' agree on the first components of their letters.

We define a projection π as $\pi(\langle i_1, P_1\rangle\langle i_2, P_2\rangle \ldots \langle i_n, P_n\rangle) = i_1 i_2 \ldots i_n$, where $i_1, \ldots, i_n \in [1, k+1]$ and $P_1, \ldots, P_n \in \{L, R\}$. We define the input similarity function d_Σ as a generalized Manhattan distance such that $d_\Sigma(s, t)$ is finite iff $\pi(s)$ is a prefix of $\pi(t)$ or vice versa. We define d_Σ using the following *symmetric* $\mathtt{diff}_\Sigma$ where $P, Q \in \{L, R\}$ and $i, j \in [1, k+1]$ with $i \neq j$:

$$\mathtt{diff}_\Sigma(\langle i, P\rangle, \langle i, P\rangle) = 0 \qquad \mathtt{diff}_\Sigma(\langle i, P\rangle, \langle j, Q\rangle) = \infty$$
$$\mathtt{diff}_\Sigma(\langle i, L\rangle, \langle i, R\rangle) = |v_i| + |w_i|, \text{ if } i \in [1, k] \qquad \mathtt{diff}_\Sigma(\langle i, P\rangle, \texttt{#}) = \infty$$
$$\mathtt{diff}_\Sigma(\langle k+1, L\rangle, \langle k+1, R\rangle) = 1$$

Thus, for all $s, t \in \mathrm{dom}(\mathcal{T})$, $d_\Sigma(s, t) < \infty$ iff one of the following holds:

(i) for some $P \in \{L, R\}$, $s = t = \langle i_1, P\rangle \ldots \langle i_n, P\rangle\langle k+1, P\rangle$, or,

(ii) $s = \langle i_1, L\rangle \ldots \langle i_n, L\rangle\langle k+1, L\rangle$ and $t = \langle i_1, R\rangle \ldots \langle i_n, R\rangle\langle k+1, R\rangle$.

In case (i), $d_\Sigma(s, t) = d_\Gamma([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) = 0$. In case (ii), $d_\Sigma(s, t) = |[\![\mathcal{T}]\!](s)| + |[\![\mathcal{T}]\!](t)| - 1$ and $d_\Gamma([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) = max(|[\![\mathcal{T}]\!](s)|, |[\![\mathcal{T}]\!](t)|) + \mathcal{N}([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t))$. Thus, $d_\Gamma([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) > d_\Sigma(s, t)$ iff $\mathcal{N}([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) = min(|[\![\mathcal{T}]\!](s)|, |[\![\mathcal{T}]\!](t)|)$. Since the letters $\langle \$, L\rangle, \langle \$, R\rangle$ occur exactly once in $[\![\mathcal{T}]\!](s)$, $[\![\mathcal{T}]\!](t)$, respectively, at the end of each word, $\mathcal{N}([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t)) = min(|[\![\mathcal{T}]\!](s)|, |[\![\mathcal{T}]\!](t)|)$ iff $|[\![\mathcal{T}]\!](s)| = |[\![\mathcal{T}]\!](t)|$ and $\pi([\![\mathcal{T}]\!](s)) = \pi([\![\mathcal{T}]\!](t))$, which holds iff $\mathcal{G}$ has a solution. Therefore, $\mathcal{T}$ is *not* 1-robust w. r. t. d_Σ, d_Γ iff $\mathcal{G}$ has a solution. ◀

We have shown that checking 1-robustness w. r. t. generalized Manhattan distances is undecidable. Observe that for every $K > 0$, K-robustness can be reduced to 1-robustness by scaling the output distance by K. We conclude that checking K-robustness is undecidable for any fixed K. In contrast, if K is not fixed, checking if there exists K such that $\mathcal{T}$ is K-robust w. r. t. d_Σ, d_Γ is decidable for transducers processing finite words.

Let us define a functional transducer $\mathcal{T}$ to be *robust* w. r. t. d_Σ, d_Γ if there exists K such that $\mathcal{T}$ is K-robust w. r. t. d_Σ, d_Γ.

▶ **Proposition 11.** *Let $\mathcal{T}$ be a given functional transducer processing finite words and d_Σ, d_Γ be instances of the generalized Manhattan distance.*

1. *Robustness of $\mathcal{T}$ is decidable in* co-NP*.*
2. *One can compute $K_\mathcal{T}$ such that $\mathcal{T}$ is robust iff $\mathcal{T}$ is $K_\mathcal{T}$-robust.*

Proof sketch. Given $\mathcal{T}$, one can easily construct a *trim*[2] functional transducer $\mathcal{P}_\mathcal{T}$ such that $[\![\mathcal{P}_\mathcal{T}]\!](s,t) = (s',t')$ iff $[\![\mathcal{T}]\!](s) = s'$ and $[\![\mathcal{T}]\!](t) = t'$. We show that $\mathcal{T}$ is not robust w. r. t. generalized Manhattan distances iff there exists some cycle in $\mathcal{P}_\mathcal{T}$ satisfying certain properties. Checking the existence of such a cycle is in NP. If such a cycle exists, one can construct paths in $\mathcal{P}_\mathcal{T}$ through the cycle, labeled with input words (s,t) and output words (s',t'), with $d_\Gamma(s',t') > K d_\Sigma(s,t)$ for *any* K. Conversely, if there exists no such cycle, one can compute $K_\mathcal{T}$ such that $\mathcal{T}$ is $K_\mathcal{T}$-robust. It follows that one can compute $K_\mathcal{T}$ such that $\mathcal{T}$ is robust iff $\mathcal{T}$ is $K_\mathcal{T}$-robust. ◀

5.1 Beyond Synchronized Transducers

In this section, we present an approach for natural extensions of Theorem 7.

Isometry approach. We say that a transducer $\mathcal{T}$ is a (d_Λ, d_Δ)-*isometry* if and only if for all $s,t \in \mathrm{dom}(\mathcal{T})$ we have $d_\Lambda(s,t) = d_\Delta([\![\mathcal{T}]\!](s), [\![\mathcal{T}]\!](t))$.

▶ **Proposition 12.** *Let $\mathcal{T}, \mathcal{T}'$ be functional transducers with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$ and $[\![\mathcal{T}']\!] \subseteq \Lambda^\omega \times \Delta^\omega$. Assume that there exist transducers $\mathcal{T}^I$ and $\mathcal{T}^O$ such that $\mathcal{T}^I$ is a (d_Σ, d_Λ)-isometry, $\mathcal{T}^O$ is a (d_Δ, d_Γ)-isometry and $[\![\mathcal{T}]\!] = [\![\mathcal{T}^O \circ (\mathcal{T}' \circ \mathcal{T}^I)]\!]$. Then, for every $K > 0$, $\mathcal{T}$ is K-robust w. r. t. d_Σ, d_Γ if and only if $\mathcal{T}'$ is K-robust w. r. t. d_Λ, d_Δ.*

▶ **Example 13** (Stuttering)**.** For a given word w we define the *stuttering pruned* word $\textsc{Stutter}(w)$ as the result of removing from w letters that are the same as the previous letter. E. g. $\textsc{Stutter}(\mathbf{ba}aa\mathbf{c}c\mathbf{a}aa\mathbf{b}) = bacab$.

Consider a transducer $\mathcal{T}$ and a similarity function d_Σ over finite words that are *stuttering invariant*, i.e., for all $s,t \in \mathrm{dom}(\mathcal{T})$, if $\textsc{Stutter}(s) = \textsc{Stutter}(t)$, then $[\![\mathcal{T}]\!](s) = [\![\mathcal{T}]\!](t)$ and for every $u \in \Sigma^*$, $d_\Sigma(s,u) = d_\Sigma(t,u)$. In addition, we assume that for every $s \in \mathrm{dom}(\mathcal{T})$, $|[\![\mathcal{T}]\!](s)| = |\textsc{Stutter}(s)|$.

Observe that these assumptions imply that: (1) the projection transducer $\mathcal{T}^\pi$ defined such that $[\![s]\!] = \textsc{Stutter}(s)$ is a (d_Σ, d_Σ)-isometry, (2) the transducer $\mathcal{T}^S$ obtained by restricting the domain of $\mathcal{T}$ to *stuttering-free* words, i.e., the set $\{w \in \mathrm{dom}(\mathcal{T}) : \textsc{Stutter}(w) = w\}$, is a synchronized transducer[3], and (3) $[\![\mathcal{T}]\!] = [\![\mathcal{T}^I \circ (\mathcal{T}^S \circ \mathcal{T}^\pi)]\!]$, where $\mathcal{T}^I$ defines the identity function over Γ^*. Therefore, by Proposition 12, in order to check K-robustness of $\mathcal{T}$, it suffices to check K-robustness of $\mathcal{T}^S$. Since $\mathcal{T}^S$ is a synchronized transducer, K-robustness of

[2] $\mathcal{P}_\mathcal{T}$ is trim if every state in $\mathcal{P}_\mathcal{T}$ is reachable from the initial state and some final state is reachable from every state in $\mathcal{P}_\mathcal{T}$.

[3] Note that any functional transducer $\mathcal{T}$ with the property: for every $s \in \mathrm{dom}(\mathcal{T})$, $|[\![\mathcal{T}]\!](s)| = |s|$, is a synchronized transducer.

$\mathcal{T}^S$ can be effectively checked, provided the similarity functions d_Σ, d_Γ satisfy the conditions of Theorem 7.

▶ **Example 14** (Letter-to-multiple-letters transducers). Consider a transducer $\mathcal{T}$ which on every transition outputs a 2-letter word[4]. Although, $\mathcal{T}$ is not synchronized, it can be transformed to a letter-to-letter transducer $\mathcal{T}^D$, whose output alphabet is $\Gamma \times \Gamma$. The transducer $\mathcal{T}^D$ is obtained from $\mathcal{T}$ by substituting each output word ab to a single letter $\langle a, b\rangle$ from $\Gamma \times \Gamma$. We can use $\mathcal{T}^D$ to decide K-robustness of $\mathcal{T}$ in the following way. First, we define transducers $\mathcal{T}^I, \mathcal{T}^{\text{pair}}$ such that $\mathcal{T}^I$ computes the identity function over Σ^ω and $\mathcal{T}^{\text{pair}}$ is a transducer representing the function $[\![\mathcal{T}^{\text{pair}}]\!](\langle a_1, b_1\rangle\langle a_2, b_2\rangle \ldots) = a_1 b_1 a_2 b_2 \ldots$. Observe that $[\![\mathcal{T}]\!] = [\![\mathcal{T}^{\text{pair}} \circ (\mathcal{T}^D \circ \mathcal{T}^I)]\!]$. Second, we define d_Γ^D as follows: $\forall s, t \in (\Sigma \times \Sigma)^\omega$, $d_\Gamma^D(s,t) = d_\Gamma([\![\mathcal{T}^{\text{pair}}]\!](s), [\![\mathcal{T}^{\text{pair}}]\!](t))$. Observe that $\mathcal{T}^I$ is a (d_Σ, d_Σ)-isometry and $\mathcal{T}^{\text{pair}}$ is a (d_Γ^D, d_Γ)-isometry. Thus, K-robustness of $\mathcal{T}$ w. r. t. d_Σ, d_Γ reduces to K-robustness of the letter-to-letter transducer $\mathcal{T}^D$ w. r. t. d_Σ, d_Γ^D, which can be effectively checked (Theorem 7).

6 Nondeterministic Transducers

Let $\mathcal{T}$ be a nondeterministic transducer with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$. Let d_Σ be an automatic similarity function for computing the similarity between input words in Σ^*. As explained in Sec. 3, the definition of K-robust nondeterministic transducers involves set-similarity functions that can compute the similarity between sets of output words in Γ^ω. In this section, we examine the K-robustness problem of $\mathcal{T}$ w. r. t. d_Σ and three classes of such set-similarity functions.

Let d_Γ be an automatic similarity function for computing the similarity between output words in Γ^ω. We first define three set-similarity functions induced by d_Γ.

▶ **Definition 15.** Given sets A, B of words in Γ^ω, we consider the following set-similarity functions induced by d_Γ:

(i) Hausdorff set-similarity function $D_\Gamma^H(A,B)$ induced by d_Γ:

$$D_\Gamma^H(A,B) = max\{\, \sup_{s\in A} \inf_{t\in B} d_\Gamma(s,t), \sup_{s\in B} \inf_{t\in A} d_\Gamma(s,t) \,\}$$

(ii) Inf-inf set-similarity function $D_\Gamma^{\inf}(A,B)$ induced by d_Γ:

$$D_\Gamma^{\inf}(A,B) = \inf_{s\in A} \inf_{t\in B} d_\Gamma(s,t)$$

(iii) Sup-sup set-similarity function $D_\Gamma^{\sup}(A,B)$ induced by d_Γ:

$$D_\Gamma^{\sup}(A,B) = \sup_{s\in A} \sup_{t\in B} d_\Gamma(s,t)$$

Of the above set-similarity functions, only the Hausdorff set-similarity function is a distance function (if d_Γ is a distance function).

Note that when $\mathcal{T}$ is a functional transducer, each set-similarity function above reduces to d_Γ. Hence, K-robustness of a functional transducer $\mathcal{T}$ w. r. t. d_Σ, D_Γ and K-robustness of $\mathcal{T}$ w. r. t. d_Σ, d_Γ coincide. As K-robustness of functional transducers in undecidable (Theorem 10), K-robustness of nondeterministic transducers w. r. t. the above set-similarity functions is undecidable as well.

Recall from Theorem 7 that K-robustness of a synchronized (functional) transducer is decidable w. r. t. certain automatic similarity functions. In particular, K-robustness of

[4] One can easily generalize this example to any fixed number.

Mealy machines is decidable when d_Σ, d_Γ are generalized Manhattan distances. In contrast, K-robustness of nondeterministic letter-to-letter transducers is undecidable w. r. t. the Hausdorff and Inf-inf set-similarity functions even when d_Σ, d_Γ are generalized Manhattan distances. Among the above defined set-similarity functions, K-robustness of nondeterministic transducers is decidable only w. r. t. the Sup-sup set-similarity function.

▶ **Theorem 16.** *Let d_Σ, d_Γ be computed by functional weighted-automata. Checking K-robustness of nondeterministic letter-to-letter transducers w. r. t. d_Σ, D_Γ induced by d_Γ is*

(i) *undecidable if D_Γ is the Hausdorff set-similarity function,*
(ii) *undecidable if D_Γ is the Inf-inf set-similarity function, and*
(iii) *decidable if D_Γ is the Sup-sup set-similarity function and d_Σ, d_Γ satisfy the conditions of Theorem 7.*

Proof of (iii). We can encode nondeterministic choices of $\mathcal{T}$, with $[\![\mathcal{T}]\!] \subseteq \Sigma^\omega \times \Gamma^\omega$, in an extended input alphabet $\Sigma \times \Lambda$. We construct a deterministic transducer $\mathcal{T}^e$ such that for every $s \in \Sigma^\omega$, $\{[\![\mathcal{T}^e]\!](\langle s, \lambda\rangle) : \langle s, \lambda\rangle \in \mathrm{dom}(\mathcal{T}^e)\} = [\![\mathcal{T}]\!](s)$. We also define d^e_Σ such that for all $\langle s, \lambda_1\rangle, \langle t, \lambda_2\rangle \in (\Sigma \times \Lambda)^\omega$, $d^e_\Sigma(\langle s, \lambda_1\rangle, \langle t, \lambda_2\rangle) = d_\Sigma(s,t)$. Then, $\mathcal{T}$ is K-robust w. r. t. $d_\Sigma, D^{\mathrm{sup}}_\Gamma$ induced by d_Γ iff $\mathcal{T}^e$ is K-robust w. r. t. d^e_Σ, d_Γ. Indeed, a nondeterministic transducer $\mathcal{T}$ is K-robust w. r. t. d_Σ, D^{sup}_Γ induced by d_Γ iff for all input words $s, t \in \mathrm{dom}(\mathcal{T})$ and for all outputs $s' \in [\![\mathcal{T}]\!](s), t' \in [\![\mathcal{T}]\!](t)$, $d_\Sigma(s,t) < \infty$ implies $d_\Gamma(s',t') \le K d_\Sigma(s,t)$. ◀

7 Related Work

In early work [19], [7, 8] on continuity and robustness analysis, the focus is on software programs manipulating numbers. In [19], the authors compute the maximum deviation of a program's output given the maximum possible perturbation in a program input. In [7], the authors formalize $\epsilon - \delta$ continuity of programs and present sound proof rules to prove continuity of programs. In [8], the authors formalize robustness of programs as Lipschitz continuity and present a sound program analysis for robustness verification. While arrays of numbers are considered in [8], the size of an array is immutable.

More recent papers have aimed to develop a notion of robustness for reactive systems. In [23], the authors present polynomial-time algorithms for the analysis and synthesis of robust transducers. Their notion of robustness is one of input-output stability, that bounds the output deviation from disturbance-free behaviour under bounded disturbance, as well as the persistence of the effect of a sporadic disturbance. Their distances are measured using cost functions that map *each* string to a nonnegative integer. In [18, 4, 2], the authors develop different notions of robustness for reactive systems, with ω-regular specifications, interacting with uncertain environments. In [9], the authors present a polynomial-time algorithm to decide robustness of sequential circuits modeled as Mealy machines, w. r. t. a *common suffix distance* metric. Their notion of robustness also bounds the persistence of the effect of a sporadic disturbance.

Recent work in [21] and [22] formalized and studied robustness of systems modeled using transducers, in the presence of bounded perturbation. The work in [21] focussed on the outputs of synchronous networks of Mealy machines in the presence of channel perturbation. The work in [22] focussed on the outputs of functional transducers in the presence of input perturbation. Both papers presented decision procedures for robustness verification w. r. t. specific distance functions such as Manhattan and Levenshtein distances.

8 Conclusion

In this paper, we studied the K-Lipschitz robustness problem for finite-state transducers. While the general problem is undecidable, we identified decidability criteria that enable reduction of K-robustness to the emptiness problem for weighted automata.

In the future, we wish to extend our work in two directions. We plan to study robustness of other computational models. We also wish to investigate synthesis of robust transducers.

References

1 S. Almagor, U. Boker, and O. Kupferman. What's Decidable about Weighted Automata? In *ATVA*, pages 482–491. LNCS 6996, Springer, 2011.

2 R. Bloem, K. Greimel, T. Henzinger, and B. Jobstmann. Synthesizing Robust Systems. In *Formal Methods in Computer Aided Design (FMCAD)*, pages 85–92, 2009.

3 R. K. Bradley and I. Holmes. Transducers: An Emerging Probabilistic Framework for Modeling Indels on Trees. *Bioinformatics*, 23(23):3258–3262, 2007.

4 P. Cerny, T. Henzinger, and A. Radhakrishna. Simulation Distances. In *Conference on Concurrency Theory (CONCUR)*, pages 253–268, 2010.

5 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Alternating weighted automata. In *FCT*, volume 5699 of *LNCS*, pages 3–13. Springer, 2009.

6 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4), 2010.

7 S. Chaudhuri, S. Gulwani, and R. Lublinerman. Continuity Analysis of Programs. In *Principles of Programming Languages (POPL)*, pages 57–70, 2010.

8 S. Chaudhuri, S. Gulwani, R. Lublinerman, and S. Navidpour. Proving Programs Robust. In *Foundations of Software Engineering (FSE)*, pages 102–112, 2011.

9 L. Doyen, T. A. Henzinger, A. Legay, and D. Ničković. Robustness of Sequential Circuits. In *Application of Concurrency to System Design (ACSD)*, pages 77–84, 2010.

10 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer Publishing Company, Incorporated, 1st edition, 2009.

11 J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag New York, Inc., New York, USA, 1996.

12 E. M. Gurari and O. H. Ibarra. A Note on Finitely-Valued and Finitely Ambiguous Transducers. *Mathematical Systems Theory*, 16(1):61–66, 1983.

13 D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

14 T. A. Henzinger. Two Challenges in Embedded Systems Design: Predictability and Robustness. *Philosophical Transactions of the Royal Society*, 366:3727–3736, 2008.

15 T. A. Henzinger, J. Otop, and R. Samanta. Lipschitz Robustness of Finite-state Transducers. *CoRR*, abs/1404.6452, 2014.

16 K. Zhou and J. C. Doyle and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

17 Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *IJAC*, 4(3):405–426, 1994.

18 R. Majumdar, E. Render, and P. Tabuada. A Theory of Robust Omega-regular Software Synthesis. *ACM Transactions on Embedded Computing Systems*, 13, 2013.

19 R. Majumdar and I. Saha. Symbolic Robustness Analysis. In *IEEE Real-Time Systems Symposium*, pages 355–363, 2009.

20 M. Mohri. Finite-state Transducers in Language and Speech Processing. *Computational Linguistics*, 23(2):269–311, 1997.

21 R. Samanta, J. V. Deshmukh, and S. Chaudhuri. Robustness Analysis of Networked Systems. In *Verification, Model Checking, and Abstract Interpretation (VMCAI)*, pages 229–247, 2013.

22 R. Samanta, J. V. Deshmukh, and S. Chaudhuri. Robustness Analysis of String Transducers. In *ATVA*, pages 427–441. LNCS 8172, Springer, 2013.

23 P. Tabuada, A. Balkan, S. Y. Caliskan, Y. Shoukry, and R. Majumdar. Input-Output Robustness for Discrete Systems. In *International Conference on Embedded Software (EMSOFT)*, 2012.

24 M. Veanes, P. Hooimeijer, B. Livshits, D. Molnar, and N. Bjørner. Symbolic Finite State Transducers: Algorithms and Applications. In *Principles of Programming Languages (POPL)*, pages 137–150, 2012.